Driving Style Classification Using a Semi-Supervised Support Vector Machine

Article *in* IEEE Transactions on Human-Machine Systems · August 2017

DOI: 10.1109/THMS.2017.2736948

4 authors, including:

Wenshuo Wang
University of California, Berkeley
50 PUBLICATIONS **405** CITATIONS

SEE PROFILE

Lin Li
Northeast Forestry University
4 PUBLICATIONS **36** CITATIONS

SEE PROFILE

Driving Style Classification Using a Semisupervised Support Vector Machine

Wenshuo Wang, *Student Member, IEEE*, Junqiang Xi, Alexandre Chong, and Lin Li

***Abstract*—Supervised learning approaches are widely used for driving style classification; however, they often require a large amount of labeled training data, which is usually scarce in a real-world setting. Moreover, it is time-consuming to manually label huge amounts of driving data due to uncertainties of driver behavior and variances among the data analysts. To address this problem, a semisupervised approach, a semisupervised support vector machine (S3VM), is employed to classify drivers into aggressive and normal styles based on a few labeled data points. First, a few data clusters are selected and manually labeled using a k-means clustering method. Then, a specific differentiable surrogate of a loss function is developed, which makes it feasible to use standard optimization tools to solve the nonconvex optimization problem. One of the most popular quasi-Newton algorithms is then used to assign the optimal label to all of the training data. Finally, we compare the S3VM method with a support vector machine method for classifying driving styles from different amounts of labeled data. Experiments show that the S3VM method can improve the classification accuracy by about 10% and reduce the labeling effort by using only a few labeled data clusters among huge amounts of unlabeled data.**

***Index Terms*—Driving style classification, longitudinal driving behavior, nonconvex optimization, quasi-Newton (QN) methods, semisupervised support vector machine (S3VM).**

NOMENCLATURE

$\mathcal{R}^d$	d-dimensional vector space.
$\mathcal{X}, \mathcal{Y}$	Set of input variables, set of labels.
$\mathcal{S}^{(l)}, \mathcal{S}^{(u)}, \mathcal{S}^{(t)}$	Labeled, unlabeled, testing datasets.
$\|(\cdot)\|_2$	2-normal of $(\cdot)$.
$\mathcal{H}$	Reproduce Hilbert kernel space.
$\mathcal{L}^1, \mathcal{L}^2$	Loss function.
$\mathcal{L}_{\text{hinge}}$	Hinge loss function.
$\mathcal{F}$	Set of available function for f.
$k(\cdot,\cdot)$	Kernel function.
$\Phi(\cdot,\cdot)$	Feature map.
$\mathbb{R}$, $\mathbb{R}^+$	Set of real numbers, set of positive numbers in range $[0,\infty)$.
$\mathbb{N}^+$	Set of positive integer.
$\mathbb{C}$	Set of complex numbers.
$\boldsymbol{H}$	Hessian approximation matrix.
$\boldsymbol{I}$	Identity matrix.
$\mathbf{x}, y$	Feature vector, data label.
v_x	Vehicle speed.
θ	Throttle opening.
σ	Width of the Gaussian RBF kernel.
λ, λ'	Weight coefficients.

Manuscript received May 16, 2016; revised October 5, 2016, March 6, 2017, and June 28, 2017; accepted July 29, 2017. This work was supported by China Scholarship Councils. This paper was recommended by Associate Editor M. Tanelli. (*Corresponding author: Junqiang Xi.*)

W. Wang is with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China, and also with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: wwsvdc2015@berkeley.edu).

J. Xi and L. Li are with the Department of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: xijunqiang@bit.edu.cn; elbert.lin.li@gmail.com).

A. Chong is with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: alex.chong@berkeley.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2017.2736948

I. INTRODUCTION

A. Motivation

DRIVING style classification plays a vital role in a large variety of realms such as human-centric vehicle control systems [1]–[6], intelligent transportation systems [7], road safety [8], and power management for electric vehicles [9]. Taking eco-driving for example, calm drivers tend to have a lower fuel rate than aggressive drivers in similar situations [10], [11]. Previous research has also shown that by incorporating the knowledge of driving styles into power management, fuel consumption can be significantly reduced [12], [13]. In addition, the National Highway Traffic Safety Administration research shows that more than 23% of deaths in traffic accidents are related to driving styles in the United States [14]. Therefore, there is a strong need for an efficient method to model, classify, and understand various driving styles.

B. Related Research

An indirect way for understanding driving styles is based on an appropriate driver model, which can be used to design personalized [2], [15]–[19], safety-oriented [20], style-oriented [21], or driving skills-oriented [22] driver assistance systems. Another way is to directly analyze driving styles using pattern classification techniques [23] based on experimental driving data. Techniques for classifying driving styles in the literature can be roughly divided into supervised methods and unsupervised methods as shown in Fig. 1.

1) Supervised Classification Method: A supervised classification method can predict a label y^* of an unknown input x^* from label sets $\mathcal{Y}$ based on given pairs of data $(x, y) \in \mathcal{X} \times \mathcal{Y}$. The training data should be first labeled according to data analysts'

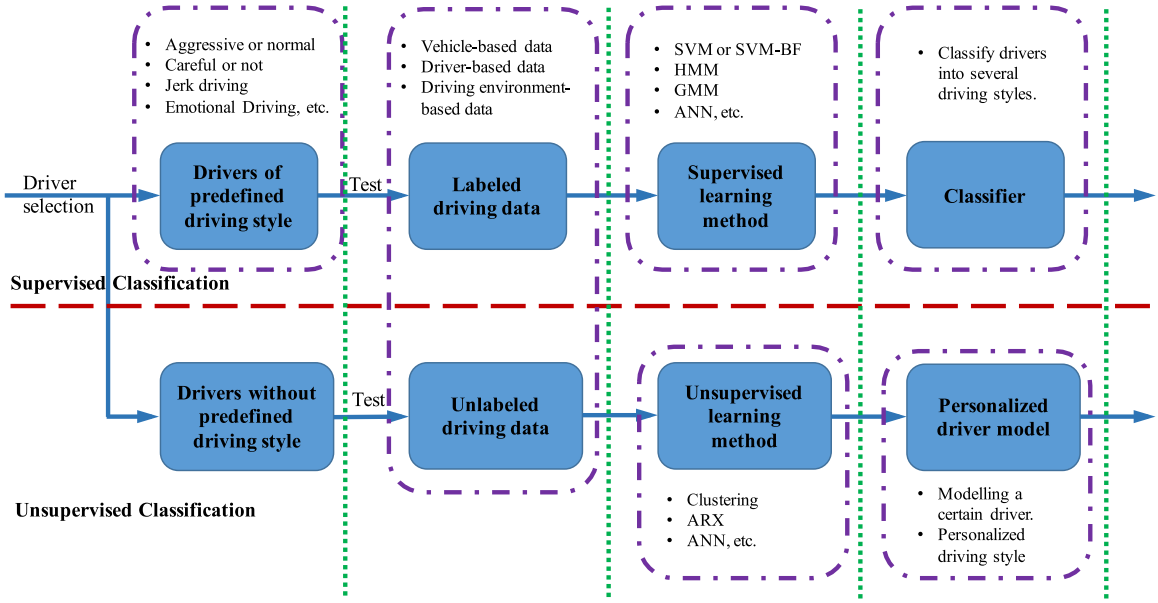


Fig. 1. Schematic diagram of supervised and unsupervised methods for driving style classification.

knowledge. For instance, Ly *et al.* [24] used a support vector machine (SVM) to recognize driving styles based on the labeled information of the vehicle's inertial sensors. Lin *et al.* [25] collected training data from drivers, who were prelabeled with six driving patterns using rule-based approaches according to data analysts' prior knowledge, in order to learn a driving style classifier. The advantages of rule-based approaches are traceability and simpler implementations for specific scenarios and driver behaviors. However, for complex driving behaviors, rule-based approaches require substantial effort in order to be extended to more general driving behavior models. Wahab *et al.* [26] applied the driving style questionnaire (DSQ) method to define individuals' driving styles and then collected driving data from these drivers to train a classifier. But usually, the DSQ method requires a significant effort to investigate driver behaviors. Aoude *et al.* [27] prelabeled the driving data into two driving styles (i.e., compliant and violating) and trained a classifier using a combination of a SVM-Bayesian filter (SVM-BF) and a hidden Markov model (HMM). Sundbom *et al.* [28] collected the labeled data from drivers who drive with normal or aggressive driving styles in order to train a classifier based on a probabilistic autoregressive eXogenous model. As mentioned above, most of the approaches in the literature normally collect the labeled driving data from the drivers of predefined types (see Fig. 1). However, uncertainties of driver behaviors and variances in data analysts' knowledge would cause the training data to be incorrectly labeled or used. For example, since an aggressive driver may show normal driving behavior at different times [11], the data collected from an aggressive driver could contain data that are labeled as normal or vice versa. In summary, the main drawbacks of methods presented in the above-mentioned literature are the following.

