

Trajectory Clustering Aided Personalized Driver Intention Prediction for Intelligent Vehicles

Dewei Yi , Student Member, IEEE, Jinya Su , Member, IEEE, Cunjia Liu , Member, IEEE, and Wen-Hua Chen , Fellow, IEEE

Abstract—Early driver intention prediction plays a significant role in intelligent vehicles. Drivers exhibit various driving characteristics impairing the performance of conventional algorithms using all drivers' data indiscriminately. This paper develops a personalized driver intention prediction system at unsignalized T intersections by seamlessly integrating clustering and classification. Polynomial regression mixture (PRM) clustering and Akaike's information criterion are applied to individual drivers trajectories for learning in-depth driving behaviors. Then, various classifiers are evaluated to link low-level vehicle states to high-level driving behaviors. CART classifier with Bayesian optimization excels others in accuracy and computation. The proposed system is validated by a real-world driving dataset. Comparative experimental results indicate that PRM clustering can discover more in-depth driving behaviors than manually defined maneuver due to its fine ability in accounting for both spatial and temporal information; the proposed framework integrating PRM clustering and CART classification provides promising intention prediction performance and is adaptive to different drivers.

Index Terms—Driver behavior prediction, intelligent vehicle, polynomial regression mixture (PRM), trajectory clustering.

I. INTRODUCTION

RECENT advancements of artificial intelligence (AI) have stimulated the rapid development of industrial informatics applications, especially in intelligent vehicles for smart cities [1]–[3]. In particular, machine learning with big data has made a great progress in perception and decision making enhancing driving safety, convenience, and fuel efficiency for intelligent vehicles [4]–[7]. A number of advanced driver assistance systems, e.g., adaptive cruise control and lane departure warning, have been developed to augment drivers' situation awareness

Manuscript received May 21, 2018; revised August 10, 2018; accepted December 18, 2018. Date of publication December 28, 2018; date of current version June 12, 2019. This work was supported by the U.K. Engineering and Physical Sciences Research Council Autonomous and Intelligent Systems program under Grant EP/J011525/1 with BAE Systems as the leading industrial partner. The work of D. Yi was supported by the Chinese Scholarship Council for his study in the U.K. Paper no. TII-18-1273. (Corresponding author: Jinya Su.)

The authors are with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough LE11 3TU, U.K. (e-mail: D.Yi@lboro.ac.uk; J.Su2@lboro.ac.uk; C.Liu5@lboro.ac.uk; W.Chen@lboro.ac.uk).

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/TII.2018.2890141

accuracy and enhance decision-making capabilities by perceiving the environmental situation in real time, alerting drivers to potential dangers, or even taking over certain driving tasks in urgent situations. Despite all the effort and progress achieved so far, there is still a long way to go to deploy driver assistance system robustly and effectively in realistic traffic environment to prevent majority of accidents.

In particular, critical challenges still remain in negotiating traffic intersection safely in urban areas. This is because maneuvering through interactions is a stressful task due to multiple directions of movements along with the need to maximally account for other drivers' possible errors [8]. According to the European Union (EU) community road accident database CARE, intersection-related fatalities account for more than 20% in the EU during the last decade (2001–2010) [9]. This is also the case in the United States, where an estimated 45% of injury crashes and 22% of roadway fatalities are intersection related [10]. It is also reported that in these accidents, driver's inability to correctly assess and/or observe the danger involved in such situations [11] is a main contributing factor.

In the past decade, a number of research and development efforts have been devoted to intersection decision support systems. For example, in [12], the problem of whether a driver will stop safely or not given the traffic signal indication (i.e., compliant or violating behaviors) was considered, where the binary classification problem was solved using discriminative support vector machine (SVM) and generative hidden Markov models based on three key features including range to intersection, speed, and longitudinal acceleration. In [13] and [14], driver's turning behavior recognition at a T intersection (e.g., straight driving, left turn, or right turn) was investigated using a discriminant analysis (DA) and a long short-term memory based recurrent neural network, where position, heading, and velocity were chosen as features. In [15], a Bayesian network is drawn to address the turning intent prediction at arbitrary intersections by incorporating knowledge about the intersection layout using contextual information extracted from a digital map. In addition to academic research efforts, various projects have also been seen in automotive sector. For example, in Europe, the InterSafe project was created by the European Commission to increase safety at road intersections [16]. In the U.S., the Intersection Decision Support project [11] and the Cooperative Intersection Collision Avoidance System project [17] were sponsored by the Department of Transportation and universities to develop intersection collision avoidance systems and their demonstration.

In contrast to the majority of preceding works where a generic (or average) system for all drivers is built by treating individual drivers' data indiscriminately [18], we aim to develop a personalized system dedicated to intersection assistance. This is done by observing that different drivers have distinct driving preferences and characteristics even for the same maneuvers at intersections. For example, for a multiple lane, some drivers may prefer to drive in the inner lane while others prefer the outer lane. Even for a single lane, an aggressive driver may exhibit distinct turning patterns in comparison to a mild one. To make the system better cooperate with individual drivers, the personalized driving characteristics should be accommodated in driving behavior recognition so that the so-called ‘‘adaptive aiding’’ [1], [19] can be provided to the drivers at the right time and in a proper manner.

To effectively accommodate driving characteristics, the personalized driving data of individual drivers are exploited including high-level trajectories and low-level vehicle states, such as velocity and heading. The proposed driver behavior prediction system comprises two layers including offline behavior learning via high-level trajectories and online behavior prediction via low-level vehicle states. The offline stage is different from the existing work [20] where classifiers are trained by labeled data. The proposed system comprises two steps. First, driving behaviors are automatically learnt by dividing the trajectories of a particular driver into various categories using clustering algorithms, where the cluster number is optimized using the classical Akaike's information criterion (AIC) [21]. This clustering-aided approach, in contrast to conventional ones where only a set of manually defined maneuvers are considered [14], automatically learns the driver behaviors using a purely unsupervised data-driven approach. Owing to this step, the proposed behavior prediction system is adaptive to different drivers and arbitrary road intersections without manually labeling the historical data. Second, based on the learnt driving behaviors represented by different clusters, low-level vehicle state data including velocity and heading in each cluster are drawn to build a mapping between low-level sensing data and high-level driving behaviors using the classical classification and regression trees (CART) algorithm. Then, in the stage of online prediction with the advent of new sensing data, the trained CART is drawn to predict individual drivers' behavior represented by different clusters.

To the best of the authors' knowledge, this paper is the first attempt to integrate trajectory clustering and behavior classifier to solve the problem of personalized driving behavior prediction, particularly the proposed framework is validated by using a recently collected dataset from real-world driving experiments with promising performance. More precisely, the main contributions are summarized as follows.

