Personalized Driver Workload Inference by Learning From Vehicle Related Measurements

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

***Abstract*—Adapting in-vehicle systems (e.g., advanced driver assistance systems and in-vehicle information systems) to individual drivers' workload can enhance both safety and convenience. To make this possible, it is a prerequisite to infer driver workload so that adaptive aiding can be provided to the driver at the right time and in an appropriate manner. Rather than developing an average model for all drivers, a personalized driver workload inference (PDWI) system considering individual drivers driving characteristics is developed using machine learning techniques via easily accessed vehicle related measurements (VRMs). The proposed PDWI system comprises two stages. In offline training, individual drivers workload is first automatically splitted into different categories according to its inherent data characteristics using fuzzy C-means (FCM) clustering. Then an implicit mapping between VRMs and different levels of workload is constructed via classification algorithms. In online implementation, VRMs samples are classified into different clusters, consequently driver workload type can be successfully inferred. A recently collected dataset from real-world naturalistic driving experiments is drawn to validate the proposed PDWI system. Comparative experimental results indicate that the proposed framework integrating FCM clustering and support vector machine classifier provides a promising workload recognition performance in terms of accuracy, precision, recall, F_1-score, and prediction time. The interindividual differences in term of workload are also identified and can be accommodated by the proposed framework due to its adaptiveness.**

***Index Terms*—Fuzzy C-means (FCM) clustering, personalized aiding, support vector machine (SVM), workload recognition.**

I. Introduction

RECENT interest in intelligent vehicles is more concentrated on how to enhance safety and convenience to drivers [1]–[4]. Different advanced functions are being developed for such a purpose. For example, advanced driver assistance systems (ADASs) such as adaptive cruise control and lane keeping system are developed to enhance the driving safety, which perceive environmental situation in real time and alert drivers to potential dangers. Moreover, in-vehicle information systems (IVISs) such as navigation systems, music, radio or even phone in certain countries [5] are also being developed, which brings much convenience to drivers by offering real-time advice and instructions.

However, the use of these advanced functions also creates problems such as increasing driver's distraction and workload, which in turn may annoy the driver or even increase the risk of traffic accidents [5]. Statistical reports from different countries and regions all indicate that there is a high correlation between traffic accidents and driver's errors induced by distractions and inappropriate workload [6]. As pointed out in [7], distraction occurs when a triggering event induces an attention shift away from the driving task, which often happens under a heavy driving workload due to multitasking with various electronic devices, such as a navigation system and a smart phone [8].

To guarantee driving safety while maintaining convenience, it is important to monitor driver workload in real time, upon which "adaptive aiding" can be supplied to the driver at the right time and in a proper manner [5], [9]. For instance, an earlier and clearer collision warning signal can be given to a driver under high workload compared to a driver under low workload [9], [10]; human machine interface can also be optimized based on the awareness of drivers and vehicles [11] (e.g., switching off certain IVIS for drivers under high workload). By doing so, driver workload can be potentially kept at an appropriate level for safe driving. Following this line of thought, this paper, rather than proposing a new ADAS or IVIS, concentrates on driver workload inference (DWI) so that existing advanced functions become more effective and more acceptable without sacrificing safety.

DWI systems, due to their significance in enhancing driving safety, are drawing increasing attention in recent years [12]–[20]. In these systems, a mapping between diagnostic measurements and different levels of driver workload is usually built using different approaches including machine learning, upon which the level of workload can be identified in real time with the advent of new diagnostic measurements. According to different types of measurements adopted in mapping building, the existing DWI systems can be broadly classified into three categories

Manuscript received July 14, 2017; accepted October 12, 2017. This work was supported by the U.K. Engineering and Physical Sciences Research Council Autonomous and Intelligent Systems Programme with BAE Systems as the leading industrial partner under Grant EP/J011525/1. The work of D. Yi was supported by the Chinese Scholarship Council. This paper was recommended by Associate Editor A. Hussain. *(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/TSMC.2017.2764263

including physiological measurement- vision- and vehicle measurements-based approaches.

In the physiological measurements-based approach [12]–[14], measurements such as electroencephalogram for brain signal, electrocardiogram for heart signal, respiration, and temperature are usually adopted. This approach is relatively accurate and promising; the major obstacle is its reliability in varied situations and for different individuals. Another approach is based on computer vision [15], [16], where eye movement metrics (e.g., eye blinks, pupil dilation, and gaze angle) and head movements are usually adopted to capture visual and cognitive related workload. This approach is undergoing a rapid development [21]; the possible challenges lie in its reliance on working condition (e.g., light intensity) and privacy issue due to the involvement of camera. Vehicle measurements-based approach is another promising approach and is drawing increasing attention recently [17]–[20]. Unlike physiological measurement and vision-based approaches which directly measure the driver, the vehicle measurements-based approach works on driving performance via vehicle related measurements (VRMs) such as lane position, speed, steering angle and accelerations. The research in this paper falls into the last category.

The diagnostic measures in this paper are VRMs comprising vehicle speed and three-axis accelerations. It should be noted that this paper mainly focuses on proposing a personalized DWI system and its validation using experimental data from real-world naturalistic driving experiments; different sources of measures can be accommodated by the proposed framework by augmenting them into the measurement vector. These measurements are chosen by the following two observations. First, they are nonintrusive, easily accessed thorough low cost hardware available on vehicles and have a lower requirement on working condition. Second, they are closely related to individual drivers' driving performance and surrounding driving situations, and have been proved to be effective for workload inference in previous works (see [2], [17]–[20] among others).

After diagnostic measurements are defined, the next step is to interpret the data. For a large dataset with multiple dimensions, it is generally hard to work out the relationship in an explicit manner. To this end, machine learning techniques are adopted [19], which extract relationships in an automatic manner. The proposed DWI system compromises two stages including offline training and online implementation. In offline training, different from the existing machine learning-based approaches [17]–[20] where only classifiers are trained, the proposed DWI system contains two steps. In this first step, individual drivers' workload is divided into several categories using clustering algorithms, where the number of clusters are determined in an automatic and optimal manner using a particular criterion. This is done by observing the following.

1) Different drivers may undergo different levels of workload even for a similar driving situation.
2) Different drivers may have different numbers of workload category.
3) Even the number of workload category is the same, the threshold may be different.

