Prediction of Vehicle Driving Conditions with Incorporation of Stochastic Forecasting and Machine Learning and A Case Study in Energy Management of Plug-in Hybrid Electric Vehicles

Yonggang Liu[1*], Jie Li[1], Jun Gao[1], Zhenzhen Lei[2], Yuanjian Zhang[3] and Zheng Chen[4,5*]
[1]State Key Laboratory of Mechanical Transmissions & School of Automotive Engineering, Chongqing University 400044, Chongqing China
[2]School of Mechanical and Power Engineering, Chongqing University of Science & Technology, Chongqing, 401331, China
[3]Sir William Wright Technology Center, Queen's University Belfast, Belfast, BT9 5BS, United Kingdom
[4]Faculty of Transportation Engineering, Kunming University of Science and Technology, Kunming, 650500, China
[5]School of Engineering and Materials Science, Queen Mary University of London, London, E1 4NS, United Kingdom
Correspondence: Zheng Chen (chen@kust.edu.cn) and Yonggang Liu (andylyg@umich.edu)

Abstract: Prediction of short-term future driving conditions can contribute to energy management of plug-in hybrid electric vehicles and subsequent improvement of their fuel economy. In this study, a fused short-term forecasting model for driving conditions is established by incorporating the stochastic forecasting and machine learning. The Markov chain is applied to calculate the transition probability of historical driving data, by which the stochastic prediction is conducted based on the Monte Carlo algorithm. Then, a neural network is employed to learn the current driving information and main knowledge after the simplified correlation of characteristic parameters, and meanwhile the genetic algorithm is adopted to optimize the initial weight and thresholds of networks. Finally, the short-term velocity prediction is achieved by combining them, and the overall performance is evaluated by four typical criteria. Simulation results indicate that the proposed fusion algorithm outperforms the single Markov model, the radial basis function neural network and the back propagation neural network with respect to the prediction precision and the difference distribution between expectation and prediction values. In addition, a case study is conducted by applying the built prediction algorithm in energy management of a plug-in hybrid electric vehicle, and simulation results highlight that the proposed algorithm can supply preferable velocity prediction, thereby facilitating improvement of the operating economy of the vehicle.

Key Words: Driving condition prediction, Markov chain, neural network, principal component analysis, energy management.

I. INTRODUCTION

Nowadays, plug-in hybrid electric vehicles (PHEVs) represent one main development direction towards transportation electrification. They usually employ two energy sources, such as an internal combustion engine (ICE) and an externally chargeable battery pack, to propel the vehicle in pure electric or hybrid mode. It is critical to employ a determined control principle, referred to as an energy management strategy (EMS), to properly allocate energy distribution of two energy sources. An EMS can not only regulate energy sources to satisfy driving power demand, but also improve vehicle's fuel economy and extend battery lifespan effectively.

Nowadays, EMSs of PHEVs can be roughly classified into three categories, i.e., rule-based schemes, global optimization algorithms, and instantaneous optimization strategies [1, 2]. The prevalent state-of-the-art EMSs are systematically summarized and compared in [3]. A rule-based scheme is comprised of a body of predetermined rules and can usually achieve energy distribution according to vehicle's status. A common knowledge is that rule-based algorithms cannot fully explore the potential fuel savings. Global optimization algorithms, such as dynamic programming (DP) [4] and quadratic programming (QP) [5], can usually find optimal solutions over a certain driving range, and yet they are fragile when subjected to time-varying driving conditions and are intractable to be implemented in practice. Instead, they are usually regarded as a benchmark to quantify and evaluate the performance of other algorithms [6].

Instantaneous optimization strategies usually conduct energy management by locally optimizing the operating cost function, such as fuel consumption minimization, emission deduction, battery lifespan extension, both or all. As a typical representative of instantaneous strategies, equivalent consumption minimization strategy (ECMS), is widely adopted to minimize fuel consumption by solving the so-called Hamiltonian function, which is built based on Pontryagin's minimum principle (PMP) [7]. To this end, trip information should be acquired in advance, and much calculation intensity will inevitably arise. Model predictive control (MPC), favored by industry and academia, can iteratively optimize the decision online in a receding horizon to coordinate energy management of powertrain components [8]. Till now, MPC algorithms have been widely adopted to manage energy distributions in different types of hybrid vehicles, such as fuel cell electric vehicles, hybrid electric vehicles (HEVs) and PHEVs.

In terms of MPC applied in energy management of PHEVs, it is of significant importance to predict driving condition timely and precisely and supply the guided reference before implementing the receding horizon control [9]. It is, to the authors' knowledge, equally important to predict road conditions, compared to EMS itself design. A precise reliable vehicle speed prediction can pave the solid way and provide useful instructions for satisfactory energy management of PHEVs. Currently, a variety of algorithms have been spurred to achieve speed prediction, including Kalman filter [10], exponential method [11], autoregressive moving average (ARMA) methods [12], particle filter [13], stochastic forecast [14], and machine learning algorithms [15]. Amongst them, Markov chain (MC) based prediction algorithms and neural networks (NNs), belonging to stochastic forest and machine learning filed, are most attractive and widely exploited [16]. Actually, future driving condition of vehicle can be estimated based on inner correlation between the current step and previous one or multi moments. Extensive studies have been conducted for improving the accuracy of MC based prediction and control performance of the EMS built upon [17]. In [18], a MC based driving cycle predictor is designed to supply the next step demand reference to operating efficiency optimization of ICE. In [19], the stochastic DP (SDP) is employed to optimize the downshifting control during regenerative braking for HEV based on the predicted braking torque by MC. In [20], a multi-step Markov prediction model is constructed to forecast the near-future driving velocity, then the MPC is applied to manage the energy flow in PHEV. In [21], a reinforcement learning (RL) based offline EMS is investigated based on the built-in stationary Markov transition probability matrix. In [22], the clustering analysis is conducted to find the characteristic parameters of transportation information, and the Markov chain model is built to identify the transportation pattern. In [23], a stochastic approach is proposed to construct the power management algorithm, in which the power demand is modeled as a MC process and estimated based on different driving cycles, then the control scenarios are generated in a stochastic MPC framework. The MC algorithm features long-term trend prediction in terms of entire driving conditions, since the probability can be globally calculated in the whole driving event. Nonetheless, it may not be qualified in short-term prediction due to its global statistical characteristics, and integration with other instantaneous prediction algorithms may solve a possible pathway to tackle this restriction [24].

Machine learning algorithms, in particular NNs, can extract nonlinear laws from the training data and can be well-suited for prediction of future driving conditions thereafter. NN can effectively fit random nonlinear data and features self-learning capability after proper parameter tuning [25]. It can discover the nonlinear law hidden underneath the data and is often employed to extrapolate extension events [26]. NN has been widely applied in prediction of traffic and short-term velocities. In [27], a time-lag recurrent NN is introduced for traffic prediction with high accuracy. In [28], short-term velocities are predicted based on the back propagation NN (BP-NN) with one hidden layer and the radial basis function NN (RBF-NN) which can automatically regulate the amount of hidden nodes. The simulation results indicate that high prediction accuracy of the vehicle velocity can improve the overall controlling performance of HEVs. In [29], a NN based trip model is proposed to help manage the energy distribution of PHEV, and simulation results validates that the built model can increase the prediction accuracy and contribute to the fuel economy improvement. Nonetheless, forecasting performance of NN models can be easily influenced by their fixed network structures, learning rates and inputs; and moreover, the evaluation metrics and the amount of collected training data can also affect the model precision. As discussed in [30], the prediction performance of NN can be discredited when encountered with over-training or over-extrapolation. Over-training occurs when the data capacity of NN is too large, while over-extrapolation means the benefit of NNs is not be fully explored when required to conduct estimation beyond the range of current training data. To sum up, it can be concluded that only a NN or MC model is intractable to predict driving conditions in a combined macro- and micro-framework, thereby meeting the requirement of long-term driving trend and short-term driving speed prediction under various conditions.

To compensate the inherent shortcomings of NN and MC, a fusion prediction model incorporating both algorithms is herein proposed to supply robust and accurate prediction for vehicle driving conditions. In addition, it does not easily fall into local optimum and can adapt to different conditions. To the best of authors' knowledge, it is seldom reported in the existing literature. To be specific, the proposed forecasting model mainly includes two parts, i.e., the driving condition prediction module and the nonlinear fusion module. There exist two NN controllers (called NN1 and NN2 hereinafter) and one MC predictor in the whole prediction framework. In it, NN1 accounts for predicting the driving condition in micro scope, and the key task of MC is to forecast the entire driving trend from macro perspective. In particular, the inputs of NN1 are simplified by means of principal component analysis (PCA) to avoid over-training. In the nonlinear fusion module, the primary velocity prediction and trend prediction are incorporated by applying NN2 to improve the prediction accuracy. In addition, the over-training and extrapolation of NN2 can also be avoided by exploiting the trend prediction result to refine the primary velocity prediction result. The initial weights and threshold of these two NNs are optimized by the genetic algorithm (GA) to prevent from being trapped into local optimum. Numerical simulations are performed to validate the effectiveness of the proposed fusion method by comparing with traditional prediction methods. In addition, to further validate the performance and highlight the benefit of the proposed algorithm, a case study is conducted, where the prediction algorithm is applied to supply the reference for predictive energy management of PHEV, and the MPC is applied with the help of the future speed information supplied by different speed prediction algorithms including the proposed fusion algorithm. The operating cost is compared to evaluate the predictor's performance, and the detailed comparison results indicate that the proposed algorithm is more effective in supplying future information for predictive EMSs and thus can facilitate operating economy improvement of PHEVs.

