A Repeated Commuting Driving Cycle Dataset With Application to Short-Term Vehicle Velocity Forecasting

Yuanzhi Liu
Department of Mechanical Engineering,
The University of Texas at Dallas,
Richardson, TX 75080
e-mail: yuanzhi.liu@utdallas.edu

Jie Zhang[1]
Department of Mechanical Engineering;
Department of Electrical and Computer Engineering,
The University of Texas at Dallas,
Richardson, TX 75080
e-mail: jiezhang@utdallas.edu

Vehicle velocity forecasting plays a critical role in operation scheduling of varying systems and devices for a passenger vehicle. The forecasted information serves as an indispensable prerequisite for vehicle energy management via predictive control algorithms or vehicle ecosystem control Co-design. This paper first generates a repeated urban driving cycle dataset at a fixed route in the Dallas area, aiming to simulate a daily commuting route and serves as a base for further energy management study. To explore the dynamic properties, these driving cycles are piecewise divided into cycle segments via intersection/stop identification. A vehicle velocity forecasting model pool is then developed for each segment, including the hidden Markov chain model, long short-term memory network, artificial neural network, support vector regression, and similarity methods. To further improve the forecasting performance, higher-level algorithms like localized model selection, ensemble approaches, and a combination of them are investigated and compared. Results show that (i) the segment-based forecast improves the forecasting accuracy by up to 20.1%, compared to the whole cycle-based forecast, and (ii) the hybrid localized model framework that combines dynamic model selection and an ensemble approach could further improve the accuracy by 9.7%. Moreover, the potential of leveraging the stopping location at an intersection to estimate the waiting time is also evaluated in this study.
[DOI: 10.1115/1.4052996]

Keywords: repeated driving cycle dataset, cycle segment, vehicle velocity forecasting, model selection, ensemble method, intersection waiting time

1 Introduction

Short-term traffic forecasting has been extensively investigated in the past decade as a potential feasible solution to mitigate the growing concern of traffic congestion, especially with the advent of connected and automated vehicles (CAV), big data, artificial intelligence, and Internet-of-things [1,2]. It has been shown that the traffic can be significantly improved by integrating the existing road network with smart traffic light control and intelligent transportation systems, based on the real-time traffic measuring and forecasting, such as the traffic flow volume, traffic density, average traffic velocity, and travel time [3].

1.1 Network Traffic Forecasting. The majority of these studies usually focus on a broader scope of traffic forecasting, from point-level, street-level, to network-level, serving to provide further insights for transportation management and policy making. At the very-beginning stage, several statistic and machine learning-based algorithms were exploited for varying traffic conditions. For instance, as a statistical analysis model, the auto-regressive integrated moving average (ARIMA) method has been broadly adopted for point-level or street-level traffic forecasting [4]. ARIMA and its families were usually adopted as a benchmark to be compared with other advanced approaches, such as machine learning-based methods (e.g., support vector regression (SVR) [5], back propagation artificial neural network (BPANN) [6], and radial basis function neural network (RBFNN)), probability-based methods (e.g., Kalman filter and hidden Markov chain (HMM) [7]), and deep learning-assisted methods (e.g., long short-term memory network (LSTM) [8,9]). Among these reported methods, the probability-based Kalman filter and HMM methods produced stochastic forecasts, while others produced deterministic forecasts. Historical driving records are usually considered as the training dataset for forecasting. Though only the temporal relationship is taken into account, the aforementioned algorithms are capable of predicting the traffic situations in most cases with reasonable accuracy. It is also worth noting that there is no published literature that indicates the superiority of any of these algorithms, due to diverse traffic data sources.