As a result, to make the DWI system effective and adaptive to individual drivers, the personal driving characteristics (reflected by driving performance data) should be accommodated [17], [22]. Owing to the presence of this step, the proposed DWI is adaptive to different drivers without further parameter tuning and hence termed Personalized DWI (PDWI). In the second step, a mapping between VRMs and different levels of driver workload is built using classification algorithms since each cluster represents a particular level of workload. In the stage of online classification, the trained classifier is adopted to process new measurements so that driver's workload category can be pinpointed in real time.

Unlike previous results where the dataset for algorithm validation is obtained in a simulated environment [13], [19], the proposed PDWI system is validated by a recently collected dataset from real-world driving experiments of about 30 min with ten participants of various backgrounds. The experiments were conducted by the human–computer-interaction (HCI) Laboratory, University of Stuttgart, in 2013 [23]. Different algorithms for different components of the framework are compared using this dataset. Particularly in unsupervised learning phase, the clustering performance between centroid-based fuzzy C-means (FCM) clustering [24] and density-based density-based spatial clustering of applications with noise (DBSCAN) algorithm [25] are compared. In online classification, Gaussian mixture model (GMM)-based classifier [26], [27] and support vector machine (SVM) [28], [29] are compared. Comparative experimental results demonstrate that the proposed PDWI system integrating FCM clustering and SVM classifier obtains a promising workload recognition performance and is able to adapt to different drivers.

To the best of the authors' knowledge, this is the first attempt to combine clustering and classification to solve the problem of personalized driver workload inference (PDWI), particularly the proposed framework is validated using a recently collected dataset from real world driving experiments with promising results. To be more exact, the main contributions of this paper are summarized.

1) Clustering and classification algorithms are integrated systematically so that a PDWI system is developed to solve the problem of PDWI.
2) Nonintrusive and easily accessed VRMs are fed into the proposed PDWI system allowing the driver workload to be inferred.
3) The proposed PDWI is adaptive to different drivers in an automatic manner due to the presence of unsupervised workload clustering.
4) A recently collected dataset from real-world driving experiments rather than simulated environment is drawn to verify the whole framework. Comparatively experimental results demonstrate that the proposed PDWI system integrating FCM clustering and SVM classifier provides a promising workload recognition performance in terms of accuracy, precision, recall, F_1-score, and prediction time.

The reminder of this paper is organized as follows. Section II discusses the framework of the proposed PDWI system. Section III introduces FCM clustering for workload

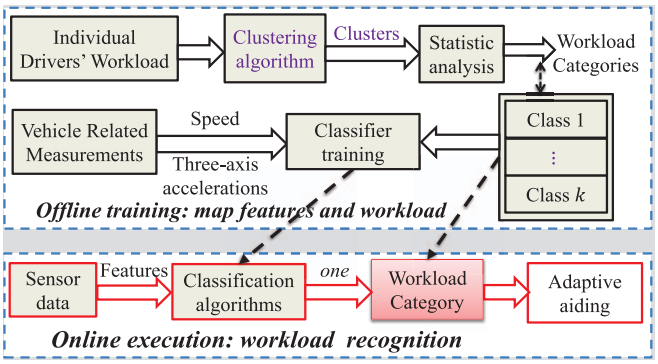


Fig. 1. Diagram of PDWI system.

categorization. Section IV discusses different classifiers for the implicit mapping establishment. In Section V, comparative experiments are conducted to validate the proposed framework using HCILab dataset. Section VI concludes this paper along with future work.

II. Personalized Driver Workload Inference

As highlighted in Section I, there exist substantial distinctions between different drivers in term of driving characteristics. Consequently, awareness of individual drivers' workload with a satisfying performance will be of particular importance, which can be seen as the basis and prerequirement for adaptive aiding systems [9]. The concept of personalized system is now drawing increasing attention in a wide range of applications. For example, to make ADAS cooperate better with human drivers, personalized driver steering model is developed in [1] by considering each driver's desired path. Personalized driver modeling is used in [2] to describe driving behavior so that driving styles are evaluated in a better way. In [22], personalized lane change models are developed by considering the dynamics and characteristics of individual driver/vehicle system during lane change maneuvers. In comparison to a model for all drivers, the personalized model is more effective and acceptable to drivers without sacrificing safety [22]. To this end, instead of proposing a new ADAS or IVIS, this paper develops a framework for PDWI. This is not easy since driver workload is not directly measurable and difficult to quantify, but only limited related measurements are available. So, machine-learning algorithms including clustering and classification are integrated. Clustering handles the adaptiveness of the proposed system, which divides driver workload into different categories automatically according to the intrinsic workload data characteristics. Classification exploits the implicit relationship between diagnostic measurements and different levels of driver workload so that given new diagnostic measurement the corresponding workload can be worked out.

The diagram of the proposed PDWI system is shown in Fig. 1, which mainly comprises two layers including offline training and online classification. All the data used in this paper are from [23] as detailed in Section V-A. In offline training, individual drivers' workload is first divided into different categories using clustering algorithms (e.g., FCM [24]). Statistical analysis (e.g., workload mean) is then performed on each cluster so that a nominal name (e.g., low, middle, and high) is given to each cluster according to its mean value and cluster number. An implicit mapping between VRMs and different levels of workload is further built using classification algorithms (e.g., SVM [30]). In online implementation, VRMs data are classified via the trained classifier so that the driver workload category can be identified. Based on the real time workload information, adaptive aiding can be provided to the driver at the right time and in an appropriate manner [5], [9]. In the following sections, different components of the proposed PDWI system will be discussed in detail.

III. Workload Clustering

In practical applications, as highlighted in Section I, it is generally challenging to determine the number of drivers' workload categories. To make the DWI system adaptive to different drivers, clustering algorithms are applied in this paper to determine the "optimal" number of workload categories in an automatic manner.

Clustering is one of the key techniques in unsupervised learning. They can be categorized into density-based and centroid-based algorithms. Centroid-based ones form clusters using the positions of cluster centroids. Several well-known centroid-based clustering algorithms are available such as k-mean clustering and FCM clustering [31]. In this paper, FCM is chosen due to its fine properties including better results for overlapped data [32] by incorporating uncertain information [33]. In density-based clustering, the data points are classified as core points, reachable points and outliers, where DBSCAN is a recognized density-based clustering algorithm [25]. To identify a more suitable algorithm for workload clustering, both centroid-based and density-based clustering algorithms are experimented, compared and analyzed. From the experimental results in Section V, it is discovered that FCM is more suitable for driver's workload clustering. In this following part, both DBSCAN and FCM clustering algorithms are introduced.