The remainder of this article is structured as follows. The methods for short-term driving condition prediction are elaborated in Section II, and the evaluation metrics are detailed in Section III. The simulation results and discussion are presented in Section IV. The case study is conducted to further validate the predictor's performance in Section V. Finally, the main conclusions are given in Section VI.

II. SHORT-TERM DRIVING CONDITION PREDICTION MODEL

A. Markov Chain Based Velocity Prediction

In this study, the velocity prediction is firstly considered as a Markov model, which describes the relationship between the actual and next states of system. Here, P_{ij} denotes the probability of state j of variable X at time step $k+1$, given the current state i at time k, and the following equation can be yielded, as:

$$P(X(k+1)=j \mid X(k)=i)=P_{ij} \tag{1}$$

The transition probability function, which provides next step information, is formulated in (2), and correspondingly, the multi-step ahead transition probability is presented in (3), as:

$$P(X(k+1) \mid (X(k))=P_{ij}(1)=P_{k,k+1} \tag{2}$$

$$P(X(k+n) \mid (X(k))=P_{ij}(n)=P_{k,k+1}^{n} \tag{3}$$

The transition probability matrix of a stationary MC model can be trained and updated according to the observation of previous states. For ease of predicting driving conditions based on MC, the driving conditions need to be divided into a finite number of states. In this paper, the velocity trajectory can be determined by the current and acceleration, which, at the end of each interval, is attributed to one specific state by approximation. Here, the acceleration can be calculated by:

$$a(t)=\frac{v(t+1)-v(t)}{3.6} \tag{4}$$

where a and v denote acceleration (unit: m/s^2) and speed (unit: km/h) of the vehicle, respectively. As such, the driving speed profile is discretized to numerous states which can be characterized by the velocity and acceleration [31], which are supposed to range from 0 to 130 km/h and -3 to 3 m/s^2, respectively. In addition, the discretization width of velocity and acceleration is set to 1 km/h and 0.05 m/s^2, respectively. Then, the speed is mapped to a specific position in the grid table via the nearest-neighbor interpolation. By this manner, the driving speed is discretized to different states, as shown in Fig. 1, and each dot highlights one state under the current driving cycle. For instance, the state with positive acceleration and high velocity indicates that the vehicle is accelerating at high speed. In addition, the state transition probability is quantified by the maximum likelihood estimation [32], which is widely adopted to estimate the probability. Its main principle is to select the highest value as the estimated probability through a number of tests. In this research, the distribution of the state transition probability for velocity is discrete, and can be calculated by (2). After calculation, the state transition probability distribution is plotted in Fig. 2, where the inputs include the current state and next state, and the output is the transition probability transferring from the current state to next state. As can be found, larger value means that the current state tends to transfer to the next state with higher probability.

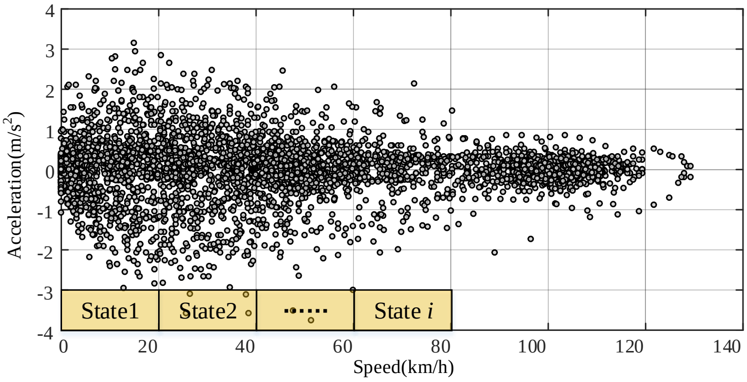


Fig. 1. State division diagram of driving conditions.

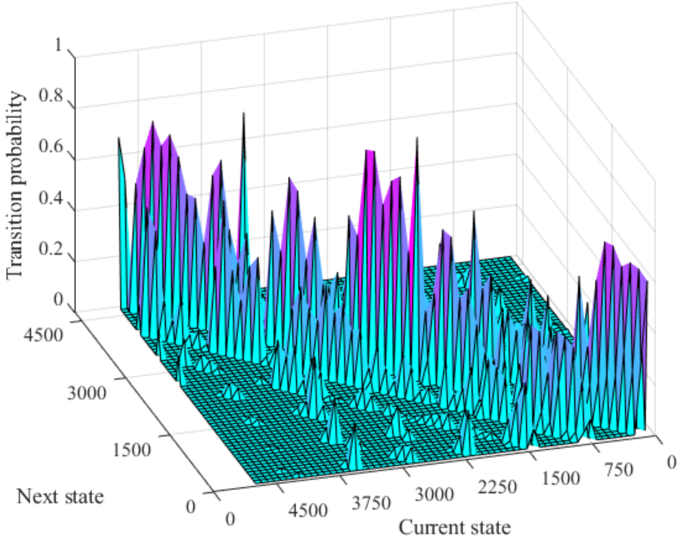


Fig. 2. State transition probability.

In the process of prediction decision, the Monte Carlo simulation [31] is performed between the state space of driving condition $M = \{1,2,3,\cdots,m\}$, transition matrix P, and homogeneous MC $\{X^{(n)}, n = 1,2,3,\cdots\}$ in the initial state i_1. The i_{th} row corresponding to the transition probability matrix is extracted according to the current driving state i. Now, a random number r, ranging from 0 to 1, is generated by applying the Monte Carlo simulation, and if the following inequation is satisfied, i.e.,

$$\sum_{j=1}^{k-1} P_{ij} < r \le \sum_{j=1}^{k} P_{ij} \quad (k \le N) \tag{5}$$

then the current state will transfer to state k_{th} in the next moment. As such, the predicted next step speed can be expressed as:

$$v_p = v_i + L_v \times n_{1\to p} \tag{6}$$

where v_p is the predicted speed, v_i is the initial speed, L_v denotes the division step of speed, and $n_{1\to p}$ is the number of state intervals in the velocity direction.

B. NN Based Velocity Prediction

NN can fully approximate any complex nonlinear mapping and characterizes strong capability of learning from and adapting to uncertain, strong robust and fault tolerant systems. Given this, NN has become a luxuriant technique to model nonlinear systems [33]. The basic requirements of NN prediction model are with capabilities of fast convergence, real-time implementation and generalization [34]. For vehicle velocity prediction, the inputs of NNs can be historical velocity sequences, and the outputs are the predicted velocity sequences in a receding horizon, as plotted in Fig. 3. Each input/output pattern is composed of a moving window with a fixed length, as:

$$\left[v_{k+1}, v_{k+2}, \cdots v_{k+H_p}\right] = f_{NN}(v_{k-H_h+1}, \cdots v_k) \tag{7}$$

where H_h denotes the dimension of input velocity sequence, f_{NN} represents the nonlinear map function of the NN based predictor, and H_p expresses the prediction horizon.

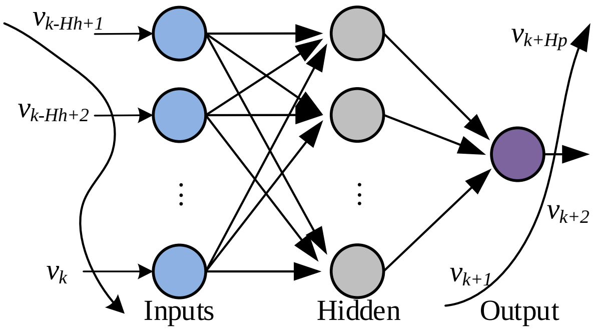


Fig. 3. Neural network prediction model.

C. The Fusion Velocity Predictor

Driving conditions are essentially stochastic time-varying sequences, and their changing rule is irregular and nonlinear. It is therefore difficult to formulate it with a determined regression model. Even MC and NN have declared to predict the future driving conditions with certain credibility, there still exists a non-negligible distance away from high prediction accuracy all the time based only on a single algorithm. A common knowledge is that each prediction method has its own pros and cons [28]. The MC stochastic forecasting model can predict the general trend of driving conditions, since it operates based on the statistical transition probability of the whole driving cycle [31]. Nonetheless, MC fails to predict the short-term prediction precisely due to its stochastic characteristics. In contrast, NN can supply the short-term prediction with high precision because of its strong nonlinear fitting and feature information excavation capabilities. However, it may easily fall into local optimum, and also over-training and extrapolation may emerge [35]. To overcome both shortcomings and advance their capabilities in respective long-term and short-term velocity prediction, a fusion prediction model is innovatively proposed with the adaptive corporation of MC and NN, thereby attaining high prediction accuracy and qualified adaption to different driving conditions.

