

Research Article

A Novel Method for Predicting Vehicle State in Internet of Vehicles

Yanting Liu ¹, Ding Cheng ², Yirui Wang ¹, Jiujun Cheng ³, and Shangce Gao ¹

¹Faculty of Engineering, University of Toyama, Toyama 930-8555, Japan

²College of Computer Science and Technology, Anhui University, Hefei 230601, China

³Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China

Correspondence should be addressed to Shangce Gao; gaosc@eng.u-toyama.ac.jp

Received 23 January 2018; Accepted 12 April 2018; Published 21 May 2018

Academic Editor: Mohamed Elhoseny

Copyright © 2018 Yanting Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the fields of advanced driver assistance systems (ADAS) and Internet of Vehicles (IoV), predicting the vehicle state is essential, including the ego vehicle's position, velocity, and acceleration. In ADAS, an early position prediction helps to avoid traffic accidents. In IoV, the vehicle state prediction is essential for the required calculation of the expected reliable communication time between two vehicles. Many approaches have emerged to perform this vehicle state prediction. However, such approaches consider limited information of the ego vehicle and its surroundings, and they may not be very effective in practice because the real situation is highly complex and complicated. Moreover, some of the approaches often lead to a delayed prediction time due to collecting and calculating the substantial history information. By assuming that the driver is a robot driver, which eliminates distinct driving behaviors of different persons when facing the same situation, this paper creates a decision tree as a new quick and reliable method adapted to all road segments, and it proposes a new method to perform the vehicle state prediction based on this decision tree.

1. Introduction

Advanced driver assistance systems (ADAS) installed in vehicles use sensing and computing technologies to assist drivers in avoiding traffic accidents. Predicting the positions of surrounding vehicles is a crucial problem, and it facilitates the early detection of potential collisions. In Internet of Vehicles (IoV), one of the most important foundations for the network connectivity is the vehicle state, including its position, velocity, and acceleration. This importance is because the vehicle state is a deterministic characteristic for communicating among vehicles and infrastructures. Therefore, a common requirement is to calculate the expected reliable communication time quickly in various road segments when two vehicles are to communicate with each other. The vehicle state can influence the network topology of IoV where the location between two vehicles determines the communication range and their velocities and accelerations affect the stability of network topology [1–3]. The routing protocol based on location is particularly important due to its adaptability for frequently changing IoV [4]. Alsaqour et al. found that the inaccurate

location obviously decreased the efficiency of the routing protocol [5]. Thus, a neighbor wireless link break prediction was proposed to predict the neighbor node's location so as to detect the ineffective node by using their velocities and accelerations [6]. However, this method is just suitable for short-time and short-distance prediction because the accelerations of vehicles may significantly change according to changing environments. Consequently, the vehicle state prediction is necessary and essential for IoV.

A decision tree is an effective method for evaluating the behaviors of vehicle drivers; some studies include a decision tree to predict or monitor vehicle drivers [7, 8]. Ahmed presented a method to predict the vehicle state, in which a decision tree was used to determine whether the vehicle changed its lane and to obtain the lane after the vehicle changed lanes [9]. Kedowide et al. used a decision tree to monitor the vehicle driver and log the driving activities, such as to evaluate whether the driver was performing the blind spot check, integrated with the behaviors of the driver [10].

The aforementioned methods are primarily used in advanced driver assistance systems (ADAS) to avoid collisions

on the planned trajectories of vehicles and considering limited information about the ego vehicle and its surroundings. Moreover, a delayed prediction time will arise when history data need to be collected. Although some studies have employed a decision tree to quickly make judgments, the decision tree does not use much road and environment information and does not perform predictions across all road segments.

This paper proposes a new approach based on a decision tree that considers more information about the ego vehicle, its surroundings, and driver behaviors in varieties of road segments and without a delayed prediction time because no history data are collected as in some previous methods. This approach saves time and reduces the precious prediction time. According to information of the ego vehicle, roads, traffic lights, other surrounding vehicles, and so forth, our approach prejudices the driving behaviors of the ego vehicle, and then a decision tree is adapted to all road segments. Thus, the state of the ego vehicle, including its position, velocity, and acceleration, can be predicted based on a previously created decision tree. Such a decision tree with considerable useful information including more road surrounding cases helps to predict the vehicle state more accurately in some complex and complicated environments and without a delayed prediction time. The decision tree has advantages such as quick situation judgment and easy extension to more complicated problems with more determination conditions to be adapted to all road segments.

The contributions of this paper can be summarized as follows: (1) This paper defines three varieties of road segments: section, intersection, and transition. Based on the definitions in System Representation, this work extracts different behaviors in distinct road segments. These defined behaviors are introduced in Driving Behavior Modeling. (2) To predict the vehicle state from the behavior, we use a decision tree in all road segments, which is illustrated in State Prediction. The decision tree includes the predefined road segment situations and has advantages such as fast situation judgment and easy extension to more complicated problems with more determination conditions to be adapted to more road segments. We discuss the state prediction by taking advantage of the decision tree, which allows our work to predict the vehicle state through the decision tree.

The remainder of this paper is organized as follows. Section 2 gives the past works in vehicle state. Section 3 presents an overview of the system. Section 4 delineates several models of driving behaviors. Our prediction approach is described in Section 5. The numerical results are presented in Section 6. Finally, the conclusions of this paper are drawn in Section 7.

2. Related Works

Researches about vehicle state can be classified as three parts, that is, environments, maneuvers, and trajectories [11]. Environments are components of conducting vehicle behaviors. Various approaches have been done to discuss vehicle behaviors in different driving environments. In [12], a detection-by-tracking method was used to detect vehicles in a spatio-temporal environment. In [13], intersection driving and

nonintersection driving were distinguished by histograms of scene flow vectors. In [14], a dynamic driving environment was established for detecting the vehicle motion. Maneuvers such as overtaking [15], turning [16], and changing lanes [17] are investigated to analyze the vehicle motion on the path. Overtaking behavior is implemented generally by using some devices to detect vehicles in front of the ego vehicle [15, 18, 19]. When overtaking conditions are satisfied in search space, vehicles will realize an overtaking maneuver. Turning behavior is another usual maneuver for vehicles. Detecting the yaw rate can judge the vehicle turning behavior [20]. Adopting a clustering of 3D points to analyze vehicle's shape can also handle a turning behavior [21]. In [17], changing lanes was achieved by establishing a dynamic Bayesian network based on practical data. Trajectories composed of a set of sequences of positions and velocities with a time window are used to extract vehicle behaviors in the past few years. In [22], a Gaussian mixture model was utilized to predict long-term trajectories of vehicles. On the highway, trajectories were constructed using a stereo vision and clustering method [23].

Besides the study of practical vehicle state, many approaches have emerged to obtain a credible vehicle state prediction. Hermes et al. predicted the position of a vehicle after several seconds using the history information of the vehicles [24]. Hermes et al. extracted a large number of vehicle trajectories to perform data training based on trajectory classification technology, in which trajectories were classified into several behaviors, such as left-handed rotation and right-handed rotation, and then they classified the existing trajectories [25]. In addition to objectivities, some researches added drivers' subjective purposes such as left turn, right turn, and changing lanes into the prediction models [26, 27]. A prediction technology for a motorcade formed by several vehicles was proposed by Pandita and Caveney, and in their approach, how a car follows was simulated using the smart driver model [28]. Additionally, a technology that combined the motion model and maneuver recognition was validated, in which probabilistic finite-state machines, fuzzy logics, and driving context recognitions were involved to predict a vehicle trajectory [29–31]. Petrich et al. used additional information from a digital map to enable a stochastic filter to select a representative set of reasonable trajectories [32]. Kumar et al. predicted the lane change intention online using a support vector machine and Bayesian filtering [33]. Yao et al. learned a simplified trajectory set using a collection of lane change trajectories from real driving data [34]. By introducing essential maneuver recognition, Houenou et al. predicted the vehicle trajectory using the constant yaw rate and acceleration motion model [31], which was widely and importantly used in [35–37]. These prediction technologies need more history information of vehicles; meanwhile, the impact of lane and traffic light on trajectories of vehicles is ignored. The trajectories of vehicles are restricted by lanes; however, a digital map based on the routing protocol can offer information of lanes to improve routing efficiency of IoV. Vehicle state prediction can also use the digital map and traffic light to enhance the accuracy of location prediction of vehicles [32, 38].