A. DBSCAN Clustering

DBSCAN is a popular density-based clustering algorithm [25], where no cluster number is required and can find arbitrary shaped cluster. In this approach, points closely packed together with many nearby neighbors are grouped as a cluster, and points lying alone in low-density regions (whose nearest neighbors are far away) are marked as outliers. There are two key parameters including the minimum points within a cluster "minPts" and the maximum radius of the neighborhood ϵ. Its details are omitted due to the lack of space; however, the main steps are summarized in Algorithm 1.

B. FCM Clustering With Optimal Cluster Number

Centroid-based FCM clustering is further discussed in this part, particularly the determination of its optimal cluster number is elaborated using a particular criterion.

Algorithm 1: Steps for DBSCAN Clustering Algorithm

(1) Select a point P and retrieve all points density-reachable from P with respect to ϵ and minPts;
(2) If P is a core point, a cluster is formed;
(3) If P is a border point, no points are density-reachable from P, then visit the next point;
(4) Continue the process until all points are processed.

1) FCM Clustering: In FCM clustering, data points are assigned to each cluster by fuzzy membership. Let $X = (x_1, \ldots, x_n)$ denote a set of n data points to be partitioned into d clusters. The algorithm is an iterative optimization that minimizes the cost function

$$J = \sum_{i=1}^{n} \sum_{j=1}^{d} u_{ij}^{m} \left\| x_i - o_j \right\|^2$$

where u_{ij} represents the membership of data point x_i in the jth cluster; o_j is the jth cluster center; $\| \bullet \|$ is a norm metric; and m is a constant controlling the fuzziness of the resulting partition. For a given data point x_i, the sum of the membership values satisfies $\sum_{j=1}^{d} u_{ij} = 1$. The cost function is minimized when data points close to the centroid of their clusters are assigned high membership values, and low membership values are assigned to data far from the centroid. The membership function represents the probability that a data point belongs to a specific cluster. In FCM clustering, the probability is dependent solely on the distance between data point and each individual cluster center. The membership functions and cluster centers are updated by

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\dfrac{\left\| x_i - o_j \right\|}{\left\| x_i - o_k \right\|} \right)^{\frac{2}{m-1}}}, \quad c_j = \frac{\sum_{i=1}^{n} u_{ij}^{m} x_i}{\sum_{i=1}^{n} u_{ij}^{m}}.$$

Starting with an initial guess for each cluster center, FCM converges to a solution o_j, where convergence can be detected by comparing the changes in the membership function or the cluster center at two successive steps.

2) Optimal Cluster Number: FCM is a centroid-based clustering algorithm; the determination of its optimal cluster number k^* is challenging. In this paper, the k^* is obtained using silhouette [13], [34]. The silhouette value is a measure of how similar that point is to points in its own cluster in comparison to points in other clusters. The silhouette value S_i for the ith point is given by

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{1}$$

where a_i is average distance from the ith point to other points in the same cluster, and b_i is the minimum average distance from the ith point to points in a different cluster, minimized over clusters. From the definition, it is clear that $-1 \leq S_i \leq 1$. A high silhouette value indicates that the ith point is well-matched to its own cluster and poorly matched to neighboring clusters. So if most points have a high silhouette value, then the clustering result is appropriate. And if many points have a low or negative silhouette value, then the clustering configuration may have either too many or too few clusters. Therefore, silhouette value is adopted as a criterion to optimize the cluster number. The main steps for the optimized FCM clustering are summarized in Algorithm 2.

Algorithm 2: Steps for FCM Clustering With Optimal Cluster Number

(1) Given a user-defined maximum clustering number k;
(2) Perform a series of FCM clustering under cluster number $i = 2, \cdots, k$ resulting in $FCM(2), \cdots, FCM(k)$ using the results in Section III-B1;
(3) Calculate the corresponding Silhouette values (all points) for each clustering configuration using the formula (1) resulting in $S(2), \cdots, S(k)$;
(4) The optimal cluster number is determined by $k^* = \arg\max_{i=2}^{k} S(i)$, with $FCM(k^*)$ being the optimal clustering configuration.

IV. Feature and Classifier Determination

Different workload clusters corresponding to different levels of driver workload have been established in Section III. In this section, different classification algorithms are then applied to build an implicit mapping between VRMs features and different levels of driver workload so that given new VRMs the corresponding workload category can be pinpointed and adaptive aiding can be performed. Before detailing the different classifiers, the input features for classification algorithms are first discussed.

A. Feature Determination

In comparison to other diagnostic measures, VRMs are nonintrusive, easily accessed and have a lower requirement on working condition, and more importantly they have been proved to be effective for workload inference in previous works [17]–[20]. For example, GPS and IMU sensors are easily available for most of modern vehicles, where position, speed, three-axis accelerations, and orientation information can be obtained. Other VRMs may include steering angle, distance from leading vehicle, lateral derivation. In this paper, speed and three-axis accelerations are chosen as the feature vector since these measurements are easily available and can reflect both driver's driving characteristics and environmental information [20]. As discussed in Section I, other sources of measurements can also be accommodated by the proposed framework by augmenting the measurement vector along with possible feature engineering (e.g., feature extraction and feature selection).

Remark: It is noted that contextual information regarding the driver (e.g., age, gender, and sleep quality), the vehicle (e.g., vehicle type) and driving environment (e.g., road type, traffic condition, and weather condition) is also important in DWI. Considering that this paper is mainly focused on PDWI using VRMs (i.e., low level sensing information), this type of information is not considered. Interested readers may refer to [35] and [36] for probabilistic approaches (i.e., Bayesian network) accommodating this type of information.

B. Classifier Determination

Given input features, the next step is to build an implicit mapping between features and different levels of workload using classification algorithms. Different algorithms are available to achieve this task, and without loss of generality two popular algorithms are considered in this paper including the parametric GMM-based classifier [26], [27] and nonparametric SVM classifier [28], [37].