The overall framework of the proposed fusion prediction method is shown in Fig. 4, which is mainly comprised of two parts: a driving condition prediction module and a nonlinear fusion module. The driving condition prediction module contains two basic predictors; including a MC predictor and a NN predictor (called NN1 hereinafter). The training data is divided into two sets, namely dataset 1 and dataset 2. The dataset 1 accounts for the training of MC and NN1 prediction model, while the dataset 2 is mainly for training of the fusion module. In the driving condition prediction module, NN1 takes charge of primary velocity prediction. Furthermore, the MC model is applied as a trend predictor to calculate the transition probability and statistical characteristics of historical driving conditions. Notably, the inputs of NN1 consist of driving condition parameters and adjacent historical velocities. Furthermore, the driving condition parameters considered as the inputs of NN1 are simplified by means of PCA to avoid over-training. In the nonlinear fusion module, as the primary prediction results and trend prediction results show complex diversity and nonlinearity, they are fused by another NN, namely NN2, with strong nonlinear mapping capability. The specific parameters of proposed prediction model are shown in Table I.

In this study, we selected the back propagation neural network (BPNN) for NN1 and NN2 and a first-order multi-scale single step MC as the trend predictor [31]. The determination of main parameters and structures are elaborated as follows.

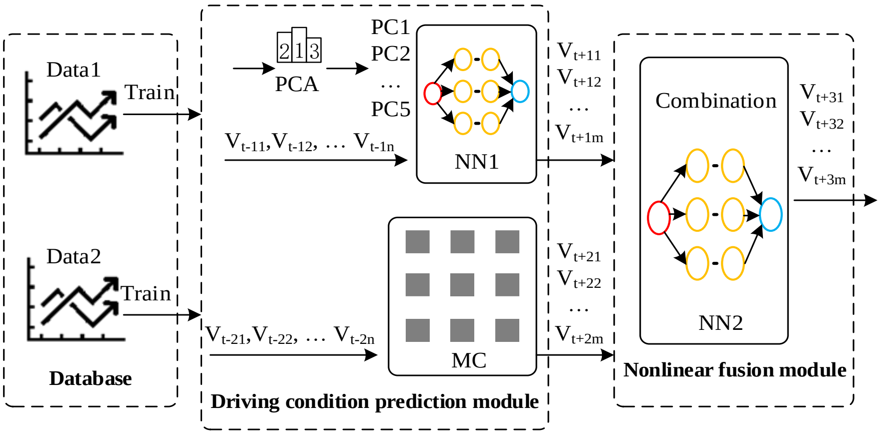


Fig. 4. The working principle of the proposed fusion prediction method.

Table I Specific parameters

Project	Nomenclature
NN1 , NN2	Primary predictor and combination
MC	Markov chain predictor
PC1, PC2 , …, PC5	Principal component
V_{t-11}, V_{t-12} , …, V_{t-1n}	Adjacent historical velocity input to NN1
V_{t-21}, V_{t-22} , …, V_{t-2n}	Adjacent historical velocity input to MC
V_{t+11} , V_{t+12} , , V_{t+1m}	Output of NN1 and input of the combiner NN2
V_{t+21} , V_{t+22} , , V_{t+2m}	Output of MC and input of the combiner NN2
V_{t+31} , V_{t+32} , , V_{t+3m}	Output of the combiner
n , m	Number of inputs or outputs

1) Extraction of Characteristic Parameters Based on PCA

In the prediction process of driving condition, if several adjacent velocities are considered as the only input of NN1, they cannot truly reflect the characteristics of certain driving conditions, thus resulting in low prediction accuracy [36]. In contrast, if too many parameters are extracted, the computational intensity cannot be tolerated. After trial and error, 12 characteristic parameters, as listed in Table II, are selected to predict the driving conditions with preferable prediction and acceptable calculation intensity. However, it is still imperative to reduce the dimension of the selected characteristic parameters, and the PCA algorithm is employed.

Table II Driving condition parameters

Driving condition parameter	Denotation	Driving condition parameter	Denotation
Average speed	v_m	% of time in idle speed	p_{idl}
Driving average speed	v_{md}	Maximum acceleration	$a_{\max}$
Variance of acceleration	a_{var}	Minimum acceleration	$a_{\min}$
% of acceleration time	p_{acc}	% of time in speed interval 0–15 km/h	$r_{0\text{-}15}$
% of deceleration time	p_{dec}	% of time in speed interval 15–30 km/h	$r_{15–30}$
% of time in certain speed	p_{con}	% of time in speed interval 50–70 km/h	$r_{50–70}$

The PCA can reduce dimensions of large amounts of data, and transform the original data into new independent components while keeping the main characteristics of original data [37]. It is usually expressed as a linear combination of original variables. Two essential principles when applying the PCA should be obeyed, including that the number of new principal components should not be more than the number of original variables, and the principal components are independent of each other. If p characteristic parameters, denoted as $x = (x_1, x_2, \cdots, x_p)'$, are introduced to evaluate a driving condition, then we can attain:

$$\begin{cases} y_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_p = \boldsymbol{a}'_1\, \boldsymbol{x} \\ y_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2p}x_p = \boldsymbol{a}'_2\, \boldsymbol{x} \\ \vdots \\ y_p = a_{p1}x_1 + a_{p2}x_2 + \cdots + a_{pp}x_p = \boldsymbol{a}'_p\, \boldsymbol{x} \end{cases} \tag{8}$$

where $\boldsymbol{a_1}, \boldsymbol{a_2}, \cdots, \boldsymbol{a_p}$ denote the unit vector, and y means the principal component. The contribution rate is the proportion of variance of the i_{th} principal component y_{i} over the total variance. The contribution rate of the principal component reflects the ability of expressing main information of the original variable. The sum of the top m $(m \le p)$ contribution rates is called the cumulative contribution rate of the first m principal components [37]. The principal component is expressed as.

$$y_j = \boldsymbol{t}'_j \boldsymbol{x} = t_{1j}x_1 + t_{2j}x_2 + \cdots + t_{pj}x_p, j = 1, 2, \cdots, p \tag{9}$$

where t_{ij} is the load coefficient of the j_{th} principal component y_j on the original variable x_i, and t_{ij} reflects the importance of x_i on y_j.

2) Short-Term Driving Condition Prediction

The general idea of the fusion prediction algorithm is described as follows. In the driving condition prediction module, the application of NN1 and MC are almost similar to those of a single prediction model. However, in the nonlinear fusion module, the prediction results of NN1 and MC are complex, random and nonlinear, and there does not exist distinct relationship between them. Thus, reliable and intelligent fusion methods are strongly encouraged to incorporate both models dynamically according to the actual driving conditions. As a typical intelligent algorithm, NN enables mapping nonlinear data with high precision and can adjust weights and thresholds according to the training data, thus it should be feasible to apply the NN based fusion method to fit the relationship between primary values and the trend values. As such, NN2 is applied to incorporate the prediction results of NN1 and MC for providing the final predicted velocity. The inputs of NN2 include two variables, i.e., the predicted velocities by NN1 and by MC; and the output of NN2 is the final predictive velocity. Note that the parameters of NN2, such as the number of layers, can impose significant influence on computational efficiency and prediction accuracy. According to our previous experiences, one hidden layer in NN2 can meet the accuracy of nonlinear velocity prediction, as is the case with most application scenarios [38]. Thus, both NN1 and NN2 have the same structure and contain three layers, including a hidden layer, an input layer and an output layer. With respect to the number of neurons and the training iterations, we determine them according to our experience and experiment iteration. The number of neurons is respectively set to 30 and 15 for NN1 and NN2 after iteration and correction; the maximum number of training iterations is defined as 200 for both NNs in this research. The specific process of the fusion prediction, as shown in Fig. 5, is mainly divided into five steps:

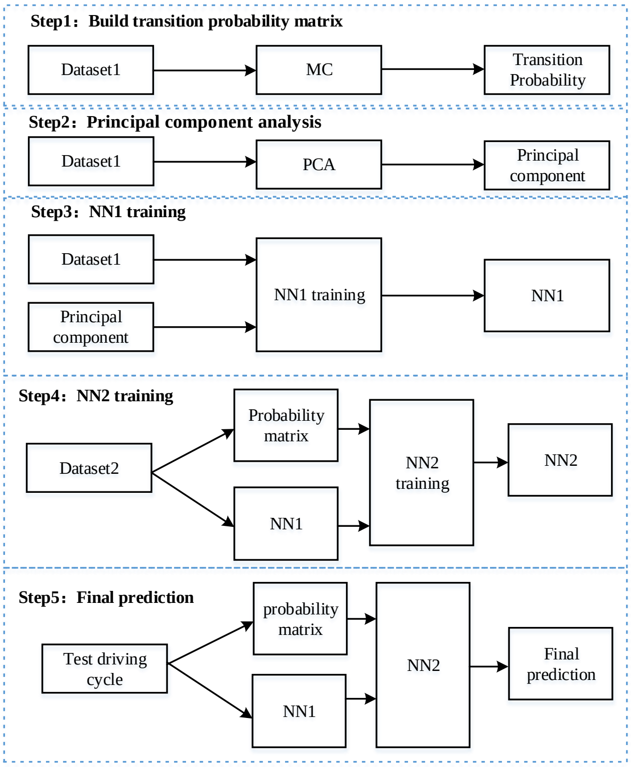


Fig. 5. Schematic diagram of the proposed method.

a) Train the MC model based on the dataset 1 and calculate the transition probability matrix;

b) Extract the characteristic parameters repeatedly from the dataset 1 according to the deterministic receding time window and apply the PCA to calculate the principal components for NN1;

c) Train NN1 according to the principal components and the historical adjacent vehicle speed to attain the primary prediction model;

d) Use dataset 2 to test NN1 and MC models to obtain the primary prediction output data. Notably, the output of NN1 and MC model are utilized as the input of NN2, which is trained to obtain the final fusion model.

e) The validation data are set as the input to the proposed fusion prediction model, and the output from the combined algorithm is considered as the result.

