

Article

# Lateral and Longitudinal Driving Behavior Prediction Based on Improved Deep Belief Network

Lei Yang<sup>1</sup>, Chunqing Zhao<sup>2</sup>, Chao Lu<sup>1</sup> , Lianzhen Wei<sup>1,3</sup> and Jianwei Gong<sup>1,3,\*</sup>

<sup>1</sup> School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China; 3120170237@bit.edu.cn (L.Y.); chaolu@bit.edu.cn (C.L.); 3120200396@bit.edu.cn (L.W.)

<sup>2</sup> China North Vehicle Research Institute, Beijing 100072, China; zhaocq2013@163.com

<sup>3</sup> Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing 314019, China

\* Correspondence: gongjianwei@bit.edu.cn

**Abstract:** Accurately predicting driving behavior can help to avoid potential improper maneuvers of human drivers, thus guaranteeing safe driving for intelligent vehicles. In this paper, we propose a novel deep belief network (DBN), called MSR-DBN, by integrating a multi-target sigmoid regression (MSR) layer with DBN to predict the front wheel angle and speed of the ego vehicle. Precisely, the MSR-DBN consists of two sub-networks: one is for the front wheel angle, and the other one is for speed. This MSR-DBN model allows ones to optimize lateral and longitudinal behavior predictions through a systematic testing method. In addition, we consider the historical states of the ego vehicle and surrounding vehicles and the driver's operations as inputs to predict driving behaviors in a real-world environment. Comparison of the prediction results of MSR-DBN with a general DBN model, back propagation (BP) neural network, support vector regression (SVR), and radical basis function (RBF) neural network, demonstrates that the proposed MSR-DBN outperforms the others in terms of accuracy and robustness.



**Citation:** Yang, L.; Zhao, C.; Lu, C.; Wei, L.; Gong, J. Lateral and Longitudinal Driving Behavior Prediction Based on Improved Deep Belief Network. *Sensors* **2021**, *21*, 8498. <https://doi.org/10.3390/s21248498>

Academic Editor: Francisco J. Martinez

Received: 1 December 2021

Accepted: 16 December 2021

Published: 20 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/>).

**Keywords:** driving behavior prediction; deep belief network; intelligent vehicles

## 1. Introduction

Intelligent transportation systems aim to improve traffic efficiency, safety, and driver comfort in various situations [1–3]. As an essential element of intelligent transportation systems, intelligent vehicles have potentials to help traffic participants make effective decisions for driving safely and efficiently. Most driver assistance systems are developed to ensure and improve driving safety in specific critical environments [1], which requires a clear and comprehensive understanding of driving behavior to achieve a high level of intelligence [4]. An accurate driving behavior prediction can leave sufficient time for driver assistance systems to deliver a warning to the driver or for autonomous systems to directly take over the vehicle to ensure traffic safety [1]. Many researchers focus on driving behavior prediction regarding elements, such as motion, gaze behavior, and intention [5–8]. Ref. [9] proposed a new evolving fuzzy systems method based on adaptive fuzzy pattern classification for the detecting the lane-change intention of a driver. Ref. [10] applied non-linear polynomial regression and recurrent hidden semi-Markov model to realize the recognition of driver lane-change intention. Here, we focus on predicting driving behaviors such as lateral and longitudinal behaviors. Some mature model-based methods have been used for prediction such as dynamic Bayesian networks, support vector machine, and hidden Markov models [11–14]. For example, Kumagai presents a driving behavior prediction method based on dynamic Bayesian networks and shows good prediction results for stop behavior [15]. A prediction approach based on a dynamic Bayesian network is proposed for lane-change maneuvers and also shows good accuracy [16]. Ref. [17] develops a prediction method for lane-change maneuver of the vehicle ahead by using a hidden

Markov model, and result shows that the model can achieve a high accuracy rate. However, model-based methods cannot adapt to the infinite complexity of driving behaviors.

Fortunately, the data-driven methods, such as artificial neural networks (ANNs), have been widely used for driving behavior prediction because of their flexible structures and powerful capability to describe non-linearity [18–21]. For instance, ref. [19] used three variables (safe speed, workload, and yaw rate) as the inputs to predict a safe speed in curve negotiation based on a two-layer back propagation (BP) neural network. In addition, the ANNs were also used to predict upcoming lane change behavior by considering three phases including lane change intention, preparation, and action [20]. Although significant achievements have been made, the existing ANN-based methods, such as BP and Radial Basis Function (RBF) neural network showed a limited accuracy in driving behavior learning and prediction because of their shallow architecture and manually-selected features [19,22,23].

The advanced on-board sensors allow obtaining plentiful data, which enables some data-hungry techniques to be practicable such as deep learning. Deep learning has attracted full attention because of its capability to automatically and genuinely extract features [24–27]. Many recent works on driving behavior prediction are primarily based on the image or video data from cameras [28–31]. For instance, researchers of [1,32] used raw images and human driving videos from camera to predict different driver actions including steering and braking/acceleration based on a recurrent neural network (RNN). Ref. [33] exploited a convolutional neural network (CNN) to extract features in a scene understanding subsystem for decision-making with considering the features from images. Ref. [34] proposed a novel method to predict lane-change maneuvers in highway scenarios by using deep learning and visual representations of the traffic scene. Additionally, some scholars studied driving behavior using mobile-based signals, for example, ref. [35–37]. However, the majority research above relied on image data rather than focused on the deep end-to-end network based on the data from on-board sensors for driving behavior prediction. CNN is also widely used in a wide range of image-related research work due to its powerful convolution structure in image feature extraction and recognition. It is suitable for dealing with 2-D data such as image data, but not suitable for working on the data obtained from on-board sensors [38].

Some existing works used the real-time and historical data collected from on-board sensors to predict driving behaviors [39]. For example, ref. [26] took road geometry, historical traffic information, and driver specific speed as inputs of a deep network stacked autoencoder to predict current vehicle speed. Based on the historical sensory data collected from natural driving, the structural RNN can predict the forthcoming lane-change behaviors of surrounding vehicles in 1~3 s ahead [27]. Ref. [40] proposed a new model to predict the lane-change maneuvers by considering the historical information and neighbor vehicles' information. Driving behavior prediction should be not only based on the understanding of traffic information but also based on the learning of driving control behavior of human drivers to realize a high level of intelligence. However, the research above did not take both human driver actions and environmental behaviors into consideration to predict the future behavior of the ego vehicle.

Although the deep learning models can extract more features than shallow ones, the above existing methods only separately predict longitudinal and lateral behaviors. For example, ref. [26] focuses on the longitudinal while [27] focuses on lane change behavior. Ref. [1] predicted lane change, steering, acceleration, and braking, but the predictions of longitudinal and lateral behaviors are conducted regardless of their mutual relationship. Among all deep learning methods, the deep belief network (DBN) is relatively effective [41], and most of the applications based on DBN were classification [42]. Then, researchers applied DBN into traffic flow prediction [43]. For example, ref. [41] predicted the traffic flow using DBN with multitask learning, which proves the deep model can achieve much high prediction accuracy for traffic flow. Considering the ability of DBN to encode network with multiple layers and avoid over-fitting and local minimum [26,41,44],

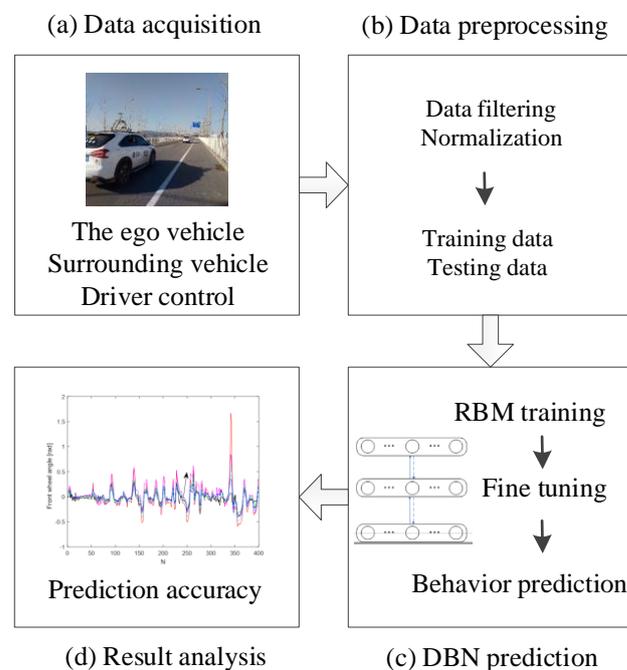
this paper attempts to develop a deep structure to predict longitudinal and lateral driving behaviors simultaneously.