3. System Overview

3.1. Motivation. Vehicle state prediction is suitable for IoV in city scenario, integrating the digital map, traffic light, and surrounding vehicles. It models the driver's behavior in different traffic environments. In this paper, according to the driver's behavior, a decision tree is established to describe vehicle state in diverse conditions. Vehicle state prediction is an important part of the connectivity model in IoV, which is proper for predicting the positions of vehicles and dynamic changes of links. Taking advantage of information IoV provides, vehicle state can be predicted to further guide the driver to adopt several operations in order to implement a better trajectory of vehicle and save time. Nevertheless, the primary purpose of the vehicle state prediction is not to find an optimal route but to predict the vehicle state in next seconds. The vehicle state contains the position, velocity, and acceleration of vehicle. The change of vehicle state can influence the topology of IoV. For example, position can determine whether two vehicles are accessible to communicate with each other, whereas velocity and acceleration influence the stability of network topology. These factors could finally affect the survival time of links among vehicles. Hence, vehicle state prediction is mainly to calculate the survival time of links so as to guarantee to achieve a better communication among several vehicles and maintain a steady structure of IoV. Meanwhile, it also offers an effective method to construct a reasonable route.

3.2. System Architecture. The vehicle state prediction proposed in this paper is designed specifically to be used in Internet of Vehicles (IoV). It is assumed that the state prediction is hosted by the server that maintains the states of vehicles on the Internet. This assumption is often considered to be a reasonable assumption. Each vehicle manages its state prediction via a virtual object. Nowadays, the virtual object plays an important role in Internet of Things to implement its virtualness and service [39]. For IoV, virtual objects implement the communication among vehicles and provide a practical application for managing vehicles. Position-based and map-based routing protocols in the previous literature are widely accepted routing protocols in IoV based on position and path. Cheng et al. [40] classify notable routing protocols into routing categories for performing routing. Both position-based and map-based routing protocols require vehicles to send their state information to the server, which is generally distributed, when a source vehicle needs to communicate with other vehicles periodically. The server destination node first queries the state information of the destination from the server and then sends data toward the vehicle at the position. The position of the destination will often change during data forwarding; thus, if the position of the destination could be predicted, it would improve the routing performance. Moreover, by predicting vehicle states in a forward routing path, the server has the ability to calculate the expected reliable communication time between two vehicles and then calculate the connectivity of the path,

which helps to select a stable path from multiple paths. When a source queries for the position of a destination, the server could send the predicted state and the optimal forward routing to the source, which will also improve the routing performance.

3.3. System Representation. The following describes the vehicle information that will be used in this work:

- (i) The position of a certain vehicle at a certain time can be represented using a two-dimensional column vector:

$$\mathbf{p}(t) = (x(t), y(t))^T. \quad (1)$$

- (ii) The velocity of a certain vehicle at a certain time can be represented as follows:

$$\mathbf{v}(t) = (v_x(t), v_y(t))^T. \quad (2)$$

- (iii) The acceleration of a certain vehicle at a certain time can be expressed as follows:

$$\mathbf{a}(t) = (a_x(t), a_y(t))^T. \quad (3)$$

- (iv) The length of a certain vehicle is l . Specifically, the length of the ego vehicle is l_α .
- (v) The number of vehicles in front of a certain vehicle at a certain time is $n(t)$.
- (vi) α refers to the ego vehicle, and $\alpha - k$ refers to the k th vehicle that is in front of the ego vehicle. For example, the directly previous vehicle is $\alpha - 1$.
- (vii) $d_{\alpha-k}$ refers to the distance between the ego vehicle and the k th vehicle in front of the ego vehicle. For example, the distance between the ego vehicle and the directly previous vehicle $\alpha - 1$ is $d_{\alpha-1}$.

Hence, the vehicle state in this work is defined as a triple:

$$\mathbf{state}(t) = \langle \mathbf{p}(t), \mathbf{v}(t), \mathbf{a}(t) \rangle, \quad (4)$$

where $\mathbf{p}(t)$, $\mathbf{v}(t)$, and $\mathbf{a}(t)$ are all mentioned above.

Additionally, the road information and vehicle surroundings should also be extracted and represented. Before extracting and representing the road information and vehicle surroundings, this work introduces a new concept of a transition between a section and an intersection. The road is divided into three segments, as shown in Figure 1. All predictions in the three segments are integrated into one decision tree. A transition is a special part of a section with information of the intersection that needs to be considered. In other words, when a vehicle is at a transition, the driver faces the intersection and is able to obtain information such as traffic lights.

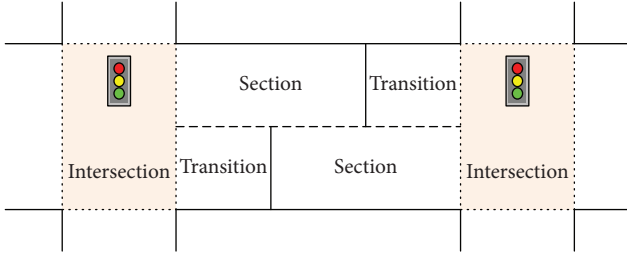


FIGURE 1: Road components.

- (i) A certain intersection is defined as a two-dimensional point:

$$\mathbf{intersec} = (x_0, y_0), \quad (5)$$

where (x_0, y_0) is its position. On an intersection, we just consider three behaviors of each vehicle, that is, left turn, right turn, and pass through. Therefore, the intersection is expressed by a point.

- (ii) A certain section between two consecutive intersections $\mathbf{intersec}_1$ and $\mathbf{intersec}_2$ that are the ends of the certain section is defined as follows:

$$\mathbf{sec} = \langle \mathbf{intersec}_1, \mathbf{intersec}_2, 0 \rangle, \quad (6)$$

where the direction of the vehicle is from intersection $\mathbf{intersec}_1$ to intersection $\mathbf{intersec}_2$.

- (iii) Consequently, a certain lane is

$$\mathbf{lane} = \langle \mathbf{sec}, n \rangle, \quad (7)$$

where \mathbf{sec} is the section to which the certain lane belongs and n is the number of the lane. In this work, the width of every lane is the same, and it is a known constant. When a vehicle faces an intersection, the driver can see three directions. Additionally, this work defines three directions: \mathbf{lane}_N is the direction of the lane in which the driver faces straight forward, \mathbf{lane}_W is the direction of the lane that the driver turns right into, and \mathbf{lane}_E is the direction of the lane that the driver turns left into.

- (iv) A certain transition between a certain section and a certain intersection that is an end of the certain section is

$$\mathbf{trans} = \langle \mathbf{intersec}_1, \mathbf{intersec}_2, 1 \rangle, \quad (8)$$

where \mathbf{trans} is very similar to \mathbf{sec} because a transition is a special part of a section, and when the vehicle is in the transition, the driver is facing intersection $\mathbf{intersec}_2$ and sees the traffic lights in the intersection. The purpose of 0 and 1 in \mathbf{sec} and \mathbf{trans} is to distinguish their mathematical definition.

Table 1 presents a summary of the aforementioned variables, including the definitions of the position, velocity,

TABLE 1: Variable summary.

Variable	Explanation
$\mathbf{p}(t)$	The position of a vehicle at time t
$\mathbf{v}(t)$	The velocity of a vehicle at time t
$\mathbf{a}(t)$	The acceleration of a vehicle at time t
$\mathbf{state}(t)$	The vehicle state of a vehicle, including its position, velocity, and acceleration
$\mathbf{intersec}$	An intersection, which is defined by a point
\mathbf{sec}	A section, which is defined by two $\mathbf{intersec}$ s
\mathbf{lane}	A lane, which is defined by $\mathbf{intersec}$ and its lane number
\mathbf{trans}	A transition, whose definition is similar to \mathbf{sec}

acceleration, and vehicle state of a vehicle and of several elementary road environments, such as an intersection, a section, a lane, and a transition.

Note that in this work, the number of lanes is based on zero, and the lane number starts from the central line of the section to which the lane belongs.