- 1) Trajectory clustering is adopted to learn individual drivers' behaviors from history trajectories resulting in a number of in-depth maneuvers/behaviors without manually labeling. Owing to this step, the proposed behavior prediction system is adaptive to different drivers and various intersection layouts automatically.
- 2) Various classifiers for driving behavior prediction are compared to identify a suitable one for the proposed

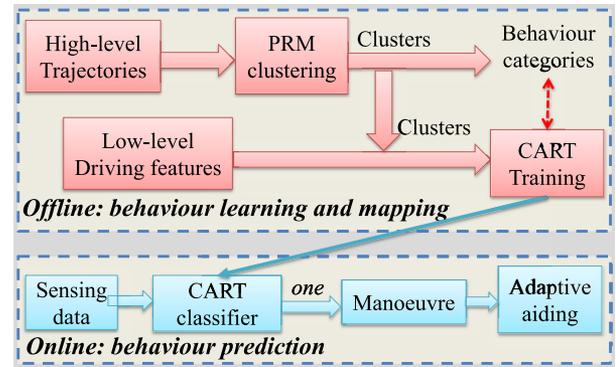


Fig. 1. Diagram of trajectory clustering aided personalized driver behavior prediction system: blocks in red are for offline behavior learning and mapping construction, whereas blocks in light blue are for online behavior prediction.

framework, where a CART classifier with Bayesian hyperparameter optimization outperforms others.

II. DRIVER BEHAVIOR PREDICTION

In order to achieve personalized driver behavior prediction, two key problems should be addressed. The first one is how to automatically label personal trajectory data since manual data labeling is painful and inefficient. Another is how to build the relationship between driving features and driver behaviors. To overcome the first problem, trajectory clustering is adopted to discover the in-depth behaviors for individual drivers in an automatic manner. To deal with the second problem, supervised learning is applied to predict driver behaviors by using vehicle-related measurements.

In this paper, the information on driving behavior is mainly grouped into two levels including ‘‘high-level trajectories’’ and ‘‘low-level driving features.’’ In particular, ‘‘low-level driving features’’ denote information related to vehicle dynamics (e.g., speed and heading, which are only functions of time), representing short-term (or instantaneous) driving behaviors [7]. Whereas ‘‘high-level trajectories’’ denote vehicle trajectories (i.e., spatial-temporal processes, which are essentially functions of both position and time) reflecting drivers' long-term behavior information (e.g., driving preference and turning pattern).

In the proposed system, ‘‘high-level history trajectories’’ are inputs of a clustering algorithm for offline learning that can automatically learn drivers' in-depth behaviors (e.g., driving preference and turning pattern). While ‘‘low-level driving features’’ are inputs of classification algorithm, which can predict driver behaviors in real time. In the remaining part of this section, the proposed personalized driver behavior prediction system aided by trajectory clustering is briefly discussed, where its overall diagram is shown in Fig. 1 including offline driving behavior learning and classifier construction and online behavior prediction.

In the offline training layer, ‘‘high-level history trajectories’’ of individual drivers are divided into different categories representing various driving behaviors (e.g., straight driving, mild turn, and aggressive turn) by using a clustering algorithm. The

trajectory clustering algorithm in this paper can consider spatial and temporal information simultaneously. At the same time, the clustering approach is adaptive to different drivers and any intersection layouts (e.g., T junction and crossroad). Considering that the length of trajectories is variable, the polynomial regression mixture (PRM) clustering algorithm is adopted that is independent of trajectory length by using a regression analysis with an expectation maximization (EM) algorithm for parameter learning. Trajectory clustering is to implement autotagging of driving behaviors and adapts to each individual driver. Then, CART classification models are trained to classify different driving behaviors by treating low-level vehicle states as the features (e.g., speed and heading). In the online execution layer, observed vehicle state data are classified into different clusters using the trained CART model, where each cluster represents a particular driving behavior. Based on the proposed framework, the driving behaviors of individual drivers can be understood by the vehicle and consequently “adaptive aiding” can be provided to the driver at the right time and in a proper manner [1]. In the following sections, each element of the framework is elaborated.

III. CLUSTERING FOR BEHAVIOR LEARNING

In this section, offline behavior learning is detailed for individual drivers for the purpose of personalized behavior learning. As highlighted in Section I, different from the existing works where drivers’ behaviors at an intersection are manually defined by a number of given maneuvers, e.g., continuing straight, turning right, or turning left, this paper relies on clustering algorithms to automatically learn driver behaviors by analyzing the high-level history trajectory data of individual drivers. As a result, this approach is adaptive to different drivers and arbitrary road intersection layouts.

Clustering is an efficient way to discover hidden patterns in the dataset by an unsupervised manner. Various clustering algorithms are available, each of which has its own pros and cons [22], [23]. For instance, center-based approaches, such as k -means clustering, are simple but cannot directly handle trajectories with various lengths. Whereas approaches such as Gaussian mixtures cannot tackle trajectories measured at different time points or that contain missing observations. As a consequence, trajectories with the same spatial pattern but in opposite directions (or trajectories with similar spatial pattern but with different velocities) may be misclassified into one cluster [24]. Therefore, it is necessary to consider the characteristics of a trajectory clustering problem in selecting an appropriate algorithm. In comparison with common clustering problems, trajectory clustering has its own characteristics that are summarized as follows: trajectories usually have different lengths due to their different time durations in the region of interest; and each trajectory is a spatial–temporal process since it is essentially a function of both position and time. As a result, two trajectories with similar spatial pattern may have distinct temporal patterns and so should not be categorized into one cluster.

Considering the aforementioned characteristics, PRM clustering is adopted, which can effectively handle the clustering problem with variable lengths while simultaneously taking

spatial and temporal information into account [24]. In addition, to determine the “optimal” cluster number, the classical AIC [21] is adopted. In the following sections, some pivotal elements of the trajectory clustering are elaborated.

A. Polynomial Regression Mixture (PRM)

PRM divides vehicle historical trajectories of a specific driver into different clusters by minimizing the differences between data and regression models. Let $Y = \{y_1, \dots, y_n\}$ be a set of the driver’s vehicle historical trajectories containing n trajectories, where each trajectory y_i ($i = 1, 2, \dots, n$) is a sequence of position measurements with length n_i observed at a set of time indexes \mathbf{x}_i (in this paper, trajectories represent historical movement of a driver at unsignalized T junctions). The conditional probabilistic model for cluster k (each cluster represents a category of driving behavior) is represented by $p_k(y_i|\mathbf{x}_i, \theta_k)$. Therefore, probability density functions (PDFs) of all clusters (i.e., all driving behaviors) can be given by the following:

$$p(y_i|\mathbf{x}_i, \Theta) = \sum_{k=1}^K \alpha_k p_k(y_i|\mathbf{x}_i, \theta_k) \quad (1)$$

where Θ is the set of θ_k and α_k ($\Theta = \{\theta_k, \alpha_k\}$). θ_k represents parameters of the k th model (driving behavior) and α_k represents the probability that the i th trajectory is assigned to cluster k (driving behavior k) satisfying $\sum_{k=1}^K \alpha_k = 1$ with K being the number of clusters.