1) GMM-Based Classifier: GMM-based classifier is a popular statistical classifier, and has been proved to be very effective and computationally inexpensive in different applications including speaker recognition and motion mode classification [26]. In this approach, there are generally two steps including GMM establishment and classification. The first step is to model feature distribution in each cluster using GMMs. GMM has the ability to form smooth approximation for general non-Gaussian probability density functions through a weighted sum of M component Gaussian functions, given by

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i)$$

where x is a D-dimensional data vector representing the features, M is the number of components, w_i represents the mixture weights for ith component, and $g(x|_i, \Sigma_i)$ is the ith component Gaussian density function. Each component density is a D-variate Gaussian function of the form

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\Sigma_i|}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$$

where μ_i, Σ_i denote mean and covariance matrix. The mixture weights w_i satisfy the constraint $\sum_{i=1}^{M} w_i = 1$. As a result, the complete GMM is parameterized by the mean vectors, covariance matrices and mixture weights of all component densities, which are collectively represented by the notation

$$\lambda = \{w_i, \mu_i, \Sigma_i\}, \quad i = 1, \ldots, M. \tag{2}$$

In this paper, each GMM corresponding to each cluster represents a class of driver's workload category such as low workload, middle workload, and high workload based on its physical meanings. In GMM establishment, there are two issues to be solved including optimal component number determination and parameter optimization so that the derived model best matches the distribution of the training data. In this paper, the optimal component number is determined using variance ratio criterion (VRC) [38]. And the parameters λ in (2) are optimized using maximum likelihood estimation via expectation-maximization (EM) algorithm [39] with initialization using k-mean++ so that the GMM likelihood given the training data is maximized.

Given different GMMs representing different levels of workload, the classification in GMM classifier is achieved by computing the distances through likelihood function between a new sample and different GMMs. And the cluster category could be determined by the corresponding GMM with smallest distance (or maximum similarity value) [26]. Suppose there are k^* clusters with corresponding GMMs, GMM(1), ..., GMM(k^*), the cluster that a test sample sequence $X = [X(1), \ldots, X(j)]$ belongs to is determined by the following formula:

$$\text{cluster}\left| X = \arg\min_{i=1}^{k^*} \sum_{k=1}^{j} -\ln\, p[X(k)|\text{GMM}(i)\right] \tag{3}$$

where $-\ln\, p[X(k)|\text{GMM}(i)]$ denotes the negative log-likelihood function of $X(k)$ given the ith GMM. The main steps for GMM classifier is summarized in Algorithm 3.

Algorithm 3: Steps for GMM Classifier With Optimal Component Number

(1) Given component number $k = 1, \cdots, M$ for $GMM(i)$ with $i = 1, \cdots, k^*$, establish a series of GMMs using EM algorithm for parameter optimization along with k-mean++ initialization;
(2) Determine the optimal component number for $GMM(i)$ with $i = 1, \cdots, k^*$ by maximizing the VRC;
(3) Perform classification using the formula in (3).

2) Support Vector Machine: Another popular classification algorithm is SVM, which is a nonparametric statistical algorithm, and no particular assumption is made on the underlying data distribution [28]. Given a training set $\mathcal{T} = \{(x_i, c_i)|1 \leq i \leq \tau\}$ with x_i the feature and c_i the class label, they can be projected into a Hilbert space $\mathcal{H}$ using a proper mapping $\Phi(.)$ resulting in $\mathcal{T} = \{((\Phi(x))_i, c_i)|1 \leq i \leq \tau\}$. The SVM separates the data using an optimal hyperplane H_p, which is determined by jointly maximizing the margin $2/\|w\|$ and minimizing the sum of classification errors $\sum_{i=1}^{\tau} \xi_i$ under the constraint $c_i(x_i w + b) - 1 \geq 0, 1 \leq i \leq \tau$

$$\Psi(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\tau} \xi_i$$

where ξ's are slack variables accounting for data nonseparability, and constant C is regularization parameter controlling the penalty assigned to errors and can effectively control the shape of decision boundary. A large value of C may lead to over-fitting and consequently requires optimization. The optimization problem can be solved by considering the dual optimization through the use of Lagrange multipliers α_i

$$\begin{cases} \max\limits_{\alpha} : & \sum_{i=1}^{\tau} \alpha_i - \dfrac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j c_i c_j \langle \Phi(x)_i, \Phi(x)_j \rangle_{\mathcal{H}} \\ \text{subject to:} & \sum_{i=1}^{\tau} \alpha_i c_i = 0, \quad 0 \leqslant \alpha_i \leqslant C, \forall i \in [1, \tau]. \end{cases} \tag{4}$$

To avoid computing the inner products in the transformed space $\langle \Phi(x)_i, \Phi(x)_j \rangle_{\mathcal{H}}$, kernel function $\mathcal{K}$ is introduced in [37] so that $\langle \Phi(x)_i, \Phi(x)_j \rangle_{\mathcal{H}} = \mathcal{K}(x_i, x_j)$. The decision rule is finally given by

$$f(x) = \text{sign}\left(\sum_{i=1}^{N_s} c_i \alpha_i \mathcal{K}(s_i, x) + b\right) \tag{5}$$

where $s_i, 1 \leq i \leq N_s$ denote the support vectors. Different kernels lead to different SVMs, where the commonly used are polynomial kernel of order p, $\mathcal{K}_{\text{poly}}(x, z) = (\langle x, z \rangle + 1)^p$, and Gaussian kernel $\mathcal{K}_{\text{gauss}}(x, z) = \exp(-\gamma\|x - z\|^2)$ with γ being a parameter inversely proportional to the width of the Gaussian

Algorithm 4: Steps for SVM Classifier With Parameter Optimization

(1) Find the optimal Hyperplane H_p using (4) along with an appropriate kernel (e.g. Gaussian kernel);
(2) Optimize the penalty C and kernel parameters (e.g. γ in Gaussian kernel);
(3) Identify the support vectors s_i, $1 \leq i \leq N_s$;
(4) Perform online prediction for a new sample using the formula (5).

kernel. Different mechanisms are available for multiclass classification, in this paper, one-versus-one is adopted due to its simplicity and effectiveness. The main steps for SVM classifier is summarized in Algorithm 4.

V. Experimental Validation

In this section, experimental validation is conducted on the proposed framework for DWI. Particularly, different algorithms for different components of the PDWI system are compared. The experiments are designed for clustering algorithms for workload categorization and classification algorithms for real-time workload inference. They are discussed in detail after the introduction of the dataset for algorithm validation in Section V-A.