1) Manually labeling large amounts of training data requires excessive effort.
2) Manually labeling driving data can also cause subjective labels.
3) Data analysts should be very familiar with driver behaviors.

2) Unsupervised Classification Method: An unsupervised classification method can assign the unknown data into categories by fully mining the underlying sources of unlabeled data. For example, a clustering-based approach was directly applied to identify driving styles [29] and to describe driver behaviors [30]. Shi *et al.* [31] trained a personalized driver model using the unlabeled data of the instrumented vehicle based on an artificial neural network (ANN). Bender *et al.* [32] inferred driving behavior from naturalistic driving data by combing a Bayesian multivariate linear model with a sequence segmentation algorithm. For most of the research regarding unsupervised methods, machine learning techniques are often employed to directly learn a model and then to predict driver behavior or driving style. The unsupervised classification method would not require as much labeling effort; however, it could cause unexpected classification results, such as not fully reflecting the real and expected driving characteristics.

C. Key Contribution

In order to enhance the performance of driving style classification and reduce labeling efforts, a semisupervised method, namely, a semisupervised support vector machine (S3VM), was developed by combining the advantages of supervised and unsupervised methods. The S3VM is able to solve the pattern classification problem with a very small amount of labeled data. It has been widely used in the fields of robotics [33], [34] and driver distraction detection [35], but it has not yet been utilized for driving style classification because most of the researchers usually collect the labeled training data from drivers of predefined styles, as shown in Fig. 1.

In this paper, we mainly focus on drivers' longitudinal driving behaviors regarding aggressive and normal driving styles when driving on curvy roads. The S3VM method takes the

additional information of unlabeled driving data into account to capture more underlying characteristics of the driving data. First, the driving data are preprocessed using a k-means clustering method, and then, a few distinct data clusters are selected and labeled initially using a rule-based approach. Second, the S3VM approach generates an optimal decision boundary by fully using the knowledge of the unlabeled data and labeled data. To solve the nonconvex optimization problem with the S3VM, we then introduce a differentiable surrogate of the loss function and a quasi-Newton (QN) algorithm, which makes it feasible to employ known optimization tools. Finally, experiments are conducted to compare the S3VM and SVM approaches.

D. Paper Organization

The remainder of this paper is organized as follows. Section II discusses the basic concepts of SVMs and S3VMs and also explains a gradient-based optimal algorithm for the problem solution. Section III presents the experiment procedure and data collection. Section IV analyzes the experiment results. Section V provides a further discussion. Finally, conclusions are stated in Section VI.

II. Methods for Driving Style Classification

In this section, two methods for driver style classification are introduced, including SVMs and S3VMs. Subsequently, an appropriate kernel selection for both of these two methods is presented. Finally, to make full use of the gradient-based optimization tools, a specific differentiable surrogate of the loss function is introduced.

A. Support Vector Machine

The basic theory of the SVM is briefly revisited for convenience based on [36] and [37]. Given a labeled dataset $\mathcal{S}^{(l)} = \{(\mathbf{x}_i^{(l)}, y_i^{(l)})\}_{i=1}^n$ with $\mathbf{x}_i^{(l)} = (x_{1,i}^{(l)}, \ldots, x_{d,i}^{(l)})^\intercal \in \mathcal{R}^d$, the supervised classification method is to learn a function $f \in \mathcal{F}: \mathbf{x}^{(l)} \mapsto y$. This function can then generate a hyperplane by maximizing the margin between different classes. Here, $y_i \in \mathcal{Y}$ is the label of some example $\mathbf{x}_i^{(l)}$ and $\mathcal{Y}$ is the set of labels. We assume that $\{(\mathbf{x}_i^{(l)}, y_i)\}_{i=1}^n$ are the samples independent and identically distributed from some set of distributions ranging over $\mathcal{X} \times \mathcal{Y}$. Thus, the SVM can be treated as an instance of regularization problems as described in [36], [38]

$$\inf_{f \in \mathcal{H}, f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y_i', f(\mathbf{x}_i^{(l)})) + \lambda \|f\|_{\mathcal{H}}^2 \right\} \tag{1}$$

where $\lambda \in \mathbb{R}^+$, $\mathcal{L}: \mathcal{Y} \times \mathbb{R} \mapsto [0, \infty)$ is the loss function, and $\|f\|_{\mathcal{H}}^2$ is the squared norm Hilbert space $\mathcal{H} \subseteq \mathbb{R}^{\mathcal{X}} = \{f: \mathcal{X} \mapsto \mathbb{R}\}$. In (1), the first term is used to measure the loss caused by the prediction function on the labeled training data and the second term is used to penalize complex functions f with a reproducing kernel Hilbert space. One of the most common hinge loss functions is chosen and defined as

$$\mathcal{L}_{\text{hinge}}(y, t) = \max\{0, 1 - yt\}, \quad y = \pm 1, t \in \mathbb{R}. \tag{2}$$

However, the labeled training data of driving behaviors are usually not linearly separable. To deal with this problem in an efficient way, we map the input space $\mathcal{R}^d$ to a feature space $\mathcal{H}$ using common kernel tricks:

$$\begin{aligned} k(\mathbf{x}^{(l)}, \mathbf{x}'^{(l)}) &= \langle \Phi(\mathbf{x}^{(l)}), \Phi(\mathbf{x}'^{(l)}) \rangle_{\mathcal{H}}, \\ \mathcal{H} &\leftarrow \mathcal{R}^d \times \mathcal{R}^d \end{aligned} \tag{3}$$

where $\Phi(\cdot, \cdot): \mathcal{X} \mapsto \mathcal{H}$ is the feature map. Then, the feature map is able to provide a linear separation of the data in a high-dimensional feature space $\mathcal{H}$, where one could find the maximum margin hyperplane as follows:

$$y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i^{(l)}) + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N \tag{4}$$

with $\mathbf{w} \in \mathcal{R}^h$ and relaxation factors $\xi_i \geq 0$. The nonlinear separating hyperplane can then be found by solving

$$\begin{aligned} \max_{f \in \mathcal{F}, f \in \mathcal{H}, \alpha_i \in \mathbb{R}} \quad & J(\alpha_i) = \sum_{i=1}^{N} \alpha_i - \lambda \|f\|_{\mathcal{H}}^2 \\ \text{s.t.} \quad & \sum_{i=1}^{N} y_i \alpha_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, N \end{aligned} \tag{5}$$

where $\|f\|_{\mathcal{H}}^2 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$.

B. Semisupervised Support Vector Machine

For the S3VM, we assume that we have a dataset, $\mathcal{S}$, consisting of labeled and unlabeled data points

$$\begin{aligned} \mathcal{S} &= \{\underbrace{\mathcal{S}^{(l)}}_{n}, \underbrace{\mathcal{S}^{(u)}}_{m}\} \\ &= \{\underbrace{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)}_{\text{Labeled data}}, \underbrace{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}}_{\text{Unlabeled data}}\}. \end{aligned}$$

Our goal is to be able to learn a model that can assign the correct label to the new test data, $\mathcal{S}^{(t)}$, by using the underlying information of the labeled and unlabeled data. Thus, we search for a function, f, and a labeling vector, $\mathbf{y}^* = (y_{n+1}^*, \ldots, y_{n+m}^*)^\intercal$, that are optimal with respect to

$$\begin{aligned} \min_{f \in \mathcal{F}, f \in \mathcal{H}, \mathbf{y}^* \in \{-1, +1\}} \quad & \underbrace{\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}^1 \left(y_i', g(\mathbf{x}_i^{(l)}) \right)}_{\text{Labeled data measurement}} \\ & + \underbrace{\frac{\lambda'}{m} \sum_{i=n+1}^{n+m} \mathcal{L}^2 \left(y_i^*, f\left(\mathbf{x}_{n+i}^{(u)} \right) \right)}_{\text{Unlabeled data measurement}} \\ & + \underbrace{\lambda \|f\|_{\mathcal{H}}^2}_{\text{Regularization term}} \end{aligned} \tag{6}$$

where the induced loss function is $\mathcal{L}^2(f(\mathbf{x}^{(u)})) := \max\{0, 1 - |f(\mathbf{x}^{(u)})|\}$, $\lambda = \frac{1}{2}$, $\lambda' = 1$. The proposed solution is formulated based on the following set of assumptions.

1) $n \ll m$, meaning that a limited amount of labeled data are used and sufficient unlabeled data are available.
2) The labeled data and unlabeled data collected from the same driver are subject to the same distribution, i.e., $\mathcal{S}^{(l)}, \mathcal{S}^{(u)} \sim \mathcal{X} \times \mathcal{Y}$.