1) *Definition of Polynomial Model:* A trajectory within a cluster can be treated as the mean trajectory of the cluster added a noise and the mean trajectory of a cluster can be obtained by (1). Therefore, it is assumed that a trajectory y_i can be represented by a p th-order polynomial regression perturbed by a Gaussian error when \mathbf{x}_i is given. The regression of y_i on \mathbf{x}_i is given by the following:

$$y_i = X_i \beta + \epsilon_i \quad (2)$$

where $\epsilon_i \sim N(0, \sigma^2 I)$ is the Gaussian error, X_i is an $n_i \times (p+1)$ regression matrix, and β is regression coefficients, which is a $(p+1)$ th vector. The p th-order regression matrix X_i evaluated at \mathbf{x}_i is given by the following:

$$X_i = \begin{bmatrix} 1 & x_{i1} & x_{i1}^2 & \dots & x_{i1}^p \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{in_i} & x_{in_i}^2 & \dots & x_{in_i}^p \end{bmatrix}.$$

For different trajectories, the row number of X_i is different to be adaptive to different trajectory lengths. The column number of X_i is the same when the order p of polynomial regression is determined. This step is to transfer a trajectory into a regression form. The regression model (2) defines the conditional PDF of y_i given \mathbf{x}_i as $\mathcal{N}(y_i|X_i \beta, \sigma^2 I)$. Integrating (2) into the mixture density (1) by incorporating dependence of this PDF on k , represented by $\{\beta_k, \sigma_k^2\}$, which results in the following PRM:

$$p(y_i|\mathbf{x}_i, \Theta) = \sum_{k=1}^K \alpha_k \mathcal{N}(y_i|X_i \beta_k, \sigma_k^2 I). \quad (3)$$

The log-likelihood of trajectories set Y is computed by summing up the conditional probability density of all n trajectories given by the following:

$$\log p(Y|X, \Theta) = \sum_{i=1}^n \log \sum_{k=1}^K \alpha_k \mathcal{N}(y_i | X_i \beta_k, \sigma_k^2 I) \quad (4)$$

where $X = \{X_1, \dots, X_n\}$ denote the set of known regression matrices, and Θ represents all unknown parameters $\{\beta_k, \sigma_k^2, \alpha_k\}$. Equation (4) is used in the EM algorithm to derive the model parameters based on maximum-likelihood estimation.

2) EM for Parameter Optimization: z_i is the cluster membership for trajectory y_i , then the joint density of y_i and z_i can be given by the following:

$$p(y_i, z_i | \mathbf{x}_i) = \alpha_{z_i} p_{z_i}(y_i | \mathbf{x}_i) = \alpha_{z_i} \mathcal{N}(y_i | X_i \beta_{z_i}, \sigma_{z_i}^2 I). \quad (5)$$

The overall log-likelihood function \mathcal{L} can be obtained by summing up all n trajectories' log joint density given by the following:

$$\mathcal{L} = \sum_{i=1}^n \log \alpha_{z_i} \mathcal{N}(y_i | X_i \beta_{z_i}, \sigma_{z_i}^2 I). \quad (6)$$

E Step: Since cluster memberships z_i are unobservable, posterior $p(z_i | y_i, \mathbf{x}_i)$ is used in the E step. The membership probability $p(z_i | y_i, \mathbf{x}_i)$ that the i th trajectory was generated from cluster z_i is calculated by the following:

$$w_{ik} = p(z_i = k | y_i, \mathbf{x}_i) \propto \alpha_k p_k(y_i | \mathbf{x}_i) = \alpha_k \mathcal{N}(y_i | X_i \beta_k, \sigma_k^2 I).$$

The posterior expectation of \mathcal{L} in (6) is computed with respect to the above-mentioned posterior given by

$$E[\mathcal{L} | y_i, \mathbf{x}_i] = \sum_{i=1}^n \sum_{k=1}^K w_{ik} \log \alpha_k \mathcal{N}(y_i | X_i \beta_k, \sigma_k^2 I). \quad (7)$$

M Step: In the M step, (7) is maximized with respect to the parameters $\Theta = \{\beta_k, \sigma_k^2, \alpha_k\}$, where the solution is given by the following:

$$\hat{\beta}_k = \left[\sum_{i=1}^n w_{ik} X_i^T X_i \right]^{-1} \sum_{i=1}^n w_{ik} X_i^T y_i \quad (8)$$

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^n w_{ik} \|y_i - X_i \beta_k\|^2}{\sum_{i=1}^n w_{ik}} \quad (9)$$

$$\hat{\alpha}_k = \frac{1}{n} \sum_{i=1}^n w_{ik}. \quad (10)$$

Remark 1: The computational complexity of the EM algorithm is linear in the number of trajectories. In the initialization phase, the membership probabilities are randomly sampled and then the M step is started. E step and M step will be repeated until a local maximum of log-likelihood is reached. However, in practice, convergence is usually detected when the incremental improvement ratio of log-likelihood drops below a threshold.

B. Optimal Cluster Number

It is usually challenging to determine the ‘‘optimal’’ cluster number k^* for PRM. In this paper, k^* is derived using AIC [21],

which is an effective measure for assessing clustering model quality. According to Akaike’s theory, the most appropriate model has the smallest AIC value, where AIC is defined by the following:

$$\begin{aligned} \text{AIC} = N * \log(\det(\frac{1}{N} \sum_1^N \epsilon(t, \hat{\theta}_N)(\epsilon(t, \hat{\theta}_N))^T)) \\ + 2n_p + N * (n_y * (\log(2\pi) + 1)) \end{aligned} \quad (11)$$

where N is the number of samples, $\epsilon(t)$ is an n_y -by-1 vector of prediction errors with n_y being the number of model outputs, and $\hat{\theta}_N$ is the estimated parameters with n_p being its number. After obtaining clustering configuration via PRM with different cluster numbers, one can calculate the corresponding AIC values, where the cluster number k^* corresponding to the smallest AIC value is selected for each driver.

AIC is usually adopted to determine ‘‘optimal’’ cluster number for clustering problems where data have the same dimension [25]. In this paper, vehicle trajectories in various clustering configurations are generally with variable lengths and therefore AIC value for each clustering configuration cannot be directly calculated. To solve the problem, trajectories of variable lengths are transferred into a fixed number of parameters using a regression analysis. Considering the characteristics of vehicle motion at a T intersection, the constant acceleration model is adopted in x and y directions, which correspond to a polynomial of order 2 given as follows:

$$x(t) = x_0 + v_{x0}t + \frac{1}{2}a_x t^2, y(t) = y_0 + v_{y0}t + \frac{1}{2}a_y t^2$$

where $x(t)$ and $y(t)$ denote the longitudinal and latitudinal movements with x_0, v_{x0}, a_x and y_0, v_{y0}, a_y representing the initial position, velocity, and acceleration in x and y directions. After fitting trajectories using polynomials of order 2, each trajectory can be represented by six coefficients $x_0, v_{x0}, a_x, y_0, v_{y0}$, and a_y . As a result, AIC values can be calculated easily under different clustering configurations.