3) Optimization of the Proposed Fusion Prediction Model Based on GA

As discussed before, the selection of initial weights and thresholds of NN is random and lacks a reasonable argument. This may lead to over-fitting and discredit the prediction accuracy, thus deteriorating the application performance of NN in the fusion model. However, the relationship between NN parameters and prediction accuracy cannot be directly formulated, and direct optimization methods are difficult to tackle this optimization problem. GA is an optimization method to find the optimal solution by simulating the natural evolution process, and can search the optimal solution adaptively without the specific formulation and models. In [39], GA has been successfully employed to effectively optimize the parameters of NNs. On this account, GA is continuously exploited in this study to optimize the initial weight and threshold of NN1 and NN2 to further improve the prediction accuracy. After repetitive adjustment, the population size, crossover rate and mutation rate of GA are set to 100, 0.7 and 0.02, respectively.

D. Evaluation Metrics

Numerous statistics criteria have been proposed and employed to evaluate the prediction performance, however, only a single variant is incomplete to characterize the performance comprehensively. In our study, four well-known variables, including average root mean square error (RMSE) [28], skewness [40], kurtosis [40] and computation time, are calculated to evaluate the prediction effect. Amongst these evaluation metrics, the RMSE is used to evaluate the overall accuracy of velocity prediction, the skewness and kurtosis can illustrate the overall trend of the prediction deviation, and the computation time is used to evaluate the real-time performance of the prediction algorithm. The average RMSE R_{e} (m/s) can be expressed as:

$$\begin{cases} e(k+i\,|\,k)=\hat{v}(k+i\,|\,k)-v(k+i), i=1,2\cdots H_P,\ \ k=1,2,\cdots N \\ R(k)=\sqrt{\sum_{i=1}^{p}(e(k+i\,|\,k))^2\,/\,H_P}, \\ R_{\mathrm{e}}=\sqrt{\sum_{k=1}^{N}R^2(k)\,/\,N} \end{cases} \tag{10}$$

where e and R denote the error and RMSE that depend on the prediction horizon H_P, N is the total number of expected values; $v(k+i)$ is the expected speed at time $k+i$; and $\hat{v}(k+i\,|\,k)$ is the prediction speed at time

k. The skewness evaluation can be formulated as:

$$\begin{cases} \hat{v}_e(i) = \dfrac{N}{(N-1)(N-2)} \sum_{k=1}^{N} \left(\dfrac{e(k+i\,|\,k) - \hat{\mu}_e(i)}{\hat{\sigma}_e(i)} \right) \\ \hat{\mu}_e(i) = \overline{e(i)} = \dfrac{1}{N} \sum_{k=1}^{N} e(k+i\,|\,k) \\ \hat{\sigma}_e(i) = \sqrt[2]{\dfrac{1}{N} \sum_{k=1}^{N} \left(e(k+i\,|\,k) - \overline{e(i)} \right)^2} \end{cases} \tag{11}$$

where $\hat{\mu}_e(i)$ and $\hat{\sigma}_e(i)$ denote the mean error and error standard deviation, respectively. Skewness is related to three central moments of distribution. A common knowledge is that the skewness status defines the degree of skewness and bias of distribution, and reflects the symmetric distribution characteristics of the error. If the skewness value is greater than zero, the error distribution is positively biased (right deviation); or else, if the skewness is less than zero, the error has a negative bias (left deviation); and moreover, if the skewness equals zero, the distribution is symmetric. In addition, the kurtosis can be calculated, as:

$$\hat{k}_e(i) = \frac{N \cdot (N-1)}{(N-1)(N-2)(N-3)} \sum_{k=1}^{N} \left(\frac{e(k+i\,|\,k) - \hat{\mu}_e(i)}{\hat{\sigma}_e(i)} \right)^4 - \frac{3 \cdot (N-1)^2}{(N-2)(N-3)} \tag{12}$$

The kurtosis is related to the fourth-order central moment of distribution. The kurtosis coefficient is used to measure the data out-migration degree. The greater the kurtosis, the values in the sequence will be more extreme. The error distribution reflects the shape information. If the kurtosis is positive, the pattern has a sharp peak and a long tail, compared with the Gaussian distribution. By contrast, if the kurtosis is negative, the pattern has a flat peak and a short tail. In addition, the calculation time of single speed point is also utilized to evaluate the real time performance of the prediction algorithm.

III. Simulation and Analysis

In this section, traditional single prediction methods are also employed, and their prediction performance is comprehensively compared with that of the proposed fusion algorithm. First, the training and data validation is conducted, followed by the performance discussion in terms of different prediction algorithms and different prediction length. Note that all the simulations and the designed NN based fusion predictor are constructed in MATLAB/Simulink.

The prediction of short-term velocity within a limited horizon is essentially a nonlinear and nonstationary stochastic process. An instinctive consideration about the construction of training dataset is that they should involve different driving conditions, such as highway, urban and urban (congested) roads, so as to strengthen the predictor's adaption to different scenarios. In this research, a variety of standard driving cycles characterizing different road conditions are combined with random sequence to form the datasets 1 and 2, of which the speed profile is shown in Fig. 6. The dataset 1 is composed of NYCC, New York Bus, ECE_EUDC_LOW, HWFET, INDIA_HWY_SAMPLE, US06_HWY, NEDC, INDIA_URBAN_SAMPLE and UNIF01 cycles; and the dataset 2 consists of MANHATTAN, LA92, SC03, WVUCITY, REP05 and Nuremberg R36 cycles. As can be found from Fig. 6, both datasets include the highway, urban and urban congested conditions. In addition, the UDDS cycle is chosen as the validation data.

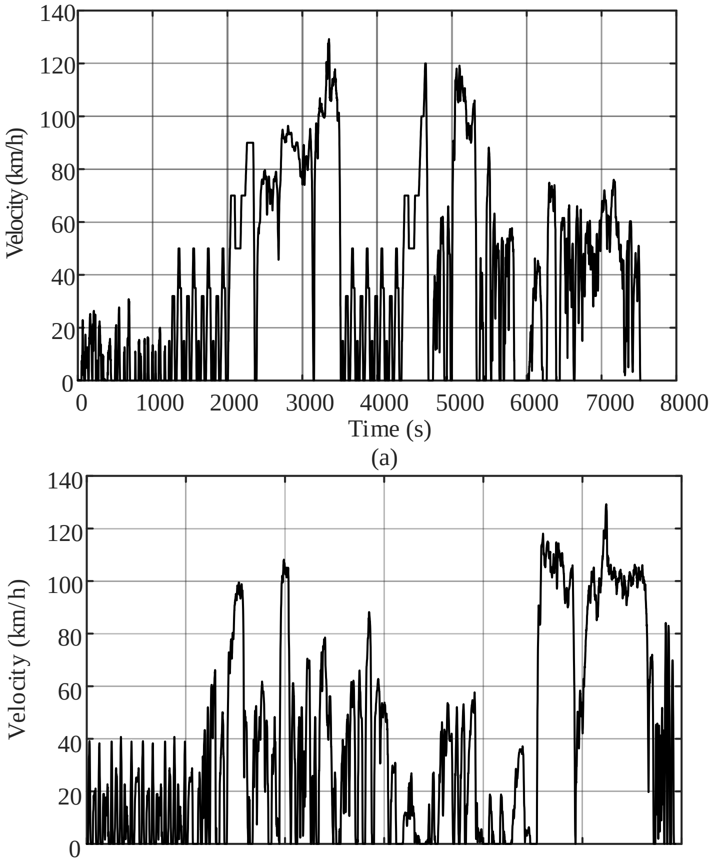


Fig. 6. Training data. (a) Training dataset 1; (b) Training dataset 2.

A. *Result Analysis of the Training Based on the Proposed Method*

Based on the collected data, the characteristic parameters such as the average speed can be calculated within a fixed time window. However, if the time window is too narrow, the historical data may not highlight the driving cycle comprehensively and lead to increased complexity of NNs, thus affecting the prediction performance; and in contrast, if the time window is too wide, the computational burden may be too heavy for real-time application. After repetitive comparison and validation, the time window is set to 175 s and the update step is set to 1 s, through trading-off the prediction accuracy and actual demand of energy management control of PHEVs.