4. Driving Behavior Modeling

In this work, the driving behaviors of a vehicle are considered as the mean motions of the vehicle, such as some sudden changes including accelerating, decelerating, changing lanes, and turning at an intersection. These driving behaviors lead to discontinuous acceleration, which causes the acceleration, velocity, and position of the vehicle to be difficult to predict using their history states. The early detection of sudden changes is necessary for predicting the vehicle state. Driving behaviors can be defined as elements in a set, and each behavior is an element of the set. To create the decision tree in all road segments in this work, it is necessary to model the driving behaviors of a vehicle. The driving behavior is divided into three cases: section prediction, intersection prediction, and transition prediction. At sections, vehicles accelerate or decelerate, which is caused by the influence of the front vehicles. Additionally, vehicles may change lanes to leave an upcoming jam or to avoid a slow vehicle that is directly in front. Only when adjacent lanes have spacing can lane changes occur. A transition, with some specific characteristics, is a certain area between a section and its intersection. Vehicles at a transition are forbidden from changing lanes, and their behaviors are mainly dependent on the traffic lights. At intersections, vehicles may turn left or right or pass through, depending on the out direction of the lane that the vehicle is in and on the traffic light.

According to the aforementioned road in various situations, this paper classifies driving behaviors into three models: section behaviors, intersection behaviors, and transition behaviors. Section behaviors occur in the section, which are relatively simple without considerations of orientation changing. Considerations of intersection behaviors include changing direction. Transition behaviors are relatively complicated. The transition situation is between section and intersection, and it contains possibilities of section's and intersection's behaviors; thus, it is difficult to predict the coming driving behavior due to various possibilities.

4.1. *Section Behaviors.* Section behaviors are always when the vehicle is far away from the front intersection and the traffic light is out of the range of the driver.

4.1.1. *Jam Leaving Intent.* When a driver realizes that a jam has occurred in the front of his current lane, he will attempt to enter adjacent lanes to avoid the jam. Lane-changing behavior is an important intent and has already been considered in the previous literature, such as by Ahmed [9]. In this work, α refers to the ego vehicle and $\alpha - m$ refers to the m th vehicle in front of the ego vehicle; for example, the vehicle directly in front of the ego vehicle is $\alpha - 1$. Here, r_α is the range of the driver in the ego vehicle α . The driver could see τ_α vehicles $\alpha - 1, \alpha - 2, \dots, \alpha - \tau_\alpha$ in the driver's range r_α , but the vehicle $\alpha - \tau_\alpha - 1$ is out of the driver's range. Thus, τ could be represented mathematically by

$$\tau_\alpha = \max_i d_{\alpha-i} < r_\alpha, \quad (9)$$

where $d_{\alpha-i}$ is the distance between the vehicle α and the i th front vehicle $\alpha - i$. The jam density ρ_α , which defines an indicator to quantify the congestion level, is as follows:

$$\rho_\alpha = \frac{\tau_\alpha}{r_\alpha}. \quad (10)$$

The driver will have a jam leaving intent if the driver cannot tolerate such a jam that $\rho_\alpha > \rho_\alpha^*$, where ρ_α^* is a tolerance threshold for the driver. In this paper, the length of vehicle is 4.3–4.7 meters; we set one vehicle within 5 meters as a tolerance threshold. Thus, this paper uses $\rho_\alpha^* = 0.2$ and $r_\alpha = 400$ m in our later numerical experiments. Then, the driver will change lanes if the condition for changing lanes is satisfied. The driver always prefers to change to the right lane, and when the condition for changing to the right lane is not satisfied, the driver considers changing to the left lane.

4.1.2. *Overtaking Intent.* For this intent, this work considers two aspects: the sizes of and the velocities between the front vehicle and the ego vehicle.

When the front vehicle, such as a truck, is considerably larger than the ego vehicle, the driver always tends to avoid following it. This work simply assumes that the width of every vehicle is the same; thus, this case is simply to compare the lengths and is presented by

$$l_{\alpha-1} > \lambda_{\text{length}} l_\alpha, \quad (11)$$

where $\lambda_{\text{length}} > 1$ is the tolerance threshold for the ratio of the length of the directly previous vehicle to the length of the ego vehicle.

In the other case, if the speed of the vehicle ahead is too slow, the driver often attempts to change lanes and overtakes the slow vehicle. Mathematically,

$$v_{\alpha-1} < \lambda_{\text{velocity}} v_\alpha, \quad (12)$$

where $\lambda_{\text{velocity}} < 1$ is the tolerance threshold. This paper uses $\lambda_{\text{length}} = 1.5$ and $\lambda_{\text{velocity}} = 0.8$.

4.1.3. *Following Intent.* In general, the driver of the ego vehicle will follow the front vehicle. However, when the driver follows the front vehicle, the driver will also adapt the ego vehicle such that it will be more comfortable and safe. For example, when the ego vehicle is too close to the front vehicle, the driver tends to brake to avoid driving into it. Mathematically,

$$t_{\text{brake}} = \frac{d_{\alpha-1}}{v_\alpha - v_{\alpha-1}}. \quad (13)$$

Here, we select a safety braking time t_{safety} as a threshold for the ratio of the velocity of the directly previous vehicle to the velocity of the ego vehicle. When $t_{\text{brake}} \leq t_{\text{safety}}$ is satisfied, the driver will decelerate with an acceleration value. In this paper, $t_{\text{safety}} = 1.5$ s and $\delta = 2$ is a correlation coefficient:

$$a = -\delta \frac{v_\alpha - v_{\alpha-1}}{d_{\alpha-1}}. \quad (14)$$

4.1.4. *Free Driving Intent.* Otherwise, the state of the driver will be maintained. This case is called *free driving intent*. For free driving, this work will consider that if the velocity of a vehicle is less than the speed limit, then the driver tends to accelerate with a constant acceleration to reach the speed limit.

4.2. *Intersection Behaviors.* Intersection behaviors are to predict the motions when the vehicle is close to or facing the front intersection. In such cases, the driver should consider the information of the front intersection, such as traffic lights. Behaviors at intersections are difficult to detect without information, including the out directions of the lane where the ego vehicle is located and the traffic lights. Existing studies always use history trajectories to recognize vehicle behavior using pattern classifications, fuzzy logics, probabilistic finite-state machines, or other technologies [29]. However, these technologies all require sufficiently long trajectories, which lead to delayed time, and these technologies have considerable computational requirements, which make them unsuitable for performing recognition of behaviors at a server with a massive number of vehicles. According to the out direction of the driving lane and traffic light, it is simple and accurate to achieve early detection of whether the vehicle is going to turn left, right, or pass through. However, there is a case in which the motion cannot be detected only using the out direction and traffic lights. This situation arises because the traffic light may occasionally allow the three directions simultaneously. In this case, the motion cannot be determined only by the directions of the lane and the traffic light. As shown in Figure 2, the **lane_s** has three out directions: S/E, S/W, and S/N. Here, S is marked as the directing lane toward the south, E is marked as the left lanes of the intersection toward the east, W is marked as the right lanes of the intersection toward the west, and N is marked as the lanes across the north. Hence, S/E means that the driver turns left from the current lane to the east lanes, and S/W and S/N have similar

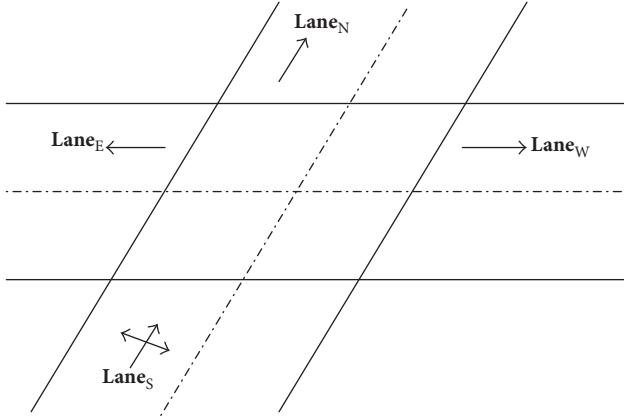


FIGURE 2: Intersection.