In practice, however, the range of different coefficients may vary a lot due to different physical meanings, consequently, Z-score [26] is adopted to normalize the coefficients. In comparison with other normalization approaches, Z-score can retain the shape properties of the original data such as skewness and kurtosis. For a vector x , Z-score is defined by $z = \frac{x - \bar{x}}{\sigma}$, where \bar{x} and σ denote the mean and standard derivation of vector x , respectively. The main steps for the optimized PRM clustering are summarized in Algorithm 1.

IV. CLASSIFICATION FOR BEHAVIOR PREDICTION

In Section III, high-level driving trajectories are automatically divided into different categories using the trajectory clustering algorithm, where each trajectory category represents one kind of driver behavior. The next step is to use a low-level vehicle state to predict the driver behavior for real time applications. With the rapid development of sensing technologies, nowadays it is very easy to assess various vehicle state data characterizing various maneuvers and driving patterns. In this paper, vehicle speed and heading are chosen as the features due to the

Algorithm 1: Steps For PRM With Optimized Cluster Number.

1. Given a user-defined maximum cluster number k_{\max} ;
 2. Perform a series of PRM clustering under $i = 1, \dots, k_{\max}$ resulting in $\text{PRM}(1), \dots, \text{PRM}(k_{\max})$;
 3. Calculate AIC value for each clustering configuration using formula (11) resulting in $\text{AIC}(1), \dots, \text{AIC}(k_{\max})$;
 4. Optimal cluster number k^* is determined by $k^* = \arg \min_{i=1}^{k_{\max}} \text{AIC}(i)$, where $\text{PRM}(k^*)$ denotes the optimal clustering configuration.
-

following main reasons: first, these measures are nonintrusive and easily accessible via Inertial Measurement Unit (IMU) and GPS sensors, and have a lower requirement on working condition in comparison with other approaches such as computer vision based ones; second, these features have been proved to be effective in characterizing driving behaviors [13]. For instance, a driver turning across traffic usually slows down and at the same time steers the vehicle in an appropriate direction.

After features are defined, the driver behavior prediction is then formulated as a classification problem building an implicit mapping between vehicle state measurements and trajectory clusters. In real-time applications with the advent of new sensing measurements, the trajectory cluster representing various driving behaviors can be predicted. In machine learning applications, it is generally not easy to select an appropriate algorithm for the task of interest. Consequently, different classifiers are first compared so that the most suitable one is identified.

A. Classifier Selection

Classifier selection is generally based on problem characteristics, personal experience, and experimental comparison. In this paper, a classifier is adopted to predict driving behavior in real time so that suitable follow-up driving assistance can be provided to the driver. In addition, the classifier should also be simple and easy to certify by industry (e.g., white box model). A number of classification algorithms are implemented and compared in this paper, which include DA [27], (CART [28], naive Bayes (NB) [29], SVM [30], and their variants. According to the problem characteristics and experimental results mentioned in Section V-E, CART with Bayesian parameter optimization [31] outperforms others significantly and therefore is adopted in the proposed framework. For the sake of completeness, the CART algorithm is briefly introduced in the following section.

B. Classification and Regression Tree (CART)

CART is a popular nonparametric statistical classifier, which can identify mutually exclusive and exhaustive subgroups of a population whose members share certain common characteristics that affect the dependent variable of interest. In this approach, the data space is recursively partitioned into smaller partitions using binary splitting according to certain splitting rules (for node and threshold selection) until certain stopping rules are reached. Then, a simple predictive model is fitted within

each partition. CART implicitly performs variable screening, i.e., selecting the most discriminatory features at the top parts of decision tree. Therefore, it can obtain better performance for the driver behavior prediction in this paper, where features (i.e., speed and heading) and labels (i.e., behavior categories) have strong correlations. In addition, in comparison to other algorithms (e.g., DA), CART does not require any assumptions of linearity or data distribution. Due to its fine properties, such as a simple structure, low computation load, and easy to interpret, this approach has found wide applications [32], [33]. The detailed algorithm is referred to [32] due to a lack of space, where the pseudocodes are given as follows for the sake of completeness.

Algorithm 2: Classification and Regression Tree (CART) Model.

1. Given a sequence of examples S and a set of discrete attributes A ;
 2. For each attribute $a^i \in A$, the set of attribute values A^i is partitioned into two disjoint subsets A_L^i and A_R^i , let X^i denote all possible A_L^i ;
 3. For each $A_L^i \in X^i$, calculate the Gini gain/mean squared error (MSE) and select the variable $(\tilde{A}_{L,q}^i = A_L^i)$, which maximizes Gini gain or minimizes MSE ($\tilde{A}_{L,q}^i$ is a split);
 4. Send data $S(\tilde{A}_{L,q}^i)$ to the “leaf node” and $S(\tilde{A}_{R,q}^i)$ to the “right node”;
 5. Recursively repeat the same process on these two “nodes” until stopping rules are satisfied;
 6. Return the CART model.
-

Remark 2: There are generally certain stopping rules in CART to avoid its adverse effects, such as overfitting. In this paper, one of the stopping rules is considered, i.e., the minimum leaf size. A smaller leaf size may make the model more prone to capturing noise in training data. In this paper, the determination of minimum leaf size is transformed into the minimization of the cross validation (CV) loss for CART, which is solved using Bayesian optimization (BO) [31], [34] (see, Section V-C).

V. EXPERIMENTAL VALIDATION

In this section, experimental validation is conducted on the proposed framework for driver behavior prediction at a T intersection. The experiments contain clustering for offline driver behavior learning and online driver behavior prediction. All aforementioned elements of the proposed system are validated by using the naturalistic real-world Intelligent Vehicle and Safety System Group (IVSSG) dataset detailed in Section V-A. In the offline phase, the determination of optimal cluster number for individual drivers is investigated in Section V-B. This is to discover in-depth driving behaviors so that the system is adaptive to individual drivers. Moreover, CART parameters are optimized by BO in Section V-C. In addition, resubstitution and k -fold CV methods are used to evaluate prediction performance. The procedure of resubstitution and k -fold CV are explained in Section V-D and prediction performance is presented in Section V-E. It is

TABLE I
SUMMARY OF DATASET

Participant No.	Test Duration	No. of Trajectory	No. of Observation
Driver1	74 mins	69	106616
Driver2	100 mins	71	152971
Driver3	51 mins	73	114456

also noted that all algorithms in this paper are implemented in MATLAB 2017a under Windows 7 Operation System and are evaluated on a PC with the following configuration: Intel Core i5-CPU at 3.20 GHz with 16 GB of RAM.