To determine the input of NN1, the historical velocity in the time window is deployed to calculate the characteristic parameters by PCA. For instance, the Pareto diagram of the first time window for dataset 1 is shown in Fig. 7. The solid line denotes the cumulative contribution rate and the gray bar indicates the contribution rate of a single principal component. The detailed contribution rate of the PCA is listed in Table III. As presented in Table III and Fig. 7, the first five principal components derived from the PCA occupy more than 96.81% of the variance of the original driving condition vectors. Thus, 12 characteristics parameters can be converted into five vectors, thereby reducing the dimension of data. The load coefficients of first five principal components are given in Table IV. As can be seen from the principal component PC1 in Table IV, the coefficient of "average speed" is relatively large. Therefore, this principal component is greatly affected by "average speed". Similarly, the load coefficients of other characteristic parameters can also be reasonably illustrated. Based on Table IV, equation (9) can be leveraged to compress the original driving condition parameters into a reduced size vector. In addition, the velocity data of adjacent driving conditions are rather continuous and supply guidelines for future speed prediction. As such, the periodical update of the first 5 s historic adjacent speed and the first five principal components are considered as the input of NN1. The input horizon n in Table I is specified as 5, and the prediction horizon m is set to 10. Similarly, the input and prediction horizon of several common prediction models are also set to n and m, respectively.

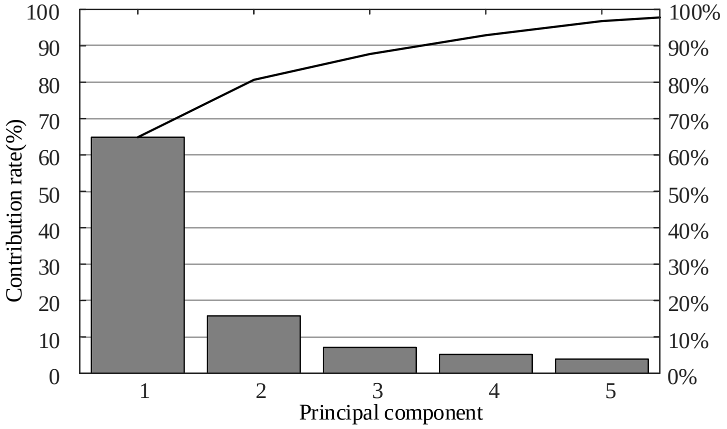


Fig. 7. Pareto diagram of PCA.

Table III Detailed results of PCA

Eigenvalues	Contribution rate (%)	Cumulative contribution rate (%)
$1.8536e^{+03}$	64.8815	64.8815
450.6235	15.7733	80.6548
202.7959	7.0985	87.7533
147.5119	5.1634	92.9167
111.2259	3.8933	96.8099
56.8498	1.9899	98.7998
25.3748	0.8882	99.6880
7.6074	0.2663	99.9543
1.1043	0.0387	99.9930
0.1156	0.0387	99.9930
0.0698	0.0024	99.9995
0.0151	$5.2943e^{-04}$	100.0000
$3.2640e^{-31}$	$1.1425e^{-32}$	100.0000
$6.7906e^{-35}$	$2.3769e^{-36}$	100.0000

Table IV Load coefficients expression

Variable	PC1	PC2	PC3	PC4	PC5
v_m	0.3835	0.0394	−0.0951	−0.0245	−0.2589
v_{md}	0.3507	0.0047	−0.0843	−0.1300	−0.2729
a_{var}	−0.1718	−0.1377	−0.1765	−0.3685	−0.2501
p_{acc}	−0.0395	0.2294	−0.3183	0.1677	−0.2615
p_{dec}	−0.0707	0.0999	−0.4273	0.0105	−0.1683
p_{con}	0.4514	0.1466	0.5124	0.2135	0.1949
p_{idl}	−0.3416	−0.4761	0.2347	−0.3915	0.1949
$a_{\max}$	−0.0041	−0.0023	−0.0146	−0.0054	−0.0076
$a_{\min}$	0.0041	$6.7906e^{-04}$	0.0159	0.0012	0.0106
$r_{0\text{-}15}$	−0.2950	−0.1267	−0.2861	0.6449	0.3533
r_{15-30}	−0.1602	0.7334	−0.1040	−0.4015	0.4552
r_{50-70}	0.5106	−0.3389	−0.5080	−0.2059	0.5227

Subsequently, the proposed fusion prediction model is optimized based on GA introduced before. The fitness value curves of NN1 and NN2 are shown in Fig. 8. In the optimization process, the fitness value gradually converges to the average fitness. The horizontal and vertical coordinates mean the number of generation and fitness value, respectively.

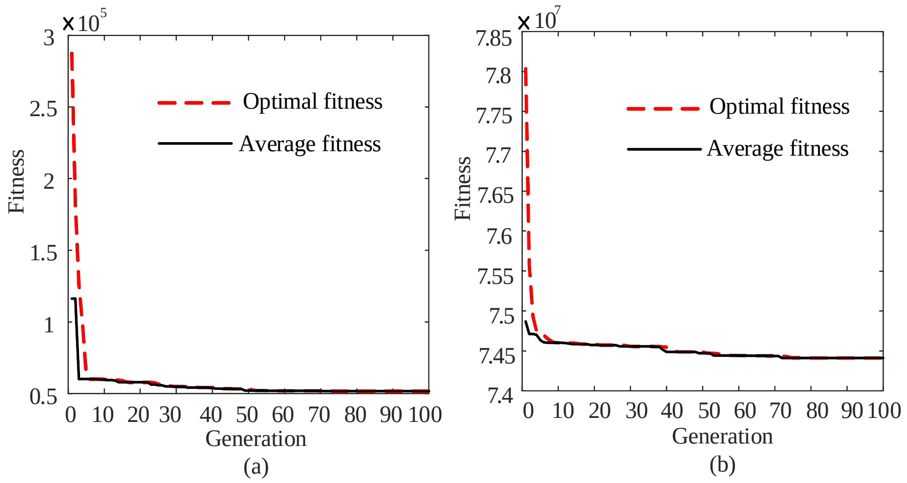
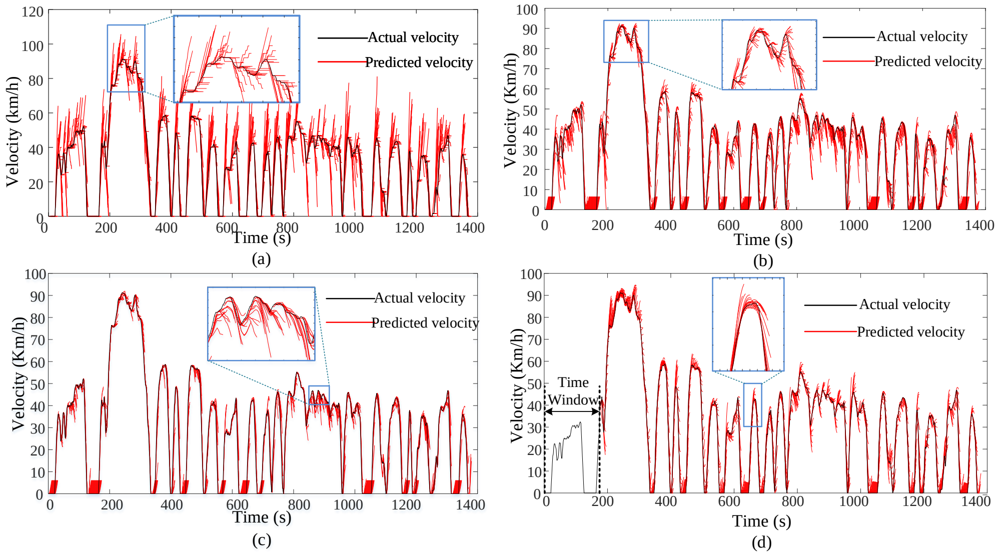


Fig. 8. Fitness curve of optimization. (a) Fitness of NN1; (b) Fitness of NN2.

B. Performance Analysis of Velocity Prediction

To verify the prediction performance of the proposed fusion algorithm, a combined simulation is performed and traditional prediction algorithms are also applied, including the MC, BP-NN and RBF-NN. The RBF-NN is a forward NN with three layers, which can approximate arbitrary nonlinear functions with high precision; therefore, it is widely used in time series analysis, pattern recognition, nonlinear control and image processing. The results, as shown in Fig. 9, depict the tentacle driving trend of the actual speed and predicted speed, where the black line is the actual speed and the red line is the short-term predictive speed. Fig. 9 (a) to (d) illustrates the velocity prediction simulation results of the MC algorithm, BP-NN, RBF-NN and the proposed algorithm, respectively. Fig. 10 shows the RMSE of the built prediction model. According to the simulation result in Table V, the average value of RMSE is 1.3801, proving the proposed method can predict the speed with preferable precision. Figs. 11 and 12 depict the error distribution by the BP-NN and the proposed model when the prediction horizon ranges from 1 s to 10 s. It can be concluded that the error distribution is symmetric; that, the skewness of two prediction models is close to 0. The results also show that the error distribution of the prediction method proposed in this study is more uniform and the distribution range is narrower. Table V lists the evaluation results in terms of different prediction models. It can be observed that the prediction performance of NNs is better than that of MC prediction models, and R_e of the optimized fusion model is the least. Owing to the stochastic property of the MC prediction model, it is possible that a low velocity may be transferred to a high value, and vice versa, thus leading to constraint violation of acceleration. In contrast, the constraint violation of NN is better than that of the MC model, whereas the performance of proposed model is a combination of the two prediction models, which can outperform each other.