3) Data from the same class lie in the same cluster. This is also known as the *cluster assumption* [39].
4) If two points x_1 and x_2 in a high-density region are close to each other, then they should have similar outputs y_1 and y_2. This is also known as the *smoothness assumption* [39].

By following [38], the optimal solution to $f(\cdot)$ in (1) and (6) is in the form

$$f(\cdot) = \sum_{i=1}^{n+m} \alpha_i k(\mathbf{x}_i, \cdot). \tag{7}$$

Therefore, the main task of searching for an optimal solution for (6) is equal to finding the optimal coefficients $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^{n+m}$.

C. Kernel Selection

To apply the kernel trick (3), an appropriate kernel selection is essential to obtaining satisfactory results using the SVM and the S3VM. Various kernel functions are available in various practical problems, and selecting an appropriate kernel function is rather important for any given application. In this research, we follow the typical procedure suggested by Ben-Hur and Weston [40]: trying a linear kernel first and then checking if the performance is better than using a nonlinear kernel such as a Gaussian or polynomial kernel. For nonlinear kernels, Ben-Hur and Weston [40] found that the Gaussian kernel usually outperforms the polynomial kernel in both accuracy and convergence time. Therefore, in this paper, the linear kernel and the Gaussian radial basis function (RBF) kernel are used. They are presented as follows.

Definition 1: The linear kernel is defined as [40]

$$k_w(x, x') = \sum_{i}^{d} w_i x_i \tag{8}$$

where $d \in \mathbb{N}$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{C}^d$ and $x' = (x'_1, x'_2, \ldots, x'_d)$.

Definition 2: The Gaussian RBF kernel is defined as [36]

$$k_{\sigma,\mathbb{C}^d} = \exp\left(-\sigma^2 \sum_{j=1}^{d} (x_j - x'_j)^2\right) \tag{9}$$

where $d \in \mathbb{N}$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{C}^d$ and $x' = (x'_1, x'_2, \ldots, x'_d)$. Then, $k_{\sigma,\mathbb{C}^d}$ is a kernel on $\mathbb{C}^d$ and its restriction $k_\sigma = (k_\sigma, \mathbb{C}^d)_{|\mathbb{R}^d \times \mathbb{R}^d}$ is an $\mathbb{R}$-valued kernel. Moreover, k_σ can be calculated by

$$k_\sigma(x, x') = \exp\left(-\frac{\|x - x'\|_2^2}{\sigma}\right), \qquad x, x' \in \mathbb{R}^d. \tag{10}$$

The width σ of the Gaussian BFR kernel is tuned using a leave-one-out cross-validation (CV) method to avoid the overfitting issues and finally set to $\sigma = 2$.

D. Gradient-Based Optimization

In terms of optimizing (5) and (6), one of the main challenges is that the hinge loss function $\mathcal{L}_{\text{hinge}}$ of the objective function is not differentiable, which strongly limits the application of most off-the-shelf optimization tools. Therefore, we introduce a modified, differentiable hinge loss function to approximate the original hinge loss function, thus making it possible to use the QN method.

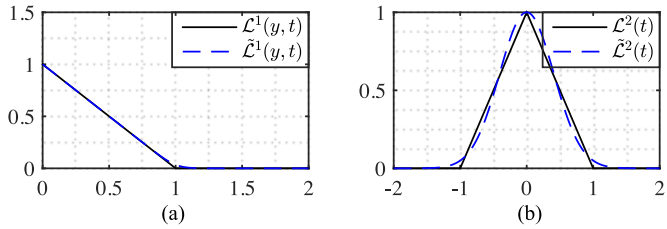


Fig. 2. (a) Hinge loss $\mathcal{L}^1(y,t)$ and its differentiable surrogate $\tilde{\mathcal{L}}^1(y,t)$. (b) Hinge loss $\mathcal{L}^2(t)$ and its differentiable surrogate $\tilde{\mathcal{L}}^2(t)$.

1) Differentiable Surrogates: Since the original objective functions are not differentiable, the differentiable surrogate for $\mathcal{L}^1(y,t)$ in (1) is approximated in [38], [41] [see Fig. 2(a)]:

$$\begin{aligned} &\mathcal{L}^1(y, f(\mathbf{x}^{(l)})) = \max\{0, 1 - yt\} \\ \Rightarrow\ &\tilde{\mathcal{L}}^1(y, f(\mathbf{x}^{(l)})) = \frac{1}{\gamma}\ln(1 + \exp(\gamma(1 - y'_i) f(\mathbf{x}_i^{(l)}))), \\ &\qquad i = 1, 2, \ldots, n, \qquad \gamma = 20 \end{aligned} \tag{11}$$

and the induced loss function $\mathcal{L}^2(f(\mathbf{x}^{(u)}))$ in (6) is approximated as shown in Fig. 2(b):

$$\begin{aligned} &\mathcal{L}^2(f(\mathbf{x}^{(u)})) = \max\{0, 1 - |f(\mathbf{x}^{(u)})|\} \\ \Rightarrow\ &\tilde{\mathcal{L}}^2(f(\mathbf{x}^{(u)})) = \exp(\zeta(f(\mathbf{x}_j^{(u)}))^2), \\ &\qquad j = n+1, \ldots, n+m, \qquad \zeta = -3. \end{aligned} \tag{12}$$

Combining (6)−(12), the objective function to be minimized is modified to be

$$\begin{aligned} G_{\lambda'}(\boldsymbol{\alpha}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\gamma}\times & \\ \underbrace{\ln\left\{1 + \exp\left[\gamma\left(1 - y'_i \sum_{i=1}^{n+m}\alpha_i k(\mathbf{x}_i^{(l)}, \cdot)\right)\right]\right\}}_{\text{Labeled data measurement}} & \\ + \underbrace{\frac{\lambda'}{m}\sum_{j=n+1}^{n+m}\exp\left\{\zeta\left[\sum_{j=1}^{n+m}\alpha_j k(\mathbf{x}_j^{(u)}, \cdot)\right]^2\right\}}_{\text{Unlabeled data measurement}} & \\ + \underbrace{\lambda\sum_{i=1}^{n+m}\sum_{j=1}^{n+m}\alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j)}_{\text{Regularization term } \|f\|_{\mathcal{H}}^2}. & \end{aligned} \tag{13}$$

2) QN Method: Due to the nonlinearity of (13) with respect to model parameters $\boldsymbol{\alpha}$, direct maximization of the function is impossible. Therefore, we use an iterative version of the QN method [38], [42] to search for the optimal solution as shown in Algorithm 1.

To get the optimal solution $\{\alpha_i\}_{i=1}^{n+m}$, the influence of the unlabeled part in (6) is increased via $\{\pi_i < \pi_{i+1}\}_{i=1}^{\tau}$ for each

Algorithm 1: Algorithm for QN-S3VM.

A labeled training set $\mathcal{S}^{(l)} = \{(\mathbf{x}_i, y_i))\}_{i=1}^{n}$, an unlabeled training set $\mathcal{S}^{(u)} = \{\mathbf{x}_{n+j}\}_{j}^{m}$, model parameters λ' and λ, an initial inverse Hessian approximation $\boldsymbol{H}_0$, and a sequence $0 < \pi_1 < \cdots < \pi_\tau$.
Initialize $\boldsymbol{\alpha}_0$ using the supervised model defined by (13) with $\lambda' = 0$, initially ignoring the unlabeled part. For $\boldsymbol{H}_0$, one can assign $\boldsymbol{H}_0 = \gamma \boldsymbol{I}$ for $\gamma > 0$.
for i=1 to τ **do**
    $k = 0$
    while $\|\nabla G_{\pi_i \cdot \lambda'}(\boldsymbol{\alpha}_k)\| \not< \epsilon$ and $k < N_{iter}^{lim}$ **do**
        Calculate the search direction $\boldsymbol{p}_k \leftarrow$ (14)
        Update $\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \beta_k \boldsymbol{p}_k$
        Update $\boldsymbol{H}_{k+1} \leftarrow$ (15)
        $k = k + 1$
    end while
    $\boldsymbol{\alpha}_0 = \boldsymbol{\alpha}_k$
end for

step. The sequence $\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \beta_k \boldsymbol{p}_k$ of candidate solutions is generated using the Broyden–Fletcher–Goldfarb–Shanno algorithm [38], where the step length β_k is computed by the line search, $\boldsymbol{p}_k$, calculated as

$$\boldsymbol{p}_k = -\boldsymbol{H}_k \nabla G_{\pi_i \cdot \lambda'}(\boldsymbol{\alpha}_k). \tag{14}$$

In addition, $\boldsymbol{H}_k$ is updated by

$$\begin{aligned} \boldsymbol{H}_{k+1} &= (\boldsymbol{I} - \rho_k \boldsymbol{s}_k \boldsymbol{z}_k^{\mathsf{T}}) \boldsymbol{H}_k (\boldsymbol{I} - \rho_k \boldsymbol{s}_k \boldsymbol{z}_k^{\mathsf{T}}) + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathsf{T}} \\ \boldsymbol{z}_k &= \nabla G_{\pi_i \cdot \lambda}(\boldsymbol{\alpha}_{k+1}) - \nabla G_{\pi_i \cdot \lambda}(\boldsymbol{\alpha}_k) \\ \boldsymbol{s}_k &= \boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k \\ \rho_k &= (\boldsymbol{z}_k^{\mathsf{T}} \boldsymbol{s}_k)^{-1}. \end{aligned} \tag{15}$$

Using Algorithm 1, the new candidate solution can be computed until the convergence criterion is fulfilled, namely, $\|\nabla G_{\pi_i \cdot \lambda'}(\boldsymbol{\alpha}_k)\| < \epsilon = 10^{-5}$ or the number of iterations exceeds a preset threshold $N_{\text{iter}}^{\text{lim}} = 500$.