A. Naturalistic Driving Dataset

The IVSSG dataset was collected on public roads near the Australian Centre for Field Robotics at the University of Sydney [13], where the field of interest (FoI) is a T intersection. During the experiments, three participants are guided to perform approximately ten passes in six possible maneuvers such as driving straight across the top of the intersection, turning, and the intersection left or right. During the data collection, drivers conducted different maneuvers following their own habits and driving characteristics. The spatial plots of data collected around the intersection for three participants are shown in Fig. 3. In the experiments, global navigation satellite system was used to record the vehicle position at a frequency of 10 Hz. In addition, a strap-down IMU was adopted to collect the vehicle inertial data, whereas vehicle's wheel encoders provided speed information. The frequency for both inertial data and speed is 100 Hz. Moreover, in the dataset, an extended Kalman filter has been adopted to fuse and filter the raw data, where the filtered vehicle state estimates were available at a rate of 100 Hz [13]. Once the data were collected, a bounding box was created around the intersection to define the FoI (i.e., the area only containing maneuvers traversing the intersection of interest). The test duration, the number of trajectories, and the number of observations for different drivers are displayed in Table I. Moreover, the IVSSG dataset adopted in this paper has been made open access and can be accessed via <http://its.acfr.usyd.edu.au/datasets/naturalistic-intersection-driving-dataset/its.acfr.usyd.edu.au/datasets/naturalistic-intersection-driving-dataset>.

B. Behavior Learning via PRM Clustering

The first step of offline training is trajectory clustering for driving behavior learning. In this paper, PRM discussed in Section III is drawn to divide driving trajectories into different categories representing various driving behaviors for individual drivers. Different parameters should be specified in this approach such as polynomial order, threshold of improvement ratio in EM algorithm, and cluster number. In this paper, the polynomial order is chosen to be 2. This is because driving through a T intersection can be represented by the constant acceleration model. Moreover, the threshold of improvement ratio for parameter optimization is chosen 1×10^{-6} following the

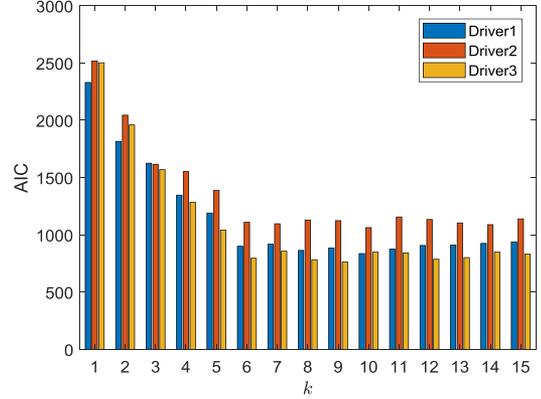


Fig. 2. AIC values under different cluster number k for different drivers.

setting in [24]; whereas the optimal cluster number is determined using AIC value as discussed in Section III-B.

Considering that the T intersection under consideration contains various maneuvers, the cluster number for AIC value calculation ranges from 1 to 15 to accommodate various driving patterns in the same maneuver and abnormal driving trajectories. The AIC values under different cluster configurations for all three participants are shown in Fig. 2. It can be seen that the “optimal” cluster numbers according to AIC values are 10, 10, and 9. Under this parameter setting, the clustering results for three participants are displayed in Fig. 3, where each color represents one type of trajectory category. Bold color lines of clusters in longitude and latitude in Fig. 3 indicate the mean trajectories of different clusters.

It follows from Fig. 3 that:

- 1) Although there are only six different maneuvers at a T intersection, more than six types of driving behaviors are learnt using the data-driven PRM clustering.
- 2) For different drivers, the number of driving behaviors may be different due to their personalized driving characteristics.
- 3) PRM clustering accounting for both spatial and temporal information can discover more in-depth behaviors than manually defined maneuvers.

Trajectories in a specific maneuver display a significantly different temporal pattern (see, right column in Fig. 3). For instance, in Fig. 3(c) and (d), trajectories within cluster 4 and cluster 8 of driver 2 are the driving passes from south-east to north-east while their speeds are different. More specifically, turning speed of cluster 4 is faster than that of cluster 8. Different from manually defined maneuvers focusing on spatial information, the proposed system considers both spatial and temporal information when defining maneuvers. Therefore, trajectories within cluster 4 and cluster 8 are divided into two different clusters rather than one.

Remark 3: With a large amount of driving trajectories for individual drivers, the PRM clustering can also be adopted to discover abnormal driving. Without a clustering analysis, it is challenging to define abnormal driving in trajectory big data.