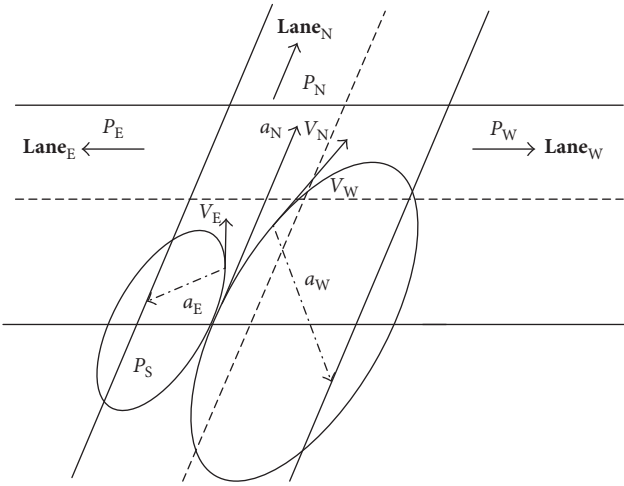


FIGURE 3: Behaviors at intersection.

meanings. The motion may be to turn left or right or pass through with the trajectories $P_S P_E$, $P_S P_W$, and $P_S P_N$, as shown in Figure 3. Here, P_E , P_S , P_W , and P_N are four positions standing for positions to the east, south, west, and north, respectively, of the certain transition. For simplicity, this work considers the curve trajectory of the motion from the ego vehicle, which is similar to 1/4 part of an ellipse, as illustrated in Figure 3, due to two accelerations changing, in which one's direction is the original direction and the other's direction is the terminal one. The details of the calculation of the two accelerations and corresponding velocity and position will be discussed in Section 4. When a vehicle is arriving from the south, it will have three probabilities: to turn left (go \mathbf{trans}_E in Figure 3), to turn right (go \mathbf{trans}_W in Figure 3), and to go straight (go \mathbf{trans}_N in Figure 3). We will discuss these three cases in the following. If

$$\frac{|\mathbf{a}^T \cdot \mathbf{v}|}{|\mathbf{a}| |\mathbf{v}|} > 1 - \epsilon, \quad (15)$$

the motion is to pass through. If

$$\mathbf{a}^T \cdot \mathbf{lane}_E > \epsilon \text{ or } \mathbf{a}^T \cdot \mathbf{lane}_W < -\epsilon, \quad (16)$$

the motion is to turn left. If

$$\mathbf{a}^T \cdot \mathbf{lane}_W > \epsilon \text{ or } \mathbf{a}^T \cdot \mathbf{lane}_E < -\epsilon, \quad (17)$$

the motion is to turn right. \mathbf{lane}_E and \mathbf{lane}_W are column vectors. The aforementioned ϵ is a positive value close to zero, indicating that it is sufficiently small. In this paper, $\epsilon = 0.01$.

4.3. Transition Behavior. When a vehicle is in a transition, the driver can see the traffic light. Different traffic lights can lead to distinct behaviors of the vehicle. Here, we consider that when a driver faces a red or yellow traffic light, the driver will give the vehicle a constant acceleration \mathbf{a}_δ . We also provide a vector \mathbf{e} to represent the traffic light information:

$$\mathbf{e} = \begin{cases} (1, 0, 0)^T, & \text{if the traffic light is red} \\ (0, 1, 0)^T, & \text{if the traffic light is yellow} \\ (0, 0, 1)^T, & \text{if the traffic light is green.} \end{cases} \quad (18)$$

Thus, the constant acceleration vector can be represented as

$$\delta_a = (\mathbf{a}_\delta, \mathbf{a}_\delta, 0)^T, \quad (19)$$

where each dimension indicates the acceleration of the vehicle in corresponding traffic light.

Hence, when the driver faces the traffic light, the acceleration that the driver will provide is

$$\mathbf{a}_\Delta = \delta_a^T \cdot \mathbf{e}. \quad (20)$$

5. State Prediction

Now, driving behaviors are modeled, and a decision tree can be created based on the surroundings and driving behaviors in varieties of road segments. Our decision tree is illustrated in Figure 4. The decision tree in Figure 4 represents the aforementioned situations in various road segments and their judgment conditions, and it will help provide quick and easy determination and extension. First, it will be considered that the ego vehicle is in a section, an intersection, or a transition, and these cases will be discussed individually. Note that in this work, when the ego vehicle is in the intersection, it means that the ego vehicle has passed the beginning line and will no longer consider traffic lights.

5.1. Prediction in Section. When the ego vehicle is in a section, it is considered whether the vehicle is changing lanes. This is because if the vehicle is changing lanes, its velocity and acceleration are not in the same direction, which will lead to a different trajectory. If the ego vehicle is not changing lanes, it is considered whether the vehicle will change lanes based on the aforementioned *jam leaving intent* and *overtaking intent*. These two intents are very common in reality. If the ego vehicle does not choose to change lanes, then there are two intents for the driver of the ego vehicle: *free driving intent* and *following intent*. To summarize, the prediction in a section could have four cases, A, B, C, and D, as indicated in Figure 4.

- (1) A: When a vehicle is in a section \mathbf{sec} and it is changing lanes, it has a velocity $\mathbf{v}_\perp(t)$ and an

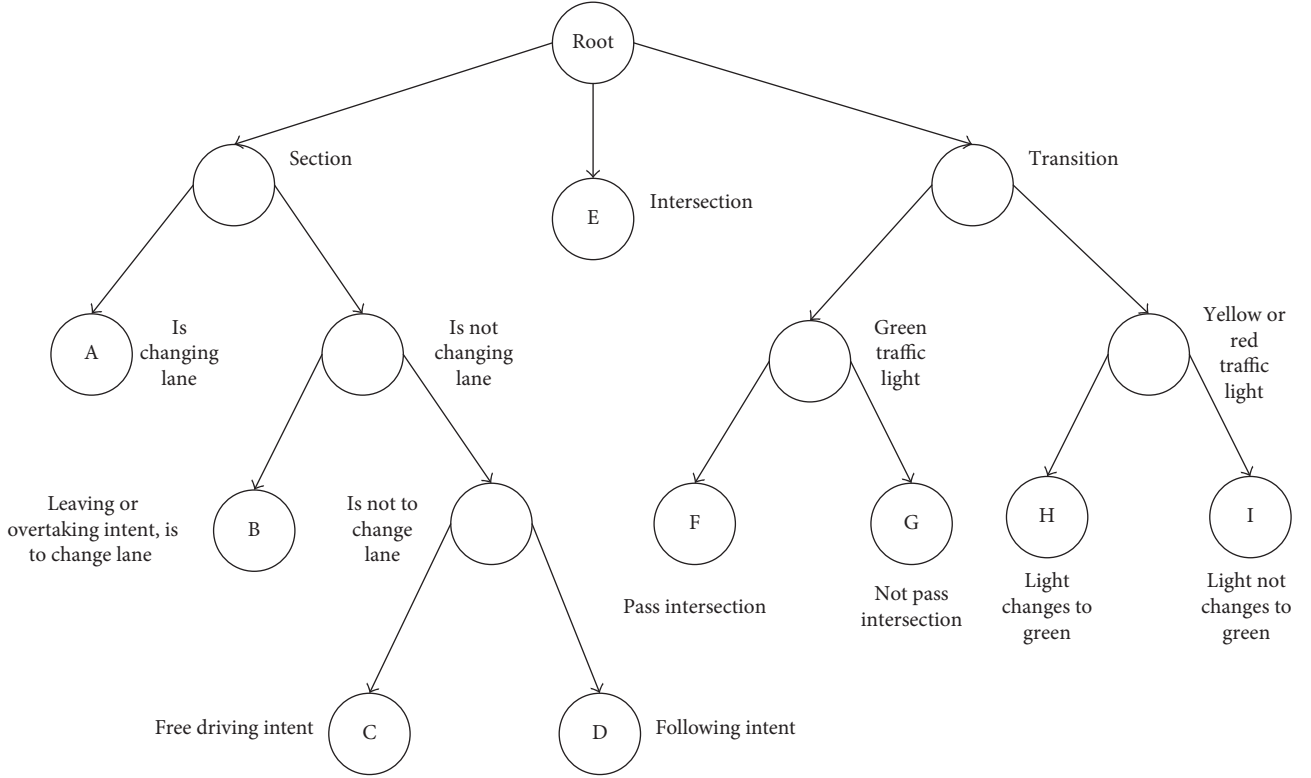


FIGURE 4: Decision tree based on driving behaviors.

acceleration $\mathbf{a}_\perp(t)$ whose directions are both perpendicular to \mathbf{sec} . During the lane change, the vehicle is supposed to have an acceleration \mathbf{a}_\perp to first accelerate and then $-\mathbf{a}_\perp$ to decelerate, where $|\mathbf{a}_\perp| > 0$ could be calculated from the data set. Therefore,

$$\mathbf{a}_\perp(t) = \begin{cases} \mathbf{a}_\perp, & t < t_0 \\ -\mathbf{a}_\perp, & t > t_0, \end{cases} \quad (21)$$

where t_0 is the time point between accelerating and decelerating and t is during the lane change. Moreover, $\mathbf{v}(t) = \mathbf{v}_\perp$ could be calculated from the data set, and the distance of changing the lane equals the lane width, which is also a known constant mentioned in the previous discussion. Hence, the $\mathbf{p}(t)$ during the lane change and the current time point could both be calculated. Then, the position after t_Δ can be determined as follows:

$$\mathbf{p}_\Delta = \begin{cases} \mathbf{p}(t + t_\Delta), & \text{if still changing} \\ \mathbf{p}(t + t') + \int_{t+t'}^{t+t_\Delta} \mathbf{v}(t) dt, & \text{if changing is done,} \end{cases} \quad (22)$$

where \mathbf{p}_Δ is the position at time point $t + t_\Delta$ and $t + t'$ is the time point when the vehicle completes the lane change.