III. Experiments and Data Collection

A. Equipments and Scenarios

All of the training and test data were collected in a stationary driving simulator consisting of five main parts (see Fig. 3): Game-type driving peripherals, virtual reality software (Vizard 5.0), vehicle dynamics models, computer, and human drivers. The game-type driving peripherals (Logitech G27) were used to collect drivers' operation signals such as the steering wheel angle, brake pedal displacement, and throttle opening. The driving scenario consisted of vehicles, roads, and traffic facilities designed by 3Ds-Max and Vizard software. An eight-degree-of-freedom four-wheel vehicle model was built in MATLAB/Simulink, including longitudinal and lateral dynamics and roll and yaw angle measurements. The vehicle model was then validated and calibrated using Carsim [43]. The simulator recorded a variety of parameters such as the throttle opening,

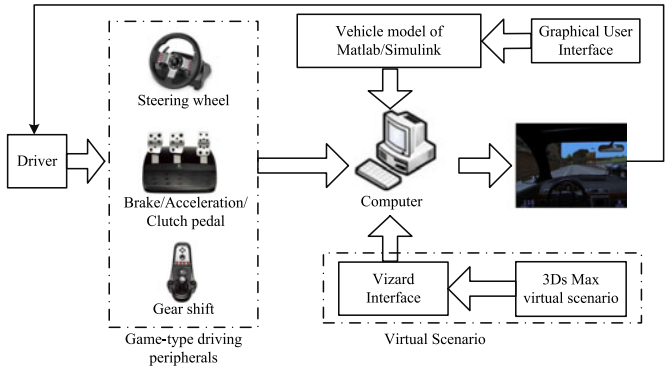


Fig. 3. Driving simulator for collecting driving data.

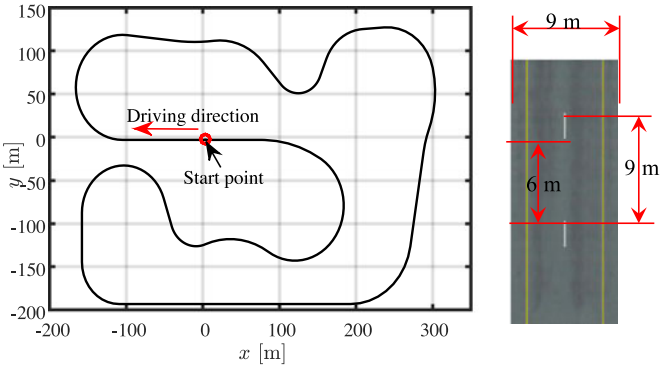


Fig. 4. Road profiles designed using 3Ds-Max.

braking force, vehicle position, steering wheel angle, longitudinal and lateral speed, yaw rate, roll angle, and acceleration at 50 Hz.

A simple curve-negotiating task on a flat road was then chosen to reflect drivers' longitudinal driving behaviors and show benefits of the proposed method. Drivers were instructed to stay in their lane and follow the reference path specified by the Carsim data (see Fig. 4). The length of curvy road was 2247 m. The path consisted of a set of simple path elements that included straight segments (zero curvature), arcs (constant curvature), and clothoids (linearly varying curvature). Due to the limitations of experimental equipment, the effects of road slope, weather condition, and traffic flow were not considered in these experiments.

B. Data Collection

1) Feature Selection: To investigate drivers' driving styles, speed and acceleration/deceleration were used to capture driver characteristics [44]. Shi *et al.* [31] found that the braking signal was not a strong enough indicator for differentiating aggressive and normal drivers. Therefore, in this work, the brake pedal signal was not set as a feature. However, the authors in [31] stated that the throttle opening could easily be used to distinguish driving styles. In this paper, we primarily focus on the longitudinal driving behavior when the vehicle is driving on a curvy road since it greatly influences fuel consumption [10]. We selected the longitudinal speed, v_x, and the throttle opening, $\theta \in [0, 1]$,

as the feature parameters $\mathbf{x} = (v_x, \theta)$ to describe driving styles. Features are detailed as follows.

1) *Vehicle Speed* (v_x)*:* Vehicle speed can reflect the drivers' longitudinal driving behaviors. If a driver prefers a high vehicle speed when driving on a curvy road, then the driver can be identified as an aggressive type. On the other hand, if a driver navigates the curvy road at a low speed, then the driver can be classified as a normal type.
2) *Throttle Opening* (θ)*:* The throttle opening is directly controlled by drivers and, therefore, is directly related to acceleration/deceleration. The longitudinal and lateral acceleration of the vehicle can also reflect driving patterns [11]. Drivers who prefer a larger longitudinal acceleration/deceleration tend to be more aggressive.

Based on the above discussion, we can conclude that the combination of vehicle speed and throttle opening can capture the driver's acceleration characteristics. For instance, driving on a flat road at low vehicle speed with a large throttle opening results in a large acceleration, and conversely, driving at high vehicle speed with a small throttle opening provides a large deceleration. Thus, in order to reduce redundant features, we do not include the acceleration because we can combine the throttle opening and vehicle speed to compute it. Note that the driver's steering characteristics specifically are not included in this work because we mainly focus on the driver's longitudinal driving behavior, even though steering characteristics do vary among individuals. For more information regarding modeling and recognizing driver steering characteristics, readers are referred to [45] and [46].

2) Driver Participant: A total of 20 licensed drivers participated in this study. Their average age was 25.60 years with a standard deviation of 2.70 years. Their average driving experience was 25.5 months with a standard deviation of 15.4 months.

3) Experiment Procedure: The experimental task was for them to follow a flat curve according to their driving preference. The task started with the vehicle stationary at the beginning of the course $(x_0, y_0) = (0, 0)$ (see Fig. 4). Participants accelerated up to a speed of about 30 km/h with a full throttle opening on the straight part of the road and then drove subsequently as they naturally would. The drivers were able to become comfortable with the experiment before entering the curve. In this paper, the vehicle's starting behavior was not considered, and thus, the data from the vehicle's starting state (i.e., $v_x \leq 30$ km/h) were removed from the training data. During the experiments, in order to ensure consistency of the used data, participants adhered to the following rules.

1) All drivers were in normal mental and physical states. Regarding mental states, participants were not allowed to be drowsy or drunk.
2) Each driver spent about 10 min familiarizing themselves with the driving simulator and driving scenarios, ensuring that they would not change their driving style drastically during their trials.
3) Other distracting activities such as talking with others or making or answering a call were not allowed.

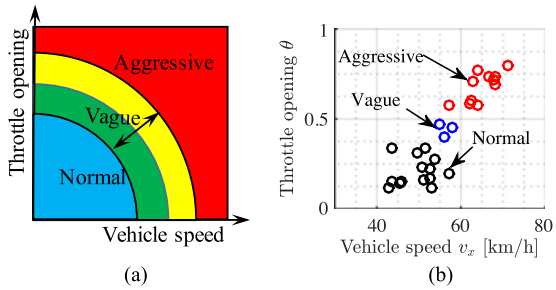


Fig. 5. Examples of (a) the rule-based scheme and (b) a few labeled data points.

4) One trial was completed when the drivers drove back to the starting point. Each driver had a (1−2 min) break between trials.
5) The driving data were invalid if the vehicle spun out of control or if the driver drove off the course. The driver was then instructed to repeat the experiment again.

Each driver conducted 20 trials with 2.5 min of each trial on average.

C. Data Preprocessing

In total, 400 datasets from 20 drivers were collected and each dataset consisted of about 6000 data points (2 min × 60 × 50 Hz = 6000). The dataset of each individual driver was equally divided into four groups to get more data samples, and eventually, 1600 datasets were obtained. To validate our proposed method, all driving datasets were divided into three disjoint parts: labeled dataset $\mathcal{S}^{(l)} = \{\mathbf{x}_i\}_{i=1}^{n}$, unlabeled dataset $\mathcal{S}^{(u)} = \{\mathbf{x}_j\}_{j=1}^{m}$, and test dataset $\mathcal{S}^{(t)} = \{\mathbf{x}_{i=1}\}_{i=1}^{r}$, where $n = 290$, $m = 1300$, and $r = 10$.