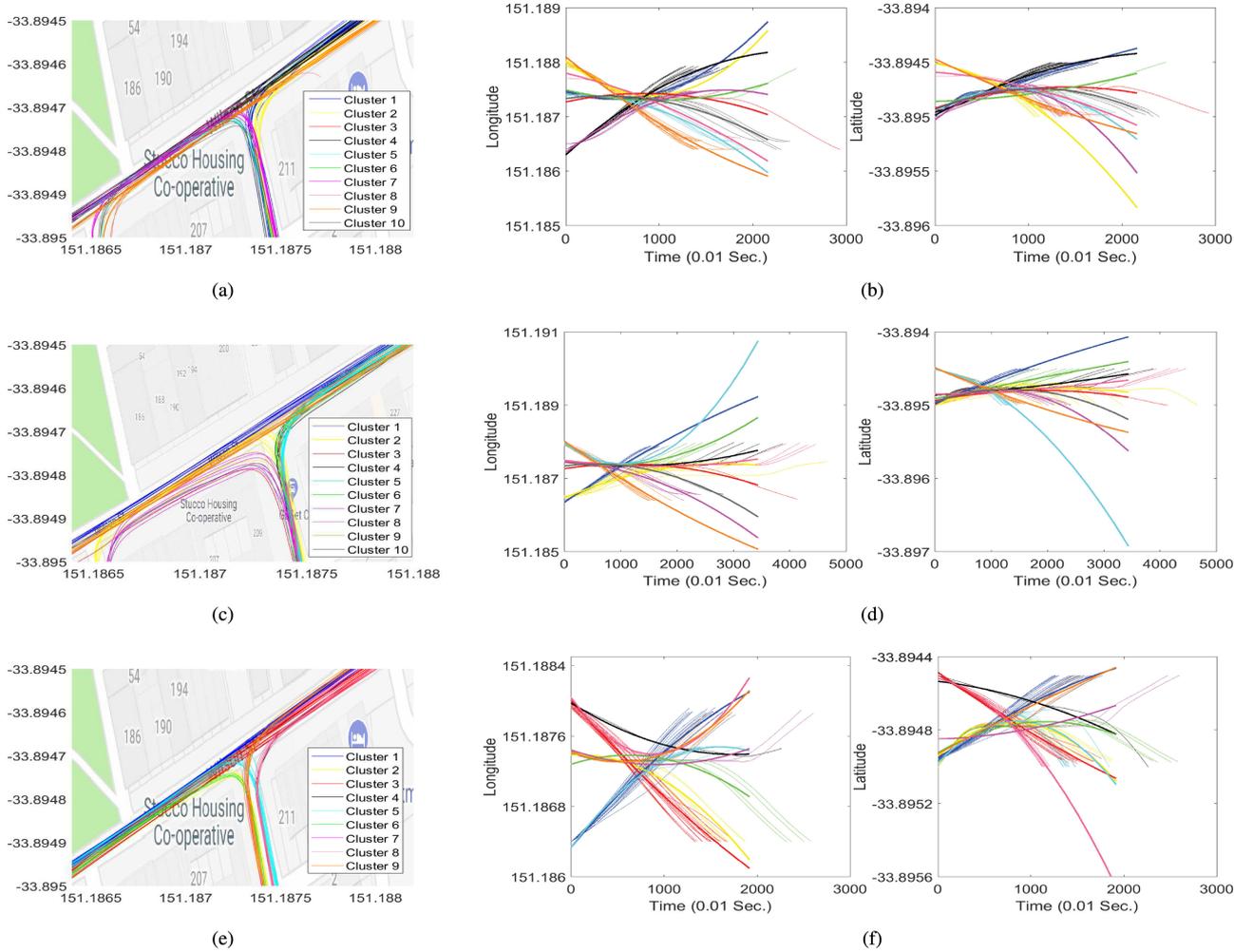


Fig. 3. Each row is the results for a driver: left column is the trajectory clustering results; right column is the trajectory clustering results in longitude and latitude and bold color lines are mean trajectories of clusters. (a) Driver1: Trajectory clustering in Fol. (b) Driver1: Clusters in longitude and latitude. (c) Driver2: Trajectory clustering in Fol. (d) Driver2: Clusters in longitude and latitude. (e) Driver3: Trajectory clustering in Fol. (f) Driver3: Clusters in longitude and latitude.

C. Optimized CART Classifier

The CART algorithm for driving behavior classification is further discussed in this section. Generally speaking, the depth of a tree can be controlled by min leaf size (MLS) and min parent size (MPS), where MLS determines the minimum number of observation per leaf and MPS determines the minimum number of observation per branch node. In this paper, these parameters are optimized by minimizing the classification loss using the BO. In BO, certain stopping rules exist, such as maximum number of iterations and maximum running time. Considering that the optimization process occurs in offline learning and so the running time is not the main concern, only the maximum number of iterations is considered, which is chosen 30. The value of MLS is searched among log scale in the range of $[1, \max(2, n - 1)]$, where n is the number of observations. And MPS is determined by the following:

$$\text{MPS} = \max[M, 2 \times \text{MLS}]$$

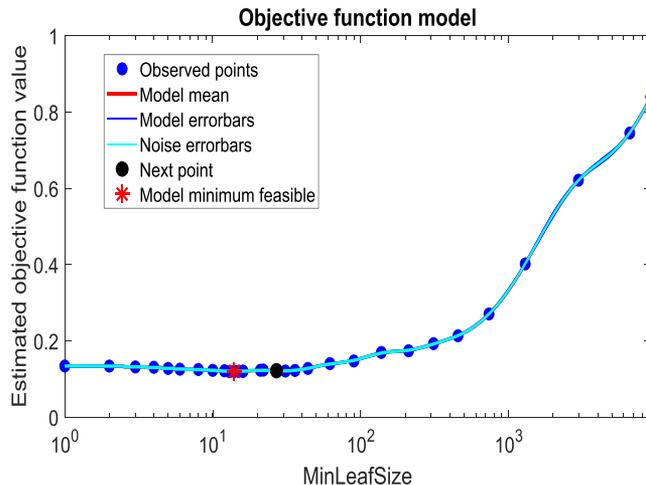


Fig. 4. Estimated objective function value under various minimum leaf sizes.

TABLE II
CLASSIFICATION PERFORMANCE COMPARISONS AMONG DIFFERENT DRIVERS

Algorithm	Driver 1			Driver 2			Driver 3		
	Resubstitution Error	CV Error	Time (sec./sample)	Resubstitution Error	CV Error	Time (sec./sample)	Resubstitution Error	CV Error	Time (sec./sample)
LDA+BO	72.44%	72.51%	8.16E-06	55.31%	55.35%	8.25E-06	48.86%	48.93%	7.16E-06
QDA	66.03%	65.96%	8.69E-06	48.41%	47.76%	1.00E-05	46.80%	46.81%	9.44E-06
CART+BO	9.77%	10.98%	3.01E-05	0.05%	0.09%	9.55E-06	0.01%	0.06%	1.08E-05
Naive Bayes+BO	16.34%	16.93%	1.42E-05	0.73%	0.74%	1.41E-05	0.94%	0.96%	1.22E-05
SVM+Linear	66.57%	66.55%	6.06E-02	49.72%	49.68%	2.22E-02	47.85%	47.87%	1.95E-02
SVM+Polynomial	48.45%	48.65%	2.27E-01	34.92%	35.02%	9.36E-02	38.10%	38.20%	1.10E-01
SVM+Gaussian	44.35%	44.40%	3.15E-02	33.71%	33.90%	1.83E-02	36.95%	36.93%	1.94E-02

where M is a fixed number in the CART model, which is 10 in this paper. Under this parameter setting, the relationships between the estimated objective function values and various minimum leaf sizes for driver 1 are shown in Fig. 4.

It follows from the figure that the optimal minimum leaf size is 15.

D. Resubstitution and k -Fold CV

In this paper, two of the most popular metrics are used to demonstrate algorithm performance, named the resubstitution error [35] and the k -fold CV error [36]. In particular, the resubstitution error is derived from training data, where a lower resubstitution error means a better fitness for given data and results in a better accuracy. However, the resubstitution error is for training data and may lead to the problem of overfitting. Therefore, the CV error is also considered. In k -fold cross validation, the original dataset is randomly divided into k subsets of equal size and holdout method is repeated k times. For each time, one of the k th subsets is retained for testing (i.e., testing set), and the remaining $(k - 1)$ subsets (i.e., training set) are used for training. As a result, every data point appears in a test set only once and appears in training set $(k - 1)$ times. The final k -fold CV error is the average of errors of each fold. The variance of the resulting estimate is reduced with an increment of k . In this experiment, tenfold CV is used for each participant following the results in [37].