(a)

(b)

(c)

(d)

Fig. 9. Simulation of speed prediction. (a) MC Prediction; (b) BP Prediction; (c) RBP Prediction; (d) Fusion Prediction

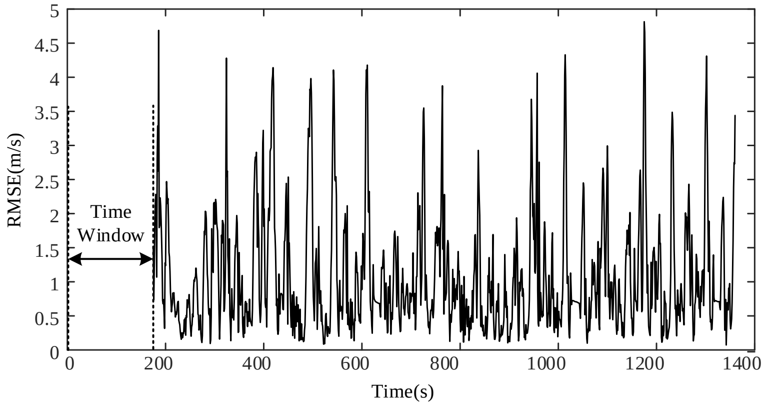


Fig. 10. RMSE of proposed fusion prediction

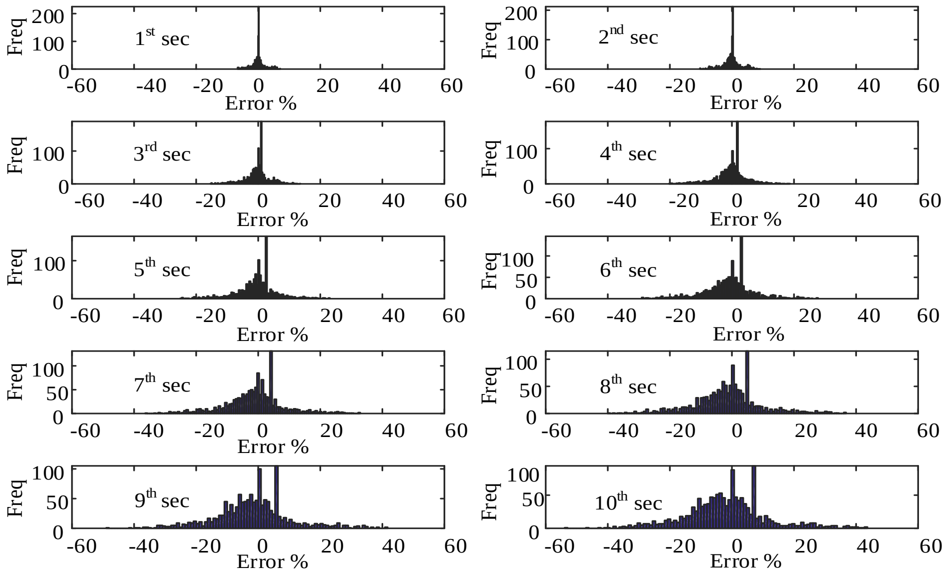


Fig. 11. Error distribution histogram of BP-NN.

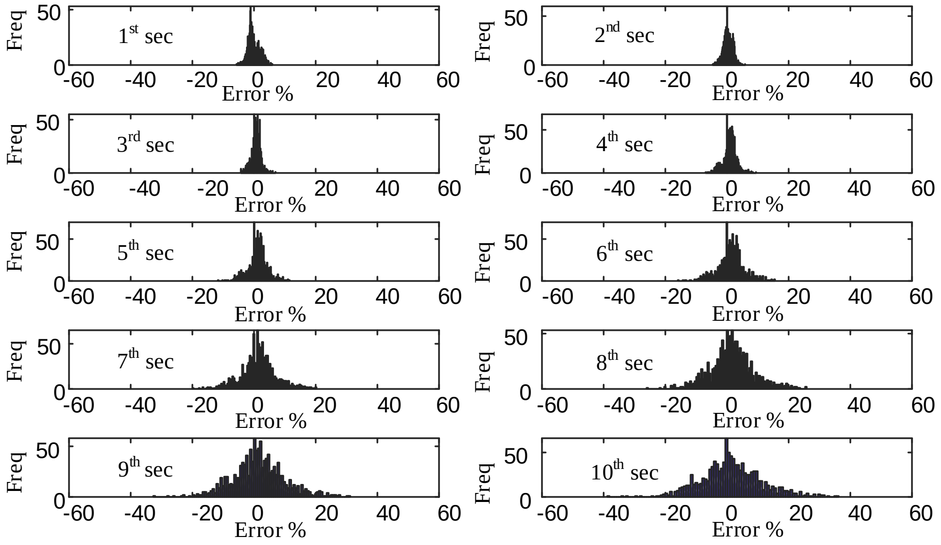


Fig. 12. Error distribution histogram of fusion prediction.

In addition, the results listed in Table V show that the kurtosis of the fusion algorithm is smaller than that of MC, and the error distribution of the proposed method is more symmetrical and shows a narrower variation range, compared with other prediction models, proving its advance and feasibility. The calculation time is more than that of the single forecasting model, as the fusion prediction model is more complex than the single prediction model. Note that, all the predictions include off-line training and online prediction. As can be seen in Table V, the online prediction time of proposed fusion prediction is 4.418 ms, which can still fully satisfy the real-time application requirement. Table VI lists the improvement of R_e, and we can find that the prediction accuracy of the proposed fusion prediction model increases by 45.32%, 39.13%, 40.56%, and 9.75%, compared with those of the first-order MC, BP-NN, RBF-NN and the fusion prediction without optimization. These results indicate that a significant improvement can be achieved when the GA optimization algorithm is employed for parameters optimization of the prediction model.

Table V Comparison of prediction results

Prediction model	Re	Skewness	Kurtosis	T_{cal}(ms)
1st multi-scale single step MC	2.5240	7.2458	−0.4798	2.2888
BP-NN	2.2674	4.1166	0.1872	0.1366
RBF-NN	2.3217	4.1104	0.2131	0.7174
Fusion prediction without optimization	1.5292	3.7676	0.2025	4.3954
Optimized fusion prediction	1.3801	3.2428	0.3602	4.4181

Table VI Comparison of prediction accuracy

Object	Fusion prediction without optimization	1st multi-scale single step MC	BP-NN	RBF-NN
Improvement	9.75%	45.32%	39.13%	40.56%

IV. Case Study in Energy Management of PHEVs

In this work, to verify the performance of velocity prediction, a case study is conducted by apply the prediction algorithm to energy management of a parallel PHEV, of which the main powertrain topology is shown in Fig. 13. As can be seen, there exists a clutch between the engine and the motor. A dual clutch transmission (DCT) is deployed between the motor and final differential reducer. The main parameters of the vehicle are specified in Table VII.

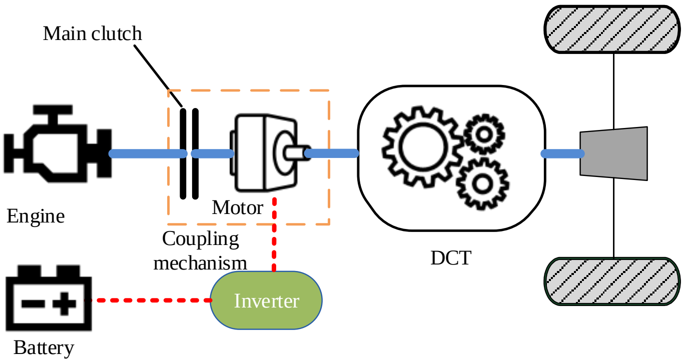


Fig. 13. Plug-in hybrid electric system model.