In addition, the driving action implemented by the driver does depend on the states of surrounding vehicles. However, the future states of surrounding vehicles are unavailable when predicting the driver's forthcoming behavior of the ego vehicle. Therefore, most behavior prediction methods are based on historical data without considering the future states of surrounding vehicles [1,32,33]. In this paper, we predict future states of surrounding vehicles by utilizing corresponding historical states. Then, we predict the future maneuver of the ego vehicle based on predicted states of surrounding vehicles and historical multi-sensor data including the states of surrounding vehicles, the ego vehicle, and driver control.

Tackling the problem of behavior prediction can simplify the problem of trajectory prediction and contribute to the decision-making of the vehicle [27]. From this perspective, this paper demonstrates that the deep learning model based on DBN can extract features to predict and understand the driving behavior and then utilizes the DBN to predict the lateral and longitudinal behaviors of an ego vehicle.

Figure 1 illustrates our proposed driving behavior prediction system, which consists of four modules: data acquisition, data preprocessing, DBN prediction, and result analysis. The main contributions of this paper are in three-fold:

- Developing a general prediction system, which allows us to consider real-world data including states of surrounding vehicles and the ego vehicle and the driver's control inputs simultaneously to predict the driving behavior in an end-to-end way;
- Proposing a systematic testing method to obtain optimal parameters of the prediction model;
- Proposing an MSR-DBN prediction model with a multi-target sigmoid regression layer to realize coupled optimization for lateral and longitudinal behavior prediction.



**Figure 1.** Proposed driving behavior prediction system.

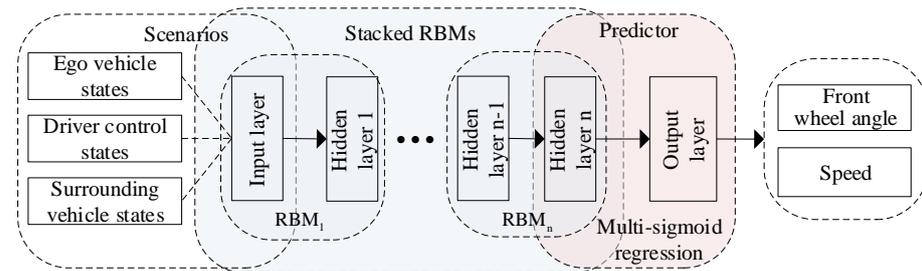
The remainder of the paper is organized as follows. Section 2 presents the DBN prediction architecture and proposes the MSR-DBN prediction model. Section 3 presents the data collection and data pre-training for the experiment. Section 4 shows the experiment process of determining the optimal structure of our proposed model. Then, the prediction

results based on MSR-DBN and comparison results are analyzed. Section 5 gives further discussion and conclusion.

## 2. Driving Behavior Prediction Model

### 2.1. MSR-DBN Prediction Model

The DBN, as a deep learning approach, usually is composed of several restricted Boltzmann machines (RBMs) [45,46]. Figure 2 displays the typical DBN prediction architecture with a multi-target sigmoid regression (MSR) layer as the predictor. The architecture is composed of one input layer,  $n$  hidden layers (i.e.,  $n$  RBMs), and one output layer. The input layer is the first RBM (denoted as  $RBM_1$ ), which is observable.



**Figure 2.** The typical DBN driving behavior prediction architecture.

Our goal is to predict the lateral and longitudinal behaviors of an ego vehicle such as the front wheel angle and speed by considering the behaviors of surrounding vehicles. To this end, we select three states as the model inputs, including the states of the ego vehicle, the driver operations, and the states of surrounding vehicles. Specifically, we denote the inputs as  $I = \{I_E, I_D, I_S\}$ , i.e., the states of the ego vehicle ( $I_E$ ), driver control ( $I_D$ ), and surrounding vehicles ( $I_S$ ), respectively. These three inputs are defined as

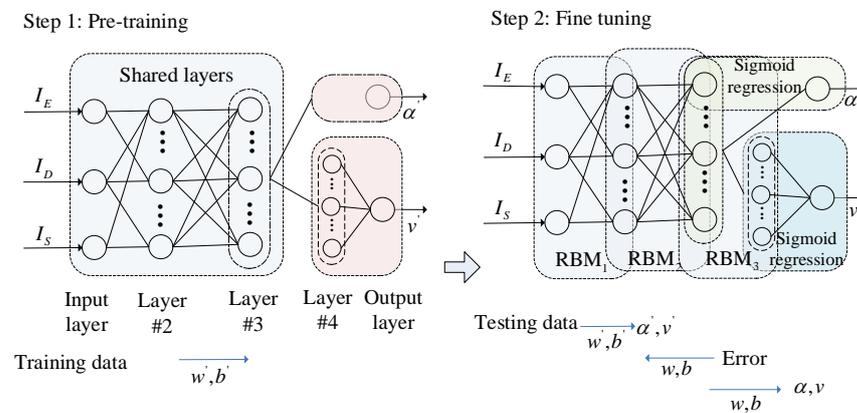
$$\begin{aligned} I_E &= v_{(t-N):t} \\ I_D &= \alpha_{(t-N):t} \\ I_S &= [\Delta x_{(t-N):(t+1)}, \Delta y_{(t-N):(t+1)}, \Delta v_{(t-N):(t+1)}] \end{aligned} \quad (1)$$

where  $t - N$  denotes the historical time  $t - N$ ,  $t$  denotes the current moment,  $t + 1$  denotes the future moment;  $v_{(t-N):t}$  and  $\alpha_{(t-N):t}$  are a series of data for speed and front wheel angle of the ego vehicle from time  $t - N$  to  $t$ ;  $\Delta x_{(t-N):(t+1)}$ ,  $\Delta y_{(t-N):(t+1)}$ , and  $\Delta v_{(t-N):(t+1)}$  are the relative lateral position, relative longitudinal position, and relative speed between the ego vehicle and surrounding vehicles from time  $t - N$  to  $t + 1$ , respectively.

We conducted a series of experiments and found that traditional architecture in Figure 2 cannot obtain an expected prediction performance for both front wheel angle and speed simultaneously. One of the reason is that the model has an integrated structure, and the same fixed layers and model parameters are obtained and used to predict lateral and longitudinal behaviors simultaneously. However, these two behaviors are significantly different from each other. To overcome this limitation, we modify the DBN structure by adopting different sub-structures for front wheel angle ( $\alpha$ ) and speed ( $v$ ) separately with the same input and output (Figure 3). Our proposed model contains two sub-networks: a four-layer sub-network for front wheel angle and a five-layer sub-network for speed, to extract features and predict the front wheel angle and speed of the ego vehicle (i.e., Step 1 in Figure 3). This modification makes it tractable to optimize the couplings of the lateral and longitudinal behaviors.

In order to improve the prediction efficiency, Layer #2 and Layer #3 share a sub-network between the front wheel angle and speed in the new model, i.e.,  $RBM_1$  and  $RBM_2$  share the RBMs to train model parameters for two sub-networks (see Step 1 and Step 2 in Figure 3). The shared layers can not only simplify the model structure and reduce model parameters but also contribute to reducing the redundant computing burden. The output

layer in the front wheel angle sub-network and the Layer #4 and output layer in speed sub-network are independent layers to extract further features and predict lateral and longitudinal behaviors. We chosen 100 hidden nodes in the shared layers (Layers #2 and #3) and Layer #4; see details in Section 4.



**Figure 3.** Improved MSR-DBN prediction model.

## 2.2. Training Procedure

Figure 3 shows the training process of our model including two steps [41,47–49]: pre-training and fine tuning. The algorithm of the proposed MSR-DBN prediction model is summarized as:

- Initialization—Initialize the weights and biases and preprocess training and testing data;
- Pre-training—Train the shared layers and independent layers for the feature extraction to obtain the model parameters  $\{w', b'\}$  based on two sub-networks (i.e., Step 1 in Figure 3);
- Fine tuning—Obtain the initial prediction  $\{\alpha', v'\}$  based on a multi-target sigmoid regression layer, and utilize a back propagation to fine tune the final prediction model (i.e., Step 2 in Figure 3);
- Prediction—Predict driving behavior and output  $\{\alpha, v\}$  for the ego vehicle.

In addition, the pre-training procedure can produce the initial model parameters by the unsupervised learning, based on which the fine-tuning procedure operates the unsupervised learning to obtain the initially predicted behaviors. Then the supervised learning method is used to predict the behaviors. Note that, the pre-training trains the initial model parameters in an unsupervised method without given driving behavior labels, while the fine tuning adjusts and updates the parameters in a supervised method based on labeled behaviors. In what follows, we will detail the training and tuning procedures.