- (2) B: This case is very similar to Case A, but the time point t is not during the lane change but rather when starting to change lanes. Moreover, in Case A, the acceleration could be calculated from the data set,

whereas in this case, the acceleration $\mathbf{a}_\perp(t)$ cannot be calculated from the data set. In this case, this work assumes that the time for changing lanes is a known constant; then, the $\mathbf{a}_\perp(t)$ could be calculated according to the distance of changing the lane, which equals the lane width. Thus, the position after t_Δ could be predicted by the method in Case A.

- (3) C: When a vehicle faces this case, the driver will choose *free driving intent*, which was previously mentioned.
- (4) D: When a vehicle faces this case, the driver will choose *following intent*, which was previously mentioned.

5.2. Prediction in Intersection

- (1) E: When a vehicle is in an intersection **intersec**, the driver could have three options: to drive straight forward, to turn left, and to turn right. The velocity \mathbf{v}_t at the current time point can be calculated from the data set, and it will be compared with \mathbf{lane}_N , \mathbf{lane}_W , and \mathbf{lane}_E to determine which direction the vehicle will go. Mathematically, the direction that the vehicle will go is given by the following equation:

$$\mathbf{direction} = \begin{cases} \mathbf{lane}_W, & \mathbf{v}(t) \cdot \mathbf{lane}_W > \epsilon \\ \mathbf{lane}_E, & \mathbf{v}(t) \cdot \mathbf{lane}_E > \epsilon \\ \mathbf{lane}_N, & \text{otherwise,} \end{cases} \quad (23)$$

where $\epsilon > 0$ is a positive value that is sufficiently small, as previously mentioned. $\epsilon = 0.01$ in this paper. If **direction** = \mathbf{lane}_N , then the intersection **intersec** could be considered as a section. If **direction** = \mathbf{lane}_W or **direction** = \mathbf{lane}_E , then the vehicle has two accelerations that have the straight forward direction and the direction same as \mathbf{lane}_W or \mathbf{lane}_E , respectively. This work denotes the first mentioned acceleration as $\mathbf{a}_o(t)$ and the second mentioned acceleration as $\mathbf{a}_1(t)$. This work assumes that in the intersection **intersec**, $\mathbf{a}_o(t)$ is linearly increasing from zero and $\mathbf{a}_1(t)$ is linearly decreasing to zero. That is, $|\mathbf{a}_o(t)| + |\mathbf{a}_1(t)|$ is a constant during turning. At some certain time point t_0 , $\mathbf{a}_o(t_0)$ and $\mathbf{a}_1(t_0)$ could be calculated from the data set; thus, we let

$$a_\Sigma = |\mathbf{a}_o(t_0)| + |\mathbf{a}_1(t_0)|. \quad (24)$$

From the data set of the map, the distance between the current time point and the time point when the vehicle completes turning can be calculated. Hence, the time t_s remaining for turning can be obtained. Therefore,

$$\begin{aligned} \mathbf{a}_o(t_0 + t) &= \left(|\mathbf{a}_o(t_0)| - \frac{|\mathbf{a}_o(t_0)|}{t_s} t \right) \mathbf{e}_o, & t < t_s, \\ \mathbf{a}_1(t_0 + t) &= \left(|\mathbf{a}_1(t_0)| + \frac{|\mathbf{a}_1(t_0)|}{t_s} t \right) \mathbf{e}_1, & t < t_s, \end{aligned} \quad (25)$$

where \mathbf{e}_o is the direction of the original direction and \mathbf{e}_1 is the direction of **direction**. The position after t_Δ is

$$\mathbf{p}(t_0 + t_\Delta) = \begin{cases} \mathbf{p}(t_0) + \int_0^{t_\Delta} \mathbf{v}_0 + \mathbf{a}(t_0 + t) dt, & t < t_s \\ \mathbf{p}(t_0 + t_s) + (t_\Delta - t_s) \mathbf{v}', & t > t_s, \end{cases} \quad (26)$$

where \mathbf{v}' is the velocity when the vehicle completes turning.

5.3. Prediction in Transition

- (1) F: When the driver faces a green traffic light and the vehicle could pass in time, the case could be in a section (when the requested time point is not sufficient to pass) or in an intersection (when the requested time point is sufficient to pass).
- (2) G: When the driver faces a green traffic light and the vehicle cannot pass in time, the driver will stop the vehicle. The vehicle knows if some vehicle is in front of it. If some vehicle is in front of it, the driver will have *following intent*. If no vehicle is in front of it, the vehicle will calculate the distance between the current position and the final stopped position, which is denoted as d_s . This work assumes that the vehicle will be stopped by a constant acceleration. The constant acceleration can be calculated as

$$\mathbf{a} = -\frac{|\mathbf{v}(t_0)|^2}{2d_s} \mathbf{e}, \quad (27)$$

where \mathbf{e} is the direction of the vehicle. Thus, the velocity and position are

$$\begin{aligned} \mathbf{v}(t_0 + t_\Delta) &= \mathbf{v}(t_0) + \mathbf{a} \cdot t_\Delta, \\ \mathbf{p}(t_0 + t_\Delta) &= \mathbf{p}(t_0) + \int_0^{t_\Delta} \mathbf{v}(t_0 + t) dt. \end{aligned} \quad (28)$$

- (3) H: This case could be separated into two time intervals: before and after the traffic light turns green. Before the traffic light turns green, the driver would choose *following intent*, while after the traffic light turns red, the case would be Case F.
- (4) I: Because the vehicle will stop as Case G, irrespective of whether some vehicle is in front of it, the vehicle has the same intent as Case G.

5.4. Prediction Summary. We call the above proposed method as driver behavior decision tree (DBDT), which obtains the relatively accurate trajectories of vehicles in a long term according to the sudden changes such as acceleration, deceleration, and turn. Moreover, to prevent the prediction from going too far, this work includes the constant yaw rate and acceleration (CYRA) [31] into our approach. CYRA is a physical kinematic-based prediction method. It assumes that within a very short term, the force on a vehicle remains unchanged and the vehicle would keep a constant accelerate vector, including its accelerate direction and value. Thus, the CYRA model regards the acceleration and direction of vehicle as a constant to predict the vehicle state. Its constant acceleration \mathbf{a}_t is formulated as follows:

$$\mathbf{a}_t = \mathbf{a}_o, \quad (29)$$

where \mathbf{a}_o is a constant value. Next, its velocity and position are calculated as follows:

$$\begin{aligned} \mathbf{v}_t &= \int_0^t \mathbf{a}_t dt, \\ \mathbf{p}_t &= \int_0^t \mathbf{v}_t dt. \end{aligned} \quad (30)$$

The linearity of its state equation achieves a transmission of state probability distribution. The next vehicle state could be predicted based on this kind of constant accelerate vector.

For a short term, the acceleration of vehicle can be considered as a constant, CYRA can effectually adapt to this situation according to its constant acceleration characteristics. Hence, CYRA can effectively handle the vehicle state prediction in a short term so as to obtain more accurate results. However, it could result in a great error for predicting the vehicle state in a long term because the acceleration of vehicle continually changes. On the contrary, DBDT can detect the sudden change of acceleration of vehicle to instantly adapt to the current state so as to obtain better results and avoid a great error, suggesting that it is more suitable for predicting the vehicle state in a long term. On the basis of both characteristics, this work finally adopts the following formula to evaluate their performances.

$$T_{\text{DBDT}}(t) = f(t)T_{\text{DBDT}'}(t) + (1 - f(t))T_{\text{CYRA}}(t), \quad (31)$$

where $T_{\text{DBDT}'}(t)$ is the result of our approach and $T_{\text{CYRA}}(t)$ is the result of another approach. $f(t)$ is an increasing function, which means driving behavior recognition is more suitable for long-term prediction and CYRA is more accurate for short-term prediction. In this paper, $f(t) = 1/4t$.