Table VII Basic parameters of the power train components

Items	Parameters	Values
ISG motor	Rated power (kW)	40
	Peak power (kW)	50
	Rated speed (r·min^{-1})	2500
	Peak speed (r·min^{-1})	6000
	Peak efficiency (%)	92
Engine	Peak power (kW)	80
	Peak speed (r·min^{-1})	6000
	Maximum torque (N·m)	140
Lithium-ion battery	Capacity (A·h)	38.5
	Rated voltage (V)	288
	Cell voltage (V)	1.2
DCT	Gear ratio i1/ i2 /i3	2.8122/1.9274/1.0310
	i4/ i5/ i6	0.8102/0.6088/0.5293
	Final drive ratio	5.24

To validate the speed predictor's performance, MPC is employed to regulate the control output in a receding horizon according to the predicted speed, and DP is regarded as a benchmark to evaluate the performance of the MPC based strategy. In addition, DP is also applied in this study to solve the local optimal control in the procedure of receding horizon optimization of MPC. By this manner, the improvement generated by prediction method can be quantified and compared. Since the PHEV can be charged from power grid and propelled by ICE, the total cost consumption, considered as the objective function in this study, is calculated by adding the electricity cost and the fuel cost, as:

$$J_k(t) = \sum\nolimits_{t=k}^{k+hp} (j_f Q_f(t) + j_{ele} Q_{ele}(t)) \tag{13}$$

where $J_k(t)$ is the total cost during step k to step $k+hp$; j_f represents the fuel price per liter, and equals CNY 7.8 in this study; j_{ele} means the unit price of electricity per kWh, which is set to CNY 0.52; $Q_f(t)$ and $Q_{ele}(t)$ denote the total fuel consumption and electricity consumption. Considering $SOC(k)$ as the state variable x_k and the motor torque $T_m(k)$ as the control input u_k, the state transition function can be formulated, as:

$$SOC(k+1) = SOC(k) + \frac{I\eta\Delta t}{3600Q} \tag{14}$$

Then, DP can be applied to solve the recursive function, as:

$$J_{k,j}^{*}(x_{k,j}) = \min_{u_{k,j,m}} \left\{ L_k(x_{k,j}, u_{k,j,m}) + J_{k+1,j}^{*}(g(x_{k,j}, u_{k,j,m})) \right\} \tag{15}$$

where $x_{k,j}$ is the j_{th} state value at step k, $u_{k,j,m}$ indicates the m_{th} decision variable of the j_{th} state, g denote the state transition function, and $L_k(x_{k,j}, u_{k,j,m})$ expresses the instantaneous function value.

To investigate influences brought by different prediction algorithms, a two-layer based EMS is built including the prediction layer and strategy layer, as shown in Fig. 14. In the prediction layer, the vehicle speed is predicted based on different algorithms. To evaluate and compare the prediction performance, the Markov model, BP-NN model, RBF-NN model and the proposed fusion model are respectively implemented to forecast the future driving speed profile, which is sent to the control layer in each step. In the decision layer, firstly, the reference SOC trajectory is determined according to the historical transportation information. In this study, a linear SOC reference curve is generated with respect to the driving distance for simplicity. In the receding horizon of MPC application, we employ the DP to solve the cost function, thereby finding the optimal control variable. Finally, the energy cost by different prediction algorithms is compared and discussed.

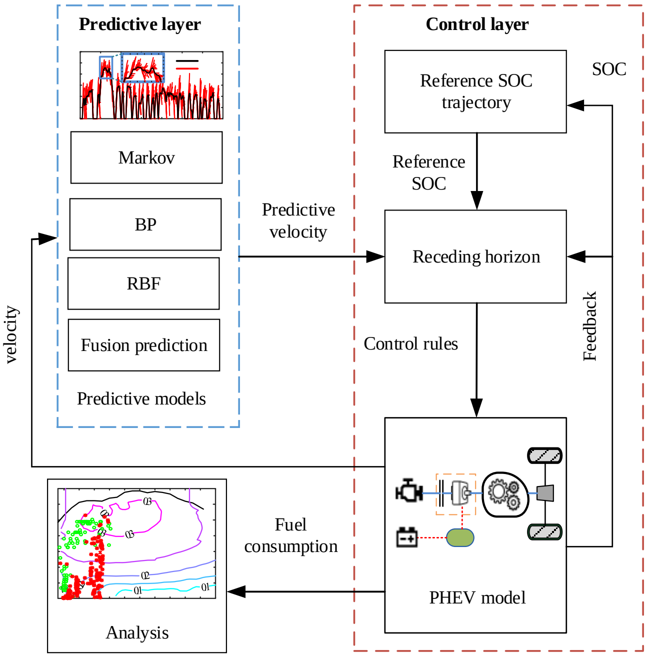


Fig. 14. Energy consumption analysis strategy

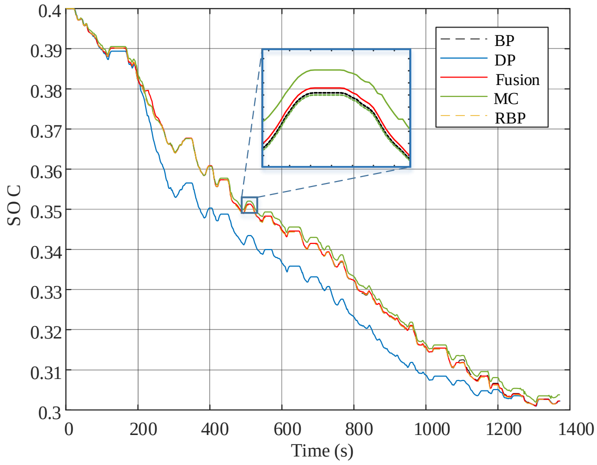


Fig. 15. Simulation results of SOC trajectories

To verify the fuel economy improvement incurred by different velocity prediction algorithms, the UDDS cycle is simulated. The corresponding SOC curves based on different prediction algorithms are shown in Fig. 15. As can be summarized, all the curves can converge to the low threshold when the trip ends, thereby manifesting the feasibility of MPC algorithm. Besides the curve regulated by the DP, all the other remaining curves remain almost the same. The engine operating points are plotted in Fig. 16. It can be easily found that there exists obvious difference in the working points between the two methods. The DP enables the engine to operate in more concentrate region, compared with the MPC algorithm. Definitely, DP can achieve the minimum cost, i.e., CNY 1.7, because of its global offline optimization capability. Table VIII lists the total cost in terms of different velocity prediction methods. Note that the fuel economy optimality is calculated through dividing the global DP result by the current cost based on the MPC algorithm with the specific speed predictor. Among all the MPC methods, the fusion velocity prediction algorithm leads to the best results: 91.18%, which highlights that the proposed prediction algorithm also has a significant effect on improvement of PHEV operation economy, thus proving the contribution of more precise speed prediction by the proposed fusion algorithm.

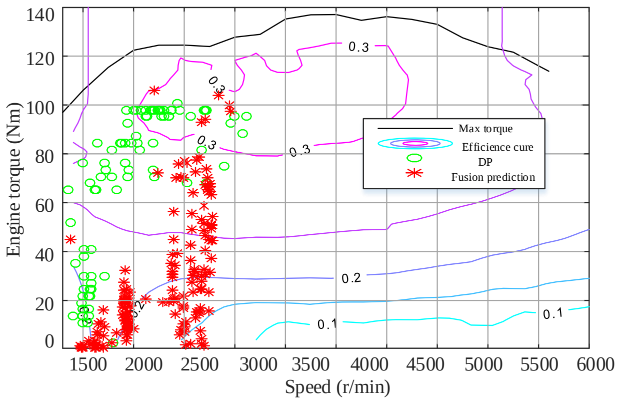


Fig. 16. Simulation results of operation points of engine

Table VIII Energy consumption under different forecasting models

Prediction models	MC	BP	RBF	Fusion
Energy cost (CNY)	1.99	1.89	1.88	1.85
Fuel economy optimality (%)	82.94	88.82	89.41	91.18

V. CONCLUSION

In this study, a novel prediction method of short-term driving condition is developed, which capitalizes the advantages of both Markov chain and artificial neural network. The proposed prediction model includes two main parts: a driving condition prediction module and a nonlinear fusion module. In the driving condition prediction module, one neural network and Markov chain are applied for primary velocity prediction. To avoid over-training of neural network, the principal component analysis is conducted to simplify the dimension of inputs. In the nonlinear fusion module, the primary prediction results are fused by another neural network to output the final prediction. The simulation results highlight that the prediction accuracy of the proposed fusion model with the parameters optimized by genetic algorithm respectively increases by 45.32%, 39.13%, 40.56%, and 9.75%, compared with those by the traditional Markov chain model, back propagation neural network, radial basis function neural network, and the fusion prediction model without optimization. In addition, a case study applied in energy management of plug-in hybrid electric vehicles verifies that the proposed prediction method can contribute more to the improvement of vehicle's operation economy, compared with other conventional velocity prediction algorithms.

Our next step work will be focused on applying the proposed fusion prediction method to further improve the fuel economy of hybrid electric vehicles. The actual hardware-in-the-loop and real vehicle validation will also be conducted in the future.

ACKNOWLEDGMENTS

The work presented in this paper is funded by the National Natural Science Foundation (No. 51775063, No. 61763021 and U1764259) in part, the EU-funded Marie Skłodowska-Curie Individual Fellowships Project under Grant 845102-HOEMEV-H2020-MSCA-IF-2018 in part, the Science Foundation of Chongqing University of Science and Technology (No. CK2017ZKYB023) in part, and the Science Foundation of Mechanical and Power Engineering of Chongqing University of Science and Technology (No. JX2018A01) in part. Any opinions expressed in this paper are solely those of the authors and do not represent those of the sponsors.