A. Driving Dataset

The dataset for algorithm validation is from a real world driving experiment of about 30 min with ten participants; the experiment was conducted by the HCI Laboratory, University of Stuttgart, in 2013 [23]. In this experiment, speed and three-axis accelerations are collected by a smart phone, where speed is recorded by a GPS at the frequency of 1 Hz and three-axis accelerations are recorded by an accelerometer at 12 Hz. The dataset also includes data from other sensors, where the highest frequency is 128 Hz. As a result, data extrapolation has been done in [23] to create a uniformed data set at 128 Hz. The feature vector in this paper includes speed and three-axis accelerations. Along with physical sensors, there are also two webcams to record the driving scenario (passenger view onto the road) and a view of the driver. After returning from the drive, each participant was guided to the lab and directly performed a post-hoc video rating evaluating the perceived workload, where workload values lie between 0 (no workload) and 1000 (maximum workload). This method is termed subjective rating approach [15], which is a common approach to deriving workload ground truth data. The driving route is shown in Fig. 2, which consists of various road types such as highway, freeway, and tunnel among others. To demonstrate the distinctions between different drivers and the adaptiveness of the proposed system to different drivers, all ten participants available in the dataset are tested, where the results are summarized in Section V-C.

To make sure that different road types and driving situations are covered in training and testing datasets, only the first 50% of data in terms of time length (equivalent to 70% of the length of trajectory, see Figs. 3 and 4 for illustration) are used for offline training including clustering and classifier training. If

Fig. 2. Driving route for the dataset shown in the open street map from [23].

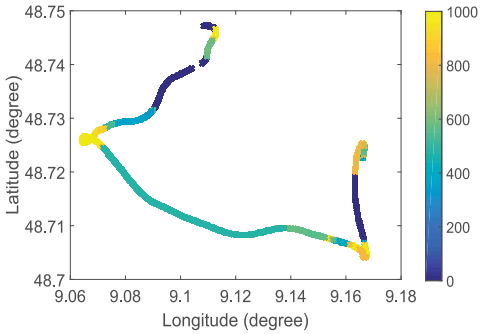


Fig. 3. Clustering results using DBSCAN approach: different colors represent different workload clusters.

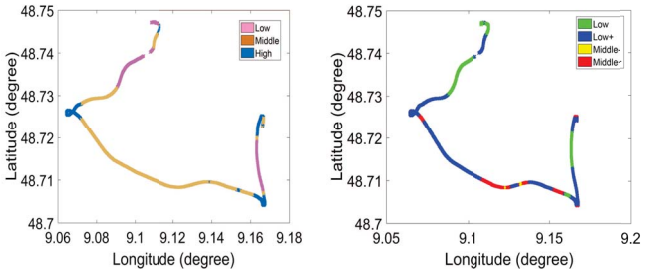


Fig. 4. Clustering results using FCM: driver 1 (left) and driver 2 (right). Different colors represent different workload clusters.

the first 70% in terms of time length of data are used for training, only a short distance of data would be available for testing, making the testing data lack of diversity. To test the algorithm, 1000 fragments (with length of 1 s per fragment) are randomly generated from the remaining data (see [14] for similar idea), and consequently each fragment contains 128 measurements of the feature vector. The fragment containing 128 feature measurements is treated as a testing sample for GMM-based classifier, and the mean value of feature vector in each fragment along with mean workload becomes a testing sample for SVM classifier. It is believed that this approach is more robust to sensor noises in real-time applications.

B. Workload Clustering Performance

As shown in Fig. 1, the first step in offline training is workload clustering so that different workload categories can be generated. In this paper, two commonly used clustering

TABLE I
DBSCAN CLUSTERING RESULTS

Cluster No	Mean workload	Cluster No	Mean workload	Cluster No	Mean workload	Cluster No	Mean workload
C1	0	C12	589	C23	519	C34	470
C2	1	C13	564	C24	540	C35	565
C3	734	C14	423	C25	562	C36	563
C4	933	C15	546	C26	720	C37	561
C5	1000	C16	390	C27	863	C38	560
C6	997	C17	322	C28	904	C39	824
C7	823	C18	359	C29	998	C40	813
C8	812	C19	360	C30	996	C41	822
C9	718	C20	378	C31	814	C42	763
C10	590	C21	396	C32	810		
C11	581	C22	432	C33	729		

TABLE II
SILHOUETTE VALUES UNDER DIFFERENT CLUSTER NUMBER k FOR DRIVER 1 (TOP) AND DRIVE 2 (BOTTOM)

k	Silhouette Value	k	Silhouette Value
2	0.7729	6	0.9192
3	**0.9317**	7	0.9179
4	0.9114	8	0.9301
5	0.8491	9	0.9182

k	Silhouette Value	k	Silhouette Value
2	0.7484	6	0.7478
3	0.7406	7	0.7517
4	**0.7676**	8	0.7522
5	0.7343	9	0.7570

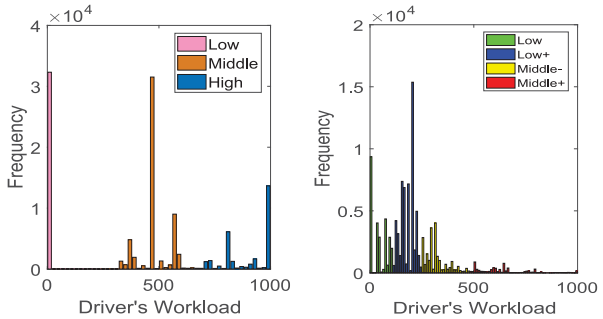


Fig. 5. Clustered workload histogram: driver 1 (left) and driver 2 (right). Different colors represent different workload clusters.

algorithms are tested and compared including density-based approach and centroid-based approach. Without loss of generality, two drivers' data are adopted to illustrate the uniqueness of individual drivers' driving characteristics, demonstrating the necessity of personalized system.