5.5. Time Complexity. A time complexity comparison between DBDT and CYRA is discussed in this subsection. For DBDT, we set C to be the number of vehicles in the same lane. The time complexity regarding prediction in section, intersection, and transition is calculated as follows:

5.5.1. Prediction in Section

(i) Motion prediction

- (1) Jam Leaving Intent: scanning vehicles in front of itself in the same lane needs the time complexity $C_{m1} = O(c)$.
- (2) Overtaking Intent: considering the vehicle in front of itself requires $C_{m2} = O(1)$.
- (3) Following Intent: calculating the vehicle in front of itself needs $C_{m3} = O(1)$.
- (4) Free Driving Intent: this situation takes $C_{m4} = O(1)$.

(ii) Vehicle state prediction

- (A) Computing the location data of the lane and state of itself costs $C_{s1} = O(1)$.
- (B) Computing the location data of the lane and vehicle states in front of itself needs $C_{s2} = O(c)$.
- (C) Computing the vehicle state in front of itself requires $C_{s3} = O(1)$.
- (D) Computing the state of itself takes $C_{s4} = O(1)$.

Thus, the time complexity of prediction in section is

$$C_{\text{sec}} = (C_{m1} + C_{m2} + C_{m3} + C_{m4}) + \max(C_{s1}, C_{s2}, C_{s3}, C_{s4}) = O(c). \quad (32)$$

5.5.2. Prediction in Intersection

(i) Motion prediction

Predicting the vehicle motions by the traffic light data and the location data of intersection lanes costs the time complexity $C_m = O(1)$.

(ii) Vehicle state prediction

- (E) Computing the vehicle state of itself and intersection lanes data needs $C_s = O(1)$.

Therefore, the time complexity of prediction in intersection is

$$C_{\text{intersec}} = C_m + C_s = O(1). \quad (33)$$

5.5.3. Prediction in Transition

(i) Motion prediction

When a vehicle is in a transition, its motion is predicted by the traffic light. This operation needs $C_m = O(1)$.

(ii) Vehicle state prediction

- (F) When the driver faces the green traffic light and the vehicle could pass in time, computing the state of itself needs $C_{s1} = O(1)$.
- (G) When the driver faces the green traffic light and the vehicle could not pass in time, computing the vehicle state and traffic light time requires $C_{s2} = O(1)$.
- (H) When the driver faces the red traffic light and the traffic light turns to green before it passes the transition, computing the vehicle state and traffic light time requires $C_{s3} = O(1)$.
- (I) When the driver faces the red traffic light and the traffic light keeps red before it passes the transition, computing the vehicle state and traffic light time costs $C_{s4} = O(1)$.

Thus, the time complexity of prediction in transition is

$$C_{\text{trans}} = C_m + \max(C_{s1}, C_{s2}, C_{s3}, C_{s4}) = O(1). \quad (34)$$

The number of vehicles is set to be n for prediction. Consequently, the whole time complexity about DBDT is $C = \max(C_{\text{sec}}, C_{\text{intersec}}, C_{\text{trans}}) = n * O(c) = O(n)$. For CYRA, each vehicle is predicted by the data of itself, its time complexity is $O(n)$ [31]. According to both time complexity, we can find that DBDT and CYRA have the same time complexity, suggesting they possess the same efficiency.

6. Results and Analysis

To test whether our work is valid, experiments are conducted in a real environment, which is based on the Lankershim Boulevard Dataset of the Next Generation Simulation (NGSIM) program [41]. The Lankershim Boulevard Dataset collects detailed vehicle trajectory data from Lankershim Boulevard in the Universal City neighborhood of Los Angeles. It provides the map of an area of Lankershim Boulevard, including three to four lane segments and covering three signalized intersections. Moreover, the traffic light data and the precise vehicle position, velocity, and acceleration in the periods of 8:30 am and 8:45 am on June 16, 2005, are available. The Lankershim Boulevard Dataset covers the driver behavior of lane changing on congested segments, overtaking, and behavior at traffic lights, which fits the experimental requirements of this work. The details of the Lankershim Boulevard Dataset are listed in Table 2. This work creates a model for the provided map in the

TABLE 2: Data set parameters.

Lankershim Boulevard Dataset	Parameters
Address	Lankershim Boulevard in Los Angeles
Time	8:28–8:45 am and 8:45–9:00 am on June 16, 2005
Road length	490 m
Intersection number	4
Sampling time	1/10 s
Lane number (same direction)	1–6
Provides traffic light data	Yes
8:28–8:45 am data amount	705294 records
8:28–8:45 am vehicle number	1375
8:45–9:00 am data amount	902025 records
8:45–9:00 am vehicle number	1601

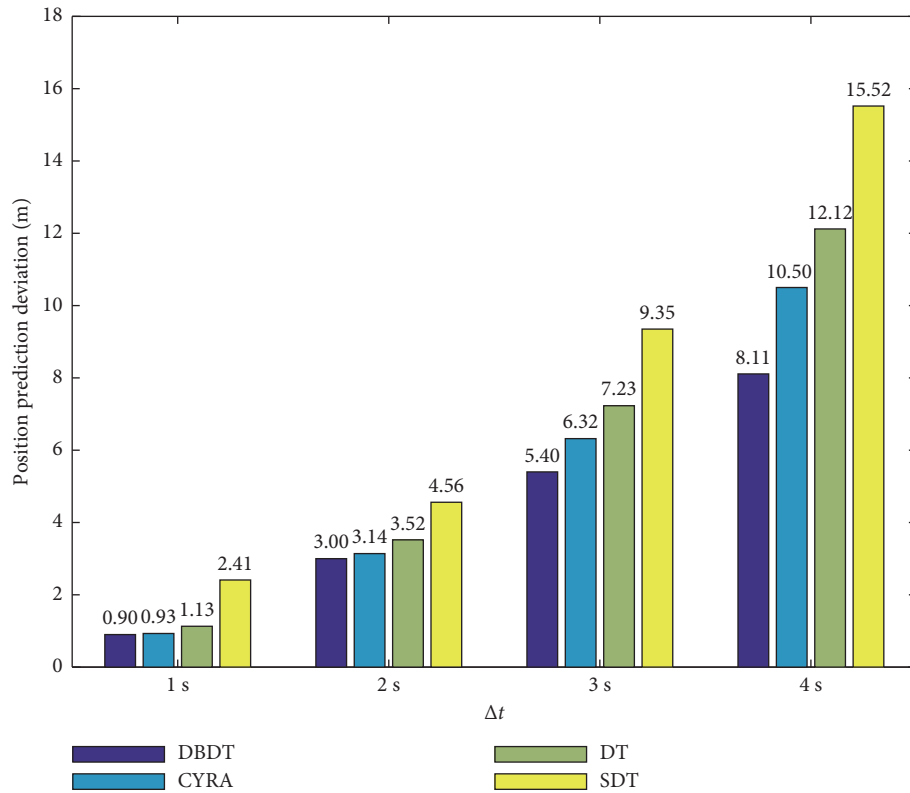


FIGURE 5: Accuracy of position prediction.

Lankershim Boulevard Dataset to extract location data of sections, intersections, and transitions. Then, this work extracts traffic light information, and thus, it obtains all road information. By inputting trajectory information of vehicles, this work will compare our approach (DBDT for short) to CYRA [31], DT which is a variant of DBDT by setting $f(t) = 1$ in (31), and SDT [7, 8] which is a self-selection threshold decision algorithm based on decision tree in four cases: $t_{\Delta} = 1$ s, $t_{\Delta} = 2$ s, $t_{\Delta} = 3$ s, and $t_{\Delta} = 4$ s, respectively.