REFERENCES

[1] J. Peng, H. He, and R. Xiong, "Rule based energy management strategy for a series–parallel plug-in hybrid electric bus optimized by dynamic programming," *Applied Energy,* vol. 185, pp. 1633-1643, 2017.

[2] Y. Liu, J. Li, Z. Chen, D. Qin, and Y. Zhang, "Research on a multi-objective hierarchical prediction energy management strategy for range extended fuel cell vehicles," *Journal of Power Sources,* vol. 429, pp. 55-66, 2019.

[3] R. Xiong, Y. Duan, J. Cao, and Q. Yu, "Battery and ultracapacitor in-the-loop approach to validate a real-time power management method for an all-climate electric vehicle," *Applied energy,* vol. 217, pp. 153-165, 2018.

[4] Z. Lei, D. Qin, P. Zhao, J. Li, Y. Liu, and Z. Chen, "A real-time blended energy management strategy of plug-in hybrid electric vehicles considering driving conditions," *Journal of Cleaner Production,* vol. 252, p. 119735, 2020.

[5] Z. Chen, C. C. Mi, R. Xiong, J. Xu, and C. You, "Energy management of a power-split plug-in hybrid electric vehicle based on genetic algorithm and quadratic programming," *Journal of Power Sources,* vol. 248, pp. 416-426, 2014.

[6] X. Hu, T. Liu, X. Qi, and M. Barth, "Reinforcement Learning for Hybrid and Plug-In Hybrid Electric Vehicle Energy Management: Recent Advances and Prospects," *IEEE Industrial Electronics Magazine,* vol. 13, no. 3, pp. 16-25, 2019.

[7] S. Xie, X. Hu, Z. Xin, and J. Brighton, "Pontryagin's minimum principle based model predictive control of energy management for a plug-in hybrid electric bus," *Applied energy,* vol. 236, pp. 893-905, 2019.

[8] S. Zhang, R. Xiong, and F. Sun, "Model predictive control for power management in a plug-in hybrid electric vehicle with a hybrid energy storage system," *Applied Energy,* vol. 185, pp. 1654-1662, 2017.

[9] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji, "Model predictive control power management strategies for HEVs: A review," *Journal of Power Sources,* vol. 341, pp. 91-106, 2017.

[10] A. Feng, L. Qian, and Y. Huang, "RFID Data-Driven Vehicle Speed Prediction Using Adaptive Kalman Filter," in *International Conference on Machine Learning and Intelligent Communications*, 2018, pp. 43-51: Springer.

[11] H. A. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, and I. V. Kolmanovsky, "Predictive energy management of a power-split hybrid electric vehicle," in *2009 American control conference*, 2009, pp. 3970-3976: IEEE.

[12] M. H. Amini, A. Kargarian, and O. Karabasoglu, "ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation," *Electric Power Systems Research,* vol. 140, pp. 378-390, 2016.

[13] R. Xiong, Y. Zhang, H. He, X. Zhou, and M. G. Pecht, "A double-scale, particle-filtering, energy state prediction algorithm for lithium-ion batteries," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 2, pp. 1526-1538, 2017.

[14] S. J. Moura, H. K. Fathy, D. S. Callaway, and J. L. Stein, "A stochastic optimal control approach for power management in plug-in hybrid electric vehicles," *IEEE Transactions on control systems technology,* vol. 19, no. 3, pp. 545-555, 2010.

[15] Y. L. Murphey, J. Park, Z. Chen, M. L. Kuang, M. A. Masrur, and A. M. Phillips, "Intelligent hybrid vehicle power control—Part I: Machine learning of optimal vehicle power," *IEEE Transactions on Vehicular Technology,* vol. 61, no. 8, pp. 3519-3530, 2012.

[16] T. Gaikwad *et al.*, "Vehicle Velocity Prediction Using Artificial Neural Network and Effect of Real World Signals on Prediction Window," SAE Technical Paper0148-7191, 2020.

[17] H. Liu, X. Li, W. Wang, L. Han, and C. Xiang, "Markov velocity predictor and radial basis function neural network-based real-time energy management strategy for plug-in hybrid electric vehicles," *Energy,* vol. 152, pp. 427-444, 2018.

[18] T. Liu, B. Wang, and C. Yang, "Online Markov Chain-based energy management for a hybrid tracked vehicle with speedy Q-learning," *Energy,* vol. 160, pp. 544-555, 2018.

[19] B. Liu, L. Li, X. Wang, and S. Cheng, "Hybrid electric vehicle downshifting strategy based on stochastic dynamic programming during regenerative braking process," *IEEE Transactions on Vehicular Technology,* vol. 67, no. 6, pp. 4716-4727, 2018.

[20] P. Shen, Z. Zhao, X. Zhan, J. Li, and Q. Guo, "Optimal energy management strategy for a plug-in hybrid electric commercial vehicle based on velocity prediction," *Energy,* vol. 155, pp. 838-852, 2018.

[21] R. Xiong, J. Cao, and Q. Yu, "Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle," *Applied energy,* vol. 211, pp. 538-548, 2018.

[22] J. Hu, J. Song, G. Yu, and Y. Zhang, "A novel networked traffic parameter forecasting method based on Markov chain model," in *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, 2003, vol. 4, pp. 3595-3600: IEEE.

[23] G. Ripaccioli, D. Bernardini, S. Di Cairano, A. Bemporad, and I. Kolmanovsky, "A stochastic model predictive control approach for series hybrid electric vehicle power management," in *Proceedings of the 2010 American Control Conference*, 2010, pp. 5844-5849: IEEE.

[24] D. Shen, C.-C. Lim, P. Shi, and P. Bujlo, "Energy management of fuel cell hybrid vehicle based on partially observable Markov decision process," *IEEE Transactions on Control Systems Technology,* 2018.

[25] Y. Zhou, A. Ravey, and M.-C. Péra, "A survey on driving prediction techniques for predictive energy management of plug-in hybrid electric vehicles," *Journal of Power Sources,* vol. 412, pp. 480-495, 2019.

[26] G. Jinquan, H. Hongwen, P. Jiankun, and Z. Nana, "A novel MPC-based adaptive energy management strategy in plug-in hybrid electric vehicles," *Energy,* vol. 175, pp. 378-392, 2019.

[27] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *European Journal of Operational Research,* vol. 131, no. 2, pp. 253-261, 2001.

[28] C. Sun, X. Hu, S. J. Moura, and F. Sun, "Velocity predictors for predictive energy management in hybrid electric vehicles," *IEEE Transactions on Control Systems Technology,* vol. 23, no. 3, pp. 1197-1204, 2014.

[29] W. Gu, D. Zhao, and B. Mason, "A review of intelligent road preview methods for energy management of hybrid vehicles," *IFAC-PapersOnLine,* vol. 52, no. 5, pp. 654-660, 2019.

[30] C. Yin, L. Rosendahl, and Z. Luo, "Methods to improve prediction performance of ANN models," *Simulation Modelling Practice and Theory,* vol. 11, no. 3-4, pp. 211-222, 2003.

[31] S. Xie, H. He, and J. Peng, "An energy management strategy based on stochastic model predictive control for plug-in hybrid electric buses," *Applied energy,* vol. 196, pp. 279-288, 2017.

[32] D. Shen, C.-C. Lim, and P. Shi, "Predictive Modeling and Control of Energy Demand for Hybrid Electric Vehicle Systems," in *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2019, pp. 1-6: IEEE.

[33] G. HWANG *et al.*, "Prediction of Hybrid Electric Bus Speed for Energy Management Using Deep Learning Method," SAE Technical Paper0148-7191, 2020.

[34] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transport reviews,* vol. 24, no. 5, pp. 533-557, 2004.

[35] S. Wang, N. Zhang, L. Wu, and Y. Wang, "Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method," *Renewable Energy,* Article vol. 94, pp. 629-636, 2016.

[36] E. Ericsson, "Independent driving pattern factors and their influence on fuel-use and exhaust emission factors," *Transportation Research Part D: Transport and Environment,* vol. 6, no. 5, pp. 325-345, 2001.

[37] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems,* vol. 2, no. 1-3, pp. 37-52, 1987.

[38] C. Xiang, F. Ding, W. Wang, and W. He, "Energy management of a dual-mode power-split hybrid electric vehicle based on velocity prediction and nonlinear model predictive control," *Applied Energy,* Article vol. 189, pp. 640-653, 2017.

[39] F. H.-F. Leung, H.-K. Lam, S.-H. Ling, and P. K.-S. Tam, "Tuning of the structure and parameters of a neural network using an improved genetic algorithm," *IEEE Transactions on Neural networks,* vol. 14, no. 1, pp. 79-88, 2003.

[40] J. Bai and S. Ng, "Tests for skewness, kurtosis, and normality for time series data," *Journal of Business & Economic Statistics,* vol. 23, no. 1, pp. 49-60, 2005.