An Intelligent Multifeature Statistical Approach for the Discrimination of Driving Conditions of a Hybrid Electric Vehicle

Xi Huang, Ying Tan, *Senior Member, IEEE*, and Xingui He

***Abstract*—As a new kind of vehicle with low fuel cost and low emissions, the hybrid electric vehicle (HEV) has been paid much attention in recent years. The key technique in the HEV is adopting the optimal control strategy for the best performance. As the premise, correct driving condition discrimination has an extremely important significance. This paper proposes an intelligent multifeature statistical approach to automatically discriminate the driving condition of the HEV. First, this approach periodically samples the driving cycle. Then, it extracts multiple statistical features and tests their significance by statistical analysis to select effective features. Afterward, it applies a support vector machine (SVM) and other machine-learning methods to intelligently and automatically discriminate the driving conditions. Compared with others, the proposed approach can compute fast and discriminate in real time during the whole HEV running mode. In our experiments, it reaches an accuracy value of 95%. As a result, our approach can completely mine the valid information from the data and extract multiple features that have clear meanings and significance. Finally, according to the prediction experiment by a neural network, the fitting experiment by the autoregressive moving average model, and the simulation results of the control strategy, it turns out that our proposed approach raises the efficiency of considerably controlling the HEV.**

***Index Terms*—Driving condition, hybrid electric vehicle (HEV), intelligent multifeature statistical discrimination (IMSD), neural network, statistical feature.**

I. Introduction

At present, faced with increasingly more resource and environmental problems, people have to pay more attention to the fuel economy (FE) and the emission of transportation such as vehicles. Developing a vehicle with lower fuel cost and lower emissions has become a goal of current vehicle industry [1].

Manuscript received June 7, 2009; revised November 16, 2009, May 18, 2010, and October 12, 2010; accepted November 9, 2010. This work was supported in part by the National High Technology Research and Development Program of China (863 Program) under Grant 2007AA01Z453, by the National Natural Science Foundation of China under Grant 60875080 and Grant 60673020, and by the Research Fund for the Doctoral Program of Higher Education in China. The Associate Editor for this paper was B. De Schutter.

X. Huang and Y. Tan are with the Key Laboratory of Machine Perception (Ministry of Education) and the Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (Corresponding author Y. Tan. e-mail: ytan@pku.edu.cn).

X. He is with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China.

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/TITS.2010.2093129

Three ways can be taken to improve engine efficiency and vehicle performance. The first way is to change the structure of the traditional engine or add some improving apparatus. This has the lowest cost because it does not change the process of vehicle production. Consequently, the performance improvement is limited. The second way is through an electric vehicle (EV) and a fuel cell vehicle [2]. These two types of vehicles, which substitute the traditional fuel with a battery and a fuel cell, and the internal combustion engine with an electric motor, are the cleanest vehicles. However, the high manufacturing cost limits their development. Combining these two ways, a hybrid EV (HEV) was developed after 1995 as the third way [3]. The HEV combines the advantages of the traditional vehicle and the EV to reach a good balance between the cost and the FE. As a way with promising performance, the HEV has increasingly become one of the main development trends in vehicle industry.

An HEV generally has two or more power sources, e.g., fuel and electric power. Its engine combines the traditional internal combustion engine, electric motor, and battery in different ways [4]. Due to the highly efficient energy storage parts, the internal combustion engine of the HEV could be smaller, more efficient, and with lower emissions. Based on the combing ways of parts, the HEV is classified into a series HEV, a parallel HEV, and a hybrid HEV [5]. Whichever type is adopted, when the HEV outputs the power, how to appropriately manage the different power sources to improve its performance becomes an extremely important problem. It is also the key problem in the HEV research—the control strategy of the HEV [6].

The control strategies of the HEV can be classified into three types. The first is the intelligent control strategy or the heuristic control strategy. This strategy usually has some predefined rules. The HEV will manage the power sources following the rules [7]–[9]. This strategy is simple, fast, and easy to implement, but its control result is often far from the optimal point. The second is the static optimization strategy. This strategy will compute the optimal power split based on the inherent parameter of the HEV [10], [11]. This strategy is much more efficient than the intelligent control strategy. However, due to lack of consideration about the driving environment, it usually performs well under some specific driving conditions, whereas it might not be optimal under others. The third is the dynamic optimization strategy. This strategy not only considers the effect of the inherent parameter but also detects the change in the external factors. Combining all the factors, the strategy will compute the best power split to dynamically manage the power sources in real time [12]–[14]. This strategy is able to continuously modify the control model and give a better performance. However, enormous computation will be a great burden and results in a lack of time.

Among the aforementioned control strategies, most of them construct a uniform control model to manage the power sources. However, the range of parameters in the control model is wide, and the same parameter in different intervals will express the different system characteristics. As a result, a uniform system model is not adequate to describe the driving process of the HEV. To construct a more precise model, we need to decompose the uniform model into a number of submodels corresponding to the different parameter intervals [15].

The driving condition of the HEV is a complicated variable that is determined by multiple factors. It will exert a significant effect on the vehicle performance. The correct discrimination of driving conditions will help decompose the uniform model into a series of submodels corresponding to the different driving conditions, which can improve the control performance. Meanwhile, there are lots of factors that influence driving conditions, e.g., wetness of the road, resistance, terrain, traffic, and even weather. The data on these factors may not be directly collected because there is no exact numerical style. Thus, how to correctly discriminate the driving conditions based on the limited data that we have collected becomes a key problem to solve.

II. Related Work

To discriminate the driving conditions, the classification standard of driving conditions should be defined first. Currently, there is no uniform standard. Based on the actual requirements, various classifications are adopted, e.g., based on the road surface condition, the driving conditions are classified into dry, wet, slushy, icy, and snowy [16]; based on the road level, they can be classified into a highway, an urban road, and an extra urban road [17]–[19]. The congestion level is also a factor to classify the driving conditions [20]. Moreover, there are some classifications by combining the aforementioned methods [21], [22].

The driving condition is determined by various factors; therefore, the data that are collected for driving condition discrimination are also different. The common data contain the following two types: The first type is an image or a video, which supplies many details such as obstacles, pedestrians, and other vehicles to distinguish the driving conditions [23]–[25]. However, such data require complicated image processing to give the result, and the enormous computation may be a great burden. The second type is time series on velocity or acceleration. Such data are most commonly used and easily collected. Many research studies are based on it [19], [26], [27]. However, these data cannot supply enough details; thus, it is not suitable for the discrimination of the complicated driving conditions. In recent years, the data collected by the automotive radar have been increasingly being used for the discrimination of driving conditions [28]–[30].

According to the data type and the classification standard, various approaches are proposed to discriminate the driving conditions, e.g., based on the pictures captured in the vehicle, image processing and computer vision approaches are used to determine the outside environment [16], [23]. Some rule-based and fuzzy logical methods are also used to make the results robust [20], [31]. In [17], the hidden Markov model is adopted to predict the future driving conditions. In [32], a neural network is also adopted.

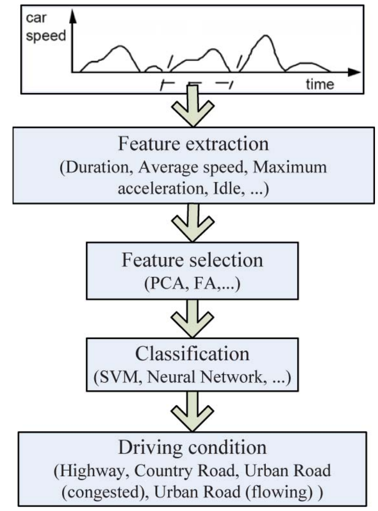

Fig. 1. Framework of the model.

When the driving conditions are discriminated, the control strategy based on the driving conditions can be built. In [18], six respective control rules are built on six driving conditions. In [32], the multimode control strategy is built based on driving patterns. In [33]–[35], various control strategies are proposed under different driving conditions.

This paper proposed a new approach of intelligent multi-feature statistical discrimination (IMSD). Based on the driving data of the HEV, it uses statistical analysis to extract and select multiple valid features. After the classifier learned the information on these features, it can intelligently discriminate the driving conditions in real time. This approach has simple data processing, definite meaning, fast computation, and high accuracy.

This paper will be organized as follows: In Section III, a framework of our model will be proposed. In Section IV, the classification standard of driving conditions will be discussed in detail. In Section V, we will analyze the extracted statistical features and their significance. In Section VI, the samples will be classified based on statistical features and different classifiers. In Section VII, the final discrimination approach will be determined. In Section VIII, some experiments will show the effectiveness of IMSD. Finally, Section IX concludes of this paper.

III. Framework of Our Model

The framework of IMSD is shown in Fig. 1, which consists of the following five steps.

A. Generation of Driving Samples

The driving cycle of the HEV is usually a long time series. In practical applications, we need to discriminate the current driving conditions in real time. Thus, we collect the driving data by sampling periodically. According to the short samples, the current driving condition can be quickly determined.

Suppose a long time series $\{S_1, S_2, \ldots, S_n\}$, we need to truncate k samples $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k\}$ of length t from it. First, we randomly select a start position n_0, i.e.,

$$n_0 = \text{random}(0, 1, \ldots, t-1). \tag{1}$$

Then, the continuous m $(m > k)$ samples of length t will be

$$\mathbf{s}_i = \{s_{ij}, j = 1, 2, \ldots, t\}, \qquad i = 1, 2, \ldots, m \tag{2}$$

where $s_{ij} = S_{n_0+(i-1)t+j}$, $n_0 + mt \leq n$.

The k samples will be randomly selected from them.

B. Feature Extraction

After collecting the samples, we need to extract the features that can describe the characteristics of driving conditions. Here, we choose the features with statistical significance and definite meaning as the elementary features, e.g., average speed and maximum acceleration.

Suppose that the multiple extracted features are $E_1, E_2, \ldots, E_m$ and that the vector that consists of them is

$$\boldsymbol{E} = (E_1, E_2, \ldots, E_m). \tag{3}$$

If the sample for feature extraction is

$$\boldsymbol{s} = (s_1, s_2, \ldots, s_t) \tag{4}$$

the process of feature extraction can be described as

$$\begin{aligned} \boldsymbol{E} &= \text{extract}(\boldsymbol{s}) \\ &= (\text{extract}_1(\boldsymbol{s}), \text{extract}_2(\boldsymbol{s}), \ldots, \text{extract}_m(\boldsymbol{s})) \end{aligned} \tag{5}$$

where $\text{extract}(\cdot)$ is a mapping from the sample to the elementary features; $\text{extract}_1(\boldsymbol{s}), \text{extract}_2(\boldsymbol{s}), \ldots, \text{extract}_m(\boldsymbol{s})$ are m components of $\boldsymbol{E}$. The specific forms of $\text{extract}(\cdot)$ can be various. If we suppose that $\boldsymbol{s}$ is a speed sample of the HEV and E_1 indicates the average speed, then

$$E_1 = \text{extract}_1(\boldsymbol{s}) = \frac{1}{t}\sum_{i=1}^{t} s_i. \tag{6}$$

C. Feature Selection

Elementary features are usually of a large quantity. Some of them may not be suitable to discriminate the driving conditions. Thus, we need to filter the elementary features and convert them into advanced features. The method for feature selection can be principal component analysis (PCA), factor analysis (FA), and so on. We will compare them in Section VI.

Suppose the advanced features after feature selection are $F_1, F_2, \ldots, F_k$ and the vector that consists of them is

$$\boldsymbol{F} = (F_1, F_2, \ldots, F_k). \tag{7}$$

The process of feature selection is described as

$$\boldsymbol{F} = \text{select}(\mathbf{E}) \tag{8}$$

where $\text{select}(\cdot)$ indicates the operation of feature selection. Based on the requirements, $\text{select}(\cdot)$ can be the different methods, e.g., PCA and FA. If we suppose that $\text{select}(\cdot)$ indicates the operation of the PCA, then

$$\boldsymbol{F} = (F_1, F_2, \ldots, F_k) = \text{PCA}(\boldsymbol{E}) \tag{9}$$

where $F_1, F_2, \ldots, F_k$ are the first k principal components that are selected.

Based on statistical analysis, we can also directly select k features from m elementary features without transformation, as we adopted in this paper. The process can be described as

$$\boldsymbol{F} = \text{select}(E_1, E_2, \ldots, E_m) = (E_{i_1}, E_{i_2}, \ldots, E_{i_k}) \tag{10}$$

where $\{E_{i_k}\}$ is a subsequence of $\{E_m\}$.

D. Classification

Based on the aforementioned advanced features, the driving conditions of the HEV can be determined by the different classifiers, e.g., k-nearest neighbor (kNN), neural network, and support vector machine (SVM).

Suppose the corresponding driving condition $R \in \{R_1, R_2, R_3, R_4\}$. R_1, R_2, R_3, and R_4 indicate the four driving conditions of the HEV, respectively. Then, our classification model will be

$$R = \text{classify}(\boldsymbol{F}) \tag{11}$$

where $\text{classify}(\cdot)$ is the classification function.

When there is a new sample to discriminate, based on its extracted feature $\boldsymbol{F}$, the result of classification R will be the current driving condition.

E. Four Driving Conditions

The aforementioned process is, in fact, a mapping from the driving cycles $\mathfrak{S}$ to the driving conditions $\mathfrak{R}$, i.e.,

$$f : \mathfrak{S} \rightarrow \mathfrak{R} \tag{12}$$

where $\mathfrak{S}$ is the set consisting of the driving cycles $\{S_n\}$, and $\mathfrak{R} = \{R_1, R_2, R_3, R_4\}$ is the set consisting of driving conditions.

In this paper, we classify the driving conditions of the HEV R into four types, including a highway $(R = R_1)$, a country road $(R = R_2)$, an urban road (congested) $(R = R_3)$, and an urban road (flowing) $(R = R_4)$. The detailed definition and explanation will be discussed in Section IV.

F. Characteristics of Our Model

From the aforementioned framework, four characteristics of the model can be seen.

1) *Dynamic determination of driving conditions:* Because the model periodically samples the whole driving process, the current driving condition can be decided in real time based on the sample.

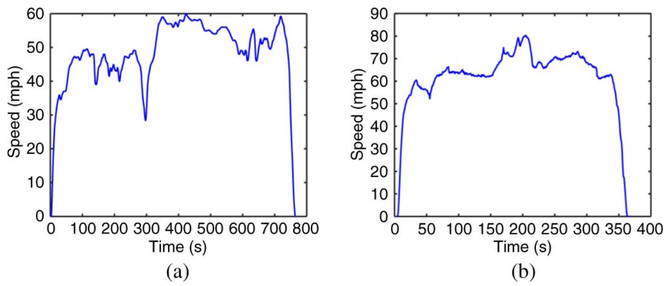

Fig. 2. Two driving cycles of a highway.

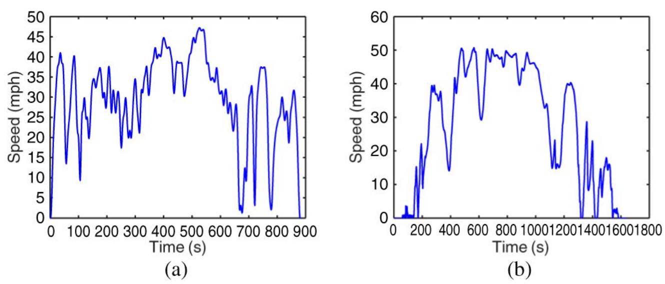

Fig. 3. Two driving cycles of a country road.

2) *Multiple features:* The model extracts more than one feature from a sample to provide accurately enough information to determine the driving condition.
3) *Statistical method for feature extraction and selection:* We use the statistical analysis and test to obtain multiple significant features, which have obvious meaning in statistics and are convenient for feature explanation.
4) *Intelligent discriminant:* The classifier in the model adopts various machine-learning algorithms, which make the model automatically learn the features and intelligently discriminate the current driving condition.

IV. Driving Conditions of the Hybrid Electric Vehicle and Their Feature Analysis

We classify the driving conditions into four types, which is the most common and representative.

A. Highway

A highway is a main road between important destinations, such as cities and towns. It has a lower limit of the driving speed and can afford the heavy traffic. In Fig. 2, the two speed sequences are collected under the real highway condition. In the figure, we can see the HEV keeps a high speed (above 50 mi/h) and drives smoothly. There is no interval of stopping (0 speed) in the cycle.

B. Country Road

A country road is a road that connects cities and countries. Compared with the highway, its speed standard and traffic capacity are lower. In Fig. 3, the sequences are collected under the country road condition. Comparatively, the speed decreases a lot (30–50 mi/h). The HEV periodically accelerates and decelerates; thus, the driving cycle becomes a form of wave.

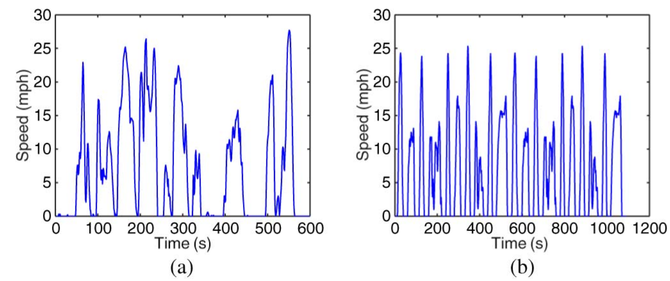

Fig. 4. Two driving cycles of an urban road (congested).

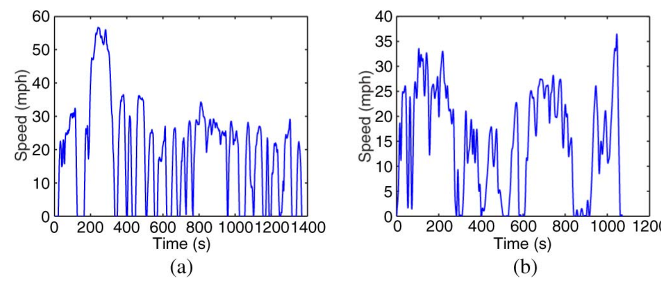

Fig. 5. Two driving cycles of an urban road (flowing).

C. Urban Road (Congested)

An urban road is the road in the city. There are numerous intersections and vehicles on the urban roads. When the traffic is heavy, the urban road usually congests. In Fig. 4, the sequences show the state of the congested urban road. Compared with the preceding two driving conditions, the HEV under this condition not only has a low speed (below 25 mi/h) but also periodically stops. Moreover, the stopping interval under this condition is usually large.