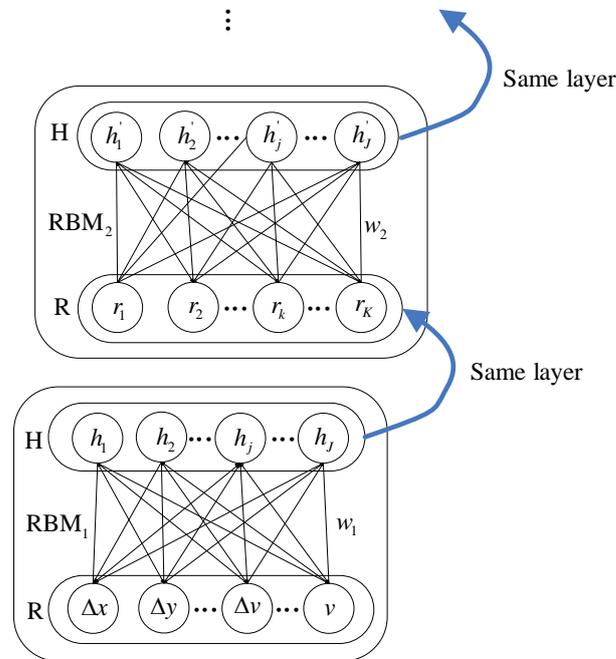
### 2.2.1. Pre-Training Procedure

Based on a stack of RBMs, pre-training is to initialize parameters through unsupervised greedy algorithms. Usually, the RBM is composed of a visible layer (denoted as R) and a hidden layer (denoted as H), as shown in Figure 4.  $RBM_1$  is the first RBM, and nodes  $\Delta x, \Delta y, \Delta v, v$  represent the visible nodes of  $RBM_1$ . The nodes between layers are interconnected while those in the same layer are disconnected with each other [41,50].

Define  $P(r, h|\theta)$  as the probability distribution of a single RBM, then the energy function of the RBM is given as

$$\begin{aligned} E(r, h|\theta) &= \ln P(r, h|\theta) \\ &= - \sum_{k=1}^K a_k r_k - \sum_{j=1}^J b_j h_j - \sum_{k=1}^K \sum_{j=1}^J w_{kj} r_k h_j \end{aligned} \quad (2)$$

where  $\theta = (w_{kj}, a_k, b_j)$  are the model parameters,  $w_{kj}$  is the weight from the  $k$ -th visible node  $r_k$  to the  $j$ -th hidden node  $h_j$ ;  $a_k$  and  $b_j$  are the biases of  $r_k$  and  $h_j$  in a single RBM, respectively. For the input layer in Figure 3, the input units including  $I_E$ ,  $I_D$ , and  $I_S$  correspond to the visible nodes  $r_k$  of the R layer for  $\text{RBM}_1$ , as shown in Figure 4 and Equation (2).



**Figure 4.** Schematic diagram for the RBM and training process for the pre-training.

Given the state of the visible unit  $r$  and the state of the hidden unit  $h$ , the activation probabilities of  $h_j$  and  $r_k$  are written as

$$\begin{aligned} P(h_j|r, \theta) &= \text{sigm}(b_j + \sum_{k=1}^K r_k w_{kj}) \\ P(r_k|h, \theta) &= \text{sigm}(a_k + \sum_{j=1}^J h_j w_{kj}) \end{aligned} \quad (3)$$

with the sigmoid function

$$\text{sigm}(z) = 1/(1 + \exp(-z)) \quad (4)$$

We update the parameter  $\theta$  using the contrastive divergence algorithm [45,51]. The update rules for weight  $w_{kj}$  and biases  $a_k$  and  $b_j$  are given as

$$\begin{aligned} \Delta w_{kj} &= \eta \cdot (\mathbb{E}(r_k h_j)_{\text{data}} - \mathbb{E}(r_k h_j)_{\text{model}}) \\ \Delta a_k &= \eta \cdot (\mathbb{E}(r_k)_{\text{data}} - \mathbb{E}(r_k)_{\text{model}}) \\ \Delta b_j &= \eta \cdot (\mathbb{E}(h_j)_{\text{data}} - \mathbb{E}(h_j)_{\text{model}}) \end{aligned} \quad (5)$$

where  $\eta$  is the learning rate for pre-training procedure which varies from zero to one,  $\mathbb{E}(\cdot)_{\text{data}}$  and  $\mathbb{E}(\cdot)_{\text{model}}$  represent the expectations of the training data and the distribution of the model, respectively.

In the DBN, the output of an RBM is the input of the next RBM, that is, the adjacent RBMs have a shared layer with the same nodes, as shown in Figure 4. The pre-training process is achieved via Step 1 in Figure 3. All the RBMs are trained one by one. Specifically, the RBMs are trained from the first RBM to the last one, and once an RBM is trained, its output will be regarded as the input of the next RBM. Thus, we can initialize the model parameters  $\{w', b'\}$  for the fine-tuning procedure. Then, the output of the last RBM is fed as the input of the sigmoid regression (Figures 2 and 3).

### 2.2.2. Fine-Tuning Procedure

The fine-tuning procedure is a supervised part based on BP network, and Step 2 in Figure 3 explains the training process. Once obtaining the initial weights and biases for the prediction model after training RBMs one by one, the fine tuning adjusts model parameters to obtain the final prediction model.

Specifically, after obtaining the initial parameters  $\theta' = (w'_{kj}, a'_i, b'_j)$  through the pre-training procedure, the network makes use of the forward propagation algorithm to compute the output of the hidden units. Then, a multi-target sigmoid regression layer is applied as shown in Figure 2 to predict the initial output  $O'_p$  (i.e., initial front wheel angle and speed) [41]. The initial output  $O'_p = \{O'_{p_\alpha}, O'_{p_v}\}$  of the multi-target sigmoid regression layer is defined as

$$\begin{aligned} O'_{p_\alpha} &= \text{sigm}(b'_\alpha + O'_{p_{\alpha-1}} w'_\alpha) \\ O'_{p_v} &= \text{sigm}(b'_v + O'_{p_{v-1}} w'_v) \end{aligned} \quad (6)$$

where  $\{b'_\alpha, w'_\alpha\}$  and  $\{b'_v, w'_v\}$  are the initial biases and weights for  $\alpha'$  (i.e.,  $O'_{p_\alpha}$ ) and  $v'$  (i.e.,  $O'_{p_v}$ ), respectively.  $O'_p$  is the output for the output layer of Step 1 in Figure 3, and  $O'_{p_{\alpha-1}}$  and  $O'_{p_{v-1}}$  are the initial outputs of Layer #3 and Layer #4 for front wheel angle and speed, as shown in Figure 3.

In order to optimize the prediction model, the BP network uses a back propagation algorithm to fine tune the parameters  $\theta'$ . We need to calculate the sensitivity  $\sigma$  of layers to modify the parameter of the model from the top layer to the bottom layer. For the predictor layer, the predicted output for the  $k$ -th node is assumed to  $O'_{p_k}$ , and the actual expected output is  $O'_{ak}$ . The sensitivity  $\sigma$  can be computed by

$$\sigma_k = O'_{p_k}(1 - O'_{p_k})(O'_{ak} - O'_{p_k}) \quad (7)$$

For the  $l$ -th hidden layer,  $\sigma$  can be written as

$$\sigma_k^l = O'_{p_k}{}^l(1 - O'_{p_k}{}^l) \sum_j w'_{kj} \sigma_j^{l+1} \quad (8)$$

Then,  $\theta'$  can be adjusted to  $\theta$  and back to each layer. Thus, the model parameters can be updated and adjusted. Similar to Equation (6), the final output  $O_p$ , the prediction result of the model, is as below

$$O_p = \{\alpha_{t+1}, v_{t+1}\} \quad (9)$$

where  $\alpha_{t+1}$  and  $v_{t+1}$  are the final front wheel angle and speed of the ego vehicle at the future moment, respectively.