E. Prediction Performance

In this section, comparative experiments are conducted to evaluate the performance of the classifiers of the proposed personalized system. Classifiers are to build up the relationship between behavior categories and vehicle-related measurements, so that the corresponding behavior category can be worked out for new vehicle related measurements. In this paper, one generic/unified framework is proposed, however, the proposed framework can automatically (i.e., without any manual parameter retuning) generate a dedicated model for each of the three drivers by using collected data of each driver. Considering the characteristics of the real-time application, accuracy and computation time are considered concurrently. As discussed in Section IV-A, various classifiers are compared in this paper, which include DA-based approaches, such as linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA), NB, and SVM. In particular, LDA is optimized by the Bayesian parameter optimization. And the parameter setting of QDA fol-

TABLE III
AVERAGE PERFORMANCE FOR THREE PARTICIPANTS

Algorithm	Resubstitution Error	CV Error	Time (sec./sample)
LDA+BO	58.87%	58.93%	7.86E-06
QDA	53.74%	53.51%	9.38E-06
CART+BO	3.28%	3.71%	1.68E-05
Naive Bayes+BO	6.00%	6.21%	1.35E-05
SVM+Linear	54.71%	54.70%	3.41E-02
SVM+Polynomial	40.49%	40.62%	1.44E-01
SVM+Gaussian	38.34%	38.41%	2.31E-02

lows [38], where $\delta = 0$ and $\gamma = 0$. Similar to LDA, CART and NB are also optimized using the BO. To identify a more suitable SVM classifier, different kernel functions are considered in this paper including linear, polynomial, and Gaussian kernel functions. The representations of linear, polynomial, and Gaussian kernel functions are shown as follows:

$$G(x_i, x_j) = x_i^T x_j, (1 + x_i^T x_j)^p \text{ and } e^{-\gamma \|x_i - x_j\|^2}$$

where x_i and x_j denote two training samples. In the polynomial kernel function, p is chosen 3 following the results in [39]. In addition, following the results in [40], parameter $\gamma = \frac{1}{2}$ and penalty parameter $C = 1$ are selected. The method to implement multiclass SVM is ‘‘One-vs-One,’’ in which approach $K(K - 2)/2$ binary classifiers are trained with K being the number of classes. The final classification results are figured out by voting through all binary classifiers, where the class owning the most votes is selected.

To quantitatively evaluate the prediction performance, resubstitution error and CV error are adopted. In particular, resubstitution error is the error for classifying training dataset and CV error is the error for classifying testing dataset in CV. The comparative results for three participants are shown in Table II. In addition, the average resubstitution error and CV error are also calculated and summarized in Table III. The algorithm with the best performance is highlighted in bold.

The following conclusions can be drawn from Tables II and III. First, regarding accuracy, CART+BO algorithm obtains the minimum resubstitution error and CV error for each driver, where the average resubstitution and CV errors are 0.0328 and 0.0371, respectively. Second, regarding computation time, DA-based approaches (e.g., LDA+BO and QDA), optimized CART, and NB+BO substantially outperform SVM classifiers and are all suitable for real-time application.

Considering accuracy and computation time concurrently, CART with BO obtains the best performance among the con-

sidered classifiers and consequently chosen as the classifier in the proposed personalized driving behavior prediction.

VI. CONCLUSION

This paper proposes a trajectory clustering aided personalized driver intention prediction system for early driver intention prediction at an unsignalized intersection to enhance driving safety and efficiency. Different from some existing studies where only manually defined maneuvers are considered using classification algorithms, this paper seamlessly integrates unsupervised clustering and supervised classification so that the proposed system is adaptive to individual drivers. The proposed system comprises offline training and online implementation. In offline training for personalized driving behavior learning, high-level driving trajectories are learnt using PRM clustering, leading to a set of trajectory clusters representing various driving behaviors. Then, a mapping between low-level vehicle state (e.g., speed and heading) and trajectory clusters is built using a classification analysis. In particular, a number of classifiers are compared, where CART with Bayesian parameter optimization outperforms others in terms of accuracy and computational load and consequently chosen as the classifier in the proposed framework. In online application, the optimized CART model is drawn to divide new samples into various trajectory clusters representing different driving behaviors.

The proposed framework is validated using a recently collected dataset (i.e., IVSSG dataset) from real-world driving experiments of different drivers. Regarding trajectory clustering analysis, the proposed framework can provide in-depth driving behavior analysis than manually defined maneuvers and is also adaptive to individual drivers. Regarding driving behavior prediction, the proposed framework adopting CART with Bayesian parameter optimization as the classifier obtains promising driving behavior prediction performance in terms of resubstitution error, CV error, and computation load.

This paper is mainly focused on developing an adaptive driving behavior prediction system and demonstrating its feasibility using recently collected real-world experimental datasets. In the future, a sliding window containing a short feature interval will be considered to improve the robustness of the system. Online learning will also be accommodated into the framework considering that drivers' driving style and patterns in executing various maneuvers may change over time. With the advent of personalized driving big data, the proposed framework can also perform abnormal driving detection.

REFERENCES

- [1] D. Yi, J. Su, C. Liu, and W.-H. Chen, "Personalized driver workload inference by learning from vehicle related measurements," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 1, pp. 159–168, Jan. 2019.
- [2] H. Guo, C. Shen, H. Zhang, H. Chen, and R. Jia, "Simultaneous trajectory planning and tracking using an MPC method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle," *IEEE Trans. Ind. Inform.*, vol. 14, no. 9, pp. 4273–4283, Sep. 2018.
- [3] C. Hu, R. Wang, and F. Yan, "Integral sliding mode-based composite nonlinear feedback control for path following of four-wheel independently actuated autonomous vehicles," *IEEE Trans. Transport. Electrification*, vol. 2, no. 2, pp. 221–230, Jun. 2016.