D. Urban Road (Flowing)

Compared with the congested urban road, the speed of the flowing urban road increases a little. However, due to the numerous intersections, the HEV still periodically stops. Under this condition, the speed of the HEV can keep for a while and will not immediately decrease. From the sequences in Fig. 5, the aforementioned characteristics are obvious.

E. Statistical Features of Different Driving Conditions

To find the features for discrimination, we extract the statistical features of driving cycles under the aforementioned driving conditions.

Some standard driving cycles that are collected in real world are adopted, including HWFET, US06-HWY, INDIA-HWY-SAMPLE, HYZEM-URBAN, HYZEM-HWY, HYZEM-SUB, CSC, WVUSUB, NYB, MANHATTAN, NYCC, CBD14, INDIA-URBAN-SAMPLE, UDDS, and WVUCITY.[1] Each driving cycle is a long speed sequence that is collected under one driving condition. The standard driving cycles belonging to the same driving condition can be combined to form a new long driving cycle under this driving condition. We create a 10 000-s

[1]More details about these cycles can be obtained from http://www.dieselnet.com/standards/cycles/.

TABLE I
STATISTICAL FEATURES OF DIFFERENT DRIVING CONDITIONS

Class	$v_{\max}$	v_{mean}	$a_{\max}$	$a_{\min}$	$(v*a)_{\max}$	$(v*a)_{\min}$	$(v*a)_{\text{mean}}$	I	σ_v	σ_a	σ_{v*a}
1	80.3	**39.77**	**6.9**	**-6.9**	209.72	-216.32	0.37	**0.019**	18.78	0.86	26.47
2	67.2	**19.61**	**6.9**	**-8.8**	142.88	-280.72	0.99	**0.12**	16.52	1.41	29.24
3	31.82	**7.01**	**6.2**	**-7.24**	99.36	-91.08	0.77	**0.38**	8.71	1.24	15.12
4	42.1	**15.17**	**3.87**	**-4.71**	100.5	-98.28	0.71	**0.14**	11.11	1.19	19.46

driving cycle for each driving condition to statistically analyze their features.

The result is shown in Table I, where various statistical features are presented, including the maximum speed $v_{\max}$; the average speed v_{mean}; the maximum acceleration $a_{\max}$; the minimum (negative maximum) acceleration $a_{\min}$; the maximum, minimum, and average values of speed multiplied by acceleration, i.e., $(v*a)_{\max}$, $(v*a)_{\min}$, and $(v*a)_{\text{mean}}$; their standard deviations σ_v, σ_a, and σ_{v*a}; and the idle rate (the percent of the stopping interval) I. In addition, the four driving conditions are indicated by Class 1 (highway), Class 2 (country road), Class 3 (urban road (congested)), and Class 4 (urban road (flowing)).

On the features v_{mean}, $a_{\max}$, $a_{\min}$, and I, the differences between classes are significant. Based on the features, the four driving conditions of cycles can be distinguished. We mark them in bold.

V. FEATURES OF SAMPLES AND THEIR SIGNIFICANCE

From the analysis of the driving cycles, we can obtain some features. However, in our model, the collected data are segments of the driving cycle. Thus, the features v_{mean}, $a_{\max}$, $a_{\min}$, and I may be not suitable. We need to generate some driving samples to analyze their validation and significance.

A. Generation of Samples

We use the method in the framework to generate the samples. Given a long driving cycle $\{S_1, S_2, \ldots, S_n\}$, we need to generate some samples $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k\}$ of length t from the driving cycle. The algorithm is given here.

Step 1: Select a start position $n_0 = \text{random}(0, 1, \ldots, t-1)$, $i = 1$.
Step 2: If $n_0 + t > n$, turn to Step 5.
Step 3: Let $s_{ij} = S_{n_0+j}, \quad j = 1, 2, \ldots, t$.
Step 4: $n_0 = n_0 + t$, $i = i + 1$, turn to Step 2.
Step 5: Select k samples randomly from i generated samples.

In the former research, we have learned that the driving period of the HEV is about 3 min [36]. When the length of the sample approaches or exceeds 3 min, it can reflect the characteristic of the current driving condition. Thus, we choose 150 s as the length of each sample and generate 300 samples under each driving condition. Then, we analyze the features of these samples.

B. Histogram of Samples

A histogram is a basic method to analyze the distribution of samples. It shows the frequency of samples at each interval on a feature.

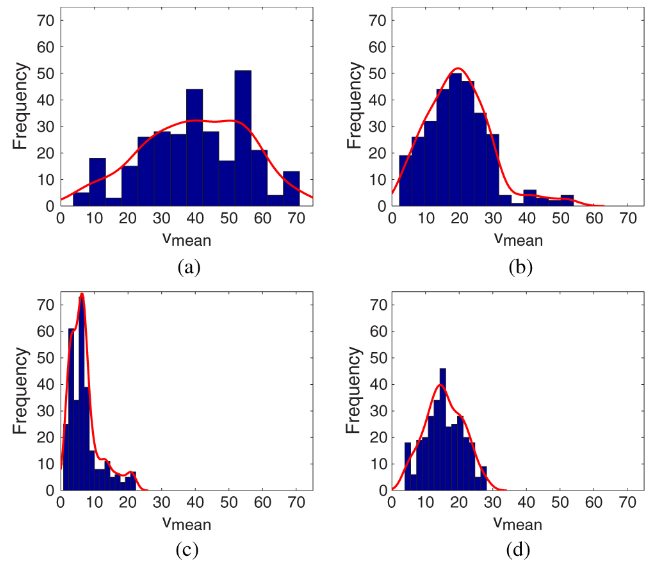

Fig. 6. Histograms on v_{mean} under the four driving conditions. (a) Highway. (b) Country road. (c) Urban road (congested). (d) Urban road (flowing).

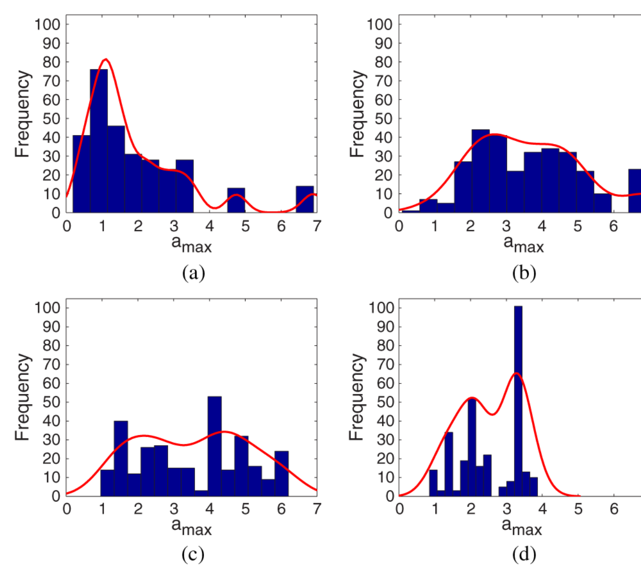

Fig. 7. Histograms on $a_{\max}$ under the four driving conditions. (a) Highway. (b) Country road. (c) Urban road (congested). (d) Urban road (flowing).

The histograms in Fig. 6 show the distributions of the samples under different driving conditions on the feature v_{mean}. In the figure, the red line is the fitting curve of the distribution density, which is obtained by the kernel density method and represents the most probable distribution form of the samples. We can see, under the four driving conditions, that the shapes and positions of the fitting curves are completely different, which means that, on the feature v_{mean}, there are significant differences in the distributions of the samples between driving

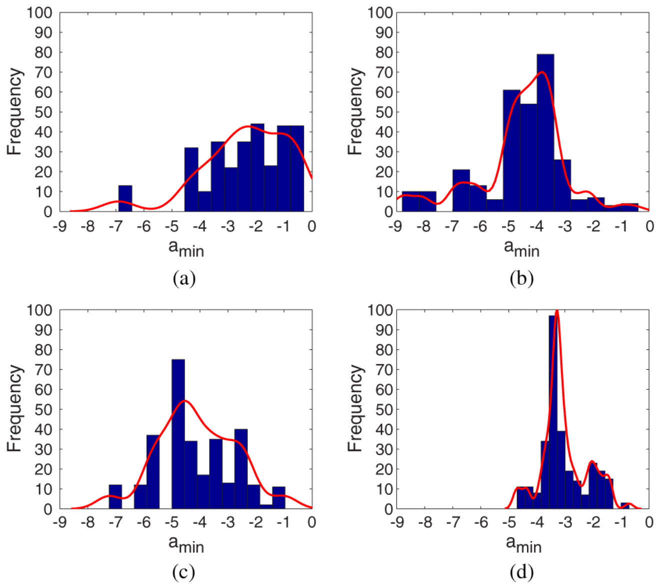

Fig. 8. Histograms on $a_{\min}$ under the four driving conditions. (a) Highway. (b) Country road. (c) Urban road (congested). (d) Urban road (flowing).

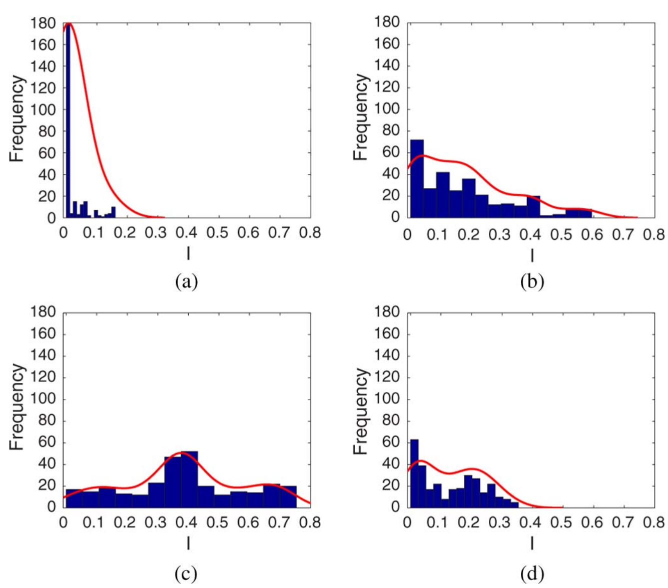

Fig. 9. Histograms on I under the four driving conditions. (a) Highway. (b) Country road. (c) Urban road (congested). (d) Urban road (flowing).

conditions. Thus, the feature v_{mean} can be used to discriminate the different driving conditions.

The histograms on the features $a_{\max}$, $a_{\min}$, and I are shown in Figs. 7–9, which express a similar characteristic as those on v_{mean}. Thus, these features can be also adopted for discrimination.

C. Boxplot and ANOVA

From the aforementioned histograms, we can see the basic distribution of the samples. To test the significance of these features, we adopt boxplot and analysis of variance (ANOVA) for a further analysis.

In a boxplot, the middle line of a box indicates the median of the samples, and the upper and lower edges of a box indicate

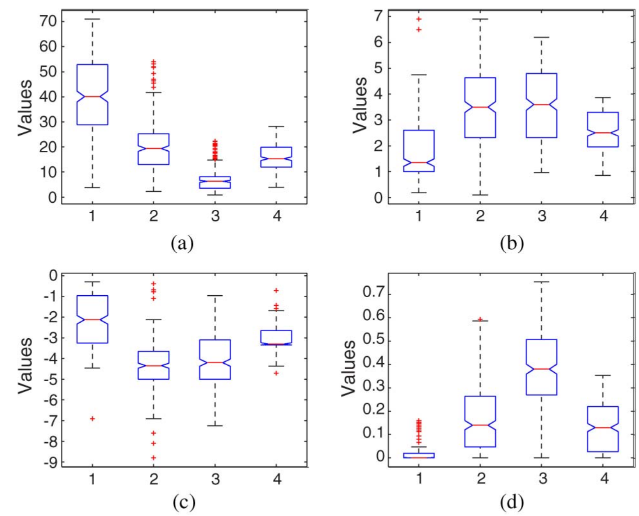

Fig. 10. Boxplots on four features, where the numbers on the horizontal axis indicate the four different driving conditions. (a) v_{mean}. (b) $a_{\max}$. (c) $a_{\min}$. (d) I.

the upper and lower quartiles, respectively. The top and the bottom of a whisker indicate the largest and smallest samples, respectively. The spacings between the different parts of the box help indicate the degree of dispersion and skewness in the samples.

The boxplots in Fig. 10 display the distributions of the samples under four driving conditions on the features v_{mean}, $a_{\max}$, $a_{\min}$, and I. It is obvious that, on each feature, the samples of different driving conditions have significant differences. Most of them are distributed in different intervals. Thus, the significance of these four features is obvious.

In addition, we use ANOVA to quantitatively analyze the significance of the features. The Kruskal–Wallis one-way ANOVA is adopted. The Kruskal–Wallis test is a nonparametric method to test for differences among two or more groups [37]. It does not assume a normal population, unlike the analogous one-way ANOVA.

First, it ranks all data from 1 to N, ignoring group membership. Then, the test statistic is given by

$$K = (N-1)\frac{\sum_{i=1}^{g} n_i(\overline{r}_i - \overline{r})^2}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}(\overline{r}_{ij} - \overline{r})^2} \tag{13}$$

where

$$\overline{r}_i = \frac{\sum_{j=1}^{n_i} r_{ij}}{n_i} \tag{14}$$

$\overline{r} = (1/2)(N+1)$ is the average of all the r_{ij}, n_i is the number of observations in group i, r_{ij} is the rank of observation j from group i, and N is the total number of observations.

The p-value is approximated by $\Pr(\chi^2_{g-1} \geq K)$. When K is large or the p-value is small enough $(p < 0.01)$, the differences between groups are significant. In our approach, the groups are the driving conditions, and the Kruskal–Wallis test on each feature is given in Table II. When the K-value is large or $p < 0.01$, we can confirm that the feature is significant, and we can differentiate the driving conditions.

TABLE II
KRUSKAL–WALLIS ANOVA ON DIFFERENT FEATURES

Feature	$v_{\max}$	v_{mean}	$a_{\max}$	$a_{\min}$	$(v*a)_{\max}$	$(v*a)_{\min}$	$(v*a)_{\text{mean}}$	I	σ_v	σ_a	σ_{v*a}
K	601.9	**621.7**	**318.26**	**397.25**	72.4	220.89	15.21	**506.42**	251.6	237.12	246.4
p	0	**0**	**0**	**0**	0	0	0.0016	**0**	0	0	0

TABLE III
CORRELATION OF DIFFERENT FEATURES

Correlation	v_{mean}	$a_{\max}$	$a_{\min}$	I
v_{mean}	1	-0.30	0.29	-0.62
$a_{\max}$	-0.30	1	-0.46	0.32
$a_{\min}$	0.29	-0.46	1	-0.17
I	-0.62	0.32	-0.17	1

From the table, all the features are significant $(p < 0.01)$; thus, the features are compared according to the K-value. v_{mean}, $v_{\max}$, I, $a_{\max}$, and $a_{\max}$ give five largest K-values, which should be adopted in our model. However, the correlation coefficient between $v_{\max}$ and v_{mean} is 0.91, which means a high correlation. Thus, we drop $v_{\max}$ to reduce the correlation between features. The K-values of $(v*a)_{\min}$, σ_v, σ_a, and σ_{v*a} are close to each other. To reduce the feature number, both of them are dropped. The experiment in Section VI-B will show that it is unnecessary to keep them in our model. The remainder of the features give small K-values; thus, all of them can be dropped. Finally, we adopt four significant features, i.e., $v_{\max}$, $a_{\max}$, $a_{\min}$, and I, as the best features for the discrimination of driving conditions. In Table III, we present their correlation coefficients. Both of them keep a low correlation with other features.

VI. EXPERIMENT OF CLASSIFICATION

After the features with an important significance are extracted, we use them to train the classifier and discriminate the driving conditions. The experimental setup is discussed here.

v_{mean}, $a_{\max}$, $a_{\min}$, and I are chosen as the input variables. The driving condition R is the output variable. One thousand two hundred new samples are generated from the driving cycles, among which there are 300 samples for each driving condition. The length of the sample is a parameter of classification. Fivefold cross validation is adopted to estimate the classification. The original sample is randomly partitioned into five subsamples. A single subsample is used for testing, and the remaining four subsamples are used as training data. The cross-validation process is then repeated five times, with each subsample used for testing once. Their average result is the final estimation.

A. Comparison of Feature Extraction Methods

In the framework of our model, the features are extracted and selected by a statistical method. To illustrate the effectiveness of our method, the common methods for feature extraction are adopted for a comparison, which are the fast Fourier transform (FFT), the discrete cosine transform (DCT), and the PCA.

1) OE: The experiment of classification with input features v_{mean}, $a_{\max}$, $a_{\min}$, and I is considered as the original experiment (OE). The multilayer neural network (MLNN) with one hidden layer is adopted as the standard classifier. Based on the conclusion of the prior experiment, we set the number of hidden nodes to 15, which will make the neural network reach a good balance between performance and complexity. In addition, according to Bishop's work [38], the number of training patterns should be around ten times as many as the weights in the network. There are about 75 weights in the network and 960 training samples, which is appropriate. The training algorithm is a Levenberg–Marquardt algorithm, and the maximum epoch is 500. The convergence goal is a mean-square error (MSE) of 0.01.

2) FFT and DCT: The FFT and DCT are two methods that are comprehensively used in digital signal processing, both of which can be used to compress the data and extract the frequency features of the data.

The FFT is the fast algorithm for the discrete Fourier transform [39]. Suppose that the input signal is $x(i)$, $i = 1, \ldots, N$ and that the FFT of the input signal is given by

$$y(k) = \sum_{n=1}^{N} x(n)\omega_N^{(n-1)(k-1)} \tag{15}$$

where $k = 1, 2, \ldots, N$, and ω_N is the Nth root of unity, which is defined as

$$\omega_N = e^{(-2\pi i)/N}. \tag{16}$$

By the FFT, the real input signal is converted into a complex frequency domain.

The DCT is defined as

$$y(k) = w(k)\sum_{n=1}^{N} x(n)\cos\frac{\pi(2n-1)(k-1)}{2N} \tag{17}$$

where $k = 1, 2, \ldots, N$, and $w(k)$ is defined as

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 1 \\ \sqrt{\frac{2}{N}}, & 2 \le k \le N. \end{cases} \tag{18}$$

From (17) and (15), the DCT uses fewer bases to transform the data. Thus, the information from the data will concentrate on a space with lower dimensionality [40].