## 3. Data Collection and Data Processing

### 3.1. Data Collection

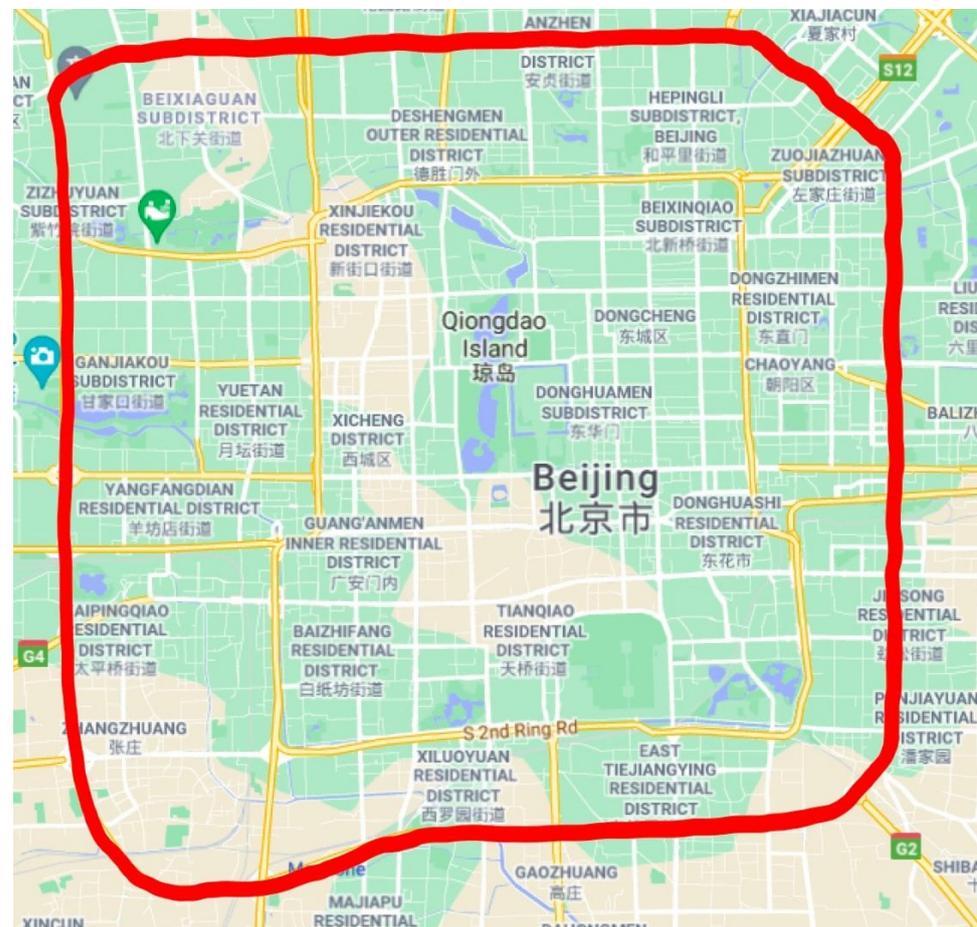
#### 3.1.1. Data Acquisition

We collected the real-vehicle data by BYD Surui, an autonomous vehicle of Intelligent Vehicle Research Center of Beijing Institute of Technology (BIT). The data acquisition platform was equipped with on-board sensors to collect vehicle and traffic data. An OxTs integrated navigation system was used to collect GPS information, such as latitude/longitude, heading angle, and time stamp of data. Vehicle CAN-bus provided throttle pedal position, brake pressure, front wheel angle, vehicle speed, and time stamp of data. Radar detected both mid-range and long-range obstacles and provided positions of states of surrounding vehicles. Additionally, a camera was used to collect the scene image information. Table 1 shows some important information of data collection.

**Table 1.** Information of data collection.

Name	Parameter	Units
power unit	combustion engine with automatic transmission	-
travel distance	48	km
travel time	0.95	h
average velocity	50.27	km/h
collection time	3 to 3.57 (off-peak)	pm

Our vehicle was driven by an experienced human driver and interacted with surrounding vehicles in the Third Ring Road in Beijing, and the route is shown by the red curve in Figure 5. We focus on the naturalistic driving scene in the straight road rather than complex scenarios, such as turning scenario and intersection scenario. Specifically, we excluded the data under complex scenarios based on the scene images captured by the camera and selected the data. Finally, 20,000 samples were selected for the prediction.

**Figure 5.** Data acquisition route.

### 3.1.2. Data Preprocessing

We focus on the surrounding vehicles that have significant effects on the ego vehicle and extract the data of three surrounding vehicles including the left front vehicle, the front vehicle, and the right front vehicle. Based on the collected data of 64 target points by radar, we used dynamic target analysis and the geometric relationship, such as relative position and relative speed to calculate and extract the states of the surrounding vehicles. In addition, we extracted the historical speed ( $v$ ) and front wheel angle ( $\alpha$ ) of the ego vehicle and the relative position ( $\Delta x$  and  $\Delta y$ ) and speed ( $\Delta v$ ) of surrounding vehicles to train our prediction model. The historical front wheel angle of the ego vehicle are treated

as the control state of the human driver. Thus, we integrated and preprocessed the driver's operations, the states of the ego vehicle, and the states of surrounding vehicles. After obtaining the filtered data, we normalized them into [0, 1] to satisfy the input requirement of DBN, and then divided them into two groups as training and testing data.

### 3.2. Performance Evaluation

Three kinds of matrices are used to evaluate the performance of the prediction model by measuring the errors of prediction results [24,52–54], including root mean square error (RMSE),

$$\text{RMSE} = \left[ \frac{1}{N} \sum_{n=1}^N (O_a - O_p)^2 \right]^{1/2} \quad (10)$$

mean absolute error (MAE),

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^N |O_a - O_p| \quad (11)$$

and mean relative error (MRE),

$$\text{MRE} = \frac{1}{N} \sum_{n=1}^N \left| \frac{O_a - O_p}{O_a} \right| \quad (12)$$

where  $N$  represents the sample number,  $O_a$  represents the observed actual behavior,  $O_p$  represents the predicted behavior. Here, we do not use MRE to evaluate the prediction performance for the front wheel angle since the front wheel angle may be equal to 0 or close to 0, thus leading to a positive infinite value of MRE.

### 3.3. Model Input Selection

As mentioned above, this paper is based on a series of historical data and current states of the ego vehicle and surrounding vehicles to predict the driving behavior of the ego vehicle [55]. More than 20,000 samples are used to train and test our prediction model (the training data account for approximately 85%). The input includes the historical data of the ego vehicle, driver operation, and surrounding vehicles, as described in Equation (1). In urban complex traffic, the surrounding vehicle behaviors would affect the state of the ego vehicle since they need to interact with each other. Additionally, the future states of the surrounding vehicles will influence the decision-making of the ego vehicle. Therefore, we take the states of surrounding vehicles at the next moment as one of input to improve the prediction accuracy by simulating real scenarios. However, the future behaviors of surrounding vehicles are unknown when predicting the next state of the ego vehicle. Therefore, this section will firstly predict the driving states of surrounding vehicles.

We will explore different time-series inputs to find the appropriate one for the prediction model. A series of experiments demonstrate that the errors are relatively low when the learning rate  $\eta$  is 0.7, and the number of hidden nodes is 100, which will be introduced explicitly in Section 4. In this section, the learning rate and hidden nodes are similarly fixed (0.7 and 100, respectively) to show our experiment design clearly.

#### 3.3.1. State Prediction of Surrounding Vehicles

To simulate the real situation, we will predict the states of surrounding vehicles for the forthcoming moment. For the surrounding vehicles, the rear vehicles drive normally and have little influences on the ego vehicle which can be neglected, and we only consider the effects of front vehicles. Since increasing the number of layers of DBN would significantly increase the training time, we choose DBN with one hidden layer to predict the states of the surrounding vehicles.

Based on different historical data of surrounding vehicles, we can compute the errors of  $\Delta x$ ,  $\Delta y$ , and  $\Delta v$  (i.e.,  $\Delta x_{\text{RMSE}}$ ,  $\Delta x_{\text{MAE}}$ ,  $\Delta y_{\text{RMSE}}$ ,  $\Delta y_{\text{MAE}}$ ,  $\Delta v_{\text{RMSE}}$ , and  $\Delta v_{\text{MAE}}$ ) for

surrounding vehicles including the left front vehicle, front vehicle, and right front vehicle, as shown in Table 2. Five cases (Case 00, Case 01, Case 02, Case 03, and Case 04) represent different historical data for previous 0 s, 1 s, 2 s, 3 s, and 4 s, respectively. In Table 2, the minimum errors for  $\Delta x_{RMSE}$ ,  $\Delta x_{MAE}$ ,  $\Delta y_{RMSE}$ ,  $\Delta y_{MAE}$ ,  $\Delta v_{RMSE}$ , and  $\Delta v_{MAE}$  among three surrounding vehicles are in red; the second-minimum errors are in blue; the value in red in the type column is the selected type for our prediction model.

Table 2 indicates that Case 01 obtains the lowest  $\Delta v_{RMSE}$  and  $\Delta v_{MAE}$  for three surrounding vehicles among all types, and the second minimum for the majority  $\Delta y_{RMSE}$  and  $\Delta y_{MAE}$ . Even though some  $\Delta x_{RMSE}$  and  $\Delta x_{MAE}$  in Case 01 do not obtain the lowest errors among all the five types, the relative lateral positions have little effects on the ego vehicle and all the vehicles driving regularly and safely on their own lanes, so we select Case 01 as the optimal one to predict surrounding vehicles in this paper. Thus, we choose Case 01 (i.e., the historical data 1 s ahead) to predict the states of surrounding vehicles at the next moment.