1) Clustering by DBSCAN: The density-based clustering algorithm DBSCAN in [25] is first tested. Following the steps in Algorithm 1, the two key parameters are chosen as $minPts = 135$ and $\epsilon = 10^{-4}$, respectively, through trial and error. The clustering results under this parameter setting for driver 1 are shown in Fig. 3 and the corresponding statistics are summarized in Table I. First it can be seen from Table I that the mean workload values are too close to each other (e.g., cluster nos. 5 and 6). Second 42 clusters are generated, which will bring challenges for real-time classification. To summarize, DBSCAN algorithm is not suitable for driver workload clustering.

2) Clustering by FCM: To this end, alternative centroid-based FCM clustering described in Algorithm 2 is exploited. In this paper, the fuzzy overlap controller is chosen $m = 2$ following the suggestion in [33]. First, silhouette values under different cluster numbers k from 2 to 9 are shown in Table II for driver 1 (left) and driver 2 (right).

One can see from Table II that the maximum silhouette values for two different drivers occur when k equals 3 and 4, respectively, consequently the driver's workload is divided into three and four clusters, respectively. Under this parameter setting, the workload clustering results for two different drivers are shown in Figs. 4 (for display on the map) and 5 (for histogram). The corresponding statistics (i.e., mean and standard derivation) for two different drivers are summarized in Table III. The cluster number for all ten participants are given in Table IV.

TABLE III
WORKLOAD CATEGORIES FOR DRIVER 1 (TOP) AND DRIVER 2 (BOTTOM)

Cluster No	C_1 (pink)	C_2 (yellow)	C_3 (blue)
Mean workload	1.9083	477.2567	908.1512
Standard derivation	12.7612	66.4750	100.6812
Workload level	Low	Middle	High

C_1 (green)	C_2 (blue)	C_3 (yellow)	C_4 (red)
44.3924	181.2466	324.7062	636.8308
38.5472	34.0122	50.7936	123.3157
Low	Low+	Middle-	Middle+

TABLE IV
CLUSTER NUMBER FOR ALL TEN PARTICIPANTS

Driver ID	1	2	3	4	5	6	7	8	9	10
Cluster number	3	5	4	4	2	2	4	5	4	3

Two observations can be drawn from the aforementioned figures and tables.

1) In comparison with density-based DBSCAN algorithm, the results of FCM clustering are more reasonable in term of cluster number and their physical meanings. One can see from Table IV that the optimal cluster numbers for all drivers are within the range [2, 5], which is consistent with the existing results in [13]. One can also see from Fig. 4(left) that high workload appears in cornering (shown in blue for driver 1), which is consistent with real scenario where corning involves careful manoeuvre.

2) By comparing the results of drivers 1 and 2 in Fig. 5 and Table III, and workload categorization for all ten participants in Table IV, one can see that there exist apparent differences between different drivers in terms of workload category number and their statistics. However, the proposed PDWI system using workload clustering is adaptive to different drivers without any parameter retuning.

C. Workload Inference Performance

In this section, online classification is validated on the basis of workload clustering and classifier off-line training. Both GMM classifier and multiclass SVM discussed in Section IV are tested and compared so that a more suitable classifier can be identified for the proposed PDWI system. All related algorithms in PDWI system 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.

1) Performance Measures: To quantitatively evaluate their performance, four commonly used criteria are adopted including accuracy, precision, recall, and F_β-score. As defined in (6), accuracy measures the proportion of true positive and true negative; precision measures the proportion of identified positives that are correctly positive as such; recall measures the proportion of positives that are correctly identified as such; and F-score F_β in (7) is widely used to evaluate overall performance of both recall and precision

$$\text{Accuracy} = \frac{t_p + f_p}{t_p + t_n + f_p + f_n}, \ \text{Precision} = \frac{t_p}{t_p + f_p}$$
$$\text{Recall} = \frac{t_p}{t_p + f_n} \tag{6}$$

where t_p is true positive, t_n is true negative, f_p is false positive, and f_n is false negative

$$F_\beta = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{\left(\beta^2 \times \text{Precision}\right) + \text{Recall}} \tag{7}$$

where β is the parameter weighting the importance of precision and recall. In this paper, $\beta = 1$ (termed harmonic average) is chosen putting same weight on precision and recall.

2) SVM Optimization: To make a fair performance comparison between GMM classifier and SVM classifier so that a more suitable classifier can be identified for the proposed PDWI system, the parameters of SVM including error penalty C in (4) and Gaussian Kernel scale γ in Section IV-B2 are optimized using a common heuristic approach in [40], where $C \in (0.01, 0.1, 1, 10, 100)$ and $\gamma \in (0.01, 0.1, 1, 10, 100)$. Without loss of generality, the overall accuracy for driver 1 under different parameter settings are taken as an illustrating example in Table V.

It can be seen that the parameter combination $C = 1$ and $\gamma = 1$ obtains the best performance in a finite number of tested parameter combinations for driver 1. Following this procedure, the optimal parameter combinations for all ten participants can be derived, given in Table VI.

TABLE V
OPTIMAL SVM PARAMETERS FOR DRIVER 1

Parameter	γ=0.01	γ=0.1	γ=1	γ=10	γ=100
C=0.01	55.9%	57.5%	78.1%	63.0%	55.9%
C=0.1	56.0%	58.2%	79.9%	67.9%	58.1%
C=1	56.0%	57.4%	**80.8%**	76.8%	62.8%
C=10	56.0%	57.4%	80.5%	77.2%	62.4%
C=100	56.0%	57.4%	80.1%	77.2%	62.9%

TABLE VI
OPTIMAL SVM PARAMETERS FOR ALL TEN PARTICIPANTS

Driver ID	Penalty (C)	Kernel Scale (γ)
1	1	1
2	10	1
3	1	1
4	1	1
5	1	1
6	100	1
7	1	1
8	10	1
9	1	1
10	1	1

3) Performance Comparison: A comparison between GMM classifier and optimized SVM classifier for all ten participants is drawn using the performance measures discussed in Section V-C1. Moreover, the average prediction time per sample is also calculated to evaluate its feasibility in real-time applications. All results are summarized in Table VII, where the algorithm with better performance in terms of different performance measures is highlighted in bold. The following conclusions can be drawn from Table VII.