In our experiment, the FFT and DCT transform the driving sample into a frequency domain and directly extract its frequency features. To compare them with our method, the first four dimensions of their output are chosen as the features of classification. The classifier is the same as the OE.

3) PCA: The PCA is a feature transform often used to reduce multidimensional data sets to lower dimensions for analysis [41]. It orthogonalizes the components of the input features; thus, output components are uncorrelated with each other. It orders the resulting orthogonal components (principal components) so that those with the largest variation come first.

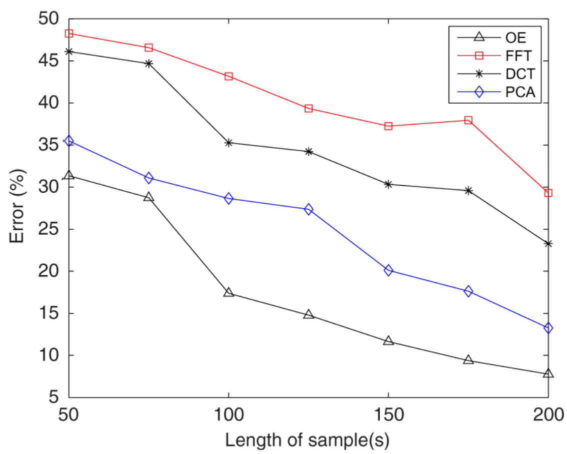

Fig. 11. Comparison of test errors by different feature extraction methods.

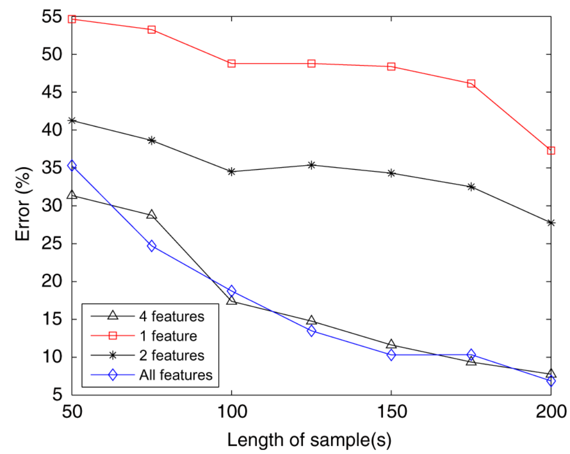

Fig. 12. Comparison of test errors under different feature numbers.

For a data matrix $\mathbf{X}^T$, where each row represents a different repetition of the experiment and each column gives the results of a particular feature, the PCA transformation is given by

$$\mathbf{Y}^T = \mathbf{X}^T\mathbf{W} = \mathbf{V}\mathbf{\Sigma} \tag{19}$$

where $\mathbf{V}\mathbf{\Sigma}\mathbf{W}^T$ is the singular value decomposition of $\mathbf{X}^T$.

In the experiment, the PCA is adopted for feature selection. It transforms 11 elementary features extracted from the samples into 11 advanced features. To keep the feature number the same as the OE, the first four principal components are chosen as the features of classification. The classifier is the same as the OE.

4) Comparison: The test errors of classification are shown in Fig. 11. We can see that the FFT and DCT obtain similar classification accuracy and that the DCT is slightly better. The test errors of the PCA and OE are much lower than those of the FFT and DCT, which illustrates that the features in the frequency domain are not enough for the classification of driving conditions and that a statistical method is more suitable for feature extraction. The test error of the OE is lower than the PCA. Compared with the PCA, the OE is simpler, and the features obtained by the OE all have clear meanings. Thus, our approach is best for feature extraction and selection.

B. Comparison of Feature Numbers

In our approach, four features, i.e., v_{max}, a_{max}, a_{min}, and I, are adopted for classification. We change the number of features to see if the result of classification is affected.

1) One Feature: Most approaches for the discrimination of driving conditions of the HEV are only based on one feature, i.e., v_{mean} [17]. We repeat the OE with only one input, i.e., v_{mean}. The result is shown in Fig. 12. Compared with our approach (four features), the test error dramatically increases. It illustrates that one feature cannot provide enough information to discriminate the driving conditions. This is also why we choose multiple features in our approach.

2) Two Features: In Section V-C, the significance of a_{max} and a_{min} is not as good as that of v_{mean} and I. We use the MLNN to test if the features a_{max} and a_{min} are necessary for our model. We repeat the OE with v_{mean} and I as the inputs. The result in Fig. 12 shows that the test error obviously increases compared with the OE (four features). Thus, we can confirm that a_{max} and a_{min} should not be dropped.

3) All Features: To test if four features are enough for our classification, we repeat the OE with all 11 features. The test error in Fig. 12 shows that its accuracy is similar to that of four features. The classification result does not obviously improve. Thus, there is no need to adopt more than four features.

4) Accuracy and Feature Number: From Fig. 12, the test accuracy is significantly improved with the feature number growing, which illustrates that the multiple-feature approach will mine more information from the data. However, when the feature number exceeds four, the accuracy improves not so obviously. There is no need to adopt more than four features.

Then, we choose the features v_{mean}, a_{max}, a_{min}, and I as the input features to test the performance of different classifiers.

C. Comparison of Classifiers

We choose the MLNN, linear classifier (LC), quadratic classifier (QC), kNN, and SVM as the classifiers to compare their capacities for the discrimination of driving conditions.

1) MLNN: In the OE, the MLNN with one hidden layer is adopted. We use its result as the result of the MLNN (see Fig. 13). We can see the test error decreases with the length of the sample growing. When the length exceeds 150 s, the test error becomes stable and remains below 12%. In fact, when the length of the sample is small, the information supplied by the sample is little, which cannot correctly reflect the driving condition. When the samples reach a certain length, all of them can reflect the current driving condition. Thus, the accuracy will increase and become stable.

2) LC and QC: The LC and QC are characterized by a simple structure and fast computation. It makes a classification decision based on the value of the linear or quadratic combination of the features. The operation of the LC or QC can be visualized as splitting a high-dimensional sample space with some hyperplanes. The samples belonging to the different driving conditions can be separated by those hyperplanes in the sample space. An LC or a QC is often used in situations where the speed of classification is an issue, particularly when the sample set is sparse. However, the LC generally cannot give

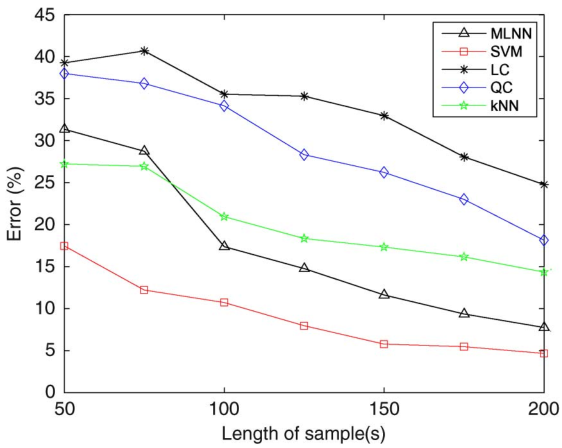

Fig. 13. Comparison of the test errors of classifiers.

TABLE IV
ACCURACY OF THE kNN BASED ON DIFFERENT k-VALUES

K-value	1	2	3	4	10	20
Accuracy	82.5%	82.5%	81.6%	80.8%	75.7%	72.9%

a good result for the linear nonseparable samples. The capacity of the QC is better than that of the LC. However, it still cannot obtain high accuracy for the complicated linear nonseparable samples.

From Fig. 13, the result of the LC and QC is obviously not as good as that of the MLNN. The QC is slightly better than the LC. It means that the samples are linear nonseparable.

3) kNN: The kNN is a basic method for classifying objects based on closest training samples. An object is classified by a majority vote of its k neighbors. The object will be assigned to the most common class among its k-nearest neighbors. If $k = 1$, then the object is simply assigned to the class of its nearest neighbor [42]. The kNN is a classifier with a simple structure and easy realization. It will give good classification accuracy when the sample is numerous and the samples of different classes are balanced. The accuracy of the kNN based on different k-values is shown in Table IV. When the k-value increases, the accuracy decreases. Thus, we set the parameter $k = 1$ in our experiment.

The classification error of the kNN is shown in Fig. 13. From the figure, the accuracy of the kNN is better than that of the LC and QC. When the length of the sample is below 100 s, the kNN is better than the MLNN. However, when the sample exceeds 100 s, the test error of the MLNN obviously decreases and remains lower than the kNN. This means that the kNN cannot mine more valid information as the sample length increases.

4) SVM: The SVM is a supervised learning method used for classification. Given the samples of two classes, the SVM will construct a separating hyperplane in the sample space, which not only classifies the samples correctly but also maximizes the margin between the two classes [43]. This is a quadratic programming optimization problem as

$$\min_{\mathbf{w},\boldsymbol{\xi}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right\} \tag{20}$$

TABLE V
CONFUSION MATRIX OF THE SVM WITH A SAMPLE LENGTH OF 150 s

Predicted	Actual				Total	Accuracy
	1	2	3	4		
1	297	23	5	0	325	95.2%
2	3	264	3	7	277	
3	0	2	288	0	290	
4	0	11	4	293	308	
Sample	300	300	300	300	1200	

subject to

$$c_i(\mathbf{w}\cdot\mathbf{x}_i - b) \geq 1 - \xi_i, \qquad \xi_i \geq 0; \quad i = 1, \ldots, n \tag{21}$$

where $\mathbf{w}$ and b are the parameters of the separating hyperplane, c_i is the class of the sample $\mathbf{x}_i$, $\boldsymbol{\xi}$ is the slack variable, and C is the penalty parameter.

The previously constructed SVM is considered a linear classifier. To classify the linear nonseparable samples, a kernel method is introduced. By mapping the samples into a feature space, the SVM can correctly separate them [44], [45]. In the experiment, we choose a radial basis function (RBF) kernel as its kernel function, which is defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\delta^2}} \tag{22}$$

where δ is the variance of the RBF, and $\mathbf{x}_i$ and $\mathbf{x}_j$ are two samples. For the SVM, two parameters δ and C need to be set. We search the optimal δ and C to reach the highest accuracy.

In Fig. 13, the SVM gives the best classification accuracy. When the length of the sample exceeds 150, the test error is below 5%, which is the best result. In addition, we give the confusion matrix of the SVM with a sample length of 150 s in Table V. From the table, the conditions of the highway and urban road (congested) can be correctly discriminated by the classifier relatively easily. The country road and urban road (flowing) are usually incorrectly discriminated. Both of them are most likely to be incorrectly classified into a highway.

5) Comparison: The comparison of the test errors of different classifiers is shown in Fig. 13, which shows that the SVM gives the smallest error. The result of the MLNN is slightly worse than that of the SVM. When the sample length exceeds 100 s, the accuracy of the MLNN is better than that of the kNN. The LC and GC are both worse than the kNN. The LC gives the biggest test error.

Moreover, we compare the classifiers based on the receiver operating characteristic (ROC) curve in Fig. 14. The ROC curve is a plot of the true positive rate versus the false positive rate for a classifier as its discrimination threshold is varied, which can be used to possibly select an optimal classifier by the area under the curve (AUC). An optimal classifier will give the biggest AUC. The curves of different colors in the figure are based on the different driving conditions. From the figure, the SVM gives the biggest AUC. The other classifiers in descending order of the AUC are the MLNN, the kNN, and the QC. The LC can be regarded as a special case of the QC and is worse than the QC. Therefore, we did not show its curves in the figure.

In Fig. 15, the time costs of five classifiers are presented. The QC and LC give a similar result, which is the lowest time cost.

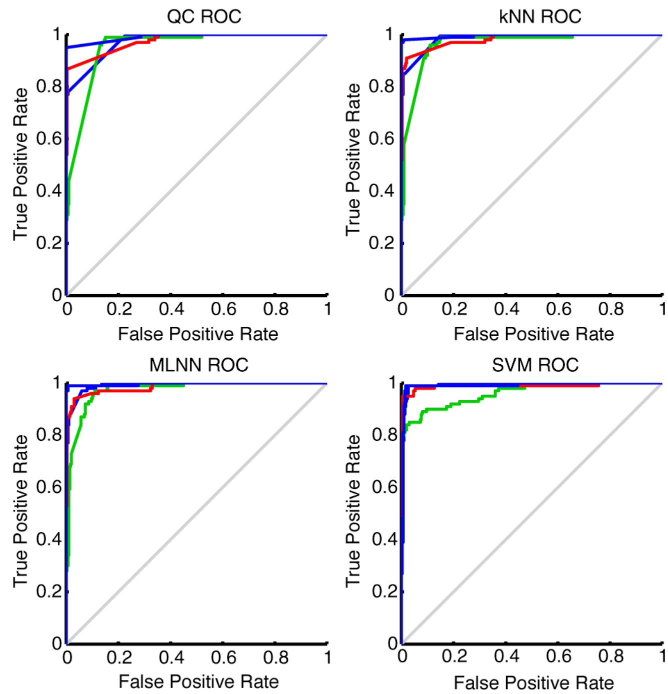

Fig. 14. ROC curves of classifiers.

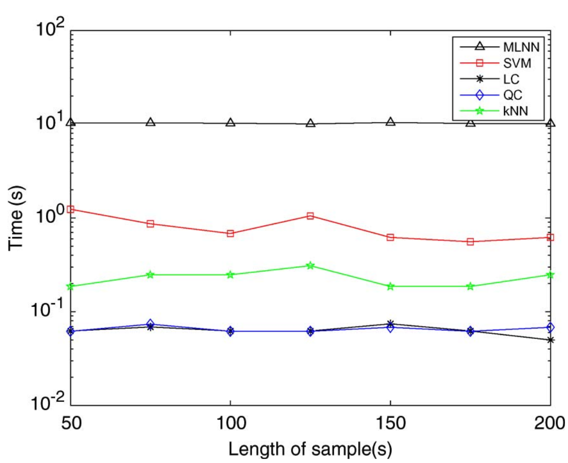

Fig. 15. Comparison of the time costs of classifiers.

The MLNN gives the highest time cost. The time cost of the SVM is slightly higher than that of the kNN. Their time costs are both between the LC and the MLNN.

From the aforementioned results, the SVM is undoubtedly the ideal classifier for the discrimination of driving conditions of the HEV.

VII. Intelligent Multifeature Statistical Discrimination Approach for the Discrimination of Driving Conditions

According to the preceding statistical analysis and classification experiments, we adopt the IMSD approach for driving condition discrimination.

- The length of the sample should be 150 s. When the HEV is running, it samples the driving cycle and determines the current driving condition every 150 s. As aforementioned, when the length of the sample exceeds 150 s, the classification will be robust and of high accuracy. On the other hand, to detect the change in the driving condition as soon as possible, the length of the sample should be reduced. Considering the foregoing reasons, 150 s is the best choice.
- Four features (i.e., v_{mean}, a_{max}, a_{min}, and I) are extracted from the sample to discriminate the current driving condition. These four features can be directly computed by a statistical method and have a clear statistical meaning. The features are suitable for discrimination.
- The SVM will be adopted as an ideal classifier. The SVM can give the smallest classification error. The time cost for classification is also the lowest, except for the QC and LC. By the SVM, the driving condition R will be correctly and efficiently discriminated.
- The classifier will be trained first based on these four features. Then, during the HEV running mode, the classifier will automatically discriminate the driving condition in real time.

VIII. Effectiveness of Intelligent Multifeature Statistical Discrimination

After IMSD was defined, it could be used in HEV control. We use the following experiments to illustrate the effectiveness of IMSD in HEV control.

A. *Neural Network Model for Prediction*

In the HEV control model, driving load prediction or velocity prediction is usually adopted to improve the vehicle performance [36]. There is a high correlation between the accuracy of prediction and the FE of the HEV. Thus, we adopted IMSD to improve the accuracy of prediction as follows.

The samples are generated from driving cycles. The length of each sample is 200 s. There are 100 samples for each driving condition in both the training and test sets. We hope to predict the average speed of the following 50 s based on the speed of the previous 150 s. Two methods are used for comparison.

1) *M1:* A uniform neural network model with one hidden layer for prediction is built. The input layer has 150 nodes, which are the speeds of the previous 150 s. The output layer has one node, which is the average speed of the following 50 s. The hidden layer has 40 nodes. The training algorithm is a Levenberg–Marquardt algorithm, and the maximum epoch is 500.
2) *M2:* Before prediction, we classify the samples into four classes by IMSD. For each class, we adopt one neural network for modeling and prediction. The structure of each neural network model is the same as the model in M1.

The result of the prediction experiment is given in Fig. 16. From the figure, M1 obtains the largest MSEs for prediction, which are 183.43 and 189.60 in the training and test sets, respectively. M2 obtains results of 60.17 and 81.20, respectively. It is obvious that the prediction accuracy is improved by adopting IMSD. From Fig. 16, in each class of M2, the obtained MSE is smaller than that of M1. As a result, M2 gives a smaller MSE than M1.

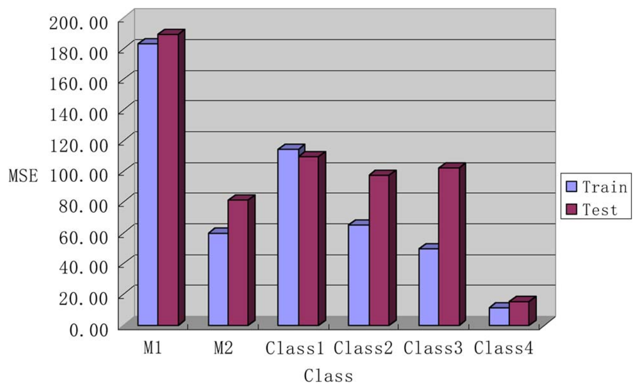

Fig. 16. Result of the prediction experiment. M1 indicates the MSE obtained by M1. M2 indicates the MSE obtained by M2. Class 1 to Class 4 indicate the MSE obtained from the samples under the four driving conditions of M2.