However, some of these forecasting algorithms may not perform well for a large area with complicated transportation networks and uncertain environmental factors. To alleviate this arising challenge, advanced deep neural networks with topological feature embedding have been employed to characterize the spatial-temporal characteristics in forecasting. Massive efforts have been performed on convolutional neural network (CNN)-based algorithms [10], which generally falls into two categories: convolution-based LSTM that integrates CNN and LSTM [11,12], and temporal graph convolutional networks that combine graph neural network with gated recurrent units/networks [13–15]. By comparing with the reported forecasting performance, these deep learning-based methods have shown overwhelming superiority over the aforementioned classic parametric or simple-structured machine learning-based approaches. This newly emerging trend is very likely to continue in traffic forecasting [16].

1.2 Individual Vehicle Velocity Forecasting. It should be noted that for the aforementioned studies, all the data have been collected from a single or a series of fixed observation locations using sensors like inductive-loop detector, wireless magnetometer, microwave radar, and video image processor. These studies emphasize

[1]Corresponding author.

Manuscript received June 11, 2021; final manuscript received November 8, 2021; published online December 2, 2021. Assoc. Editor: Marcello Canova.

more on the networked vehicles rather than an individual passenger vehicle. However, velocity forecasting for individual vehicles has drawn significant attention in the past decade, especially along with the fast-growing demand for vehicle electrification. Velocity forecasting plays a critical role in improving the energy efficiency for electric or hybrid vehicles. Generally, velocity forecasting serves as the system input for a model predictive control-assisted or reinforced learning-based energy management system to optimize the charging/discharging schedule, the regenerative power harvest, and the operation of an on-board air-conditioning system, especially for repeated fixed routes inside or between cities [17–20]. Additionally, forecasted velocity is also regarded as an indispensable prerequisite to generate varying scenarios and networked/individual vehicle ecosystem for multidisciplinary control Co-design [21–23].

There are generally three major discrepancies that the individual vehicle velocity forecasting differs from the network traffic forecasting. First, individual vehicle velocity forecasting utilizes the floating velocity trajectory as the data source instead of the network traffic records. Second, individual vehicle velocity forecasting requires a significantly shorter prediction horizon at seconds, compared to the network traffic forecasting at minutes or hours timescales. Third, the networked traffic forecasting can be facilitated by local/cloud-based powerful computing tools with sophisticated deep learning-based structures; while the velocity forecasting for individual vehicles tends to directly utilize on-board computing devices, leaving no alternative but to implement computationally efficient forecasting algorithms.

The traffic forecasting algorithms discussed above are still applicable and practicable to individual vehicle velocity forecasting, which can generally be categorized into stochastic and deterministic approaches. For instance, as one of the most popularly used stochastic methods, HMM was modified by Jing et al. [24] with a fuzzy logistic model to predict individual vehicle speed 8 s ahead, and Zhou et al. [25] developed a self-learning multi-step Markov chain model based on simulated data. However, stochastic methods are usually eclipsed by deterministic approaches regarding prediction accuracy. For example, Sun et al. [26] revealed that RBFNN and ANN performed significantly better than HMM. Liu et al. [27] reported similar results that both LSTM and ARIMA outperformed HMM in 10 s ahead speed forecasting based on a real urban driving dataset. Moreover, among all the deterministic methods, it seems that LSTM possesses prevailing advantages for a same driving dataset. For instance, a comparative study between LSTM and other deep learning-based algorithms like CNN and CNN-LSTM conducted by Rabinowitz et al. [28] revealed that LSTM dominates other deep learning-based and machine learning-based forecasting technologies with a considerable higher accuracy. The feasibility of embedding LSTM on board has also been tested and verified by Gaikwad et al. [29] with an on-board processor. Besides, other deep learning networks such as deep belief network and stacked auto-encoder have also been investigated for speed forecasting for a highway speed dataset [30].