1) Manually Labeling Data: The data mining techniques including clustering methods [30], [44] showed their powerful capability to elaborate and analyze the similarities and differences between driving behavior characteristics regarding the longitudinal driving behavior. In this paper, to make labeling work easier, each dataset in $\mathcal{S}^{(l)}$ was clustered using the k-means clustering method. For example, clustered points with a larger speed (e.g., $v_x = 80$ km/h) and a smaller throttle opening (e.g., $\theta \lesssim 0.2$) could be directly labeled as aggressive. Based on this, all distinct clusters were manually labeled by $y = \{-1, 1\}$, where -1 and 1 represent an aggressive and a normal driver, respectively. Fig. 5 provides a diagram of manually labeling the clustering points. It can be seen that some of the data can easily be labeled, while data in the "vague" region shown in Fig. 5(a) are difficult to label. On the contrary, the developed S3VM approach in this paper is able to determine a more distinct decision boundary by utilizing the underlying characteristics of the labeled and unlabeled data. We only needed to label a few clustered data that were obviously subject to aggressive or normal driving styles with red and black circles as can be seen in Fig. 5(b).

2) Training and Testing Procedures: A moving window with width $T_d = 6$ s [11] was then applied to all of the training data

in $\mathcal{S}^{(l)}$ and $\mathcal{S}^{(u)}$. All feature data located within the specified window were clustered into a single point. These data consisted of the vehicle speed and throttle opening. All of test data $\mathcal{S}^{(t)}$ were also preprocessed in the same way as the training data.

IV. Classification Results and Analysis

A. Performance Index

A widely used evaluation scheme for pattern classification is called CV, which requires that the data used for training a classifier are disjoint from the data used for testing. Therefore, we adopted the CV method to assess the classification performance for each driving style, using the SVM and S3VM approaches. To apply the CV method and show the benefits of the developed S3VM, the labeled data were used to train an SVM classifier, while the partially labeled datasets $\mathcal{S}_p^{(l)} = \{\mathbf{x}_i\}_{i=1}^{P} \subset \mathcal{S}^{(l)}$ and unlabeled datasets $\mathcal{S}^{(u)}$ were used to learn the S3VM classifier. Here, $P \in \mathbb{N}^+ < n$ is the amount of labeled data. Ten test datasets were selected from $\mathcal{S}^{(t)}$ as the test objects. We assume that for each dataset, we have

$$(\mathbf{x}_i, \mathbf{y}^{\text{real}}) = \{(\mathbf{x}_{i,1}, y_{i,1}^{\text{real}}), (\mathbf{x}_{i,2}, y_{i,2}^{\text{real}}), \ldots, (\mathbf{x}_{i,M}, y_{i,M}^{\text{real}})\}^{\intercal}$$
$$i = 1, 2, \ldots, r$$
$$j = 1, 2, \ldots, M$$

where $M \in \mathbb{N}^+$ is the length of each dataset $\mathbf{x}_i$, $\mathbf{x}_{i,j}$ is the jth element of $\mathbf{x}_i$, and $y_{i,j}^{\text{real}}$ is the corresponding real label of the jth element of $\mathbf{x}_i$. The classification accuracy for each driving dataset, $\mathbf{x}_i$, was computed by

$$\ell_i = \frac{1}{M}\sum_{j=1}^{M} \delta_{i,j} \qquad i = 1, 2, \ldots, r \tag{16}$$

with

$$\delta_{i,j} = \begin{cases} 1, & \text{if} \quad y_{i,j}^{\text{real}} = y_{i,j}^{\text{pre}} \\ 0, & \text{if} \quad y_{i,j}^{\text{real}} \neq y_{i,j}^{\text{pre}} \end{cases}$$

where $y_{i,j}^{\text{real}}, y_{i,j}^{\text{pre}} \in \{-1, 1\}$ and $y_{i,j}^{\text{pre}}$ is the predicted label of the jth data of $\mathbf{x}_i$ by the trained classifier. The average accuracy of all test data $\mathbf{x} \in \mathcal{S}^{(t)}$ is calculated by

$$\mu = \frac{1}{r}\sum_{i=1}^{r} \ell_i, \qquad r = 10. \tag{17}$$

B. Results and Analysis

Insufficient labeled data for training may not take full advantage of unlabeled data to properly reflect the reality of various driving styles. Hence, to further demonstrate and evaluate the S3VM's capability of exploiting the available data, comparative experiments were conducted using different amounts of labeled datasets $\mathcal{S}_p^{(l)}$, where $P \in \{9, 25, 45, 70, 145, 290\}$. Fig. 6 shows the comparative accuracy of SVM using the labeled data and S3VM using both labeled and unlabeled data. In other words, we varied the amount of labeled training data and then compared accuracy results of SVM and S3VM using the aforementioned

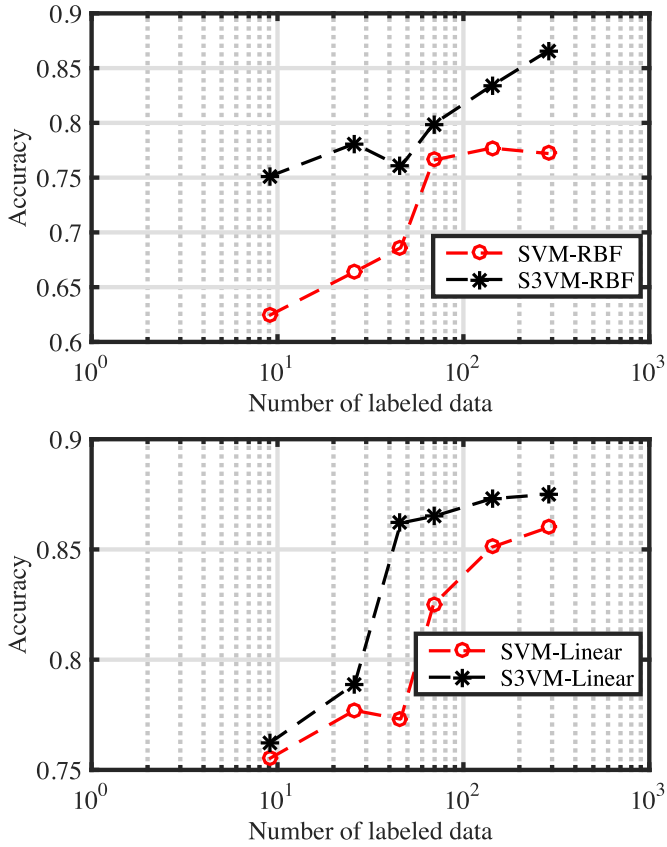


Fig. 6. Classification accuracy of using S3VM and SVM based on an RBF kernel and a linear kernel.

method. The vertical axis is the average classification accuracy μ and the horizontal axis is the amount of labeled training data on a log scale.

Figs. 7–10 show the optimal classifier boundaries for different amounts of labeled data using SVM and S3VM with a linear kernel and an RBF kernel for both. Red circles with black filled in and green circles with white filled in represent the correct classification of aggressive drivers and normal drivers, respectively. Note, however, that red circles with white filled in or green circles with black filled in represent a wrong classification. Red regions represent normal driving styles and blue regions represent aggressive driving styles. In addition, a deeper red indicates a higher probability of being normal and a deeper blue represents a higher probability of being aggressive.

1) Driving Style Analysis: From Fig. 6, we see that S3VM is able to take full advantage of both labeled and unlabeled data and, therefore, provides an optimal classification boundary, which creates a more objective result than that from rule-based approaches. Even though a small amount of labeled data must be manually and subjectively predefined, the classification boundary can be tuned and optimized by exploiting the underlying knowledge of the unlabeled data (refer to Section V-D for more discussions). Based on the classification results with $P = 290$ from Figs. 7 to 10, we can conclude that when driving on a flat curvy road without considering the influence of traffic:

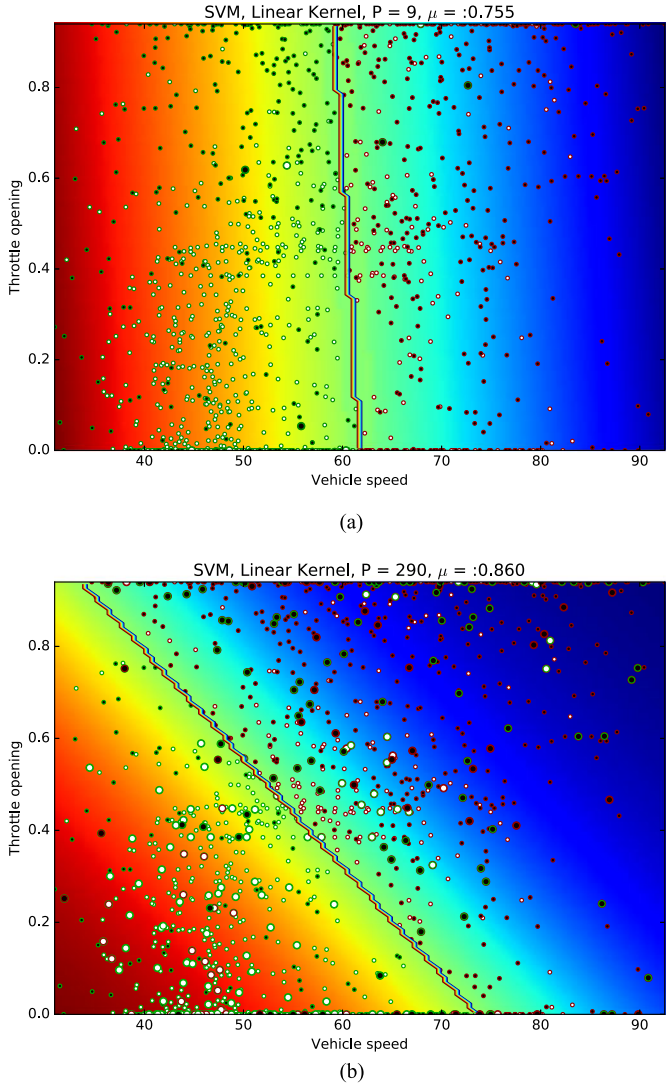


Fig. 7. Examples of results using SVM with a linear kernel. (a) $\mu = 0.755$, $P = 9$. (b) $\mu = 0.860$, $P = 290$.

1) if the vehicle speed is $30 \leq v_x \lesssim 60$ km/h and the throttle opening is $\theta \lesssim 0.45$, the driver tends to prefer small accelerations and decelerations, which indicates a normal driving style;
2) if the vehicle speed is $30 \leq v_x \lesssim 60$ km/h, but the throttle opening is $\theta \gtrsim 0.45$, then the driver tends to prefer large accelerations, which indicates an aggressive driving style;
3) if the vehicle speed is $v_x \gtrsim 60$, but the throttle opening is $\theta \lesssim 0.45$, the driver tends to prefer large decelerations, which indicates an aggressive driving style;
4) if the vehicle speed is $v_x \gtrsim 60$, but the throttle opening is $\theta \gtrsim 0.45$, the driver tends to prefer high vehicle speeds, which indicates an aggressive driving style.

We also demonstrated that the S3VM method was able to generate a more objective decision boundary, compared to rule-based methods, especially for cases where the decision boundaries between two driving styles are not clear. In addition, S3VM requires a very small amount of labeled data and hence can significantly reduce the labeling effort for training a classifier.

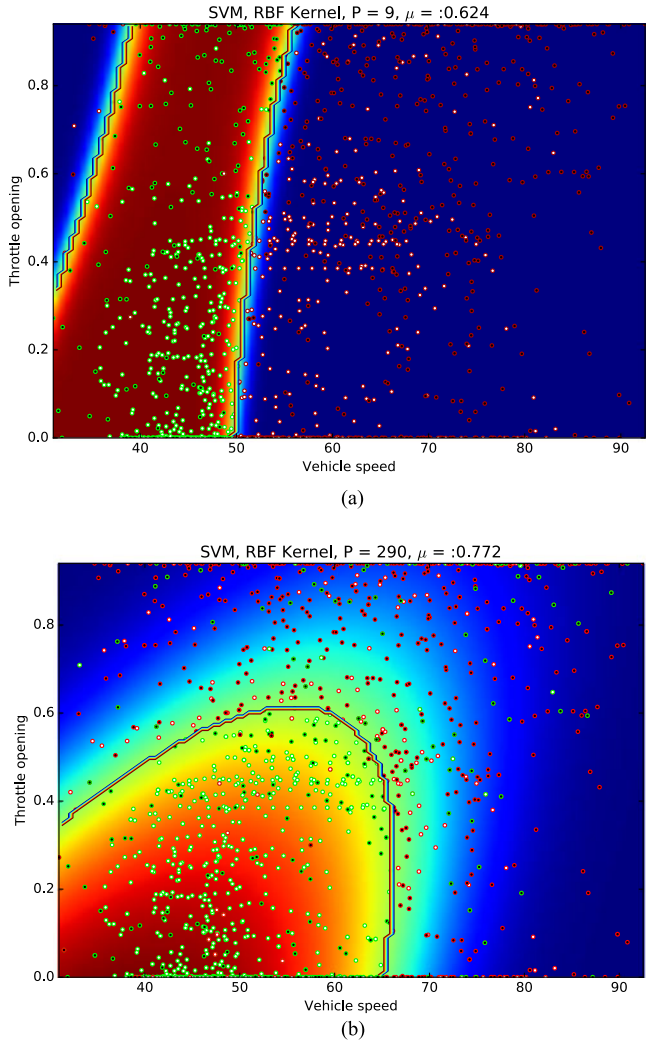


Fig. 8. Examples of results using SVM with an RBF kernel. (a) $\mu = 0.624$, $P = 9$. (b) $\mu = 0.772$, $P = 290$.

2) Performance Index Analysis: Fig. 6 indicates that S3VM generally performs better than SVM. For example, with nine labeled data points using an RBF kernel and a linear kernel, SVM achieves accuracies of 62.4% and 75.5%, respectively, while S3VM achieves accuracies of 75.1% and 76.2%, respectively. We also found that S3VM still performs well even with a very small amount of labeled data. With only 25 labeled data points, S3VM obtains accuracies of 78.1% and 76.9% using the two kernels mentioned above, while SVM obtains accuracies of 66.4% and 76.7%. As expected, the performance of S3VM converges to a constant when more labeled data are utilized. In other words, adding more unlabeled data points does not help improve classification performance significantly because there are enough labeled data points to accurately estimate the classifier parameters. For example, with 290 driving data points using the RBF kernel, the classification accuracy increases from 77.2% with the SVM method [see Fig. 8(b)] to 86.6% with the S3VM method [see Fig. 10(b)].

We see that S3VM performs better than SVM when using a linear kernel. For example, when using 45, 70, and 145 labeled points, the accuracy increases by 8.9%, 4.1%, and 3.2%,

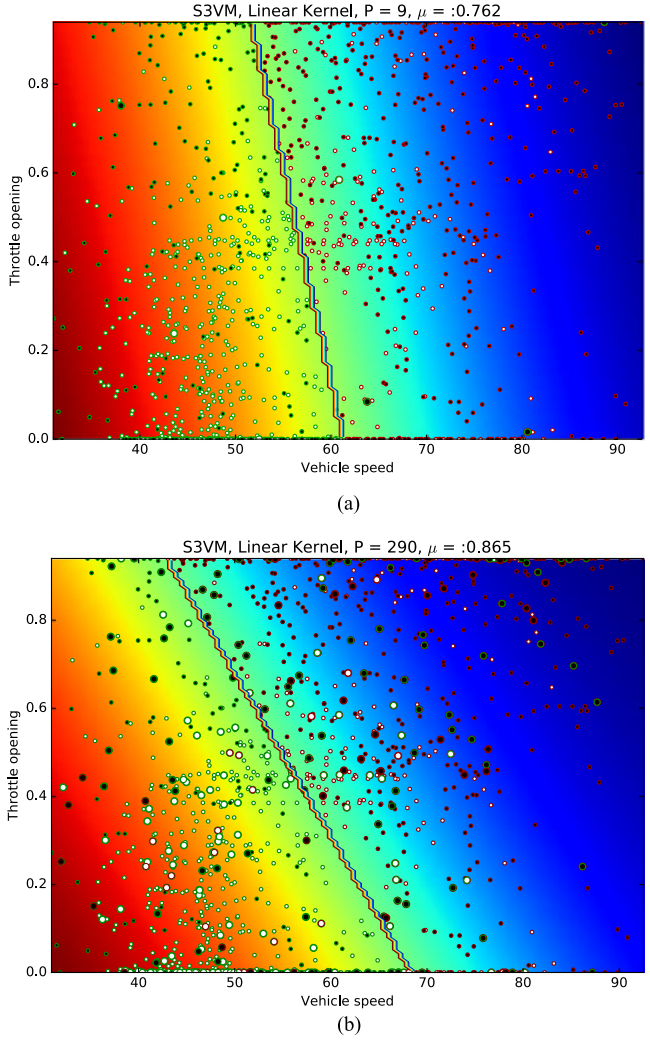


Fig. 9. Examples of results using S3VM with a linear kernel. (a) $\mu = 0.762$, $P = 9$. (b) $\mu = 0.865$, $P = 290$.