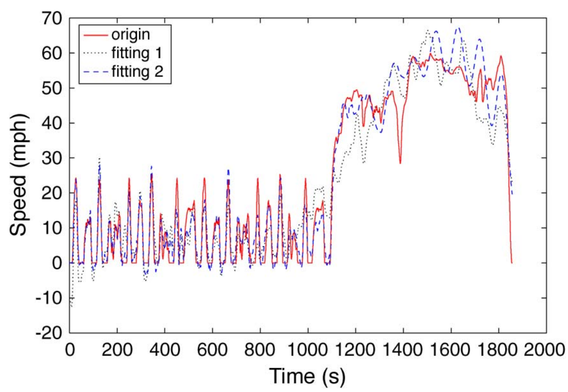

Fig. 17. Fitting result of the ARMA model. (Red line) Original data. (Black dotted line) Fitting curve obtained by M1. (Blue dashed line) Fitting curve obtained by M2.

B. *ARMA Model for Fitting*

In the simulation process of the HEV, we usually build a simple model to simulate the complicated process. Here, we use the autoregressive moving average (ARMA) model and IMSD to fit the driving cycle of the HEV.

The ARMA model is usually used to fit the time series. It uses the past p-values to approximate the current value of the time series, which is described as follows:

$$Y_t = c + \sum_{i=1}^{p} b_i Y_{t-i} + \sum_{j=1}^{q} a_j \varepsilon_{t-j} \tag{23}$$

where $\{Y_t\}$ is the time series for fitting, $\{\varepsilon_t\}$ is a random noise, and a_i $(i = 1, 2, \ldots, p)$, b_j $(j = 1, 2, \ldots, q)$, and c are the parameters. A big p or q usually means a complicated model.

In Fig. 17, the red solid line indicates a driving cycle combining the driving cycles under urban road and highway conditions. Two methods are adopted to fit it.

1) *M1:* A uniform ARMA(p, q) model is built. The values of p and q are both 200; thus, the model will be ARMA(200,200). The fitting result is shown by the black dotted line in Fig. 17.
2) *M2:* IMSD is used to determine the driving condition of each driving part first. Then, we build the different fitting models for the driving parts under different driving conditions. The urban part (previous 1100 s) and the highway part (following 750 s) are fitted by ARMA(50,50) and ARMA(100,100), respectively, which is shown by a blue dashed line in Fig. 17.

Obviously, from Fig. 17, M2 is closer to the original data than M1. The model of M2 is simpler than that of M1 because it uses less parameters. The accuracy of M2 is also better than that of M1. The MSE of fitting obtained by M1 is 54.96. M2 obtains an MSE of 27.08. The result of M2 is only half of that of M1. It is obvious that the fitting accuracy increases by using IMSD.

C. *Control Strategy Based on IMSD*

Based on IMSD, we can build different control rules according to the different driving conditions of the HEV. The control strategy in our experiment is based on two basic control strategies for a parallel HEV, namely, the motor assistant control strategy (MACS) [46] and the real-time optimization control strategy (RTOCS) [47].

In the MACS, the engine outputs the torque request of the vehicle when the torque request is below the maximum engine torque. The motor assists the torque if the required torque exceeds the maximum engine torque. The engine in this control strategy usually runs in low efficiency; thus, the control performance is not optimal. In the RTOCS, the engine will output the optimal engine torque based on the current engine speed and the engine efficiency map. When the engine torque exceeds the torque request of the vehicle, the excess torque will be recycled to charge the battery. The fuel efficiency of the engine in this control strategy is much better than that in the MACS. However, when the HEV is in an urban condition, the vehicle will keep a low driving load and a frequent change in the torque request. The optimal engine torque is usually much larger than the torque request. Thus, the excess energy should be recycled into the battery, and there will be an energy loss in the process of energy conversion. In this case, the performance of the RTOCS is worse than that of the MACS.

We build a mixed control strategy (MCS) based on them. The control strategy will first discriminate the current driving condition by IMSD. For the driving condition of an urban road (congested or flowing), the HEV will be controlled by the MACS. When the driving condition is a highway or a country road, the HEV will be controlled by the RTOCS.

The aforementioned control strategies were simulated on the software of ADVISOR [48]. In the simulation, the default parallel HEV model was adopted, and the vehicle ran over a driving cycle that is composed of standard driving cycles, including US06, NYCC, and NEDC.

The control results are presented in Table VI. In the table, the FE and the emissions are shown. The emissions contain three regular emissions, which are hydrocarbons (HC), carbon monoxide (CO), and nitrous oxides (NOx). We can see that the FE of the MCS is higher than that of both the RTOCS and the MACS, and the emission of the MCS is close to their lowest value. This means that the MCS gives a better control performance.

TABLE VI
COMPARISON OF THE FE AND THE EMISSIONS BASED ON DIFFERENT CONTROL STRATEGIES

Control Strategy	FE mpgge	HC g/ml	CO	NOx
MACS	34.81	0.73	6.57	0.50
RTOCS	32.39	0.80	4.67	0.78
MCS	**36.3**	**0.74**	**4.84**	**0.59**

According to the aforementioned experiments in HEV control, it can be seen that by using IMSD, the complicated model can be decomposed into multiple simple submodels. These submodels are specific and precise, which are helpful to reach a higher control performance.

IX. CONCLUSION

This paper has proposed a new IMSD approach. Combining statistical analysis and machine learning, this approach can automatically analyze the HEV driving data, extract multiple features, and dynamically discriminate the driving conditions, which is helpful for the best control strategy of the HEV.

During the HEV running mode, this approach periodically samples the driving cycle. Based on the samples, multiple features for the discrimination of driving conditions are extracted. Their significance is proved by histogram, boxplot, and ANOVA. Based on the extracted features, IMSD learns the information on the labeled samples by machine learning. Then, it can automatically discriminate the current driving condition of the HEV.

Compared with the current methods, the IMSD approach extracts more features to obtain more information about the driving conditions. It can accurately and dynamically discriminate the driving condition with a fast speed.

From the prediction experiment by a neural network, the fitting experiment by the ARMA model, and the control strategy by IMSD, the IMSD approach can be used to decompose the uniform model into multiple submodels, which can improve the efficiency and accuracy of the model and obtain the best control effect.

REFERENCES

[1] V. Wouk, "Hybrids: Then and now," *IEEE Spectr.*, vol. 32, no. 7, pp. 16–21, Jul. 1995.

[2] C. C. Chan, "An overview of electric vehicle technology," *Proc. IEEE*, vol. 81, no. 9, pp. 1202–1213, Sep. 1993.

[3] F. A. Wyczalek, "Hybrid electric vehicles: Year 2000 status," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 16, no. 3, pp. 15–25, Mar. 2001.

[4] C. C. Chan, "The state of the art of electric and hybrid vehicles," *Proc. IEEE*, vol. 90, no. 2, pp. 247–275, Feb. 2002.

[5] A. Kalberlah, "Electric hybrid drive systems for passenger cars and taxis," *SAE Trans.*, vol. 100, pp. 404–413, 1991.

[6] K. T. Chau and Y. S. Wong, "Overview of power management in hybrid electric vehicles," *Energy Convers. Manage.*, vol. 43, no. 15, pp. 1953–1968, Oct. 2002.

[7] N. Jalil, N. A. Kheir, and M. Salman, "A rule-based energy management strategy for a series hybrid vehicle," in *Proc. Amer. Control Conf.*, Jun. 1997, vol. 1, pp. 689–693.

[8] N. J. Schouten, M. A. Salman, and N. A. Kheir, "Fuzzy logic control for parallel hybrid vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 3, pp. 460–468, May 2002.

[9] F. U. Syed and D. Filev, "Real time advisory system for fuel economy improvement in a hybrid electric vehicle," in *Proc. Annu. Meeting NAFIPS*, 2008, pp. 1–6.

[10] A. Piccolo, L. Ippolito, V. Galdi, and A. Vaccaro, "Optimization of energy flow management in hybrid electric vehicles via genetic algorithms," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2001, vol. 1, pp. 434–439.

[11] C. C. Lin, H. Peng, J. W. Grizzle, and J. M. Kang, "Power management strategy for a parallel hybrid electric truck," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 6, pp. 839–849, Nov. 2003.

[12] R. Saeks, C. J. Cox, J. Neidhoefer, P. R. Mays, and J. J. Murray, "Adaptive control of a hybrid electric vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 4, pp. 213–234, Dec. 2002.

[13] L. Johannesson, M. Asbogard, and B. Egardt, "Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 366–371, Mar. 2007.

[14] M. Gao and M. Zhou, "Control strategy selection for autonomous vehicles in a dynamic environment," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2005, vol. 2, pp. 1651–1656.

[15] M. H. Huang, "Optimal multilevel hierarchical control strategy for parallel hybrid electric vehicle," in *Proc. Veh. Power Propulsion Conf.*, 2006, pp. 1–4.

[16] M. Yamada, K. Ueda, I. Horiba, and N. Sugie, "Discrimination of the road condition toward understanding of vehicle driving environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 1, pp. 26–31, Mar. 2001.

[17] M. Montazeri-Gh, A. Ahmadi, and M. Asadi, "Driving condition recognition for genetic–fuzzy HEV control," in *Proc. 3rd Int. Workshop GEFS*, Mar. 2008, pp. 65–70.

[18] C. C. Lin, H. Peng, S. Jeon, and J. M. Lee, "Control of a hybrid electric truck based on driving pattern recognition," in *Proc. Adv. Veh. Control Conf.*, Hiroshima, Japan, Sep. 2002.

[19] M. Montazeri-Gh and M. Naghizadeh, "Development of the Tehran car driving cycle," *Int. J. Environ. Pollut.*, vol. 30, no. 1, pp. 106–118, 2007.

[20] R. Langari and J. S. Won, "Intelligent energy management agent for a parallel hybrid vehicle—Part I: System architecture and design of the driving situation identification process," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 925–934, May 2005.

[21] A. Poursamad, "Optimal gain scheduling control strategy for parallel HEV based on traffic conditions," *Int. J. Elect. Hybrid Veh.*, vol. 1, no. 3, pp. 221–237, Aug. 2008.

[22] W. T. Hung, H. Y. Tong, C. P. Lee, K. Ha, and L. Y. Pao, "Characterization of vehicle driving patterns and development of driving cycles in Chinese cities," *Transp. Res. Part D: Transp. Environ.*, vol. 13, no. 5, pp. 289–297, Jul. 2008.

[23] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, T. Graf, and R. Schmidt, "Driving environment perception using stereovision," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 331–336.

[24] A. Barth and U. Franke, "Estimating the driving state of oncoming vehicles from a moving platform using stereo vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 560–571, Dec. 2009.

[25] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 425–437, Sep. 2008.

[26] M. Borner, L. Andreani, P. Albertos, and R. Isermann, "Detection of lateral vehicle driving conditions based on the characteristic velocity," in *Proc. Triennial World Congr. Int. Fed. Autom. Control*, Jul. 2002.

[27] Y. C. Ma, M. Chowdhury, A. Sadek, and M. Jeihani, "Real-time highway traffic condition assessment framework using vehicle-infrastructure integration (VII) with artificial intelligence (AI)," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 615–627, Dec. 2009.

[28] V. V. Viikari, T. Varpula, and M. Kantanen, "Road-condition recognition using 24-GHz automotive radar," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 639–648, Dec. 2009.

[29] S. H. Tsang, P. S. Hall, E. G. Hoare, and N. J. Clarke, "Advance path measurement for automotive radar applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 273–281, Sep. 2006.

[30] G. Alessandretti, A. Broggi, and P. Cerri, "Vehicle and guard rail detection using radar and vision data fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 95–105, Mar. 2007.

[31] J. M. Nigro and M. Rombaut, "IDRES: A rule-based system for driving situation recognition with uncertainty management," *Inf. Fusion*, vol. 4, no. 4, pp. 309–317, Dec. 2003.

[32] S. Jeon, S. Jo, Y. Park, and J. M. Lee, "Multi-mode driving control of a parallel hybrid electric vehicle using driving pattern recognition," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 124, no. 1, pp. 141–148, Mar. 2002.

[33] R. Langari and J. S. Won, "Intelligent energy management agent for a parallel hybrid vehicle," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 935–953, May 2005.

[34] M. Montazeri-Gh and M. Asadi, “Influence of the road grade on the optimization of fuzzy based HEV control strategy,” presented at the Powertrain and Fluid Systems Conf. Exhibition, Toronto, ON, Canada, 2006, SAE Technical Paper series, Paper 2006-01-3293.
[35] M. Montazeri-Gh and M. Asadi, “Intelligent control for HEV power management based on driving conditions,” in *Proc. 2nd ICMSAO*, Abu Dhabi, UAE, 2007.
[36] J. Yang, X. Huang, Y. Tan, and X. G. He, “Forecast of driving load of hybrid electric vehicles by using discrete cosine transform and support vector machine,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 2227–2234.
[37] W. Kruskal and W. Wallis, “Use of ranks in one-criterion variance analysis,” *J. Amer. Stat. Assoc.*, vol. 47, no. 260, pp. 583–621, Dec. 1952.
[38] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon, 1995, p. 380.
[39] N. Brenner and C. Rader, “A new principle for fast Fourier transformation,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-24, no. 3, pp. 264–266, Jun. 1976.
[40] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.
[41] I. T. Jolliffe, *Principal Component Analysis,* 2nd ed. New York: Springer-Verlag, 2002, ser. Springer Series in Statistics.
[42] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision*. Cambridge, MA: MIT Press, 2005.
[43] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[44] J. S. Taylor and N. Cristianini, *Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[45] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, “An introduction to kernel-based learning algorithms,” *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, May 2001.
[46] V. Wouk, “Hybrids: Then and now,” *IEEE Spectr.*, vol. 32, no. 7, pp. 16–21, 1995.
[47] V. H. Johnson, K. B. Wipke, and D. J. Rausen, “HEV control strategy for real-time optimization of fuel economy and emissions,” presented at the Future Car Congr., Washington, DC, 2000, SAE Paper 2000-01-1543.
[48] A. Brooker, T. Hendricks, V. Johnson, K. Kelly, B. Kramer, M. O’Keefe, S. Sprik, and K. Wipke, “ADVISOR: A systems analysis tool for advanced vehicle modeling,” *J. Power Sources*, vol. 110, no. 2, pp. 255–266, Aug. 2002.

Xi Huang received the B.S. degree in mathematics in 2006 from Peking University, Beijing, China, where he is currently working toward the Ph.D. degree in computer science with the Key Laboratory of Machine Perception (Ministry of Education) and the Department of Machine Intelligence, School of Electronics Engineering and Computer Science.

His recent research interests include neural networks, machine learning, and statistics.

Ying Tan (M’98–SM’02) received the B.S. degree in electronic engineering from the Electronic Engineering Institute, Hefei, China, in 1985, the M.S. degree in electronic engineering from Xidian University, Xi’an, China, in 1988, and the Ph.D. degree in signal and information processing from Southeast University, Nanjing, China, in 1997.

In 1997, he became a Postdoctoral Research Fellow and then an Associate Professor with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei. He was a Full Professor, an advisor of Ph.D. candidates, and the Director of the Institute of Intelligent Information Science, University of Science and Technology of China. He was with the Chinese University of Hong Kong, Shatin, Hong Kong, in 1999 and during 2004–2005. He was an electee of the 100-Talent Program of the Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Full Professor and an advisor of Ph.D. candidates with the Key Laboratory of Machine Perception (Ministry of Education) and the Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing. He is also the Head of the Computational Intelligence Laboratory, Peking University. He has authored or coauthored more than 200 academic papers in refereed journals and conferences and several books and book chapters. His current research interests include computational intelligence, swarm intelligence, artificial immune systems, intelligent information processing, pattern recognition, bioinformatics, statistical learning theory, and their applications.

Dr. Tan was the Program Committee Chair for the 2008 International Symposium on Neural Networks and the General Chair for the 2010 International Conference on Swarm Intelligence (ICSI). He is the General Chair of ICSI 2011. He is an Associate Editor of the *International Journal of Swarm Intelligence Research* and the *IES Journal B: Intelligent Devices and Systems* and an Associate Editor-in-Chief of the *International Journal of Intelligent Information Processing*. He is a member of the Advisory Board of the *International Journal of Knowledge-Based and Intelligent Engineering Systems* and of the Editorial Board of the *Journal of Computer Science and Systems Biology* and *Applied Mathematical and Computational Sciences*. He is also the Editor of Springer Lecture Notes on Computer Science, LNCS 5263, 5264, 6145, and 6146, and the Guest Editor of several referred journals, including *Information Science*, *Softcomputing*, the *Computer Journal*, and the *International Journal of Artificial Intelligence*. He was the recipient of a number of academic and research achievement awards from his country and universities due to his outstanding contributions and distinguished works, including the 2009 National Natural Science Prize of China.

Xingui He received the Bachelor’s and the Ph.D. degrees in approximation theory from Peking University, Beijing, China, in 1960 and 1967, respectively.

In the 1980s, he studied database and database machines with the Department of Computer and Information Science, Ohio State University, Columbus. Since 2001, he has been an Academician with the Chinese Academy of Engineering and the leader of the Consultative Expert Group on information science with the Chinese National Programs for Fundamental Research and Development (973 Program). From 2002 to 2006, he was the Dean of the School of Electronics Engineering and Computer Science, Peking University. He is currently a Professor and the Chair of the Academic Committee of the School of Electronics Engineering and Computer Science, Peking University. He has been engaged in the theoretical research and engineering practices of computer software and artificial intelligence for a long time. He is one of the pioneers of computer software in China. He has made creative and systematic contributions in the fields of fuzzy theory and technology, computational intelligence, and databases. As of 2010, he has published more than 150 academic papers and 11 Chinese and English books, including *Fuzzy Knowledge Processing Theory and Technology*, *Fuzzy Database Systems, Knowledge Processing and Expert Systems*, *Process Neural Networks—Theory and Applications*, etc. He is also the Editorial Director of multiple dictionaries and encyclopedia. He is the Editor-in-Chief for *Computer Engineering and Design*, the *CAAI Transactions on Intelligent Systems*, and *Computer Science and Exploration*. From 1998 to 2010, he was a Co-Editor-in-Chief for the *Journal of Computers*. He is also the member of the editorial committee of the *Journal of Software* and many other prestigious journals.

Dr. He was the President of Beijing Computer Association from 1999 to 2004. During 1999–2008, he was the Chair of the Anti-Harsh Environment Committee of the China Computer Society. In 2008, he became the Director of the Database Organization Committee of the China Computer Society. In 2010, he was elected Deputy Chairman of the China Association of Artificial Intelligence. He was the recipient of 19 national or ministerial-level scientific and technological progress awards, including the Guanghua Science and Technology Award and the Science and Technology Progress Award of the He-Liang-He-Li Fund in China in 2007.