1) The SVM classifier obtains consistently better results over the GMM classifier in terms of accuracy, precision, recall, and F_1-score for all drivers except drivers 1, 4, and 9; for driver 1, the GMM classifier obtains slightly better results in terms of different performance measures except precision; for drivers 4 and 9, GMM classifier obtains slightly better results than SVM classifier in terms of Precision and F_1-score but poorer results in term of Accuracy.
2) In terms of average performance for all ten testing drivers, SVM classifier obtains much better results than GMM classifier in terms of accuracy, precision, recall, and F_1-score, particularly the average accuracy has increased from 67.22% to 76.51%.
3) The computation time of SVM classifier is 27 times of GMM classifier. This is because there are a large number of support vectors in SVM classifier (non-parametric approach) while the log-likelihood function calculation in GMM classifier is fast due to a limited number of parameters (parametric classification approach). However, both of them are feasible for real-time computation.
4) Considering classification performance and computation time for real-time applications simultaneously, the optimized SVM is chosen as the classifier for the proposed PDWI system.

To summarize, the proposed PDWI system integrating FCM clustering for workload categorization and SVM classifier

TABLE VII
CLASSIFICATION RESULTS FOR TEN DRIVERS: GMM CLASSIFIER VERSUS SVM CLASSIFIER

	GMM classifier				SVM classifier			
Driver ID	Accuracy	Precision	Recall	F_1-score	Accuracy	Precision	Recall	F_1-score
1	**81.1%**	77.4%	**77.2%**	**0.7727**	80.8%	**84.9%**	70.5%	0.7283
2	63.0%	47.8%	51.2%	0.4861	**71.1%**	**72.8%**	**56.9%**	**0.5915**
3	53.3%	63.1%	54.8%	0.5754	**62.0%**	**68.2%**	**64.9%**	**0.6480**
4	78.8%	**85.5%**	54.6%	**0.6285**	**90.3%**	80.3%	**55.8%**	0.6273
5	59.9%	68.1%	66.8%	0.5980	**80.4%**	**81.0%**	**71.1%**	**0.7329**
6	62.6%	58.4%	60.4%	0.5803	**72.4%**	**74.1%**	**68.1%**	**0.6834**
7	59.7%	67.0%	69.2%	0.6679	**69.9%**	**78.8%**	**75.4%**	**0.7529**
8	66.3%	65.2%	56.9%	0.5650	**77.6%**	**74.1%**	**62.8%**	**0.6667**
9	69.3%	**72.5%**	**58.9%**	**0.6282**	**72.8%**	67.8%	49.2%	0.5347
10	78.3%	82.7%	76.4%	0.7853	**87.8%**	**91.2%**	**79.8%**	**0.8317**
Average	67.22%	68.8%	62.7%	0.6287	**76.51%**	**77.30%**	**65.40%**	**0.6797**
Time/sample	**4.4302**$\times 10^{-4}$ sec				1.2054×10^{-2} sec			

for real-time workload inference can effectively infer driver workload with an accuracy of 76.51% using all ten drivers' data available in the real-world dataset collected by HCI Laboratory. More importantly, the proposed framework is adaptive to individual drivers without parameter retuning due to the introduction of unsupervised workload clustering.

VI. CONCLUSION

To adapt driver assistance systems to individual drivers' workload level so that these systems become more effective and more acceptable to drivers, this paper developed a framework for PDWI by learning from easily accessed VRMs. This is achieved by integrating FCM clustering algorithm and SVM classifier, where FCM is to automatically split individual drivers' workload into different categories according to its intrinsic data characteristics, and SVM is to build an implicit mapping between VRMs features and different levels of driver workload.

The proposed framework is validated by a recently collected dataset (i.e., HCILab dataset) from a real-world driving experiment of ten drivers of different backgrounds. The interindividual differences in term of workload (i.e., number of categories) for different drivers are first identified, which demonstrates the necessity of a personalized learning system. Then comparatively experimental results using all ten drivers' data demonstrate that the proposed framework obtains a promising workload recognition performance in terms of accuracy (76.51%), precision, recall, F_1-score, and computation time. Moreover, it is also adaptive to different drivers due to its special hierarchical structure.

This paper mainly focused on proposing a systematic framework for PDWI system and initially demonstrated its feasibility using a recently collected real-world naturalistic driving dataset. There is still room for further development, where the following aspects are identified.

1) Different advanced clustering, feature engineering and classification algorithms can fit into the proposed framework so that workload inference performance can be further improved.
2) More VRMs (e.g., steering angle, vehicle lateral deviation, and vehicle headway) can be accommodated into the framework; this can be done by augmenting them into the measurement vector along with possible feature engineering.
3) The driving environmental information (e.g., road types, traffic condition, and weather condition) and driver's characteristics (e.g., age, gender, and sleep quality) providing certain prior information on workload may also be incorporated; data fusion techniques such as Bayesian network may be useful.
4) Online learning may be added to the proposed framework, since individual drivers' workload categories may change with more driving experience being accumulated; this inevitably requires new driver's labeled data.

REFERENCES

[1] 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.
[2] 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.
[3] R. Ding, M. Yu, H. Oh, and W.-H. Chen, "New multiple-target tracking strategy using domain knowledge and optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 605–616, Apr. 2017.
[4] J. Su and W.-H. Chen, "Model-based fault diagnosis system verification using reachability analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2710132.
[5] J. Ziegler, T. Hussein, D. Münter, J. Hofmann, and T. Linder, "Generating route instructions with varying levels of detail," in *Proc. 3rd Int. Conf. Automotive User Interfaces Interact. Veh. Appl.*, Salzburg, Austria, 2011, pp. 31–38.
[6] J.-S. Wang, R. R. Knipling, and M. J. Goodman, "The role of driver inattention in crashes: New statistics from the 1995 crashworthiness data system," in *Proc. 40th Annu. Assoc. Adv. Automotive Med.*, 1996, pp. 377–392.
[7] T. Horberry, J. Anderson, M. A. Regan, T. J. Triggs, and J. Brown, "Driver distraction: The effects of concurrent in-vehicle tasks, road environment complexity and age on driving performance," *Accident Anal. Prevent.*, vol. 38, no. 1, pp. 185–191, 2006.
[8] G. Cena, A. Valenzano, and S. Vitturi, "Advances in automotive digital communications," *Comput. Stand. Interfaces*, vol. 27, no. 6, pp. 665–678, 2005.
[9] W. Hajek, I. Gaponova, K. H. Fleischer, and J. Krems, "Workload-adaptive cruise control—A new generation of advanced driver assistance systems," *Transp. Res. F Traffic Psychol. Behav.*, vol. 20, pp. 108–120, Sep. 2013.
[10] 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. 17, no. 4, pp. 1108–1120, Apr. 2016.
[11] C. J. D. Patten, A. Kircher, J. Östlund, L. Nilsson, and O. Svenson, "Driver experience and cognitive workload in different traffic environments," *Accident Anal. Prevent.*, vol. 38, no. 5, pp. 887–894, 2006.