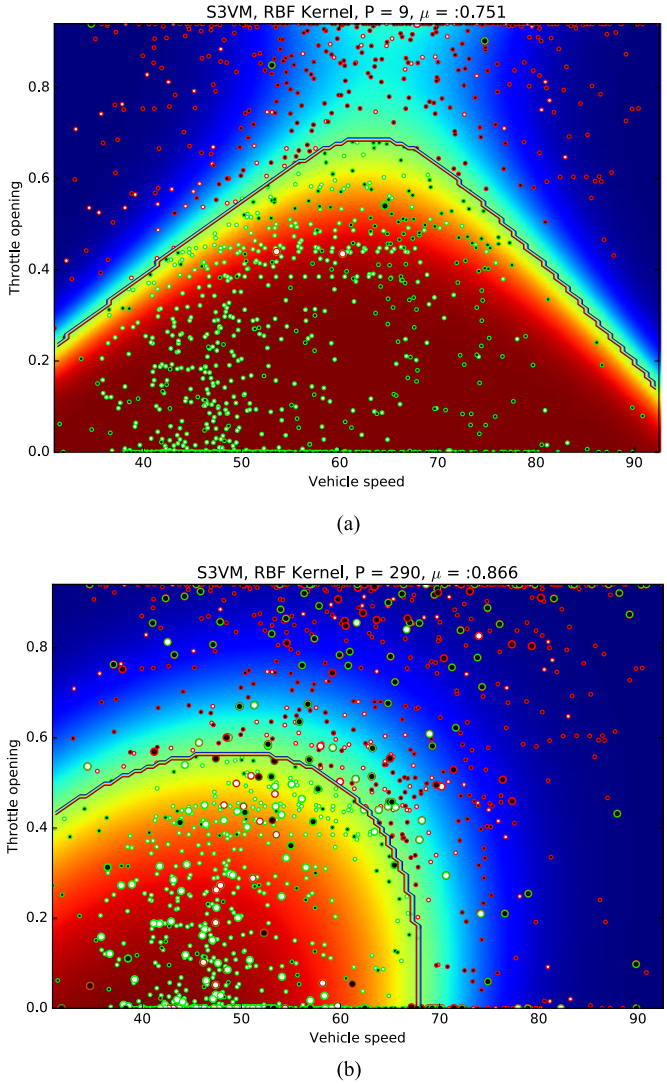


Fig. 10. Examples of results using S3VM with a RBF kernel. (a) $\mu = 0.751$, $P = 9$. (b) $\mu = 0.866$, $P = 290$.

respectively. With the RBF kernel, even though only a few labeled data points are used (e.g., $P = 9$ and $P = 25$), S3VM can utilize unlabeled data information to obtain a well-trained classifier, which is not possible with the SVM method.

In summary, S3VM shows better performance than SVM for driving style classifications where there are limited labeled data and a large amount of unlabeled data.

V. Additional Discussion and Future Work

In this paper, we present a semisupervised approach for driving style classification, which requires a very low labeling effort. There may be potential improvement in classification performance using other semisupervised approaches such as transductive SVM, Laplacian SVM, or semisupervised extreme learning machine. However, our emphasis in this paper was to demonstrate the use of semisupervised classification approaches for reducing labeling effort rather than to provide detailed comparisons of various semisupervised approaches. A number of research issues need to be addressed in order to fully bring semisupervised classification approaches into practical use.

A. Amount of Driving Data

The amount of driving data are crucial for training a model or classifier. In this work, we recorded driving data from 20 drivers. Each driver drove about 60 min with data being collected at a rate of 50 Hz. The amount of driving data obtained were sufficient to train a classifier for a particular driving style, according to [11], [31], and [47], but this amount might not necessarily be the optimal amount. Thus, the problem "*How much driving data is enough to train a model or learn a classifier*?" should be explored.

B. Experiment in the Real World

A driving simulator offers many advantages such as reproducibility, ease of data collection, and safety by not having to encounter any dangerous driving situations [48]. In addition, it is possible to replay the same scenario and compare different drivers' behaviors. A driving simulator is also able to offer purpose-built scenarios, allowing one to analyze specific

driver behaviors in concerned scenarios such as path following, lane changing, and car following. However, a driving simulator, compared to on-road experiments, lacks physical, perceptual, and behavioral factors, which all have an effect on driver behavior. Hence, experiments with a real vehicle will be conducted for our future work.

C. Influences of Labeled Data

In this paper, only a few data points were needed to be labeled initially, which did not need to lie in precise clusters—reasonably separated regions were sufficient—but the data should not be labeled randomly. And even though we do not label the data according to distinct clusters, based on the assumptions in Section II-B and how the gradient-based optimization algorithm works, we can always generate an optimal decision boundary by fully exploiting the underlying resources of unlabeled data. However, completely incorrect labeling of data could result in unsatisfactory results.

D. Influences of Other Factors

The classification of driving styles is a multidimensional problem, which could be influenced by many factors such as weather, road profiles, and traffic information. For example, driver behavior on a flat road could differ from that on a steep road. In this paper, due to the limitations of our experimental equipment and types of driving scenarios, we mainly focused on path-following behaviors on a curvy road without considering weather or traffic. However, these parameters should be included in future work.

VI. Conclusion

To reduce the initial labeling effort and enhance the classification accuracy, in this paper, we apply a semisupervised approach, namely a semisupervised support machine (S3VM), to classify various driving styles. The experimental results show that the S3VM method performs better than the SVM method when we use a small amount of labeled data and a large amount of unlabeled data. More specifically, this paper demonstrated that S3VM improves classification accuracy by about 10%, compared to SVM, in such situations. Thus, this paper has shown that not only does S3VM improve classification performance in general, but also the data labeling required a prior is significantly reduced.

References

[1] W. Wang, J. Xi, and H. Chen, "Modeling and recognizing driver behavior based on driving data: A survey," *Math. Probl. Eng.*, vol. 2014, Feb. 2014, Art. no. 245641.

[2] S. Lefèvre, A. Carvalho, Y. Gao, H. E. Tseng, and F. Borrelli, "Driver models for personalized driving assistance," *Veh. Syst. Dyn.*, vol. 53, no. 12, pp. 1705–1720, Dec. 2015.

[3] K. Bengler, K. Dietmayer, B. Färber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 980–992, Apr. 2016.

[4] W. Wang, J. Xi, and J. Wang, "Human-centered feed-forward control of a vehicle steering system based on a driver's steering model," in *Proc. Amer. Control Conf.*, Chicago, IL, USA, Jul. 2015, pp. 3361–3366.

[5] W. Wang, J. Xi, C. Liu, and X. Li, "Human-centered feed-forward control of a vehicle steering system based on a driver's path-following characteristics," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1440–1453, Jun. 2017.

[6] M. Corno, P. Giani, M. Tanelli, and S. M. Savaresi, "Human-in-the-loop bicycle control via active heart rate regulation," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1029–1040, May 2015.

[7] A. Bolovinou, A. Amditis, and F. Bellotti, "Driving style recognition for co-operative driving: A survey," in *Proc. 6th Int. Conf. Adapt. Self-Adapt. Syst. Appl.*, 2014, pp. 73–78.

[8] F. Sagberg, G. F. B. Piccinini, and J. Engström, "A review of research on driving styles and road safety," *Human Factors Ergon. Soc.*, vol. 57, no. 7, pp. 1248–1275, Nov. 2015.

[9] W. Dib, A. Chasse, P. Moulin, A. Sciarretta, and G. Corde, "Optimal energy management for an electric vehicle in eco-driving applications," *Control Eng. Pract.*, vol. 29, pp. 299–307, 2014.

[10] C. Bingham, C. Walsh, and S. Carroll, "Impact of driving characteristics on electric vehicle energy consumption and range," *IET Intell. Transp. Syst.*, vol. 6, no. 1, pp. 29–35, 2012.

[11] Y. L. Murphey, R. Milton, and L. Kiliaris, "Driver's style classification using jerk analysis," in *Proc. IEEE Workshop Comput. Intell. Vehicles Veh. Syst.*, 2009, pp. 23–28.

[12] S. Zhang and R. Xiong, "Adaptive energy management of a plug-in hybrid electric vehicle based on driving pattern recognition and dynamic programming," *Appl. Energy*, vol. 155, pp. 68–78, 2015.

[13] M. Zhou, H. Jin, and W. Wang, "A review of vehicle fuel consumption models to evaluate eco-driving and eco-routing," *Transp. Res. Part D: Transp. Environ.*, vol. 49, pp. 203–218, 2016.

[14] 2014. [Online]. Available: http://www.nhtsa.gov/About+NHTSA/Press+Releases/2015/2014-traffic-death s-drop-but-2015-trending-higher

[15] W. Wang, D. Zhao, J. Xi, and W. Han, "A learning-based approach for lane departure warning systems with a personalized driver model," *arXiv: 1702.01228*, Feb. 2017.