An Intelligent Multifeature Statistical Approach for the Discrimination of Driving Conditions of a Hybrid Electric Vehicle

Xi Huang, Ying Tan, *Senior Member, IEEE*, and Xingui He

***Abstract*—As a new kind of vehicle with low fuel cost and low emissions, the hybrid electric vehicle (HEV) has been paid much attention in recent years. The key technique in the HEV is adopting the optimal control strategy for the best performance. As the premise, correct driving condition discrimination has an extremely important significance. This paper proposes an intelligent multifeature statistical approach to automatically discriminate the driving condition of the HEV. First, this approach periodically samples the driving cycle. Then, it extracts multiple statistical features and tests their significance by statistical analysis to select effective features. Afterward, it applies a support vector machine (SVM) and other machine-learning methods to intelligently and automatically discriminate the driving conditions. Compared with others, the proposed approach can compute fast and discriminate in real time during the whole HEV running mode. In our experiments, it reaches an accuracy value of 95%. As a result, our approach can completely mine the valid information from the data and extract multiple features that have clear meanings and significance. Finally, according to the prediction experiment by a neural network, the fitting experiment by the autoregressive moving average model, and the simulation results of the control strategy, it turns out that our proposed approach raises the efficiency of considerably controlling the HEV.**

***Index Terms*—Driving condition, hybrid electric vehicle (HEV), intelligent multifeature statistical discrimination (IMSD), neural network, statistical feature.**

I. Introduction

At PRESENT, faced with increasingly more resource and environmental problems, people have to pay more attention to the fuel economy (FE) and the emission of transportation such as vehicles. Developing a vehicle with lower fuel cost and lower emissions has become a goal of current vehicle industry [1].

Manuscript received June 7, 2009; revised November 16, 2009, May 18, 2010, and October 12, 2010; accepted November 9, 2010. This work was supported in part by the National High Technology Research and Development Program of China (863 Program) under Grant 2007AA01Z453, by the National Natural Science Foundation of China under Grant 60875080 and Grant 60673020, and by the Research Fund for the Doctoral Program of Higher Education in China. The Associate Editor for this paper was B. De Schutter.

X. Huang and Y. Tan are with the Key Laboratory of Machine Perception (Ministry of Education) and the Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (Corresponding author Y. Tan. e-mail: ytan@pku.edu.cn).

X. He is with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China.

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/TITS.2010.2093129

Three ways can be taken to improve engine efficiency and vehicle performance. The first way is to change the structure of the traditional engine or add some improving apparatus. This has the lowest cost because it does not change the process of vehicle production. Consequently, the performance improvement is limited. The second way is through an electric vehicle (EV) and a fuel cell vehicle [2]. These two types of vehicles, which substitute the traditional fuel with a battery and a fuel cell, and the internal combustion engine with an electric motor, are the cleanest vehicles. However, the high manufacturing cost limits their development. Combining these two ways, a hybrid EV (HEV) was developed after 1995 as the third way [3]. The HEV combines the advantages of the traditional vehicle and the EV to reach a good balance between the cost and the FE. As a way with promising performance, the HEV has increasingly become one of the main development trends in vehicle industry.

An HEV generally has two or more power sources, e.g., fuel and electric power. Its engine combines the traditional internal combustion engine, electric motor, and battery in different ways [4]. Due to the highly efficient energy storage parts, the internal combustion engine of the HEV could be smaller, more efficient, and with lower emissions. Based on the combing ways of parts, the HEV is classified into a series HEV, a parallel HEV, and a hybrid HEV [5]. Whichever type is adopted, when the HEV outputs the power, how to appropriately manage the different power sources to improve its performance becomes an extremely important problem. It is also the key problem in the HEV research—the control strategy of the HEV [6].

The control strategies of the HEV can be classified into three types. The first is the intelligent control strategy or the heuristic control strategy. This strategy usually has some predefined rules. The HEV will manage the power sources following the rules [7]–[9]. This strategy is simple, fast, and easy to implement, but its control result is often far from the optimal point. The second is the static optimization strategy. This strategy will compute the optimal power split based on the inherent parameter of the HEV [10], [11]. This strategy is much more efficient than the intelligent control strategy. However, due to lack of consideration about the driving environment, it usually performs well under some specific driving conditions, whereas it might not be optimal under others. The third is the dynamic optimization strategy. This strategy not only considers the effect of the inherent parameter but also detects the change in the external factors. Combining all the factors, the strategy will compute the best power split to dynamically manage the power sources in real time [12]–[14]. This strategy is able to continuously modify the control model and give a better performance. However, enormous computation will be a great burden and results in a lack of time.

Among the aforementioned control strategies, most of them construct a uniform control model to manage the power sources. However, the range of parameters in the control model is wide, and the same parameter in different intervals will express the different system characteristics. As a result, a uniform system model is not adequate to describe the driving process of the HEV. To construct a more precise model, we need to decompose the uniform model into a number of submodels corresponding to the different parameter intervals [15].

The driving condition of the HEV is a complicated variable that is determined by multiple factors. It will exert a significant effect on the vehicle performance. The correct discrimination of driving conditions will help decompose the uniform model into a series of submodels corresponding to the different driving conditions, which can improve the control performance. Meanwhile, there are lots of factors that influence driving conditions, e.g., wetness of the road, resistance, terrain, traffic, and even weather. The data on these factors may not be directly collected because there is no exact numerical style. Thus, how to correctly discriminate the driving conditions based on the limited data that we have collected becomes a key problem to solve.

II. Related Work

To discriminate the driving conditions, the classification standard of driving conditions should be defined first. Currently, there is no uniform standard. Based on the actual requirements, various classifications are adopted, e.g., based on the road surface condition, the driving conditions are classified into dry, wet, slushy, icy, and snowy [16]; based on the road level, they can be classified into a highway, an urban road, and an extra urban road [17]–[19]. The congestion level is also a factor to classify the driving conditions [20]. Moreover, there are some classifications by combining the aforementioned methods [21], [22].

The driving condition is determined by various factors; therefore, the data that are collected for driving condition discrimination are also different. The common data contain the following two types: The first type is an image or a video, which supplies many details such as obstacles, pedestrians, and other vehicles to distinguish the driving conditions [23]–[25]. However, such data require complicated image processing to give the result, and the enormous computation may be a great burden. The second type is time series on velocity or acceleration. Such data are most commonly used and easily collected. Many research studies are based on it [19], [26], [27]. However, these data cannot supply enough details; thus, it is not suitable for the discrimination of the complicated driving conditions. In recent years, the data collected by the automotive radar have been increasingly being used for the discrimination of driving conditions [28]–[30].

According to the data type and the classification standard, various approaches are proposed to discriminate the driving conditions, e.g., based on the pictures captured in the vehicle, image processing and computer vision approaches are used to determine the outside environment [16], [23]. Some rule-based and fuzzy logical methods are also used to make the results robust [20], [31]. In [17], the hidden Markov model is adopted to predict the future driving conditions. In [32], a neural network is also adopted.

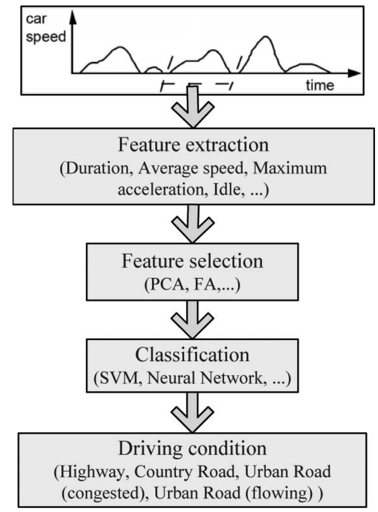

Fig. 1. Framework of the model.

When the driving conditions are discriminated, the control strategy based on the driving conditions can be built. In [18], six respective control rules are built on six driving conditions. In [32], the multimode control strategy is built based on driving patterns. In [33]–[35], various control strategies are proposed under different driving conditions.

This paper proposed a new approach of intelligent multi-feature statistical discrimination (IMSD). Based on the driving data of the HEV, it uses statistical analysis to extract and select multiple valid features. After the classifier learned the information on these features, it can intelligently discriminate the driving conditions in real time. This approach has simple data processing, definite meaning, fast computation, and high accuracy.

This paper will be organized as follows: In Section III, a framework of our model will be proposed. In Section IV, the classification standard of driving conditions will be discussed in detail. In Section V, we will analyze the extracted statistical features and their significance. In Section VI, the samples will be classified based on statistical features and different classifiers. In Section VII, the final discrimination approach will be determined. In Section VIII, some experiments will show the effectiveness of IMSD. Finally, Section IX concludes of this paper.

III. Framework of Our Model

The framework of IMSD is shown in Fig. 1, which consists of the following five steps.

A. Generation of Driving Samples

The driving cycle of the HEV is usually a long time series. In practical applications, we need to discriminate the current driving conditions in real time. Thus, we collect the driving data by sampling periodically. According to the short samples, the current driving condition can be quickly determined.

Suppose a long time series $\{S_1, S_2, \ldots, S_n\}$, we need to truncate k samples $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k\}$ of length t from it. First, we randomly select a start position n_0, i.e.,

$$n_0 = \text{random}(0, 1, \ldots, t-1). \tag{1}$$

Then, the continuous m $(m > k)$ samples of length t will be

$$\mathbf{s}_i = \{s_{ij}, j = 1, 2, \ldots, t\}, \qquad i = 1, 2, \ldots, m \tag{2}$$

where $s_{ij} = S_{n_0+(i-1)t+j}$, $n_0 + mt \leq n$.

The k samples will be randomly selected from them.

B. Feature Extraction

After collecting the samples, we need to extract the features that can describe the characteristics of driving conditions. Here, we choose the features with statistical significance and definite meaning as the elementary features, e.g., average speed and maximum acceleration.

Suppose that the multiple extracted features are E_1, $E_2, \ldots, E_m$ and that the vector that consists of them is

$$\boldsymbol{E} = (E_1, E_2, \ldots, E_m). \tag{3}$$

If the sample for feature extraction is

$$\boldsymbol{s} = (s_1, s_2, \ldots, s_t) \tag{4}$$

the process of feature extraction can be described as

$$\begin{aligned}\boldsymbol{E} &= \text{extract}(\boldsymbol{s}) \\ &= (\text{extract}_1(\boldsymbol{s}), \text{extract}_2(\boldsymbol{s}), \ldots, \text{extract}_m(\boldsymbol{s}))\end{aligned} \tag{5}$$

where extract$(\cdot)$ is a mapping from the sample to the elementary features; $\text{extract}_1(\boldsymbol{s}), \text{extract}_2(\boldsymbol{s}), \ldots, \text{extract}_m(\boldsymbol{s})$ are m components of $\boldsymbol{E}$. The specific forms of extract$(\cdot)$ can be various. If we suppose that $\boldsymbol{s}$ is a speed sample of the HEV and E_1 indicates the average speed, then

$$E_1 = \text{extract}_1(\boldsymbol{s}) = \frac{1}{t}\sum_{i=1}^{t} s_i. \tag{6}$$

C. Feature Selection

Elementary features are usually of a large quantity. Some of them may not be suitable to discriminate the driving conditions. Thus, we need to filter the elementary features and convert them into advanced features. The method for feature selection can be principal component analysis (PCA), factor analysis (FA), and so on. We will compare them in Section VI.

Suppose the advanced features after feature selection are $F_1, F_2, \ldots, F_k$ and the vector that consists of them is

$$\boldsymbol{F} = (F_1, F_2, \ldots, F_k). \tag{7}$$

The process of feature selection is described as

$$\boldsymbol{F} = \text{select}(\mathbf{E}) \tag{8}$$

where select$(\cdot)$ indicates the operation of feature selection. Based on the requirements, select$(\cdot)$ can be the different methods, e.g., PCA and FA. If we suppose that select$(\cdot)$ indicates the operation of the PCA, then

$$\boldsymbol{F} = (F_1, F_2, \ldots, F_k) = \text{PCA}(\boldsymbol{E}) \tag{9}$$

where $F_1, F_2, \ldots, F_k$ are the first k principal components that are selected.

Based on statistical analysis, we can also directly select k features from m elementary features without transformation, as we adopted in this paper. The process can be described as

$$\boldsymbol{F} = \text{select}(E_1, E_2, \ldots, E_m) = (E_{i_1}, E_{i_2}, \ldots, E_{i_k}) \tag{10}$$

where $\{E_{i_k}\}$ is a subsequence of $\{E_m\}$.

D. Classification

Based on the aforementioned advanced features, the driving conditions of the HEV can be determined by the different classifiers, e.g., k-nearest neighbor (kNN), neural network, and support vector machine (SVM).

Suppose the corresponding driving condition $R \in \{R_1, R_2, R_3, R_4\}$. R_1, R_2, R_3, and R_4 indicate the four driving conditions of the HEV, respectively. Then, our classification model will be

$$R = \text{classify}(\boldsymbol{F}) \tag{11}$$

where classify$(\cdot)$ is the classification function.

When there is a new sample to discriminate, based on its extracted feature $\boldsymbol{F}$, the result of classification R will be the current driving condition.

E. Four Driving Conditions

The aforementioned process is, in fact, a mapping from the driving cycles $\mathfrak{S}$ to the driving conditions $\mathfrak{R}$, i.e.,

$$f : \mathfrak{S} \rightarrow \mathfrak{R} \tag{12}$$

where $\mathfrak{S}$ is the set consisting of the driving cycles $\{S_n\}$, and $\mathfrak{R} = \{R_1, R_2, R_3, R_4\}$ is the set consisting of driving conditions.

In this paper, we classify the driving conditions of the HEV R into four types, including a highway $(R = R_1)$, a country road $(R = R_2)$, an urban road (congested) $(R = R_3)$, and an urban road (flowing) $(R = R_4)$. The detailed definition and explanation will be discussed in Section IV.

F. Characteristics of Our Model

From the aforementioned framework, four characteristics of the model can be seen.

1) *Dynamic determination of driving conditions:* Because the model periodically samples the whole driving process, the current driving condition can be decided in real time based on the sample.

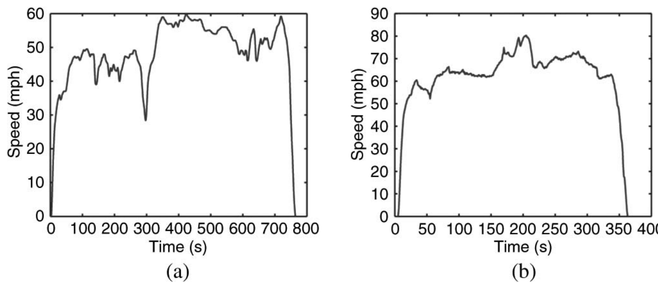

Fig. 2. Two driving cycles of a highway.

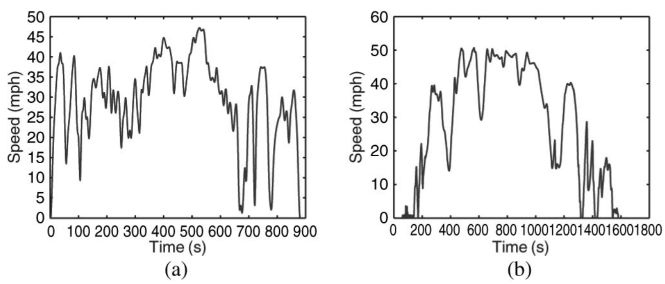

Fig. 3. Two driving cycles of a country road.

2) *Multiple features:* The model extracts more than one feature from a sample to provide accurately enough information to determine the driving condition.
3) *Statistical method for feature extraction and selection:* We use the statistical analysis and test to obtain multiple significant features, which have obvious meaning in statistics and are convenient for feature explanation.
4) *Intelligent discriminant:* The classifier in the model adopts various machine-learning algorithms, which make the model automatically learn the features and intelligently discriminate the current driving condition.

IV. Driving Conditions of the Hybrid Electric Vehicle and Their Feature Analysis

We classify the driving conditions into four types, which is the most common and representative.

A. Highway

A highway is a main road between important destinations, such as cities and towns. It has a lower limit of the driving speed and can afford the heavy traffic. In Fig. 2, the two speed sequences are collected under the real highway condition. In the figure, we can see the HEV keeps a high speed (above 50 mi/h) and drives smoothly. There is no interval of stopping (0 speed) in the cycle.

B. Country Road

A country road is a road that connects cities and countries. Compared with the highway, its speed standard and traffic capacity are lower. In Fig. 3, the sequences are collected under the country road condition. Comparatively, the speed decreases a lot (30–50 mi/h). The HEV periodically accelerates and decelerates; thus, the driving cycle becomes a form of wave.

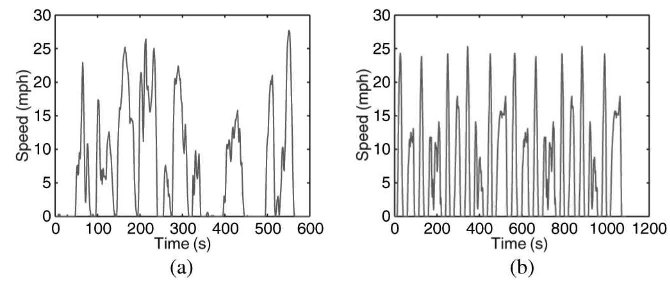

Fig. 4. Two driving cycles of an urban road (congested).

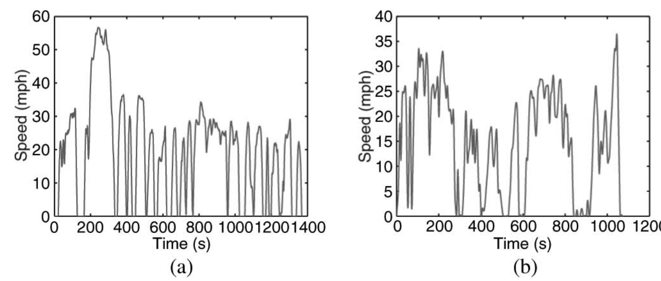

Fig. 5. Two driving cycles of an urban road (flowing).

C. Urban Road (Congested)

An urban road is the road in the city. There are numerous intersections and vehicles on the urban roads. When the traffic is heavy, the urban road usually congests. In Fig. 4, the sequences show the state of the congested urban road. Compared with the preceding two driving conditions, the HEV under this condition not only has a low speed (below 25 mi/h) but also periodically stops. Moreover, the stopping interval under this condition is usually large.