[12] G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia, and F. Babiloni, "Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness," *Neurosci. Biobehav. Rev.*, vol. 44, pp. 58–75, Jul. 2014.

[13] J. Zhang, Z. Yin, and R. Wang, "Recognition of mental workload levels under complex human–machine collaboration by using physiological features and adaptive support vector machines," *IEEE Trans. Human–Mach. Syst.*, vol. 45, no. 2, pp. 200–214, Apr. 2015.

[14] J. A. Healey and R. W. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 156–166, Jun. 2005.

[15] J. L. Harbluk, Y. I. Noy, P. L. Trbovich, and M. Eizenman, "An on-road assessment of cognitive distraction: Impacts on drivers' visual behavior and braking performance," *Accident Anal. Prevent.*, vol. 39, no. 2, pp. 372–379, 2007.

[16] S. Benedetto *et al.*, "Driver workload and eye blink duration," *Transp. Res. F Traffic Psychol. Behav.*, vol. 14, no. 3, pp. 199–208, 2011.

[17] Y. Zhang, Y. Owechko, and J. Zhang, "Driver cognitive workload estimation: A data-driven perspective," in *Proc. 7th Int. IEEE Conf. Intell. Transp. Syst.*, 2004, pp. 642–647.

[18] Y. Liang, M. L. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 340–350, Jun. 2007.

[19] F. Tango and M. Botta, "Real-time detection system of driver distraction using machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 894–905, Jun. 2013.

[20] G. A. M. Meiring and H. C. Myburgh, "A review of intelligent driving style analysis systems and related artificial intelligence algorithms," *Sensors*, vol. 15, no. 12, pp. 30653–30682, 2015.

[21] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[22] 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.

[23] S. Schneegass, B. Pfleging, N. Broy, F. Heinrich, and A. Schmidt, "A data set of real world driving to assess driver workload," in *Proc. 5th Int. Conf. Automotive User Interfaces Interact. Veh. Appl.*, Eindhoven, The Netherlands, 2013, pp. 150–157.

[24] E. Backer and A. K. Jain, "A clustering performance measure based on fuzzy set decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 1, pp. 66–75, Jan. 1981.

[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96. Portland, OR, USA, 1996, pp. 226–231.

[26] Y. Huang, K. B. Englehart, B. Hudgins, and A. D. C. Chan, "A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 11, pp. 1801–1811, Nov. 2005.

[27] B. Fernando, E. Fromont, D. Muselet, and M. Sebban, "Supervised learning of gaussian mixture models for visual vocabulary generation," *Pattern Recognit.*, vol. 45, no. 2, pp. 897–907, 2012.

[28] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, Pittsburgh, PA, USA, 1992, pp. 144–152.

[29] L. He *et al.*, "Common Bayesian network for classification of EEG-based multiclass motor imagery BCI," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 843–854, Jun. 2016.

[30] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2003.

[31] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, nos. 2–3, pp. 191–203, 1984.

[32] D. Yi, J. Su, C. Liu, and W.-H. Chen, "Data-driven situation awareness algorithm for vehicle lane change," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Rio de Janeiro, Brazil, 2016, pp. 998–1003.

[33] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.

[34] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[35] Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *IEEE Trans. Veh. Technol.*, vol. 53, no. 4, pp. 1052–1068, Jul. 2004.

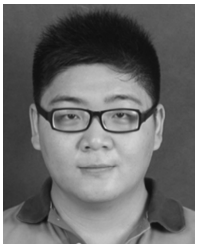
[36] Q. Ji, P. Lan, and C. Looney, "A probabilistic framework for modeling and real-time monitoring human fatigue," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 5, pp. 862–875, Sep. 2006.

[37] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998.

[38] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat. Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[39] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Soc. B (Methodol.)*, vol. 38, no. 1, pp. 1–38, 1977.

[40] Y. Aphinyanaphongs *et al.*, "A comprehensive empirical comparison of modern supervised classification and feature selection methods for text categorization," *J. Assoc. Inf. Sci. Technol.*, vol. 65, no. 10, pp. 1964–1987, 2014.



Dewei Yi (S'16) was born in Ürümqi, China, in 1991. He received the B.Eng. degree in software engineering from the Zhejiang University of Technology, Zhejiang, China, in 2014, and the M.Sc. degree (with distinction) from the Department of Computer Science, Loughborough University, Loughborough, U.K., in 2015, where he is currently pursuing the Ph.D. degree with the Department of Aeronautical and Automotive Engineering.

His current research interests include personalized driving assistance, autonomous vehicle, vehicular network, and advanced driver assistance systems.

Jinya Su (M'16) was born in Liaocheng, China, in 1990. He received the B.Sc. degree from the School of Mathematics and Statistics, Shandong University, Weihai, China, in 2011, and the Ph.D. degree from the Centre for Autonomous Systems, Loughborough University, Loughborough, U.K., in 2016.

He has been a Post-Doctoral Research Associate with the Centre for Autonomous Systems, Loughborough University, since 2015. His current research interests include advanced control strategies, Kalman filter, machine learning, and their applications to autonomous systems, such as intelligent vehicle and agricultural information system.

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, and the Ph.D. degree in autonomous vehicle control from Loughborough University, Loughborough, U.K., in 2011.

He has been a Lecturer of flight dynamics and control with Loughborough University, since 2013. 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 (F'06) received the M.Sc. and Ph.D. degrees from Northeast University, Shenyang, China, in 1989 and 1991, respectively.

In 2000, he became a Lecturer with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K., where he was appointed as a Professor in 2012. His current research interests include development of advanced control strategies, such as nonlinear model predictive control and disturbance-observer-based control, and their applications in aerospace and automotive engineering, and development of unmanned autonomous intelligent systems.

Dr. Chen is a fellow of the Institution of Engineering and Technology and the Institution of Mechanical Engineers.