- [4] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transport. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.
- [5] C. Tran and M. M. Trivedi, "3-D posture and gesture recognition for interactivity in smart spaces," *IEEE Trans. Ind. Inform.*, vol. 8, no. 1, pp. 178–187, Feb. 2012.
- [6] D. Yi, J. Su, C. Liu, and W.-H. Chen, "Data-driven situation awareness algorithm for vehicle lane change," in *Proc. 19th Int. IEEE Conf. Intell. Transport. Syst.*, 2016, pp. 998–1003.
- [7] J. Wang, J. Wang, R. Wang, and C. Hu, "A framework of vehicle trajectory replanning in lane exchanging with considerations of driver characteristics," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 3583–3596, May 2017.
- [8] V. A. Butakov and P. Ioannou, "Personalized driver assistance for signalized intersections using V2I communication," *IEEE Trans. Intell. Transport. Syst.*, vol. 17, no. 7, pp. 1910–1919, Jul. 2016.
- [9] J. Broughton *et al.*, "Traffic safety basic facts 2012: Junctions," Deliverable D3.9 of the EC FP7 project DaCoTA, 2013.
- [10] 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. Transport. Syst.*, vol. 13, no. 2, pp. 724–736, Jun. 2012.
- [11] B. Bougler, D. Cody, and C. Nowakowski, "California intersection decision support: A driver-centered approach to left-turn collision avoidance system design," California Partners Adv. Transit Highways, California Path Program, Inst. Transport. Studies, Univ. California, Berkeley, CA, USA, California PATH Res. Rep. UCB-ITS-PRR-2008-1, 2008.
- [12] L. Guo, P.-S. Ge, M.-H. Zhang, L.-H. Li, and Y.-B. Zhao, "Pedestrian detection for intelligent transportation systems combining adaboost algorithm and support vector machine," *Expert Syst. Appl.*, vol. 39, no. 4, pp. 4274–4286, 2012.
- [13] A. Bender, J. R. Ward, S. Worrall, and E. M. Nebot, "Predicting driver intent from models of naturalistic driving," in *Proc. 18th Int. IEEE Conf. Intell. Transport. Syst.*, 2015, pp. 1609–1615.
- [14] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in *Proc. IEEE Conf. Intell. Veh. Symp.*, 2017, pp. 1484–1489.
- [15] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Exploiting map information for driver intention estimation at road intersections," in *Proc. IEEE Conf. Intell. Veh. Symp.*, 2011, pp. 583–588.
- [16] K. C. Fuerstenberg, "A new European approach for intersection safety—The EC-project INTERSAFE," in *Proc. 8th Int. IEEE Conf. Intell. Transport. Syst.*, 2005, pp. 432–436.
- [17] I. Phase, "Cooperative intersection collision avoidance system limited to stop sign and traffic signal violations (CICAS-V)," Res. Innov. Technol. Admin., U.S. Dept. Transport., Washington, DC, USA, Phase I Final Rep., 2008.
- [18] 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.
- [19] W. B. Rouse, "Adaptive aiding for human/computer control," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 30, no. 4, pp. 431–443, 1988.
- [20] A. Doshi and M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Trans. Intell. Transport. Syst.*, vol. 10, no. 3, pp. 453–462, Sep. 2009.
- [21] I. Tomasic, A. Andersson, and P. Funk, "Mixed-effect models for the analysis and optimization of sheet-metal assembly processes," *IEEE Trans. Ind. Inform.*, vol. 13, no. 5, pp. 2194–2202, Oct. 2017.
- [22] C. Sung, D. Feldman, and D. Rus, "Trajectory clustering for motion prediction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 1547–1552.
- [23] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*, vol. 20. Philadelphia, PA, USA: SIAM, 2007.
- [24] S. J. Gaffney, "Probabilistic curve-aligned clustering and prediction with regression mixture models," Ph.D. dissertation, Dept. Comput. Sci., Univ. California, Irvine, CA, USA, 2004.
- [25] S. Ragothaman, S. Narasimhan, M. G. Basavaraj, and R. Dewar, "Unsupervised segmentation of cervical cell images using Gaussian mixture model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 70–75.
- [26] E. Kreyszig, *Advanced Engineering Mathematics*. Hoboken, NJ, USA: Wiley, 2010.

- [27] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 188–194, Jan. 2016.
- [28] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.
- [29] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive Bayes—Which naive Bayes?" in *Proc. 3rd Conf. Email Anti-Spam*, 2006, vol. 17, pp. 28–69.
- [30] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [31] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2951–2959.
- [32] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, "The cart decision tree for mining data streams," *Inf. Sci.*, vol. 266, pp. 1–15, 2014.
- [33] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, no. Mar., pp. 1157–1182, 2003.
- [34] A. D. Bull, "Convergence rates of efficient global optimization algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2879–2904, Oct. 2011.
- [35] P. Gupta and T. Dallas, "Feature selection and activity recognition system using a single triaxial accelerometer," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1780–1786, Jun. 2014.
- [36] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 569–575, Mar. 2010.
- [37] L. Wang, Z. Zhang, H. Long, J. Xu, and R. Liu, "Wind turbine gearbox failure identification with deep neural networks," *IEEE Trans Ind. Inform.*, vol. 13, no. 3, pp. 1360–1368, Jun. 2017.
- [38] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2001.
- [39] D. J. Sebald and J. A. Bucklew, "Support vector machine techniques for nonlinear equalization," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3217–3226, Nov. 2000.
- [40] C. Gold and P. Sollich, "Model selection for support vector machine classification," *Neurocomputing*, vol. 55, no. 1, pp. 221–249, 2003.



Dewei Yi (S'16) received the B.Eng. degree in software engineering from Zhejiang University of Technology, Zhejiang, China, in 2014, the M.Sc. (with distinction) degree in computer science from the Department of Computer Science, Loughborough University, Loughborough, U.K., in 2015, and the Ph.D. degree in cognitive vehicles from the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K., in 2018.

Since 2018, he has been a Postdoctoral Research Associate with the Centre for Autonomous Systems, Loughborough University. His current research interests include personalized driving assistance, autonomous vehicle, vehicular network, and advanced driver assistance systems.

Dr. Yi was the recipient of the grant of China Scholarship Council. He was also the recipient of the CommAgility MSc Project Prize for best wireless communication project in 2015.



Jinya Su (M'16) received the B.Sc. degree in mathematics from the School of Mathematics and Statistics, Shandong University, Weihai, China, in 2011, and the Ph.D. degree in autonomous systems from the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K., in 2016.

Since 2015, he has been a Research Associate with the Centre for Autonomous Systems, Loughborough University. His research interests include Kalman filter and machine learning and their applications to autonomous systems, such as intelligent vehicle and agricultural information system.

Dr. Su was the recipient of the Best Student Paper Award at the 19th International Conference on Automation and Computing (2013), the IEEE-IES Student Paper Travel Award at the 17th International Conference on Industrial Technology (2016), and the Annual ICI Prize from the Institute of Measurement and Control in 2016. He was the recipient of the prestigious Chinese Government Award for Outstanding Self-financed Students Abroad in 2015.



Cunjia Liu (M'16) received the B.Eng. and M.Sc. degrees in guidance, navigation, and control from Beihang University, Beijing, China, in 2005 and 2008, respectively, and the Ph.D. degree in autonomous vehicle control from Loughborough University, Loughborough, U.K.

From 2011, he was a Research Associate with the Department of Aeronautical and Automotive Engineering, Loughborough University, where he was appointed as a Lecturer in flight dynamics and control of unmanned vehicles in 2013 and a Senior Lecturer in 2018. His current research interests include optimization-based control, disturbance-observer based control, Bayesian information fusion, and their applications to autonomous vehicles for flight control, path planning, decision making, and situation awareness.



Wen-Hua Chen (M'00–SM'06–F'17) received the M.Sc. and Ph.D. degrees from Northeast University, Shenyang, China, in 1989 and 1991, respectively.

From 1991 to 1996, he was a Lecturer and then an Associate Professor with the Department of Automatic Control, Nanjing University of Aeronautics and Astronautics, Nanjing, China. From 1997 to 2000, he held a research position and then a Lecturer in control engineering with the Centre for Systems and Control, University of Glasgow, Glasgow, U.K. In 2000, he moved to the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K., as a Lecturer, where he was appointed as a Professor in 2012. As a Professor in autonomous vehicles, he is currently mainly working on the development of unmanned autonomous systems. His research interests include the development of advanced control, signal processing, and decision making methods and their applications in aerospace engineering.