**Table 2.** Prediction errors for surrounding vehicles based on DBN.

Type	Surrounding Vehicles \ Errors	$\Delta x_{RMSE}$	$\Delta x_{MAE}$	$\Delta y_{RMSE}$	$\Delta y_{MAE}$	$\Delta v_{RMSE}$	$\Delta v_{MAE}$
Case 00	Left front vehicle	0.1793	0.1223	2.6915	2.2376	1.3976	0.8935
	Front vehicle	0.2865	0.2139	2.3622	1.6879	1.2073	0.8054
	Right front vehicle	0.2875	0.2293	3.5747	1.9832	1.5416	1.1863
Case 01	Left front vehicle	0.4094	0.3794	3.9323	2.5273	0.9838	0.7563
	Front vehicle	0.1396	0.1036	3.6175	3.0658	0.5568	0.4278
	Right front vehicle	0.4806	0.4685	2.5734	2.0658	0.8483	0.6464
Case 02	Left front vehicle	0.2568	0.1527	3.6497	2.5632	1.5037	1.0070
	Front vehicle	0.2391	0.1758	9.7533	6.8699	1.3156	0.9809
	Right front vehicle	0.2899	0.2175	9.8397	5.5563	1.8377	1.2637
Case 03	Left front vehicle	0.1814	0.1068	6.2141	4.4135	1.8260	1.6235
	Front vehicle	0.1909	0.1352	5.1570	3.1565	2.0950	1.7989
	Right front vehicle	0.1901	0.1217	3.8158	2.2171	1.5692	1.3391
Case 04	Left front vehicle	0.2912	0.1947	11.3866	9.5590	2.0985	1.4397
	Front vehicle	0.2587	0.2040	9.0340	8.0025	1.5723	1.2682
	Right front vehicle	0.2613	0.1882	5.4806	3.4477	2.8110	2.4085

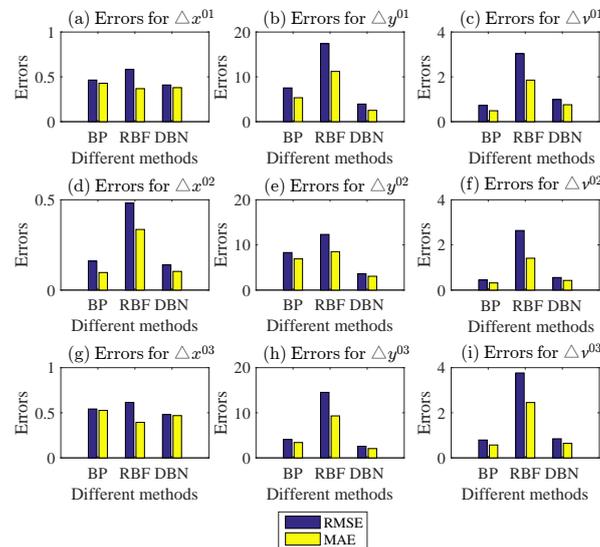
Figure 6 presents the comparison results with BP and RBF to illustrate the prediction accuracy for surrounding vehicles based on DBN, where superscripts 01, 02, and 03 represent the relative states of the left front vehicle ( $vehicle_1$ ), front vehicle ( $vehicle_2$ ) and right front vehicle ( $vehicle_3$ ) with respect to the ego vehicle ( $vehicle_0$ ), associated with Table 2. Most RMSE and MAE values of  $\Delta x$ ,  $\Delta y$  and  $\Delta v$  for three surrounding vehicles are slightly low, compared to those of BP and RBF. Thus, we utilize the prediction results of the surrounding vehicles to improve the driving state prediction for the ego vehicle.

### 3.3.2. Different Historical Data

To obtain an optimal prediction performance of driving behavior for the ego vehicle, we test different inputs with different historical data of the ego vehicle and surrounding vehicles. Table 3 shows the prediction errors of  $\alpha$  and  $v$  of DBN with one RBM (0.7 learning rate and 100 hidden layer nodes), where Case 0, Case 1, Case 2, Case 3, and Case 4 represent  $N = 0$ ,  $N = 1$ ,  $N = 2$ ,  $N = 3$ , and  $N = 4$  in Equation (1), respectively. In order to describe the results efficiently, we term the prediction errors of the front wheel angle as  $\alpha_{RMSE}$  and  $\alpha_{MAE}$ . Additionally, the RMSE, MAE, and MRE of the speed are abbreviated as  $v_{RMSE}$ ,  $v_{MAE}$ , and  $v_{MRE}$ .

Table 3 shows that Case 0 obtains the lowest  $v_{RMSE}$ ,  $v_{MAE}$  and  $v_{MRE}$  while Case 4 and 2 obtain the lowest  $\alpha_{RMSE}$ , and  $\alpha_{MAE}$ . The errors of speed are relatively higher than those of front wheel angle, so we consider the errors of speed in the first place. Even though Case 0 obtain the lowest value of the errors of speed, it obtains a more significant error of front wheel angle, compared to Case 4. Thus, we choose Case 1 since most of the errors of speed and front wheel angle are acceptable. The rest of this paper will regard the current

states and historical states 1 s ahead of surrounding vehicles and the ego vehicle and the predicted states of surrounding vehicles at next moment as input to predict the driving behavior of the ego vehicle.



**Figure 6.** Prediction errors for surrounding vehicles based on different methods.

**Table 3.** Prediction errors for ego vehicle based on DBN with one RBM.

Type \ Errors	$\alpha$ RMSE	$\alpha$ MAE	$v$ RMSE	$v$ MAE	$v$ MRE
Case 0	0.1935	0.1874	0.3055	0.2364	0.0172
Case 1	0.0946	0.0514	0.4668	0.3920	0.0654
Case 2	0.0323	0.0212	0.6665	0.5054	0.0317
Case 3	0.0617	0.0516	0.5971	0.4851	0.0395
Case 4	0.0279	0.0228	1.4551	1.2334	0.0968

## 4. Results

### 4.1. Experiment of DBN Structure

Previous studies have obtained some prediction models based on DBN by determining different parameters, such as the prediction model for traffic flow [41]; however, it is still unknown which kind of structure is appropriate for driving behavior prediction, and there is not a general method to guide the model design. To obtain an optimal DBN structure for prediction, we explore a systematic testing method and train the model using the cut-and-try method which is similar to the training method used by [54]. Specifically, the performance of one parameter of the model is investigated by keeping other parameters fixed. The parameters of our prediction model mainly include the learning rate, the number of hidden layers, and the number of nodes in hidden layers [56,57]. The prediction model has 31 input nodes and 2 output nodes as described above.

#### 4.1.1. Learning Rates

The number of hidden layers and hidden nodes are determined by firstly testing different learning rates. In this subsection, we set the DBN structure with one hidden layer (i.e., one RBM) consisting of 100 hidden nodes. The prediction results of our model for different learning rates are obtained based on training data, as shown in Table 4. Both  $\alpha$ MAE (0.0939) and  $\alpha$ RMSE (0.0493) reach the best performance at the learning rate 0.5, and  $v$ RMSE,  $v$ MAE, and  $v$ MRE obtain the best performance with the learning rate 0.9, respectively. However, a large learning rate would cause a poor prediction performance because of over-fitting [51]. Additionally, Table 4 demonstrates that the error of speed is larger than that of the front wheel angle. Thus, we chose the learning rate as 0.7 since all the errors are close to their associated lowest ones.

**Table 4.** Prediction results of different learning rates for front wheel angle and speed.

Learning Rate	$\alpha_{RMSE}$	$\alpha_{MAE}$	$v_{RMSE}$	$v_{MAE}$	$v_{MRE}$
0.1	1.0406	1.0118	15.1516	15.0381	2.5666
0.3	0.1029	0.0547	0.5516	0.4731	0.0814
0.5	<b>0.0939</b>	<b>0.0493</b>	0.5019	0.4335	0.0741
<b>0.7</b>	0.0946	0.0514	0.4668	0.3920	0.0654
0.9	0.0942	0.0494	<b>0.4166</b>	<b>0.3381</b>	<b>0.0571</b>

#### 4.1.2. Hidden Layers

The number of hidden layers is explored by fixing the number of hidden nodes in each hidden layer to 100. Table 5 shows the errors based on different hidden layers. The limited training data size causes an unsatisfied prediction performance of complex structure with over 3 layers for front wheel angle and speed, which may be caused by under-fitting. When the number of RBM is 3, all the prediction errors of speed ( $v_{RMSE}$  of 0.3858,  $v_{MAE}$  of 0.2965, and  $v_{MRE}$  of 0.0531) are the lowest values, i.e., it achieves the best performance.  $\alpha_{RMSE}$  and  $\alpha_{MAE}$  are the lowest values when setting the hidden layers as two and one, but larger errors for speed than those of three layers. For the three-hidden-layer structure,  $\alpha_{RMSE} = 0.1431$  is slightly larger than the average error of 0.1290, but  $\alpha_{MAE} = 0.0811$  is lower than the average value of 0.0862. In addition, this structure performs the best for speed as mentioned above. Therefore, we choose the hidden layer as 3.