Acting as an indispensable element, individual vehicle velocity forecasting has been deeply integrated with the connected and automated system by leveraging vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technologies. It is worth noting that the forecasting performance can be further improved by considering the surrounding traffic situations via vehicle-to-everything communications. For example, Moser et al. [31] and Zhang et al. [32] proved that individual vehicle velocity forecasting could yield more accurate predictions by knowing the states of traffic lights in advance. Upon perceiving the traffic conditions of the local area, velocity forecasting will become a dynamic process by choosing an optimal eco-routing [33]. There is also a growing trend to predict the vehicle behaviors at intersections including acceleration and deceleration, aiming to achieve optimal velocity trajectory for energy control [34].

The electric vehicle industry has witnessed huge advancements in embedding with advanced driver assistance systems (ADAS) and predictive optimal energy management strategies (POEMSs). However, one of the challenges is that the policy-making and construction of intelligent transportation system are falling far behind the electrification and intellectualization of passenger vehicles. It is still economically and technologically prohibitive to enable all passenger vehicles being connected to the intelligent transportation system and receive real-time traffic information. Based on the existing urban transportation infrastructure and vehicle installations, how to improve the velocity forecasting accuracy is still a stringent and challenging problem for vehicle energy management.

1.3 Research Objective. To further improve the performance of individual vehicle velocity forecasting, a hybrid velocity forecasting algorithm is developed in this study, by leveraging the fact that most of the newly released electric or hybrid vehicles are equipped with on-board GPS devices. To validate the proposed algorithm, a repeated urban driving cycle dataset is first generated by the same driver (with the same driving habits) in the area of Dallas, TX. The driving patterns between weekdays and weekends are investigated, and road segments are also identified. A forecasting pool that consists of HMM, LSTM, ANN, SVR, and similarity methods is established, and a localized model selection and ensemble framework is developed to dynamically choose the appropriate forecasting models for each road segment. This study seeks to enhance the forecasting accuracy with the currently available urban transportation infrastructure, and our contributions are threefold: (i) generate a publicly available commuting dataset with repeated driving cycles for energy management and control co-design, (ii) develop a segment-based vehicle speed forecasting model, in conjunction with a localized model selection and ensemble framework; (iii) explore the feasibility of leveraging the stopping position at an intersection for estimating the waiting time and thus improving the forecasting accuracy.

The remainder of the paper is organized as follows. First, the driving cycle dataset, pattern recognition, and road segments are analyzed in Sec. 2. The developed segment-based forecasting algorithms including the stochastic and deterministic approaches are applied to the dataset in Sec. 3. A comparative study among the dynamic model selection, the ensemble approach, and the combined hybrid method is conducted in Sec. 4. The future research insights of leveraging image detection and piecewise traffic network are discussed in Sec. 5. Concluding remarks are summarized in Sec. 6.

2 Data Collection and Analyses

2.1 Data Collection. There exist several repeated driving cycles based on a fixed route published in the literature, such as the Connected Ann Arbor (A2) dataset [27], the Gothenburg driving dataset [35], the CSU driving cycle [28], the MTU driving cycle [36], and the Fort Collins repeated driving cycle dataset [29]. However, few datasets are publicly available at the current stage. Another piratical way to obtain repeated driving cycles is to extract the repeated routes from large-scale traffic or vehicle energy consumption dataset that covers a certain period of time, such as the Geolife Trajectories,[2] the vehicle energy dataset (VED) [37],[3] the performance measurement system (PeMS) dataset,[4] the Roma taxi dataset,[5] and the San Francisco bay area taxi dataset.[6] These datasets are readily available online, but one of the prominent drawbacks is that the extracted cycles are usually inconsecutive and scattered with a very short period of time, making it challenging to be integrated with any predictive control implementation. Moreover, a large dataset with repeated routes is also beneficial to route planning and decision-making for

[2]https://www.microsoft.com/en-us/download/details.aspx?id=52367
[3]https://github.com/gsoh/VED/blob/master/README.md
[4]https://archive.ics.uci.edu/ml/datasets/PEMS-SF
[5]https://crawdad.org/roma/taxi/20140717/
[6]https://crawdad.org/epfl/mobility/20090224/cab/

autonomous driving development. By capturing the surrounding environment and analyzing dynamic traffic conditions via image recognition, the autonomous driving system with improved confidence can be potentially advanced to a higher level regarding reliability and safety for a specific repeated route in a limited area [38].