D. Urban Road (Flowing)

Compared with the congested urban road, the speed of the flowing urban road increases a little. However, due to the numerous intersections, the HEV still periodically stops. Under this condition, the speed of the HEV can keep for a while and will not immediately decrease. From the sequences in Fig. 5, the aforementioned characteristics are obvious.

E. Statistical Features of Different Driving Conditions

To find the features for discrimination, we extract the statistical features of driving cycles under the aforementioned driving conditions.

Some standard driving cycles that are collected in real world are adopted, including HWFET, US06-HWY, INDIA-HWY-SAMPLE, HYZEM-URBAN, HYZEM-HWY, HYZEM-SUB, CSC, WVUSUB, NYB, MANHATTAN, NYCC, CBD14, INDIA-URBAN-SAMPLE, UDDS, and WVUCITY.[1] Each driving cycle is a long speed sequence that is collected under one driving condition. The standard driving cycles belonging to the same driving condition can be combined to form a new long driving cycle under this driving condition. We create a 10 000-s

[1]More details about these cycles can be obtained from http://www.dieselnet.com/standards/cycles/.

TABLE I
STATISTICAL FEATURES OF DIFFERENT DRIVING CONDITIONS

Class	$v_{\max}$	v_{mean}	$a_{\max}$	$a_{\min}$	$(v*a)_{\max}$	$(v*a)_{\min}$	$(v*a)_{\text{mean}}$	I	σ_v	σ_a	σ_{v*a}
1	80.3	**39.77**	**6.9**	**-6.9**	209.72	-216.32	0.37	**0.019**	18.78	0.86	26.47
2	67.2	**19.61**	**6.9**	**-8.8**	142.88	-280.72	0.99	**0.12**	16.52	1.41	29.24
3	31.82	**7.01**	**6.2**	**-7.24**	99.36	-91.08	0.77	**0.38**	8.71	1.24	15.12
4	42.1	**15.17**	**3.87**	**-4.71**	100.5	-98.28	0.71	**0.14**	11.11	1.19	19.46

driving cycle for each driving condition to statistically analyze their features.

The result is shown in Table I, where various statistical features are presented, including the maximum speed $v_{\max}$; the average speed v_{mean}; the maximum acceleration $a_{\max}$; the minimum (negative maximum) acceleration $a_{\min}$; the maximum, minimum, and average values of speed multiplied by acceleration, i.e., $(v*a)_{\max}$, $(v*a)_{\min}$, and $(v*a)_{\text{mean}}$; their standard deviations σ_v, σ_a, and σ_{v*a}; and the idle rate (the percent of the stopping interval) I. In addition, the four driving conditions are indicated by Class 1 (highway), Class 2 (country road), Class 3 (urban road (congested)), and Class 4 (urban road (flowing)).

On the features v_{mean}, $a_{\max}$, $a_{\min}$, and I, the differences between classes are significant. Based on the features, the four driving conditions of cycles can be distinguished. We mark them in bold.

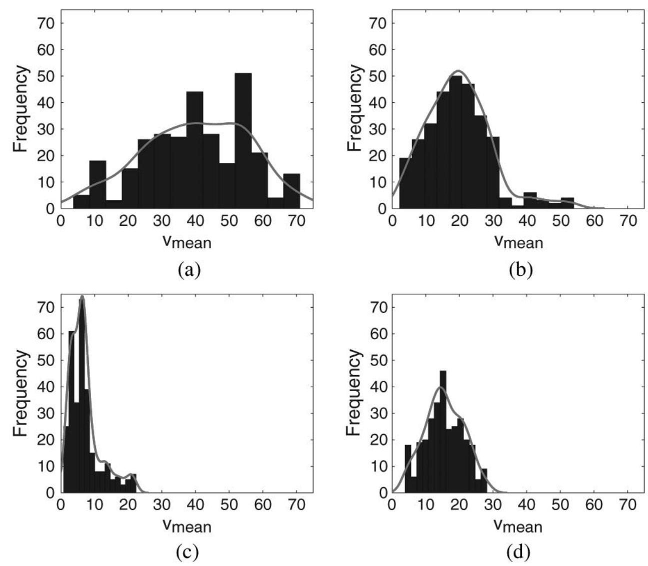

Fig. 6. Histograms on v_{mean} under the four driving conditions. (a) Highway. (b) Country road. (c) Urban road (congested). (d) Urban road (flowing).

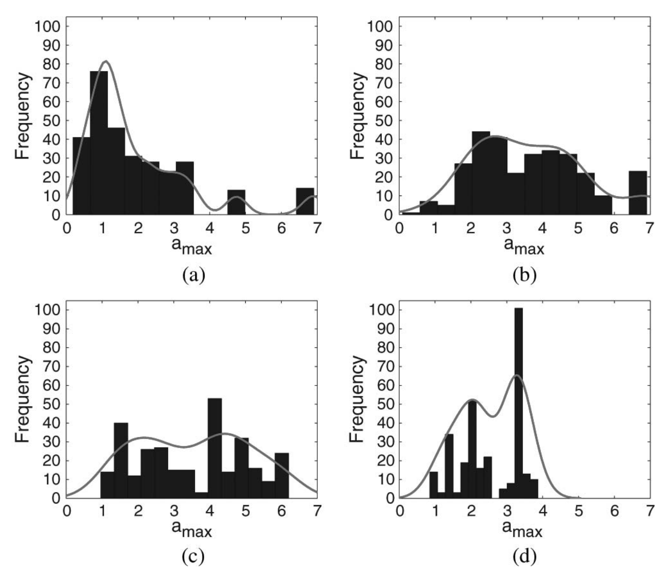

Fig. 7. Histograms on $a_{\max}$ under the four driving conditions. (a) Highway. (b) Country road. (c) Urban road (congested). (d) Urban road (flowing).

V. FEATURES OF SAMPLES AND THEIR SIGNIFICANCE

From the analysis of the driving cycles, we can obtain some features. However, in our model, the collected data are segments of the driving cycle. Thus, the features v_{mean}, $a_{\max}$, $a_{\min}$, and I may be not suitable. We need to generate some driving samples to analyze their validation and significance.

A. Generation of Samples

We use the method in the framework to generate the samples. Given a long driving cycle $\{S_1, S_2, \ldots, S_n\}$, we need to generate some samples $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k\}$ of length t from the driving cycle. The algorithm is given here.

Step 1: Select a start position $n_0 = \text{random}(0, 1, \ldots, t-1)$, $i = 1$.
Step 2: If $n_0 + t > n$, turn to Step 5.
Step 3: Let $s_{ij} = S_{n_0+j}, \qquad j = 1, 2, \ldots, t$.
Step 4: $n_0 = n_0 + t$, $i = i + 1$, turn to Step 2.
Step 5: Select k samples randomly from i generated samples.

In the former research, we have learned that the driving period of the HEV is about 3 min [36]. When the length of the sample approaches or exceeds 3 min, it can reflect the characteristic of the current driving condition. Thus, we choose 150 s as the length of each sample and generate 300 samples under each driving condition. Then, we analyze the features of these samples.

B. Histogram of Samples

A histogram is a basic method to analyze the distribution of samples. It shows the frequency of samples at each interval on a feature.

The histograms in Fig. 6 show the distributions of the samples under different driving conditions on the feature v_{mean}. In the figure, the red line is the fitting curve of the distribution density, which is obtained by the kernel density method and represents the most probable distribution form of the samples. We can see, under the four driving conditions, that the shapes and positions of the fitting curves are completely different, which means that, on the feature v_{mean}, there are significant differences in the distributions of the samples between driving

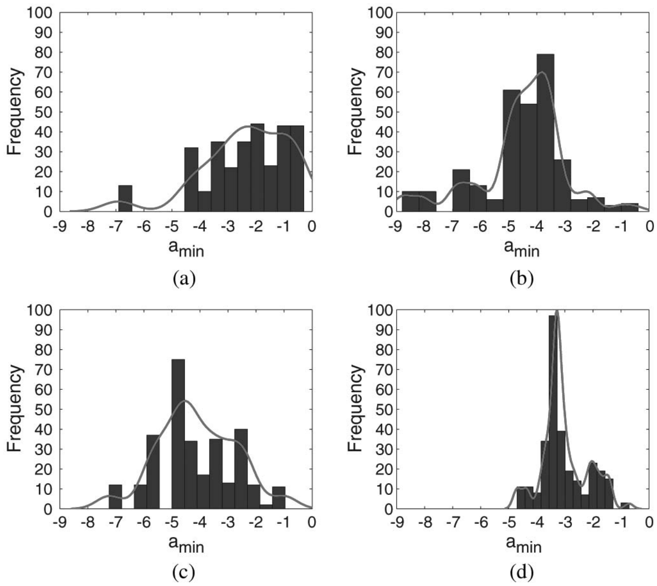

Fig. 8. Histograms on $a_{\min}$ under the four driving conditions. (a) Highway. (b) Country road. (c) Urban road (congested). (d) Urban road (flowing).

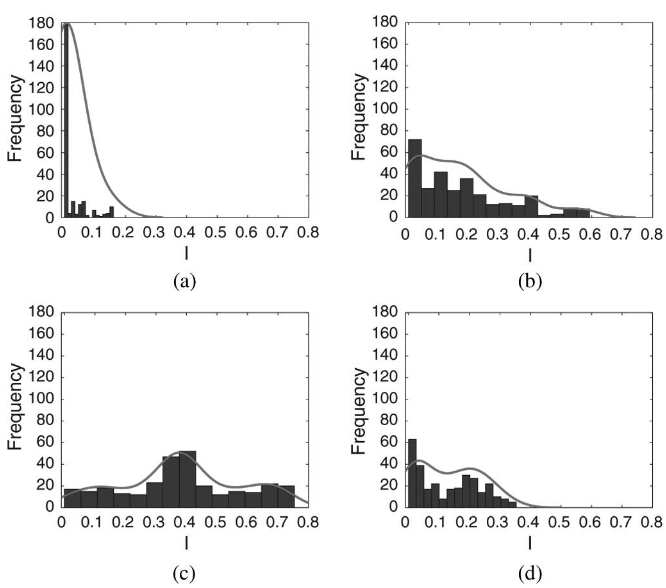

Fig. 9. Histograms on I under the four driving conditions. (a) Highway. (b) Country road. (c) Urban road (congested). (d) Urban road (flowing).

conditions. Thus, the feature v_{mean} can be used to discriminate the different driving conditions.

The histograms on the features $a_{\max}$, $a_{\min}$, and I are shown in Figs. 7–9, which express a similar characteristic as those on v_{mean}. Thus, these features can be also adopted for discrimination.

C. Boxplot and ANOVA

From the aforementioned histograms, we can see the basic distribution of the samples. To test the significance of these features, we adopt boxplot and analysis of variance (ANOVA) for a further analysis.

In a boxplot, the middle line of a box indicates the median of the samples, and the upper and lower edges of a box indicate

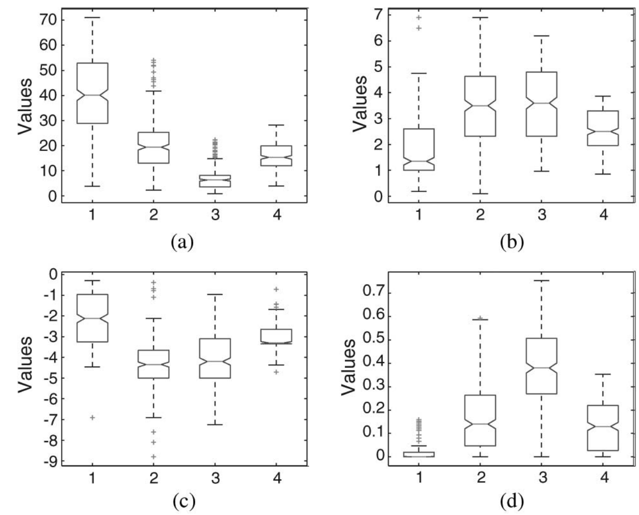

Fig. 10. Boxplots on four features, where the numbers on the horizontal axis indicate the four different driving conditions. (a) v_{mean}. (b) $a_{\max}$. (c) $a_{\min}$. (d) I.

the upper and lower quartiles, respectively. The top and the bottom of a whisker indicate the largest and smallest samples, respectively. The spacings between the different parts of the box help indicate the degree of dispersion and skewness in the samples.

The boxplots in Fig. 10 display the distributions of the samples under four driving conditions on the features v_{mean}, $a_{\max}$, $a_{\min}$, and I. It is obvious that, on each feature, the samples of different driving conditions have significant differences. Most of them are distributed in different intervals. Thus, the significance of these four features is obvious.

In addition, we use ANOVA to quantitatively analyze the significance of the features. The Kruskal–Wallis one-way ANOVA is adopted. The Kruskal–Wallis test is a nonparametric method to test for differences among two or more groups [37]. It does not assume a normal population, unlike the analogous one-way ANOVA.

First, it ranks all data from 1 to N, ignoring group membership. Then, the test statistic is given by

$$K = (N-1)\frac{\sum_{i=1}^{g} n_i(\overline{r}_i - \overline{r})^2}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}(\overline{r}_{ij} - \overline{r})^2} \tag{13}$$

where

$$\overline{r}_i = \frac{\sum_{j=1}^{n_i} r_{ij}}{n_i} \tag{14}$$

$\overline{r} = (1/2)(N+1)$ is the average of all the r_{ij}, n_i is the number of observations in group i, r_{ij} is the rank of observation j from group i, and N is the total number of observations.

The p-value is approximated by $\Pr(\chi^2_{g-1} \geq K)$. When K is large or the p-value is small enough ($p < 0.01$), the differences between groups are significant. In our approach, the groups are the driving conditions, and the Kruskal–Wallis test on each feature is given in Table II. When the K-value is large or $p < 0.01$, we can confirm that the feature is significant, and we can differentiate the driving conditions.

TABLE II
KRUSKAL–WALLIS ANOVA ON DIFFERENT FEATURES

Feature	$v_{\max}$	v_{mean}	$a_{\max}$	$a_{\min}$	$(v*a)_{\max}$	$(v*a)_{\min}$	$(v*a)_{\text{mean}}$	I	σ_v	σ_a	σ_{v*a}
K	601.9	**621.7**	**318.26**	**397.25**	72.4	220.89	15.21	**506.42**	251.6	237.12	246.4
p	0	**0**	**0**	**0**	0	0	0.0016	**0**	0	0	0

TABLE III
CORRELATION OF DIFFERENT FEATURES

Correlation	v_{mean}	$a_{\max}$	$a_{\min}$	I
v_{mean}	1	-0.30	0.29	-0.62
$a_{\max}$	-0.30	1	-0.46	0.32
$a_{\min}$	0.29	-0.46	1	-0.17
I	-0.62	0.32	-0.17	1

From the table, all the features are significant $(p < 0.01)$; thus, the features are compared according to the K-value. v_{mean}, $v_{\max}$, I, $a_{\max}$, and $a_{\max}$ give five largest K-values, which should be adopted in our model. However, the correlation coefficient between $v_{\max}$ and v_{mean} is 0.91, which means a high correlation. Thus, we drop $v_{\max}$ to reduce the correlation between features. The K-values of $(v*a)_{\min}$, σ_v, σ_a, and σ_{v*a} are close to each other. To reduce the feature number, both of them are dropped. The experiment in Section VI-B will show that it is unnecessary to keep them in our model. The remainder of the features give small K-values; thus, all of them can be dropped. Finally, we adopt four significant features, i.e., $v_{\max}$, $a_{\max}$, $a_{\min}$, and I, as the best features for the discrimination of driving conditions. In Table III, we present their correlation coefficients. Both of them keep a low correlation with other features.

VI. EXPERIMENT OF CLASSIFICATION

After the features with an important significance are extracted, we use them to train the classifier and discriminate the driving conditions. The experimental setup is discussed here.

v_{mean}, $a_{\max}$, $a_{\min}$, and I are chosen as the input variables. The driving condition R is the output variable. One thousand two hundred new samples are generated from the driving cycles, among which there are 300 samples for each driving condition. The length of the sample is a parameter of classification. Fivefold cross validation is adopted to estimate the classification. The original sample is randomly partitioned into five subsamples. A single subsample is used for testing, and the remaining four subsamples are used as training data. The cross-validation process is then repeated five times, with each subsample used for testing once. Their average result is the final estimation.

A. Comparison of Feature Extraction Methods

In the framework of our model, the features are extracted and selected by a statistical method. To illustrate the effectiveness of our method, the common methods for feature extraction are adopted for a comparison, which are the fast Fourier transform (FFT), the discrete cosine transform (DCT), and the PCA.

1) OE: The experiment of classification with input features v_{mean}, $a_{\max}$, $a_{\min}$, and I is considered as the original experiment (OE). The multilayer neural network (MLNN) with one hidden layer is adopted as the standard classifier. Based on the conclusion of the prior experiment, we set the number of hidden nodes to 15, which will make the neural network reach a good balance between performance and complexity. In addition, according to Bishop's work [38], the number of training patterns should be around ten times as many as the weights in the network. There are about 75 weights in the network and 960 training samples, which is appropriate. The training algorithm is a Levenberg–Marquardt algorithm, and the maximum epoch is 500. The convergence goal is a mean-square error (MSE) of 0.01.

2) FFT and DCT: The FFT and DCT are two methods that are comprehensively used in digital signal processing, both of which can be used to compress the data and extract the frequency features of the data.

The FFT is the fast algorithm for the discrete Fourier transform [39]. Suppose that the input signal is $x(i)$, $i = 1, \ldots, N$ and that the FFT of the input signal is given by

$$y(k) = \sum_{n=1}^{N} x(n)\omega_N^{(n-1)(k-1)} \tag{15}$$

where $k = 1, 2, \ldots, N$, and ω_N is the Nth root of unity, which is defined as

$$\omega_N = e^{(-2\pi i)/N}. \tag{16}$$

By the FFT, the real input signal is converted into a complex frequency domain.

The DCT is defined as

$$y(k) = w(k)\sum_{n=1}^{N} x(n)\cos\frac{\pi(2n-1)(k-1)}{2N} \tag{17}$$

where $k = 1, 2, \ldots, N$, and $w(k)$ is defined as

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 1 \\ \sqrt{\frac{2}{N}}, & 2 \le k \le N. \end{cases} \tag{18}$$

From (17) and (15), the DCT uses fewer bases to transform the data. Thus, the information from the data will concentrate on a space with lower dimensionality [40].