[16] V. A. Butakov and P. Ioannou, "Personalized driver/vehicle lane change models for ADAS," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4422–4431, Oct. 2015.

[17] X. Fu and D. Soeffker, "Modeling of individualized human driver model for automated personalized supervision," *SAE Tech.* Paper 2010–01-0458, 2010.

[18] E. Gilman, A. Keskinarkaus, S. Tamminen, S. Pirttikangas, J. Röning, and J. Riekki, "Personalised assistance for fuel-efficient driving," *Transp. Res. Part C: Emerg. Technol.*, vol. 58, pp. 681–705, Sep. 2015.

[19] S. Schnelle, J. Wang, H. Su, and R. Jagacinski, "A driver steering model with personalized desired path generation," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 47, no. 1, pp. 111–120, Jan. 2017.

[20] R. Castro, M. Tanelli, R. E. Araújo, and S. M. Savaresi, "Design of safety-oriented control allocation strategies for overactuated electric vehicles," *Veh. Syst. Dyn.*, vol. 52, no. 8, pp. 1017–1046, Aug. 2014.

[21] L. Xu, J. Hu, H. Jiang, and W. Meng, "Establishing style-oriented driver models by imitating human driving behaviors," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2522–2530, Oct. 2015.

[22] T. Qu, H. Chen, D. Cao, H. Guo, and B. Gao, "Switching-based stochastic model predictive control approach for modeling driver steering skill," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 365–375, Feb. 2015.

[23] W. Wang and J. Xi, "A rapid pattern-recognition method for driving styles using clustering-based support vector machines," in *Proc. IEEE Int. Conf. Amer. Control Conf.*, Boston, MA, USA, Jul. 2016, pp. 5270–5275.

[24] M. V. Ly, S. Martin, and M. M. Trivedi, "Driver classification and driving style recognition using inertial sensors," in *Proc. IEEE Intell. Vehicles Symp.*, Gold Coast, Australia, Jun. 2013, pp. 1040–1045.

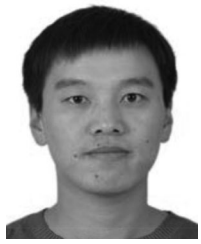
[25] C. C. Lin, S. Jeon, H. Peng, and J. M. Lee, "Driving pattern recognition for control of hybrid electric trucks," *Veh. Syst. Dyn.*, vol. 42, nos. 1/2, pp. 41–58, Dec. 2004.

[26] A. Wahab, C. Quek, C. K. Tan, and K. Takeda, "Driving profile modeling and recognition based on soft computing approach," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 563–582, Apr. 2009.

[27] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Driver behavior classification at intersections and validation on large naturalistic data set," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 724–736, Jun. 2012.

[28] M. Sundbom, P. Falcone, and J. Sjoberg, "Online driver behavior classification using probabilistic ARX models," in *Proc. 16th Int. IEEE Annu. Conf. Intell. Transp. Syst.*, Hague, The Netherlands, Oct. 2013, pp. 1107–1112.

[29] Z. Constantinescu, C. Marinoiu, and M. Vladoiu, "Driving style analysis using data mining techniques," *Int. J. Comput. Commun. Control*, vol. 5, no. 5, pp. 654–663, Dec. 2012.
[30] B. Higgs and M. Abbas, "Segmentation and clustering of car-following behavior: Recognition of driving patterns," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 81–90, Feb. 2015.
[31] B. Shi *et al.*, "Evaluating driving styles by normalizing driving behavior based on personalized driver modeling," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 45, no. 12, pp. 1502–1508, Dec. 2015.
[32] A. Bender, G. Agamennoni, J. R. Ward, S. Worrall, and E. M. Nebot, "An unsupervised approach for inferring driver behavior from naturalistic driving data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3325–3336, Dec. 2015.
[33] I. Misra, A. Shrivastava, and M. Hebert, "Watch and learn: Semi-supervised learning of object detectors from videos," in *Proc. IEEE Conf. Comput. Vision. Pattern Recognit.*, 2015, pp. 3593–3602.
[34] A. Teichman and S. Thrun, "Tracking-based semi-supervised learning," *Int. J. Robot. Res.*, vol. 31, no. 7, pp. 804–818, Jun. 2012.
[35] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1108–1120, Apr. 2016.
[36] I. Steinwart and A. Christman, *Support Vector Machines*. Berlin, Germany: Springer, 2008.
[37] S. Abe, *Support Vector Machines for Pattern Classification*. Berlin, Germany: Springer, 2005.
[38] F. Gieseke, A. Airola, T. Pahikkala, and O. Kramer, "Fast and simple gradient-based optimization for semi-supervised support vector machine," *Neurocomputing*, vol. 123, pp. 23–32, Jan. 2014.
[39] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
[40] A. Ben-Hur and J. Weston, "A user's guide to support vector machines," *Data Mining Techniques for the Life Sciences*. New York, NY, USA: Springer, 2010, pp. 223–239.
[41] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. 10th Int. Workshop Artif. Intell. Statist.*, Jan. 2005, pp. 57–64.
[42] J. Nocedal and S. Wright, *Numerical Optimization*. Berlin, Germany: Springer, 2006.
[43] J. Xi, Y. Zong, and W. Wang, "Research on virtual experimental teaching platform of vehicle electronic control," *Res. Exploration Lab.*, vol. 34, no. 1, pp. 79–83, 2015.
[44] G. Qi, Y. Du, J. Wu, and M. Xu, "Leveraging longitudinal driving behaviour data with data mining techniques for driving style analysis," *IET Intell. Transp. Syst.*, vol. 9, no. 8, pp. 792–801, 2015.
[45] J. Wang, G. Zhang, R. Wang, S. C. Schnelle, and J. Wang, "A gain-scheduling driver assistance trajectory following an algorithm considering different driver steering characteristics," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1097–1108, May 2017, doi: 10.1109/TITS.2016.2598792.
[46] X. Na and D. J. Cole, "Linear quadratic game and non-cooperative predictive methods for potential application to modelling driver–AFS interactive steering control," *Veh. Syst. Dyn.*, vol. 51, no. 2, pp. 165–198, 2013.
[47] W. Wang, C. Liu, and D. Zhao, "How much data is enough? A statistical approach with case study on longitudinal driving behavior," *IEEE Trans. Intell. Veh.*, vol. 2, no. 2, pp. 85–98, 2017.
[48] I. Karl, G. Berg, F. Ruger, and B. Farber, "Driving behavior and simulator sickness while driving the vehicle in the loop: Validation of longitudinal driving behavior," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 1, pp. 42–57, Spring 2013.

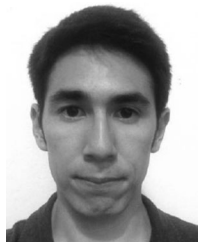


Wenshuo Wang (S'15) received the B.S. degree in transportation engineering from Shandong University of Technology, Shandong, China, in 2012. He is currently working toward the Ph.D. degree in mechanical engineering with Beijing Institute of Technology (BIT), Beijing, China.

He is a visiting scholar with the School of Mechanical Engineering, University of California at Berkeley (UCB), Berkeley, CA, USA. He conducts research under the supervisor of Prof. J. Xi of BIT and Prof. K. Hedrick of the University of California at Berkeley. His research interests include vehicle dynamics control, adaptive control, driver model, human–vehicle interaction, and recognition and application of human driving characteristics. His work focuses on modeling and recognizing drivers behavior, making intelligent control systems between human driver and vehicles.

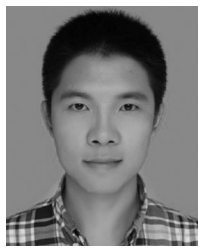
Junqiang Xi received the B.S. degere in automotive engineering from Harbin Institute of Technology, Harbin, China, in 1995, and the Ph.D. degree in vehicle engineering from Beijing Institute of Technology (BIT), Beijing, China, in 2001.

In 2001, he joined the State Key Laboratory of Vehicle Transmission, BIT. During 2012–2013, he conducted research as an Advanced Research Scholar with the Vehicle Dynamic and Control Laboratory, Ohio State University, USA. He is currently a Professor and Director of the Automotive Research Center, BIT. His research interests include vehicle dynamic and control, powertrain control, mechanics, intelligent transportation systems, and intelligent vehicles.



Alexandre Chong is a fourth-year undergraduate student in mechanical and electrical engineering with the University of California, Berkeley, CA, USA.

His research interests include advanced driver assistance systems and autonomous driving.



Lin Li received the M.S. degree in mechanical engineering from Guizhou University, Guiyang, China, in 2015. He is currently working toward the Ph.D. degree in vehicle dynamics and control with Beijing Institute of Technology, Beijing, China.

His research interests include driver model with artificial intelligence, pattern recognition of human driver characteristics, and human-intelligent vehicle collaboration.