**Table 5.** Prediction results of different hidden layers for front wheel angle and speed.

Number of RBM	$\alpha_{RMSE}$	$\alpha_{MAE}$	$v_{RMSE}$	$v_{MAE}$	$v_{MRE}$
1	0.0946	<b>0.0514</b>	0.4668	0.3920	0.0654
2	<b>0.0900</b>	0.0586	0.5763	0.4576	0.0878
<b>3</b>	0.1431	0.0811	<b>0.3858</b>	<b>0.2965</b>	<b>0.0531</b>
4	0.1171	0.0906	1.8569	1.5595	0.2867
5	0.1866	0.1310	1.5647	1.2909	0.2387
6	0.1427	0.1044	1.7684	1.4769	0.2716
Average	0.1290	0.0862	1.1032	0.9122	0.1672

#### 4.1.3. Hidden Nodes

Based on the analysis above, we test the model with different hidden nodes when the learning rate is 0.7, and the hidden layer is 3. The number of hidden nodes is set to 32, 50, 64, 100, 128, 150, 200, and 256, respectively. Table 6 shows the test results and indicates that for the front wheel angle, fewer hidden nodes in each hidden layer can achieve better performance. Specifically,  $\alpha_{RMSE}$  and  $\alpha_{MAE}$  obtain the lowest errors for the front wheel angle when the hidden nodes are [32, 32, 32] and [64, 64, 64]. For the speed, the structure with 100 hidden nodes in each layer is the best choice. Although  $\alpha_{RMSE}$  and  $\alpha_{MAE}$  are 0.1431 and 0.0811 with 100 hidden nodes, and are slightly larger than the lowest value, they are still lower than the average ones. However, inappropriate hidden nodes (e.g., 32 and 200) for the speed would lead to an unsatisfied performance because too few nodes are insufficient to extract representative features and too many nodes would lead to extra burden to the training model. Therefore, we set 100 nodes in each hidden layer.

Finally, we can obtain a three-hidden-layer DBN prediction model with 100 hidden nodes and the learning rate of 0.7, denoted as  $DBN_3^{100}$ . The subscript represents the number of hidden layers and superscript represents the number of nodes in each hidden layer.

**Table 6.** Prediction results of different hidden nodes in a layer.

Hidden Nodes	$\alpha$ RMSE	$\alpha$ MAE	$v$ RMSE	$v$ MAE	$v$ MRE
32	0.1160	0.0766	1.8241	1.5283	0.2829
50	0.1279	0.0768	1.6475	1.3799	0.2543
64	0.1212	0.0717	1.0640	0.8687	0.1592
100	0.1431	0.0811	0.3858	0.2965	0.0531
128	0.2090	0.1555	1.2237	0.9832	0.1830
150	0.1426	0.0778	0.5932	0.4585	0.0778
200	0.1751	0.1260	1.5396	1.2659	0.2316
256	0.1614	0.1093	1.1938	0.9601	0.1720
Average	0.1495	0.0968	1.1840	0.9676	0.1767

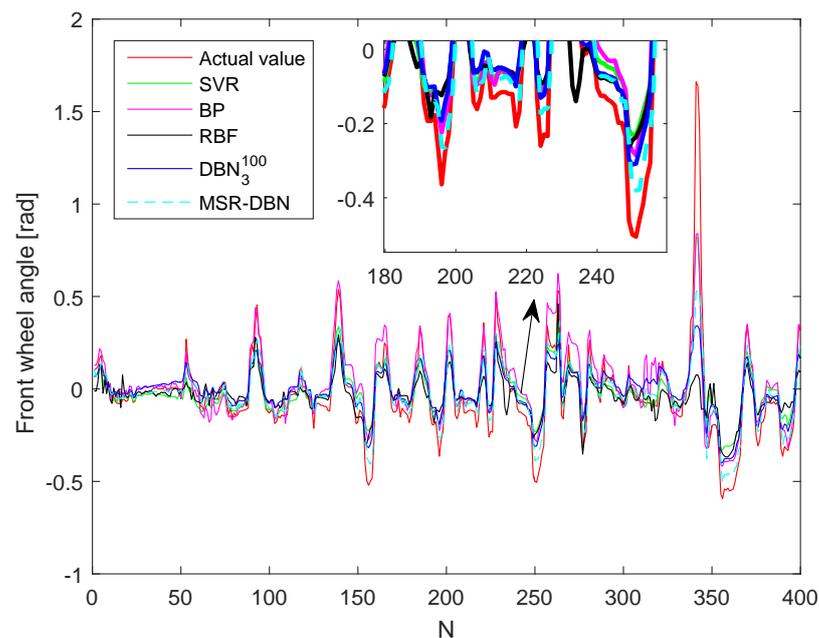
#### 4.2. Structure of MSR-DBN

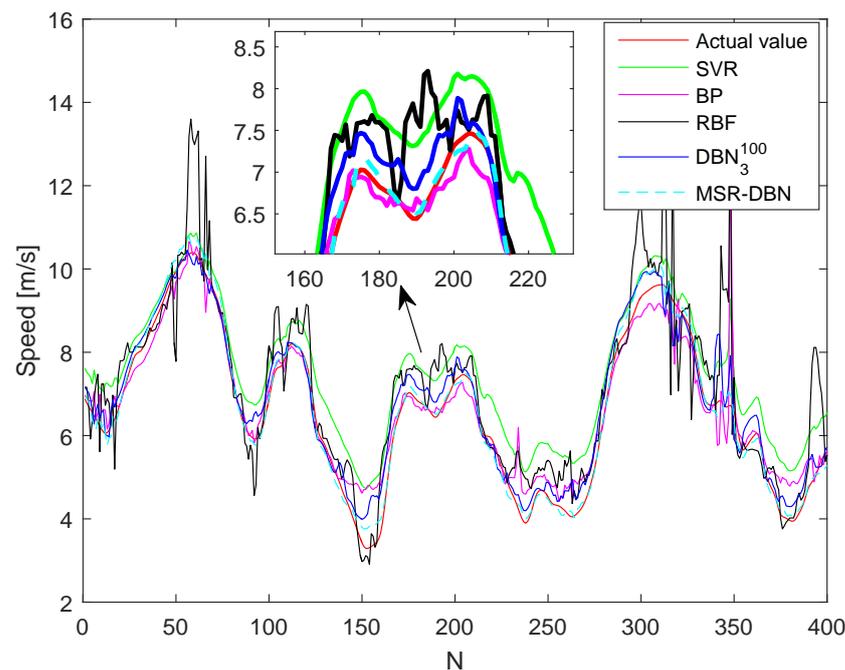
Based on the obtained  $DBN_3^{100}$ , we should optimize the prediction performance further. Tables 5 and 6 show that the  $DBN_3^{100}$  performs best for speed, but not for the front wheel angle. In addition, the two-hidden-layer structure with 100 nodes in Table 5 achieves much better performance for the front wheel angle, which is even better than the three-hidden-layer model with 32 and 64 nodes.

Considering both front wheel angle and speed, we modify the prediction model  $DBN_3^{100}$  and proposes a novel MSR-DBN prediction model which is a four-five-layer prediction model for the driving behavior of the ego vehicle as shown in Figure 3. Thus, the MSR-DBN consists of two sub-systems: a two-hidden-layers sub-network for front wheel angle and a three-hidden-layers sub-network for speed.

#### 4.3. Result Analysis and Comparison

To demonstrate the performance of our improved MSR-DBN for driving behavior prediction, we analyze and compare the prediction results with  $DBN_3^{100}$ , SVR, BP, RBF, and the actual values [24,44,54]. Figures 7 and 8 show the prediction results of the front wheel angle and speed. We can see that these five prediction results have the same tendency with the actual value, indicating that all these five methods can predict driving behaviors for intelligent vehicles.

**Figure 7.** Prediction results of the front wheel angle based on different methods.



**Figure 8.** Prediction results of the speed based on different methods.

Figure 7 shows that SVR, BP, and RBF could achieve a relatively close performance. The prediction results based on  $\text{DBN}_3^{100}$  are closer to the actual values than SVR, BP, and RBF in the light of its deep architecture. However, the performance of MSR-DBN outperforms all of these four methods. For example, in the 250-th test sample, the MSR-DBN obtains the performance ( $-0.38$  rad) that is the closest one to the actual value ( $-0.54$  rad) than the others.

However, different results occur for speed, as shown in Figure 8. The RBF obtains the worst prediction performance, which even appears many big jumps for speed (such as the test samples 50 and 350). In addition, BP has more fluctuations while it obtains a satisfying performance for speed. However, the DBN achieves a better prediction performance, compared with RBF and SVR. Take the 190-th sample for example, SVR cannot obtain a prediction performance of speed as good as the front wheel angle, and the prediction result is approximately 7.35 m/s which is higher than the actual speed. The predictions of  $\text{DBN}_3^{100}$  and MSR-DBN (6.85 m/s and 6.5 m/s, respectively) are slightly larger than the actual one (6.45 m/s). Therefore, the MSR-DBN is more robust, compared with the other four models.