The results for the accuracy of position prediction are shown in Figure 5, those for the accuracy of velocity prediction are shown in Figure 6, and those for the accuracy of acceleration prediction are shown in Figure 7. The results show that although the state predicted by our approach is not very accurate at the beginning, the state is more accurate than that of CYRA as time passes. This is because our

approach provides early detection of the driving behavior, which leads to changing the state at the very beginning of the prediction time point. Moreover, the vehicle state includes the ego vehicle's position, velocity, and acceleration, for which the importances are decreasing in many fields. For example, to avoid traffic accidents, the vehicle position prediction is the most essential. Considering the discontinuous acceleration, the three vehicle state components, which are the position, the velocity, and the acceleration, are becoming more difficult. Thus, it is expected that from the numerical results, the position prediction is the best, the velocity prediction is not good when $\Delta t = 1$ s, and the acceleration prediction is not good when $\Delta t = 1$ s or $\Delta t = 2$ s. As time passes, the numerical results become better. From the results, the difference value between the previous one second and the next one second becomes increasingly

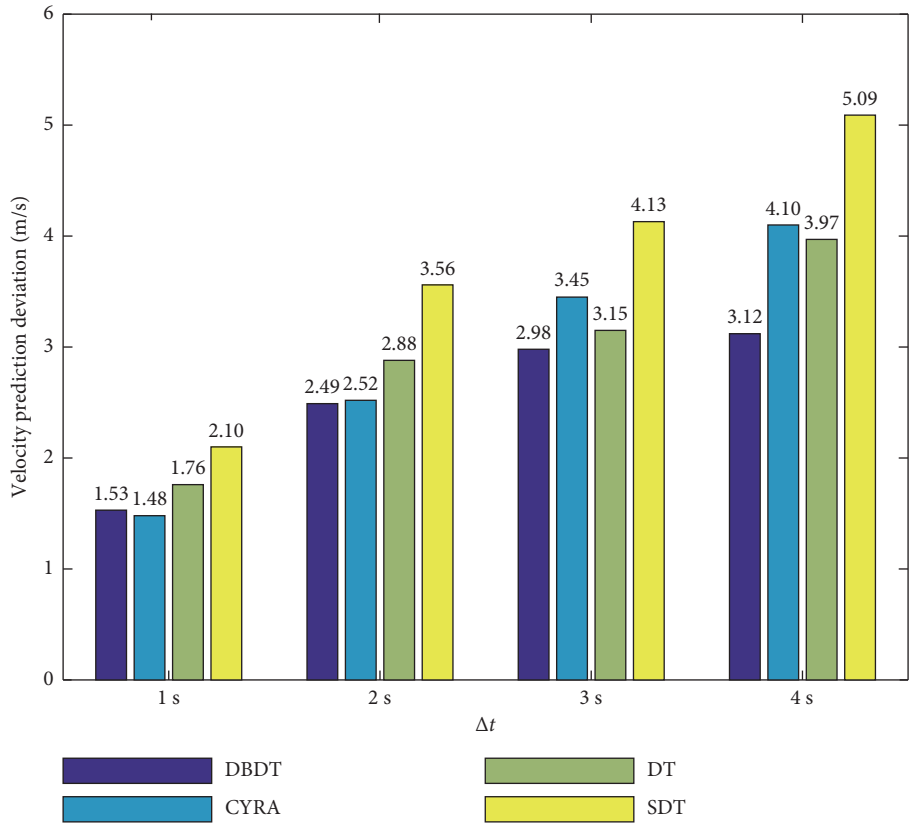


FIGURE 6: Accuracy of velocity prediction.

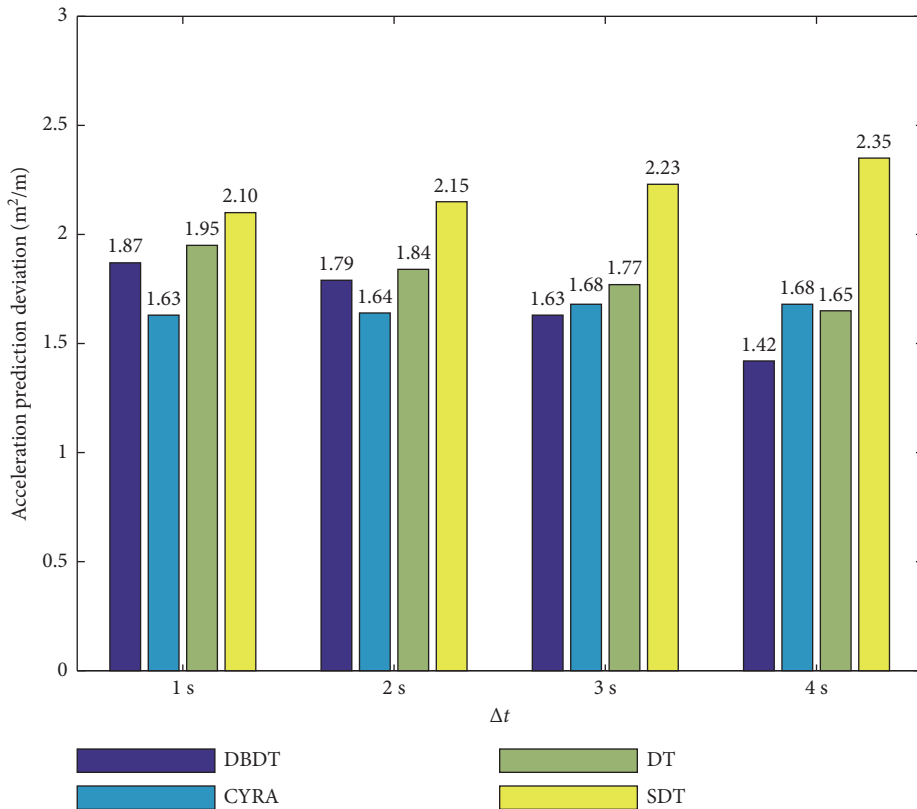


FIGURE 7: Accuracy of acceleration prediction.

smaller. Thus, as time lasts past a certain range, the state prediction will be more accurate than that of CYRA.

Additionally, the results of DBDT are better than those of DT, suggesting that the key to the good performance of our proposal is the incorporation and extension of the decision tree and CYRA. In comparison with SDT which generally utilizes thresholds to determine the state selection in a decision tree, our proposal performs better with the aid of accurate modeling of the driving behaviors. Moreover, in light of the results, we believe that more research on how to use driving behaviors of vehicle in the varieties of the all road segments to predict or monitor vehicle drivers by decision trees is warranted.

7. Conclusion

This paper highlights that the previous approaches for predicting the vehicle states using the substantial history information have a delayed prediction time. Some trajectory prediction methods based on lane changing recognition are proposed. Although a validation method for complicated environments such as multilanes and intersections is not currently available, this paper proposes a new method for the prediction by using a decision tree in varieties of road segments generated by the driving behaviors. This decision tree helps to detect driving behaviors and predict the vehicle state in all road segments, including sections with multilanes, transition segments, and intersections. The driving behavior recognition improves the accuracy of vehicle state prediction in long-term cases. Our approach shows advantages in the provided real environments.

Social Internet of Vehicles is an important and intelligent transport network [42]. It has more characteristics and more complicated circumstances. Thus, to predict this kind of IoV is more meaningful and challenging in the future work. Furthermore, the proposed technique might lead to the development of vehicle networking and intelligentization [43], as well as to provide effective methods to solve vehicle routing problems in dynamic environments [44].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

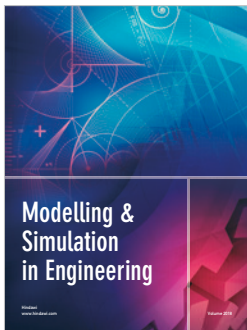
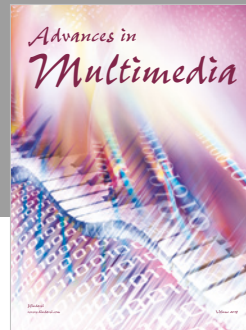
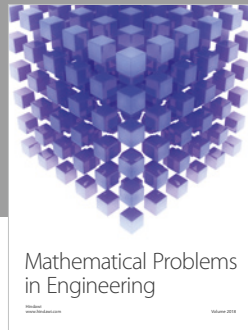
This research was partially supported by the JSPS KAKENHI Grant no. JP17K12751 and the National Natural Science Foundation of China (Grant no. 61472284).