In our experiment, the FFT and DCT transform the driving sample into a frequency domain and directly extract its frequency features. To compare them with our method, the first four dimensions of their output are chosen as the features of classification. The classifier is the same as the OE.

3) PCA: The PCA is a feature transform often used to reduce multidimensional data sets to lower dimensions for analysis [41]. It orthogonalizes the components of the input features; thus, output components are uncorrelated with each other. It orders the resulting orthogonal components (principal components) so that those with the largest variation come first.

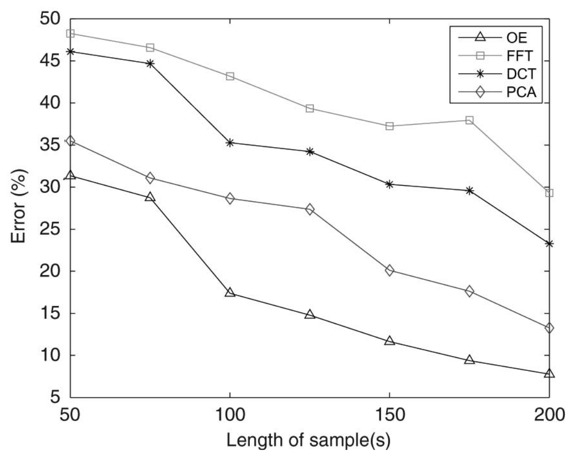

Fig. 11. Comparison of test errors by different feature extraction methods.

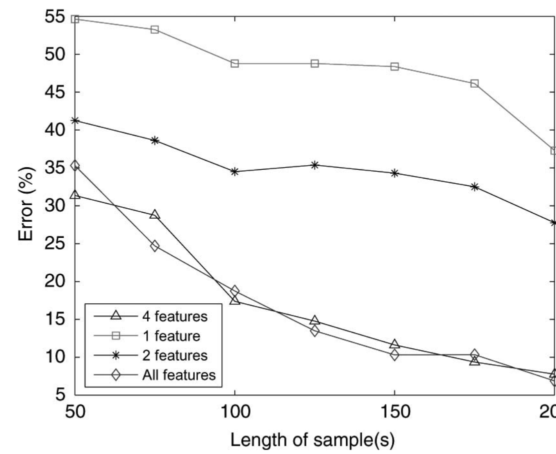

Fig. 12. Comparison of test errors under different feature numbers.

For a data matrix $\mathbf{X}^T$, where each row represents a different repetition of the experiment and each column gives the results of a particular feature, the PCA transformation is given by

$$\mathbf{Y}^T = \mathbf{X}^T\mathbf{W} = \mathbf{V}\boldsymbol{\Sigma} \tag{19}$$

where $\mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T$ is the singular value decomposition of $\mathbf{X}^T$.

In the experiment, the PCA is adopted for feature selection. It transforms 11 elementary features extracted from the samples into 11 advanced features. To keep the feature number the same as the OE, the first four principal components are chosen as the features of classification. The classifier is the same as the OE.

4) Comparison: The test errors of classification are shown in Fig. 11. We can see that the FFT and DCT obtain similar classification accuracy and that the DCT is slightly better. The test errors of the PCA and OE are much lower than those of the FFT and DCT, which illustrates that the features in the frequency domain are not enough for the classification of driving conditions and that a statistical method is more suitable for feature extraction. The test error of the OE is lower than the PCA. Compared with the PCA, the OE is simpler, and the features obtained by the OE all have clear meanings. Thus, our approach is best for feature extraction and selection.

B. Comparison of Feature Numbers

In our approach, four features, i.e., $v_{\max}$, $a_{\max}$, $a_{\min}$, and I, are adopted for classification. We change the number of features to see if the result of classification is affected.

1) One Feature: Most approaches for the discrimination of driving conditions of the HEV are only based on one feature, i.e., v_{mean} [17]. We repeat the OE with only one input, i.e., v_{mean}. The result is shown in Fig. 12. Compared with our approach (four features), the test error dramatically increases. It illustrates that one feature cannot provide enough information to discriminate the driving conditions. This is also why we choose multiple features in our approach.

2) Two Features: In Section V-C, the significance of $a_{\max}$ and $a_{\min}$ is not as good as that of v_{mean} and I. We use the MLNN to test if the features $a_{\max}$ and $a_{\min}$ are necessary for our model. We repeat the OE with v_{mean} and I as the inputs. The result in Fig. 12 shows that the test error obviously increases compared with the OE (four features). Thus, we can confirm that $a_{\max}$ and $a_{\min}$ should not be dropped.

3) All Features: To test if four features are enough for our classification, we repeat the OE with all 11 features. The test error in Fig. 12 shows that its accuracy is similar to that of four features. The classification result does not obviously improve. Thus, there is no need to adopt more than four features.

4) Accuracy and Feature Number: From Fig. 12, the test accuracy is significantly improved with the feature number growing, which illustrates that the multiple-feature approach will mine more information from the data. However, when the feature number exceeds four, the accuracy improves not so obviously. There is no need to adopt more than four features.

Then, we choose the features v_{mean}, $a_{\max}$, $a_{\min}$, and I as the input features to test the performance of different classifiers.

C. Comparison of Classifiers

We choose the MLNN, linear classifier (LC), quadratic classifier (QC), kNN, and SVM as the classifiers to compare their capacities for the discrimination of driving conditions.

1) MLNN: In the OE, the MLNN with one hidden layer is adopted. We use its result as the result of the MLNN (see Fig. 13). We can see the test error decreases with the length of the sample growing. When the length exceeds 150 s, the test error becomes stable and remains below 12%. In fact, when the length of the sample is small, the information supplied by the sample is little, which cannot correctly reflect the driving condition. When the samples reach a certain length, all of them can reflect the current driving condition. Thus, the accuracy will increase and become stable.

2) LC and QC: The LC and QC are characterized by a simple structure and fast computation. It makes a classification decision based on the value of the linear or quadratic combination of the features. The operation of the LC or QC can be visualized as splitting a high-dimensional sample space with some hyperplanes. The samples belonging to the different driving conditions can be separated by those hyperplanes in the sample space. An LC or a QC is often used in situations where the speed of classification is an issue, particularly when the sample set is sparse. However, the LC generally cannot give

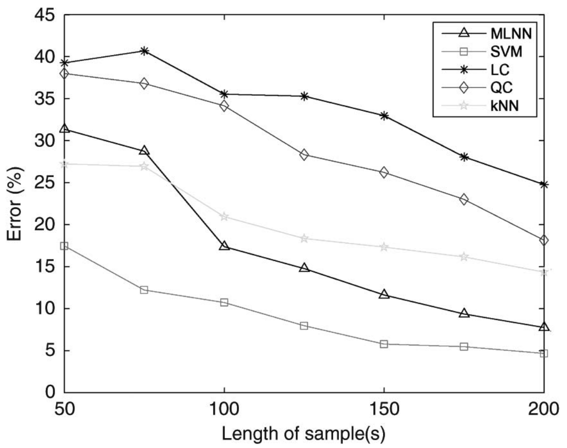

Fig. 13. Comparison of the test errors of classifiers.

TABLE IV
ACCURACY OF THE kNN BASED ON DIFFERENT k-VALUES

K-value	1	2	3	4	10	20
Accuracy	82.5%	82.5%	81.6%	80.8%	75.7%	72.9%

a good result for the linear nonseparable samples. The capacity of the QC is better than that of the LC. However, it still cannot obtain high accuracy for the complicated linear nonseparable samples.

From Fig. 13, the result of the LC and QC is obviously not as good as that of the MLNN. The QC is slightly better than the LC. It means that the samples are linear nonseparable.

3) kNN: The kNN is a basic method for classifying objects based on closest training samples. An object is classified by a majority vote of its k neighbors. The object will be assigned to the most common class among its k-nearest neighbors. If $k = 1$, then the object is simply assigned to the class of its nearest neighbor [42]. The kNN is a classifier with a simple structure and easy realization. It will give good classification accuracy when the sample is numerous and the samples of different classes are balanced. The accuracy of the kNN based on different k-values is shown in Table IV. When the k-value increases, the accuracy decreases. Thus, we set the parameter $k = 1$ in our experiment.

The classification error of the kNN is shown in Fig. 13. From the figure, the accuracy of the kNN is better than that of the LC and QC. When the length of the sample is below 100 s, the kNN is better than the MLNN. However, when the sample exceeds 100 s, the test error of the MLNN obviously decreases and remains lower than the kNN. This means that the kNN cannot mine more valid information as the sample length increases.

4) SVM: The SVM is a supervised learning method used for classification. Given the samples of two classes, the SVM will construct a separating hyperplane in the sample space, which not only classifies the samples correctly but also maximizes the margin between the two classes [43]. This is a quadratic programming optimization problem as

$$\min_{\mathbf{w},\boldsymbol{\xi}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right\} \tag{20}$$

TABLE V
CONFUSION MATRIX OF THE SVM WITH A SAMPLE LENGTH OF 150 S

Predicted	Actual				Total	Accuracy
	1	2	3	4		
1	297	23	5	0	325	95.2%
2	3	264	3	7	277	
3	0	2	288	0	290	
4	0	11	4	293	308	
Sample	300	300	300	300	1200	

subject to

$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i, \qquad \xi_i \geq 0; \quad i = 1, \ldots, n \tag{21}$$

where $\mathbf{w}$ and b are the parameters of the separating hyperplane, c_i is the class of the sample $\mathbf{x}_i$, $\boldsymbol{\xi}$ is the slack variable, and C is the penalty parameter.

The previously constructed SVM is considered a linear classifier. To classify the linear nonseparable samples, a kernel method is introduced. By mapping the samples into a feature space, the SVM can correctly separate them [44], [45]. In the experiment, we choose a radial basis function (RBF) kernel as its kernel function, which is defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\delta^2}} \tag{22}$$

where δ is the variance of the RBF, and $\mathbf{x}_i$ and $\mathbf{x}_j$ are two samples. For the SVM, two parameters δ and C need to be set. We search the optimal δ and C to reach the highest accuracy.

In Fig. 13, the SVM gives the best classification accuracy. When the length of the sample exceeds 150, the test error is below 5%, which is the best result. In addition, we give the confusion matrix of the SVM with a sample length of 150 s in Table V. From the table, the conditions of the highway and urban road (congested) can be correctly discriminated by the classifier relatively easily. The country road and urban road (flowing) are usually incorrectly discriminated. Both of them are most likely to be incorrectly classified into a highway.

5) Comparison: The comparison of the test errors of different classifiers is shown in Fig. 13, which shows that the SVM gives the smallest error. The result of the MLNN is slightly worse than that of the SVM. When the sample length exceeds 100 s, the accuracy of the MLNN is better than that of the kNN. The LC and GC are both worse than the kNN. The LC gives the biggest test error.

Moreover, we compare the classifiers based on the receiver operating characteristic (ROC) curve in Fig. 14. The ROC curve is a plot of the true positive rate versus the false positive rate for a classifier as its discrimination threshold is varied, which can be used to possibly select an optimal classifier by the area under the curve (AUC). An optimal classifier will give the biggest AUC. The curves of different colors in the figure are based on the different driving conditions. From the figure, the SVM gives the biggest AUC. The other classifiers in descending order of the AUC are the MLNN, the kNN, and the QC. The LC can be regarded as a special case of the QC and is worse than the QC. Therefore, we did not show its curves in the figure.

In Fig. 15, the time costs of five classifiers are presented. The QC and LC give a similar result, which is the lowest time cost.

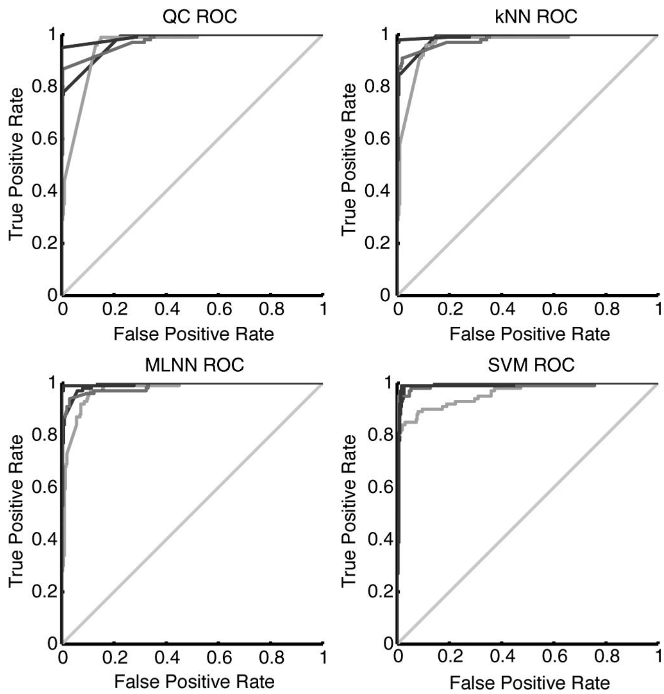

Fig. 14. ROC curves of classifiers.

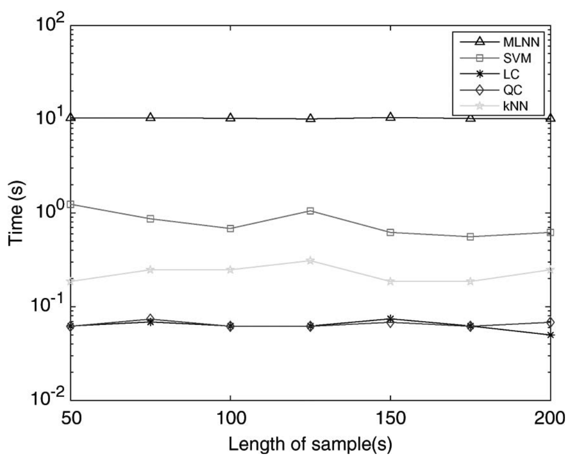

Fig. 15. Comparison of the time costs of classifiers.

The MLNN gives the highest time cost. The time cost of the SVM is slightly higher than that of the kNN. Their time costs are both between the LC and the MLNN.

From the aforementioned results, the SVM is undoubtedly the ideal classifier for the discrimination of driving conditions of the HEV.

VII. Intelligent Multifeature Statistical Discrimination Approach for the Discrimination of Driving Conditions

According to the preceding statistical analysis and classification experiments, we adopt the IMSD approach for driving condition discrimination.

- The length of the sample should be 150 s. When the HEV is running, it samples the driving cycle and determines the current driving condition every 150 s. As aforementioned, when the length of the sample exceeds 150 s, the classification will be robust and of high accuracy. On the other hand, to detect the change in the driving condition as soon as possible, the length of the sample should be reduced. Considering the foregoing reasons, 150 s is the best choice.
- Four features (i.e., v_{mean}, $a_{\max}$, $a_{\min}$, and I) are extracted from the sample to discriminate the current driving condition. These four features can be directly computed by a statistical method and have a clear statistical meaning. The features are suitable for discrimination.
- The SVM will be adopted as an ideal classifier. The SVM can give the smallest classification error. The time cost for classification is also the lowest, except for the QC and LC. By the SVM, the driving condition R will be correctly and efficiently discriminated.
- The classifier will be trained first based on these four features. Then, during the HEV running mode, the classifier will automatically discriminate the driving condition in real time.

VIII. Effectiveness of Intelligent Multifeature Statistical Discrimination

After IMSD was defined, it could be used in HEV control. We use the following experiments to illustrate the effectiveness of IMSD in HEV control.

A. Neural Network Model for Prediction

In the HEV control model, driving load prediction or velocity prediction is usually adopted to improve the vehicle performance [36]. There is a high correlation between the accuracy of prediction and the FE of the HEV. Thus, we adopted IMSD to improve the accuracy of prediction as follows.

The samples are generated from driving cycles. The length of each sample is 200 s. There are 100 samples for each driving condition in both the training and test sets. We hope to predict the average speed of the following 50 s based on the speed of the previous 150 s. Two methods are used for comparison.

1) *M1:* A uniform neural network model with one hidden layer for prediction is built. The input layer has 150 nodes, which are the speeds of the previous 150 s. The output layer has one node, which is the average speed of the following 50 s. The hidden layer has 40 nodes. The training algorithm is a Levenberg–Marquardt algorithm, and the maximum epoch is 500.
2) *M2:* Before prediction, we classify the samples into four classes by IMSD. For each class, we adopt one neural network for modeling and prediction. The structure of each neural network model is the same as the model in M1.

The result of the prediction experiment is given in Fig. 16. From the figure, M1 obtains the largest MSEs for prediction, which are 183.43 and 189.60 in the training and test sets, respectively. M2 obtains results of 60.17 and 81.20, respectively. It is obvious that the prediction accuracy is improved by adopting IMSD. From Fig. 16, in each class of M2, the obtained MSE is smaller than that of M1. As a result, M2 gives a smaller MSE than M1.

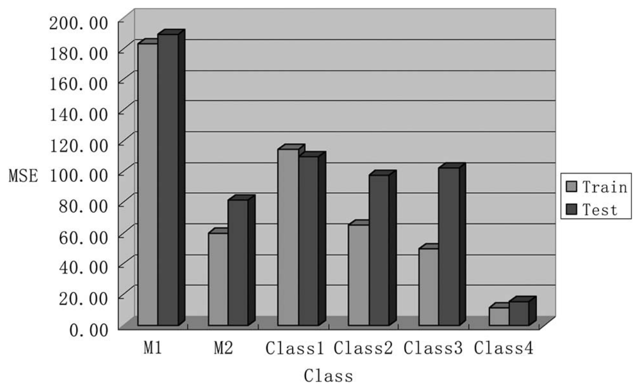

Fig. 16. Result of the prediction experiment. M1 indicates the MSE obtained by M1. M2 indicates the MSE obtained by M2. Class 1 to Class 4 indicate the MSE obtained from the samples under the four driving conditions of M2.