We also compare the prediction errors of MSR-DBN with the other four models to further illustrate the prediction performance, as shown in Table 7. Specifically, the RBF model obtains higher  $\alpha_{\text{RMSE}}$  and  $\alpha_{\text{MAE}}$  than SVR and BP. The  $\text{DBN}_3^{100}$  model obtains a larger error of front wheel angle, compared with other three models. However, the proposed MSR-DBN outperforms  $\text{DBN}_3^{100}$  with  $\alpha_{\text{RMSE}}$  (0.1165) and  $\alpha_{\text{MAE}}$  (0.0593), which is consistent with the results in Figures 7 and 8.

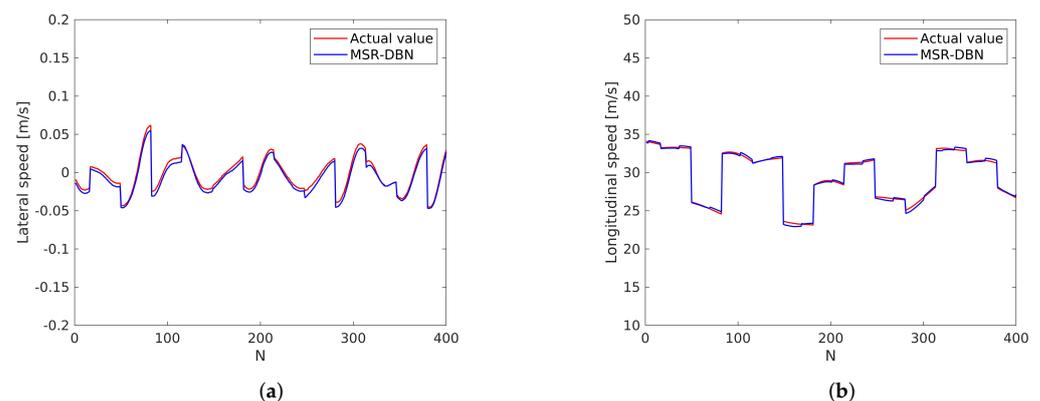
For the prediction of speed, the SVR obtains the worst performance with  $v_{\text{RMSE}} = 0.9142$ ,  $v_{\text{MAE}} = 0.8447$ , and  $v_{\text{MRE}} = 0.1541$ , followed by RBF and BP. Compared to SVR, RBF, and BP,  $\text{DBN}_3^{100}$  obtains a smaller error, and MSR-DBN outperforms  $\text{DBN}_3^{100}$ . Thus, compared to  $\text{DBN}_3^{100}$ , the MSR-DBN can not only improve the prediction performance of front wheel angle, but reduce the errors of speed, which shows a strong capability of predicting driving behavior.

**Table 7.** Error comparison for front wheel angle and speed based on different models.

Methods \ Errors	$\alpha$ RMSE	$\alpha$ MAE	$v$ RMSE	$v$ MAE	$v$ MRE
SVR	0.1256	0.0813	0.9142	0.8447	0.1541
BP	0.1277	0.0860	0.5799	0.3684	0.0695
RBF	0.1806	0.1009	1.0403	0.6894	0.1090
DBN <sub>3</sub> <sup>100</sup>	0.1431	0.3858	0.0811	0.2965	0.0531
MSR-DBN	0.1165	0.0593	0.2067	0.1626	0.0259

#### 4.4. Generalization Performance Analysis

In order to analysis the generalization performance of the MSR-DBN, test in different dataset is necessary. The highD dataset is a new dataset used by many researchers, it uses a drone to record the naturalistic vehicle trajectories on German [58]. It can provide higher-precision trajectory information by using state-of-the-art computer vision algorithms. Different from the urban road scenario data collected by ourselves, highD dataset is collected from the highway scenario. We can extract the position, speed and other information of the ego vehicle and surrounding vehicles from the dataset, but the steering wheel angle information is not provided. We choose the longitudinal velocity of x-axis as the speed, and the lateral velocity of y-axis to present the steering wheel angle. As shown in Figure 9, the MSR-DBN still has a good prediction performance.



**Figure 9.** Prediction results on highD dataset. (a) Prediction result of lateral speed. (b) Prediction result of longitudinal speed.

## 5. Discussion

This paper presented a driving behavior prediction system with four sub-systems, data acquisition, data preprocessing, DBN prediction, and result analysis. This system can utilize multi-resource data including states of the ego vehicle, states of driver control, and states of surrounding vehicles. In addition, we used a systematic testing method obtaining an optimal DBN structure DBN<sub>3</sub><sup>100</sup> and then developed a new DBN (called MSR-DBN, consisting of two sub-networks) by integrating a multi-target sigmoid regression to predict the longitudinal and lateral driving behavior simultaneously. A series of comparison results demonstrate that our proposed MSR-DBN can reduce the prediction errors of the speed and front wheel angle with more stable performance and higher accuracy for driving behavior.

Although our proposed MSR-DBN shows promising results, there still exist some work to improve further prediction performance, such as increasing more data. Additionally, more complex predictions will be conducted in the near future such as real-time behavior prediction, specific behavior prediction in different scenarios, and interaction behavior prediction.

**Author Contributions:** Conceptualization, C.Z. and C.L.; methodology, C.Z. and C.L.; software, L.Y. and C.Z.; validation, L.Y. and C.Z.; writing—original draft preparation, L.Y. and C.Z.; writing—review and editing, C.L., L.Y., C.Z., L.W. and J.G.; supervision, J.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported in part by the National Natural Science Foundation of China under Grant U19A2083.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Olabiyi, O.; Martinson, E.; Chintalapudi, V.; Guo, R. Driver Action Prediction Using Deep (Bidirectional) Recurrent Neural Network. *arXiv* **2017**, arXiv:1706.02257.
2. Sun, L.; Yin, Y. Discovering themes and trends in transportation research using topic modeling. *Transp. Res. Part C Emerg. Technol.* **2017**, *77*, 49–66. [[CrossRef](#)]
3. Meyer, J.; Becker, H.; Bösch, P.M.; Axhausen, K.W. Autonomous vehicles: The next jump in accessibilities? *Res. Transp. Econ.* **2017**, *62*, 80–91. [[CrossRef](#)]
4. Lu, C.; Wang, H.; Lv, C.; Gong, J.; Xi, J.; Cao, D. Learning Driver-specific Behaviour for Overtaking: A Combined Learning Framework. *IEEE Trans. Veh. Technol.* **2018**, *67*, 6788–6802.
5. Schwehr, J.; Willert, V. Driver's gaze prediction in dynamic automotive scenes. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–8.
6. 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.
7. Li, X.; Wang, W.; Roetting, M. Estimating Driver's Lane-Change Intent Considering Driving Style and Contextual Traffic. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 3258–3271. [[CrossRef](#)]
8. Wang, W.; Xi, J.; Zhao, D. Learning and inferring a driver's braking action in car-following scenarios. *IEEE Trans. Veh. Technol.* **2018**, *67*, 3887–3899. [[CrossRef](#)]
9. Bocklisch, F.; Bocklisch, S.F.; Beggato, M.; Krems, J.F. Adaptive fuzzy pattern classification for the online detection of driver lane change intention. *Neurocomputing* **2017**, *262*, 148–158. [[CrossRef](#)]
10. Liu, Q.; Xu, S.; Lu, C.; Yao, H.; Chen, H. Early Recognition of Driving Intention for Lane Change Based on Recurrent Hidden Semi-Markov Model. *IEEE Trans. Veh. Technol.* **2020**, *69*, 10545–10557. [[CrossRef](#)]
11. Maye, J.; Triebel, R.; Spinello, L.; Siegart, R. Bayesian on-line learning of driving behaviors. In Proceedings of the Robotics and Automation (ICRA), 2011 IEEE International Conference, Shanghai, China, 9–13 May 2011; pp. 4341–4346.
12. Sun, L.; Chen, X.; He, Z.; Miranda-Moreno, L.F. Pattern Discovery and Anomaly Detection of Individual Travel Behavior Using License Plate Recognition Data. In Proceedings of the Transportation Research Board 98th Annual Meeting, Washington, DC, USA, 13–17 January 2019.
13. Dou, Y.; Yan, F.; Feng, D. Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers. In Proceedings of the 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, Canada, 12–15 July 2016; pp. 901–906.
14. Wang, W.; Zhao, D.; Han, W.; Xi, J. A learning-based approach for lane departure warning systems with a personalized driver model. *IEEE Trans. Veh. Technol.* **2018**, *67*, 9145–9157. [[CrossRef](#)]
15. Kumagai, T.; Akamatsu, M. Prediction of human driving behavior using dynamic Bayesian networks. *IEICE Trans. Inf. Syst.* **2006**, *89*, 857–860. [[CrossRef](#)]
16. Li, J.; Dai, B.; Li, X.; Xu, X.; Liu, D. A dynamic bayesian network for vehicle maneuver prediction in highway driving scenarios: Framework and verification. *Electronics* **2019**, *8*, 40. [[CrossRef](#)]
17. Yuan, W.; Li, Z.; Wang, C. Lane-change prediction method for adaptive cruise control system with hidden Markov model. *Adv. Mech. Eng.* **2018**, *10*. [[CrossRef](#)]
18. Peng, J.; Guo, Y.; Fu, R.; Yuan, W.; Wang, C. Multi-parameter prediction of drivers' lane-changing behaviour with neural network model. *Appl. Ergon.* **2015**, *50*, 207–217. [[CrossRef](#)] [[PubMed](#)]
19. Chen, X.; Chi, R.; Wang, J.; Pang, C.; Xiao, Q. Driver safe speed model based on BP neural network for rural curved roads. In Proceedings of the International Conference on Computer and Computing Technologies in Agriculture, Beijing, China, 29–31 October 2011; pp. 92–102.
20. Leonhardt, V.; Wanielik, G. Neural network for lane change prediction assessing driving situation, driver behavior and vehicle movement. In Proceedings of the Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference, Yokohama, Japan, 16–19 October 2017; pp. 1–6.
21. Yang, S.; Wang, W.; Liu, C.; Deng, W. Scene Understanding in Deep Learning-Based End-to-End Controllers for Autonomous Vehicles. *IEEE Trans. Syst. Man, Cybern. Syst.* **2019**, *49*, 53–63. [[CrossRef](#)]