References

- [1] T. Taleb, M. Ochi, A. Jamalipour, N. Kato, and Y. Nemoto, "An efficient vehicle-heading based routing protocol for

- VANET networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2006)*, vol. 4, pp. 2199–2204, Las Vegas, NV, USA, April 2006.
- [2] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, "A stable routing protocol to support ITS services in VANET networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3337–3347, 2007.
- [3] M. Dixit, R. Kumar, and A. K. Sagar, "VANET: architectures, research issues, routing protocols, and its applications," in *Proceedings of the IEEE 2016 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 555–561, Greater Noida, India, April 2016.
- [4] J. Liu, J. Wan, Q. Wang, P. Deng, K. Zhou, and Y. Qiao, "A survey on position-based routing for vehicular ad hoc networks," *Telecommunication Systems*, vol. 62, no. 1, pp. 15–30, 2016.
- [5] R. Alsaqour, M. Abdelhaq, and T. Abdullah, "Modeling the position information inaccuracy in MANET position-based routing protocols," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 3, no. 9, pp. 971–976, 2011.
- [6] R. A. Alsaqour, M. S. Abdelhaq, and O. A. Alsukour, "Effect of network parameters on neighbor wireless link breaks in GPSR protocol and enhancement using mobility prediction model," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 171, 2012.
- [7] S. Wang, C. Fan, C. H. Hsu, Q. Sun, and F. Yang, "A vertical handoff method via self-selection decision tree for internet of vehicles," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1183–1192, 2016.
- [8] B. Ma, D. Wang, S. Cheng, and X. Xie, "Modeling and analysis for vertical handoff based on the decision tree in a heterogeneous vehicle network," *IEEE Access*, vol. 5, pp. 8812–8824, 2017.
- [9] K. I. Ahmed, *Modeling Drivers' Acceleration and Lane Changing Behavior*, Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.
- [10] C. Kedowide, C. Gouin-Vallerand, and É. Vallières, "Recognizing blind spot check activity with car drivers based on decision tree classifier approach," in *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 22–26, Québec City, QC, Canada, July 2014.
- [11] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.
- [12] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, "Vehicle detection and tracking in car video based on motion model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 583–595, 2011.
- [13] A. Geiger and B. Kitt, "Object flow: a descriptor for classifying traffic motion," in *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium (IV)*, pp. 287–293, La Jolla, CA, USA, June 2010.
- [14] S. Cherng, C. Y. Fang, C. P. Chen, and S. W. Chen, "Critical motion detection of nearby moving vehicles in a vision-based driver-assistance system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 70–82, 2009.
- [15] F. Garcia, P. Cerri, A. Broggi, A. de la Escalera, and J. M. Armingol, "Data fusion for overtaking vehicle detection based on radar and optical flow," in *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium (IV)*, pp. 494–499, Alcalá de Henares, Madrid, Spain, June 2012.
- [16] A. Barth and U. Franke, "Tracking oncoming and turning vehicles at intersections," in *Proceedings of the 2010 13th*

- International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 861–868, Funchal, Madeira Island, Portugal, September 2010.
- [17] D. Kasper, G. Weidl, T. Dang et al., “Object-oriented Bayesian networks for detection of lane change maneuvers,” *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 3, pp. 19–31, 2012.
- [18] Y. Zhu, D. Comaniciu, M. Pellkofer, and T. Koehler, “Reliable detection of overtaking vehicles using robust information fusion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 401–414, 2006.
- [19] J. Wang, G. Bebis, and R. Miller, “Overtaking vehicle detection using dynamic and quasi-static background modeling,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops (CVPR’2005)*, p. 64, San Diego, CA, USA, June 2005.
- [20] A. Barth and U. Franke, “Estimating the driving state of oncoming vehicles from a moving platform using stereo vision,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 560–571, 2009.
- [21] B. Barrois and C. Wöhler, “3D pose estimation of vehicles using stereo camera,” in *Encyclopedia of Sustainability Science and Technology*, pp. 10589–10612, Springer, Berlin, Germany, 2012.
- [22] J. Wiest, M. Höffken, U. Kresel, and K. Dietmayer, “Probabilistic trajectory prediction with Gaussian mixture models,” in *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium (IV)*, pp. 141–146, Alcalá de Henares, Madrid, Spain, June 2012.
- [23] S. Sivaraman, B. Morris, and M. Trivedi, “Learning multi-lane trajectories using vehicle-based vision,” in *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 2070–2076, Barcelona, Spain, November 2011.
- [24] C. Hermes, J. Einhaus, M. Hahn, C. Wöhler, and F. Kummert, “Vehicle tracking and motion prediction in complex urban scenarios,” in *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium (IV)*, pp. 26–33, La Jolla, CA, USA, June 2010.
- [25] C. Hermes, C. Wohler, K. Schenk, and F. Kummert, “Long-term vehicle motion prediction,” in *Proceedings of the 2009 IEEE Intelligent Vehicles Symposium*, pp. 652–657, Xi’an, China, June 2009.
- [26] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïzi, “Model-based predictive motion cueing strategy for vehicle driving simulators,” *Control Engineering Practice*, vol. 17, no. 9, pp. 995–1003, 2009.
- [27] J. Sorstedt, L. Svensson, F. Sandblom, and L. Hammarstrand, “A new vehicle motion model for improved predictions and situation assessment,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1209–1219, 2011.
- [28] R. Pandita and D. Caveney, “Preceding vehicle state prediction,” in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1000–1006, Gold Coast City, Australia, June 2013.
- [29] T. Hülhnagen, I. Dengler, A. Tamke, T. Dang, and G. Breuel, “Maneuver recognition using probabilistic finite-state machines and fuzzy logic,” in *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium (IV)*, pp. 65–70, La Jolla, CA, USA, June 2010.
- [30] B. Morris, A. Doshi, and M. Trivedi, “Lane change intent prediction for driver assistance: on-road design and evaluation,” in *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 895–901, Baden-Baden, Germany, June 2011.
- [31] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, “Vehicle trajectory prediction based on motion model and maneuver recognition,” in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4363–4369, Tokyo, Japan, November 2013.
- [32] D. Petrich, T. Dang, D. Kasper, G. Breuel, and C. Stiller, “Map-based long term motion prediction for vehicles in traffic environments,” in *Proceedings of the 2013 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 2166–2172, The Hague, Netherlands, October 2013.
- [33] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier, “Learning-based approach for online lane change intention prediction,” in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 797–802, Gold Coast City, Australia, June 2013.
- [34] W. Yao, H. Zhao, F. Davoine, and H. Zha, “Learning lane change trajectories from on-road driving data,” in *Proceedings of the 2012 IEEE Vehicles Symposium (IV)*, pp. 885–890, Alcalá de Henares, Madrid, Spain, June 2012.
- [35] R. Schubert, E. Richter, and G. Wanielik, “Comparison and evaluation of advanced motion models for vehicle tracking,” in *Proceedings of the IEEE 2008 11th International Conference on Information Fusion*, pp. 1–6, Cologne, Germany, June–July 2008.
- [36] A. Berthelot, A. Tamke, T. Dang, and G. Breuel, “Handling uncertainties in criticality assessment,” in *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 571–576, Baden-Baden, Germany, June 2011.
- [37] A. Tamke, T. Dang, and G. Breuel, “A flexible method for criticality assessment in driver assistance systems,” in *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 697–702, Baden-Baden, Germany, June 2011.
- [38] N. Mattern, R. Schubert, and G. Wanielik, “High-accurate vehicle localization using digital maps and coherency images,” in *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium (IV)*, pp. 462–469, La Jolla, CA, USA, June 2010.
- [39] M. Nitti, V. Pilloni, G. Colistra, and L. Atzori, “The virtual object as a major element of the internet of things: a survey,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1228–1240, 2016.
- [40] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, “Routing in internet of vehicles: a review,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2339–2352, 2015.
- [41] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, “Estimation of vehicle trajectories from observed discrete positions and next-generation simulation program (ngsim) data,” in *Proceedings of the TRB 2009 Annual Meeting*, Washington, DC, USA, January 2009.
- [42] K. M. Alam, M. Saini, and A. El Saddik, “Toward social internet of vehicles: concept, architecture, and applications,” *IEEE Access*, vol. 3, pp. 343–357, 2015.
- [43] F. Wang, S. Wang, J. Li, Z. Liu, and Q. Sun, “An overview of internet of vehicles,” *China Communications*, vol. 11, no. 10, pp. 1–15, 2014.
- [44] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, and Z. Tang, “Ant colony optimization with clustering for solving the dynamic location routing problem,” *Applied Mathematics and Computation*, vol. 285, pp. 149–173, 2016.



Hindawi

Submit your manuscripts at
www.hindawi.com