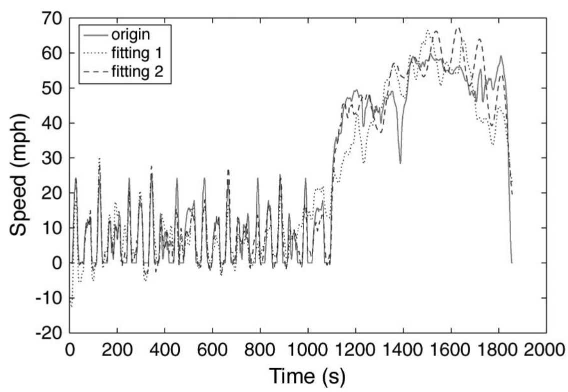

Fig. 17. Fitting result of the ARMA model. (Red line) Original data. (Black dotted line) Fitting curve obtained by M1. (Blue dashed line) Fitting curve obtained by M2.

B. ARMA Model for Fitting

In the simulation process of the HEV, we usually build a simple model to simulate the complicated process. Here, we use the autoregressive moving average (ARMA) model and IMSD to fit the driving cycle of the HEV.

The ARMA model is usually used to fit the time series. It uses the past p-values to approximate the current value of the time series, which is described as follows:

$$Y_t = c + \sum_{i=1}^{p} b_i Y_{t-i} + \sum_{j=1}^{q} a_j \varepsilon_{t-j} \tag{23}$$

where $\{Y_t\}$ is the time series for fitting, $\{\varepsilon_t\}$ is a random noise, and a_i $(i = 1, 2, \ldots, p)$, b_j $(j = 1, 2, \ldots, q)$, and c are the parameters. A big p or q usually means a complicated model.

In Fig. 17, the red solid line indicates a driving cycle combining the driving cycles under urban road and highway conditions. Two methods are adopted to fit it.

1) *M1:* A uniform ARMA(p, q) model is built. The values of p and q are both 200; thus, the model will be ARMA(200,200). The fitting result is shown by the black dotted line in Fig. 17.
2) *M2:* IMSD is used to determine the driving condition of each driving part first. Then, we build the different fitting models for the driving parts under different driving conditions. The urban part (previous 1100 s) and the highway part (following 750 s) are fitted by ARMA(50,50) and ARMA(100,100), respectively, which is shown by a blue dashed line in Fig. 17.

Obviously, from Fig. 17, M2 is closer to the original data than M1. The model of M2 is simpler than that of M1 because it uses less parameters. The accuracy of M2 is also better than that of M1. The MSE of fitting obtained by M1 is 54.96. M2 obtains an MSE of 27.08. The result of M2 is only half of that of M1. It is obvious that the fitting accuracy increases by using IMSD.

C. Control Strategy Based on IMSD

Based on IMSD, we can build different control rules according to the different driving conditions of the HEV. The control strategy in our experiment is based on two basic control strategies for a parallel HEV, namely, the motor assistant control strategy (MACS) [46] and the real-time optimization control strategy (RTOCS) [47].

In the MACS, the engine outputs the torque request of the vehicle when the torque request is below the maximum engine torque. The motor assists the torque if the required torque exceeds the maximum engine torque. The engine in this control strategy usually runs in low efficiency; thus, the control performance is not optimal. In the RTOCS, the engine will output the optimal engine torque based on the current engine speed and the engine efficiency map. When the engine torque exceeds the torque request of the vehicle, the excess torque will be recycled to charge the battery. The fuel efficiency of the engine in this control strategy is much better than that in the MACS. However, when the HEV is in an urban condition, the vehicle will keep a low driving load and a frequent change in the torque request. The optimal engine torque is usually much larger than the torque request. Thus, the excess energy should be recycled into the battery, and there will be an energy loss in the process of energy conversion. In this case, the performance of the RTOCS is worse than that of the MACS.

We build a mixed control strategy (MCS) based on them. The control strategy will first discriminate the current driving condition by IMSD. For the driving condition of an urban road (congested or flowing), the HEV will be controlled by the MACS. When the driving condition is a highway or a country road, the HEV will be controlled by the RTOCS.

The aforementioned control strategies were simulated on the software of ADVISOR [48]. In the simulation, the default parallel HEV model was adopted, and the vehicle ran over a driving cycle that is composed of standard driving cycles, including US06, NYCC, and NEDC.

The control results are presented in Table VI. In the table, the FE and the emissions are shown. The emissions contain three regular emissions, which are hydrocarbons (HC), carbon monoxide (CO), and nitrous oxides (NOx). We can see that the FE of the MCS is higher than that of both the RTOCS and the MACS, and the emission of the MCS is close to their lowest value. This means that the MCS gives a better control performance.

TABLE VI
COMPARISON OF THE FE AND THE EMISSIONS BASED ON DIFFERENT CONTROL STRATEGIES

Control Strategy	FE mpgge	HC g/ml	CO	NOx
MACS	34.81	0.73	6.57	0.50
RTOCS	32.39	0.80	4.67	0.78
MCS	**36.3**	**0.74**	**4.84**	**0.59**

According to the aforementioned experiments in HEV control, it can be seen that by using IMSD, the complicated model can be decomposed into multiple simple submodels. These submodels are specific and precise, which are helpful to reach a higher control performance.

IX. CONCLUSION

This paper has proposed a new IMSD approach. Combining statistical analysis and machine learning, this approach can automatically analyze the HEV driving data, extract multiple features, and dynamically discriminate the driving conditions, which is helpful for the best control strategy of the HEV.

During the HEV running mode, this approach periodically samples the driving cycle. Based on the samples, multiple features for the discrimination of driving conditions are extracted. Their significance is proved by histogram, boxplot, and ANOVA. Based on the extracted features, IMSD learns the information on the labeled samples by machine learning. Then, it can automatically discriminate the current driving condition of the HEV.

Compared with the current methods, the IMSD approach extracts more features to obtain more information about the driving conditions. It can accurately and dynamically discriminate the driving condition with a fast speed.

From the prediction experiment by a neural network, the fitting experiment by the ARMA model, and the control strategy by IMSD, the IMSD approach can be used to decompose the uniform model into multiple submodels, which can improve the efficiency and accuracy of the model and obtain the best control effect.

REFERENCES

[1] V. Wouk, "Hybrids: Then and now," *IEEE Spectr.*, vol. 32, no. 7, pp. 16–21, Jul. 1995.

[2] C. C. Chan, "An overview of electric vehicle technology," *Proc. IEEE*, vol. 81, no. 9, pp. 1202–1213, Sep. 1993.

[3] F. A. Wyczalek, "Hybrid electric vehicles: Year 2000 status," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 16, no. 3, pp. 15–25, Mar. 2001.

[4] C. C. Chan, "The state of the art of electric and hybrid vehicles," *Proc. IEEE*, vol. 90, no. 2, pp. 247–275, Feb. 2002.

[5] A. Kalberlah, "Electric hybrid drive systems for passenger cars and taxis," *SAE Trans.*, vol. 100, pp. 404–413, 1991.

[6] K. T. Chau and Y. S. Wong, "Overview of power management in hybrid electric vehicles," *Energy Convers. Manage.*, vol. 43, no. 15, pp. 1953–1968, Oct. 2002.

[7] N. Jalil, N. A. Kheir, and M. Salman, "A rule-based energy management strategy for a series hybrid vehicle," in *Proc. Amer. Control Conf.*, Jun. 1997, vol. 1, pp. 689–693.

[8] N. J. Schouten, M. A. Salman, and N. A. Kheir, "Fuzzy logic control for parallel hybrid vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 3, pp. 460–468, May 2002.

[9] F. U. Syed and D. Filev, "Real time advisory system for fuel economy improvement in a hybrid electric vehicle," in *Proc. Annu. Meeting NAFIPS*, 2008, pp. 1–6.

[10] A. Piccolo, L. Ippolito, V. Galdi, and A. Vaccaro, "Optimization of energy flow management in hybrid electric vehicles via genetic algorithms," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2001, vol. 1, pp. 434–439.

[11] C. C. Lin, H. Peng, J. W. Grizzle, and J. M. Kang, "Power management strategy for a parallel hybrid electric truck," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 6, pp. 839–849, Nov. 2003.

[12] R. Saeks, C. J. Cox, J. Neidhoefer, P. R. Mays, and J. J. Murray, "Adaptive control of a hybrid electric vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 4, pp. 213–234, Dec. 2002.

[13] L. Johannesson, M. Asbogard, and B. Egardt, "Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 366–371, Mar. 2007.

[14] M. Gao and M. Zhou, "Control strategy selection for autonomous vehicles in a dynamic environment," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2005, vol. 2, pp. 1651–1656.

[15] M. H. Huang, "Optimal multilevel hierarchical control strategy for parallel hybrid electric vehicle," in *Proc. Veh. Power Propulsion Conf.*, 2006, pp. 1–4.

[16] M. Yamada, K. Ueda, I. Horiba, and N. Sugie, "Discrimination of the road condition toward understanding of vehicle driving environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 1, pp. 26–31, Mar. 2001.

[17] M. Montazeri-Gh, A. Ahmadi, and M. Asadi, "Driving condition recognition for genetic–fuzzy HEV control," in *Proc. 3rd Int. Workshop GEFS*, Mar. 2008, pp. 65–70.

[18] C. C. Lin, H. Peng, S. Jeon, and J. M. Lee, "Control of a hybrid electric truck based on driving pattern recognition," in *Proc. Adv. Veh. Control Conf.*, Hiroshima, Japan, Sep. 2002.

[19] M. Montazeri-Gh and M. Naghizadeh, "Development of the Tehran car driving cycle," *Int. J. Environ. Pollut.*, vol. 30, no. 1, pp. 106–118, 2007.

[20] R. Langari and J. S. Won, "Intelligent energy management agent for a parallel hybrid vehicle—Part I: System architecture and design of the driving situation identification process," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 925–934, May 2005.

[21] A. Poursamad, "Optimal gain scheduling control strategy for parallel HEV based on traffic conditions," *Int. J. Elect. Hybrid Veh.*, vol. 1, no. 3, pp. 221–237, Aug. 2008.

[22] W. T. Hung, H. Y. Tong, C. P. Lee, K. Ha, and L. Y. Pao, "Characterization of vehicle driving patterns and development of driving cycles in Chinese cities," *Transp. Res. Part D: Transp. Environ.*, vol. 13, no. 5, pp. 289–297, Jul. 2008.

[23] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, T. Graf, and R. Schmidt, "Driving environment perception using stereovision," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 331–336.

[24] A. Barth and U. Franke, "Estimating the driving state of oncoming vehicles from a moving platform using stereo vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 560–571, Dec. 2009.

[25] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 425–437, Sep. 2008.

[26] M. Borner, L. Andreani, P. Albertos, and R. Isermann, "Detection of lateral vehicle driving conditions based on the characteristic velocity," in *Proc. Triennial World Congr. Int. Fed. Autom. Control*, Jul. 2002.

[27] Y. C. Ma, M. Chowdhury, A. Sadek, and M. Jeihani, "Real-time highway traffic condition assessment framework using vehicle-infrastructure integration (VII) with artificial intelligence (AI)," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 615–627, Dec. 2009.

[28] V. V. Viikari, T. Varpula, and M. Kantanen, "Road-condition recognition using 24-GHz automotive radar," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 639–648, Dec. 2009.

[29] S. H. Tsang, P. S. Hall, E. G. Hoare, and N. J. Clarke, "Advance path measurement for automotive radar applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 273–281, Sep. 2006.

[30] G. Alessandretti, A. Broggi, and P. Cerri, "Vehicle and guard rail detection using radar and vision data fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 95–105, Mar. 2007.

[31] J. M. Nigro and M. Rombaut, "IDRES: A rule-based system for driving situation recognition with uncertainty management," *Inf. Fusion*, vol. 4, no. 4, pp. 309–317, Dec. 2003.

[32] S. Jeon, S. Jo, Y. Park, and J. M. Lee, "Multi-mode driving control of a parallel hybrid electric vehicle using driving pattern recognition," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 124, no. 1, pp. 141–148, Mar. 2002.

[33] R. Langari and J. S. Won, "Intelligent energy management agent for a parallel hybrid vehicle," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 935–953, May 2005.

[34] M. Montazeri-Gh and M. Asadi, "Influence of the road grade on the optimization of fuzzy based HEV control strategy," presented at the Powertrain and Fluid Systems Conf. Exhibition, Toronto, ON, Canada, 2006, SAE Technical Paper series, Paper 2006-01-3293.
[35] M. Montazeri-Gh and M. Asadi, "Intelligent control for HEV power management based on driving conditions," in *Proc. 2nd ICMSAO*, Abu Dhabi, UAE, 2007.
[36] J. Yang, X. Huang, Y. Tan, and X. G. He, "Forecast of driving load of hybrid electric vehicles by using discrete cosine transform and support vector machine," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 2227–2234.
[37] W. Kruskal and W. Wallis, "Use of ranks in one-criterion variance analysis," *J. Amer. Stat. Assoc.*, vol. 47, no. 260, pp. 583–621, Dec. 1952.
[38] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon, 1995, p. 380.
[39] N. Brenner and C. Rader, "A new principle for fast Fourier transformation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-24, no. 3, pp. 264–266, Jun. 1976.
[40] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.
[41] I. T. Jolliffe, *Principal Component Analysis,* 2nd ed. New York: Springer-Verlag, 2002, ser. Springer Series in Statistics.
[42] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision*. Cambridge, MA: MIT Press, 2005.
[43] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[44] J. S. Taylor and N. Cristianini, *Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[45] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, May 2001.
[46] V. Wouk, "Hybrids: Then and now," *IEEE Spectr.*, vol. 32, no. 7, pp. 16–21, 1995.
[47] V. H. Johnson, K. B. Wipke, and D. J. Rausen, "HEV control strategy for real-time optimization of fuel economy and emissions," presented at the Future Car Congr., Washington, DC, 2000, SAE Paper 2000-01-1543.
[48] A. Brooker, T. Hendricks, V. Johnson, K. Kelly, B. Kramer, M. O'Keefe, S. Sprik, and K. Wipke, "ADVISOR: A systems analysis tool for advanced vehicle modeling," *J. Power Sources*, vol. 110, no. 2, pp. 255–266, Aug. 2002.

Xi Huang received the B.S. degree in mathematics in 2006 from Peking University, Beijing, China, where he is currently working toward the Ph.D. degree in computer science with the Key Laboratory of Machine Perception (Ministry of Education) and the Department of Machine Intelligence, School of Electronics Engineering and Computer Science.

His recent research interests include neural networks, machine learning, and statistics.

Ying Tan (M'98–SM'02) received the B.S. degree in electronic engineering from the Electronic Engineering Institute, Hefei, China, in 1985, the M.S. degree in electronic engineering from Xidian University, Xi'an, China, in 1988, and the Ph.D. degree in signal and information processing from Southeast University, Nanjing, China, in 1997.

In 1997, he became a Postdoctoral Research Fellow and then an Associate Professor with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei. He was a Full Professor, an advisor of Ph.D. candidates, and the Director of the Institute of Intelligent Information Science, University of Science and Technology of China. He was with the Chinese University of Hong Kong, Shatin, Hong Kong, in 1999 and during 2004–2005. He was an electee of the 100-Talent Program of the Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Full Professor and an advisor of Ph.D. candidates with the Key Laboratory of Machine Perception (Ministry of Education) and the Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing. He is also the Head of the Computational Intelligence Laboratory, Peking University. He has authored or coauthored more than 200 academic papers in refereed journals and conferences and several books and book chapters. His current research interests include computational intelligence, swarm intelligence, artificial immune systems, intelligent information processing, pattern recognition, bioinformatics, statistical learning theory, and their applications.

Dr. Tan was the Program Committee Chair for the 2008 International Symposium on Neural Networks and the General Chair for the 2010 International Conference on Swarm Intelligence (ICSI). He is the General Chair of ICSI 2011. He is an Associate Editor of the *International Journal of Swarm Intelligence Research* and the *IES Journal B: Intelligent Devices and Systems* and an Associate Editor-in-Chief of the *International Journal of Intelligent Information Processing*. He is a member of the Advisory Board of the *International Journal of Knowledge-Based and Intelligent Engineering Systems* and of the Editorial Board of the *Journal of Computer Science and Systems Biology* and *Applied Mathematical and Computational Sciences*. He is also the Editor of Springer Lecture Notes on Computer Science, LNCS 5263, 5264, 6145, and 6146, and the Guest Editor of several referred journals, including *Information Science*, *Softcomputing*, the *Computer Journal*, and the *International Journal of Artificial Intelligence*. He was the recipient of a number of academic and research achievement awards from his country and universities due to his outstanding contributions and distinguished works, including the 2009 National Natural Science Prize of China.

Xingui He received the Bachelor's and the Ph.D. degrees in approximation theory from Peking University, Beijing, China, in 1960 and 1967, respectively.

In the 1980s, he studied database and database machines with the Department of Computer and Information Science, Ohio State University, Columbus. Since 2001, he has been an Academician with the Chinese Academy of Engineering and the leader of the Consultative Expert Group on information science with the Chinese National Programs for Fundamental Research and Development (973 Program). From 2002 to 2006, he was the Dean of the School of Electronics Engineering and Computer Science, Peking University. He is currently a Professor and the Chair of the Academic Committee of the School of Electronics Engineering and Computer Science, Peking University. He has been engaged in the theoretical research and engineering practices of computer software and artificial intelligence for a long time. He is one of the pioneers of computer software in China. He has made creative and systematic contributions in the fields of fuzzy theory and technology, computational intelligence, and databases. As of 2010, he has published more than 150 academic papers and 11 Chinese and English books, including *Fuzzy Knowledge Processing Theory and Technology*, *Fuzzy Database Systems, Knowledge Processing and Expert Systems*, *Process Neural Networks—Theory and Applications*, etc. He is also the Editorial Director of multiple dictionaries and encyclopedia. He is the Editor-in-Chief for *Computer Engineering and Design*, the *CAAI Transactions on Intelligent Systems*, and *Computer Science and Exploration*. From 1998 to 2010, he was a Co-Editor-in-Chief for the *Journal of Computers*. He is also the member of the editorial committee of the *Journal of Software* and many other prestigious journals.

Dr. He was the President of Beijing Computer Association from 1999 to 2004. During 1999–2008, he was the Chair of the Anti-Harsh Environment Committee of the China Computer Society. In 2008, he became the Director of the Database Organization Committee of the China Computer Society. In 2010, he was elected Deputy Chairman of the China Association of Artificial Intelligence. He was the recipient of 19 national or ministerial-level scientific and technological progress awards, including the Guanghua Science and Technology Award and the Science and Technology Progress Award of the He-Liang-He-Li Fund in China in 2007.