22. Wang, L.; Liu, X.m.; Zhong, X.m. A car-following model based on artificial neural networks in urban expressway sections. In Proceedings of the 85th TRB Annual Meeting, Transportation Research Board, National Research Council, Washington, DC, USA, 22–26 January 2006.
23. Dogan, Ü.; Edelbrunner, J.; Iossifidis, I. Autonomous driving: A comparison of machine learning techniques by means of the prediction of lane change behavior. In Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics, Karon Beach, Thailand, 7–11 December 2011; pp. 1837–1843.
24. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873. [[CrossRef](#)]
25. Polson, N.G.; Sokolov, V.O. Deep learning for short-term traffic flow prediction. *Transp. Res. Part C Emerg. Technol.* **2017**, *79*, 1–17. [[CrossRef](#)]
26. Lemieux, J.; Ma, Y. Vehicle speed prediction using deep learning. In Proceedings of the Vehicle Power and Propulsion Conference (VPPC), Montreal, QC, Canada, 19–22 October 2015; pp. 1–5.
27. Patel, S.; Griffin, B.; Kusano, K.; Corso, J.J. Predicting Future Lane Changes of Other Highway Vehicles using RNN-based Deep Models. *arXiv* **2018**, arXiv:1801.04340.
28. Ciregan, D.; Meier, U.; Schmidhuber, J. Multi-column deep neural networks for image classification. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3642–3649.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
30. Li, S.Z.; Yu, B.; Wu, W.; Su, S.Z.; Ji, R.R. Feature learning based on SAE-PCA network for human gesture recognition in RGBD images. *Neurocomputing* **2015**, *151*, 565–573. [[CrossRef](#)]
31. Liu, H.L.; Taniguchi, T.; Tanaka, Y.; Takenaka, K.; Bando, T. Visualization of Driving Behavior Based on Hidden Feature Extraction by Using Deep Learning. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2477–2489. [[CrossRef](#)]
32. Chi, L.; Mu, Y. Deep Steering: Learning End-to-End Driving Model from Spatial and Temporal Visual Cues. *arXiv* **2017**, arXiv:1708.03798.
33. Li, L.; Ota, K.; Dong, M. Human-Like Driving: Empirical Decision-Making System for Autonomous Vehicles. *IEEE Trans. Veh. Technol.* **2018**, *67*, 6814–6823. [[CrossRef](#)]
34. Mänttari, J.; Folkesson, J.; Ward, E. Learning to Predict Lane Changes in Highway Scenarios Using Dynamic Filters On a Generic Traffic Representation. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1385–1392. [[CrossRef](#)]
35. You, C.W.; Montes-de Oca, M.; Bao, T.J.; Lane, N.D.; Lu, H.; Cardone, G.; Torresani, L.; Campbell, A.T. CarSafe: A driver safety app that detects dangerous driving behavior using dual-cameras on smartphones. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; pp. 671–672.
36. Romera, E.; Bergasa, L.M.; Arroyo, R. A real-time multi-scale vehicle detection and tracking approach for smartphones. In Proceedings of the Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference, Gran Canaria, Spain, 1–15 September 2015; pp. 1298–1303.
37. Kaplan, S.; Guvensan, M.A.; Yavuz, A.G.; Karalurt, Y. Driver behavior analysis for safe driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3017–3032. [[CrossRef](#)]
38. Sharma, A.; Kumar, D. Non-image Data Classification with Convolutional Neural Networks. *arXiv* **2020**, arXiv:2007.03218.
39. He, Z.; Hu, J.; Park, B.B.; Levin, M.W. Vehicle Sensor Data-Based Transportation Research: Modeling, Analysis, and Management. *J. Intell. Transp. Syst.* **2019**, *23*, 99–102.
40. Su, S.; Muelling, K.; Dolan, J.; Palanisamy, P.; Mudalige, P. Learning Vehicle Surrounding-aware Lane-changing Behavior from Observed Trajectories. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1412–1417. [[CrossRef](#)]
41. Huang, W.; Song, G.; Hong, H.; Xie, K. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2191–2201. [[CrossRef](#)]
42. Lee, H.; Pham, P.; Largman, Y.; Ng, A.Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. *Adv. Neural Inf. Process. Syst.* **2009**, *22*, 1096–1104.
43. Bengio, Y. Learning deep architectures for AI. *Found. Trends® Mach. Learn.* **2009**, *2*, 1–127. [[CrossRef](#)]
44. Huang, H.B.; Huang, X.R.; Li, R.X.; Lim, T.C.; Ding, W.P. Sound quality prediction of vehicle interior noise using deep belief networks. *Appl. Acoust.* **2016**, *113*, 149–161. [[CrossRef](#)]
45. Hinton, G.E.; Osindero, S.; Teh, Y.W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2014**, *18*, 1527–1554. [[CrossRef](#)]
46. Le Roux, N.; Bengio, Y. Representational power of restricted Boltzmann machines and deep belief networks. *Neural Comput.* **2008**, *20*, 1631–1649. [[CrossRef](#)]
47. Kim, J.; Kang, U.; Lee, Y. Statistics and Deep Belief Network-Based Cardiovascular Risk Prediction. *Healthc. Inf. Res.* **2017**, *23*, 169–175. [[CrossRef](#)] [[PubMed](#)]
48. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)] [[PubMed](#)]
49. Mao, Y.; Shen, J.; Gui, X. A Study on Deep Belief Net for Branch Prediction. *IEEE Access* **2017**, *6*, 10779–10786.

50. Wang, L.; Xia, X.; Cao, J.; Liu, X. Modeling and predicting remanufacturing time of equipment using deep belief networks. *Clust. Comput.* **2017**, *22*, 2677–2688. [[CrossRef](#)]
51. Lv, Q.; Dou, Y.; Niu, X.; Xu, J.; Xia, F. Remote sensing image classification based on DBN model. *J. Comput. Res. Dev.* **2014**, *51*, 1911–1918.
52. Shen, F.; Chao, J.; Zhao, J. Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing* **2015**, *167*, 243–253. [[CrossRef](#)]
53. Duan, Y.; Lv, Y.; Liu, Y.; Wang, F. An efficient realization of deep learning for traffic data imputation. *Transp. Res. Part C Emerg. Technol.* **2016**, *72*, 168–181. [[CrossRef](#)]
54. Huang, H.B.; Li, R.X.; Yang, M.L.; Lim, T.C.; Ding, W.P. Evaluation of vehicle interior sound quality using a continuous restricted Boltzmann machine-based DBN. *Mech. Syst. Signal Process.* **2017**, *84*, 245–267. [[CrossRef](#)]
55. Zhao, C.; Gong, J.; Lu, C.; Xiong, G.; Mei, W. Speed and steering angle prediction for intelligent vehicles based on deep belief network. In Proceedings of the Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference, Yokohama, Japan, 16–19 October 2017; pp. 301–306.
56. Huang, L.; Xiang, L.Y. Method for Meteorological Early Warning of Precipitation-Induced Landslides Based on Deep Neural Network. *Neural Process. Lett.* **2018**, *48*, 1243–1260.
57. Zhang, X.L.; Wu, J. Deep Belief Networks Based Voice Activity Detection. *IEEE Trans. Audio Speech Lang. Process.* **2013**, *21*, 697–710. [[CrossRef](#)]
58. 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, 4–7 November 2018; pp. 2118–2125. [[CrossRef](#)]