To better understand the impacts of vehicle speed forecasting on energy management, we have generated a Dallas repeated driving cycle (DRD) dataset.[7] In this dataset, dozens of driving cycle tests have been performed on a fixed route in the Dallas area to simulate a typical commuting route for passenger vehicles, which consists of an expressway test of 5 km and a local urban road test of 20 km, as shown in Fig. 1. The dataset was acquired between Dec. 2020 and Jan. 2021 at around 4:45 PM to 6:00 PM, in which each cycle takes approximately 30 min using a conventional internal combustion engine passenger vehicle. All the testing cycles were conducted with a fixed route by the same driver with similar driving manners. It needs to be noted that some of the traffic signals in this area are adaptively controlled in a daily or continuous real-time manner, according to the progress report of a regional traffic retiming program [39]. More repeated commuting cycles by different drivers are expected to be included in the DRD dataset in the coming future.

Several critical dynamic driving parameters and indices are recorded using a GPS logger (brand: Garmin, model: eTrex 10) and a camera. All the data are time-stamped with a time interval of 1 s for the sake of accuracy and consistency, including the vehicle velocity, altitude and longitude information, elevation, heading direction, and traffic light picture/video. Compared with other datasets, one of the remarkable merits of this DRD dataset is that it covers a broader set of road types with tens of intersections, and traffic light images can be used as a potential tool for intersection waiting time prediction.

2.2 Data Processing. As a preprocessing step, clustering plays an important role in improving forecasting accuracy. For velocity forecasting using historical data, there are two general approaches to cluster the data and identify major spatial-temporal patterns: (i) cluster the cycles for pattern recognition, split the grouped cycles into segments, perform another round of clustering for the segments, and then establish the forecasting models; and (ii) skip the first-round clustering and follow the rest steps [40,41]. Note that unsupervised clustering can be transformed into supervised classification by manually labeling the data with expert knowledge, e.g., the cycles can be classified directly into weekday and weekend/holiday conditions based on daily driving experiences.

To quantify the traffic discrepancies between weekdays and weekends, a congestion index ε_{it} (of a location i at time t) is modified here by comparing the measured floating vehicle velocity with the free-flow velocity [42], as defined in Eq. (1).

$$\varepsilon_{it} = \frac{s_{it}}{v_{it}} - 1 \tag{1}$$

where the parameter S_{it} represents the free-flow velocity defined by the maximum value recorded, while v_{it} represents the current vehicle velocity. A larger index ε_{it} indicates relatively more severe traffic congestion. Here, a total of 18 locations away from the intersections are selected randomly using the stratified sampling algorithm, as shown in Fig. 1. The discrepancies between weekdays and weekends/holidays are illustrated in Fig. 2. It is seen that there do not exist many significant differences between weekdays and weekends/holidays, except for the saturated section ranging from location 2 to 5 on the upper-right corner, which illustrates an improvement of the highway traffic on weekends and holidays. The possible reason behind this phenomenon is the current remote-work environment due to the COVID-19 pandemic. It is also noticed that the whole trip takes an averaged time of 1801 s on weekends and holidays, which is 193 s shorter on weekdays. By investigating the details of route segments, it is observed that moving through the intersections on weekdays requires extra time, as shown in Fig. 3.

2.3 Intersection/Stop Identification. Given the preceding analysis, the second clustering approach discussed in Sec. 2.2 (i.e., split the grouped cycles into segments, perform another round of clustering for the segments, and then establish the forecasting models) is employed here for feature identification in the study. The driving cycles are directly divided into segments, followed by a time sequence clustering of the segments if necessary. Specifically, all the intersections are extracted from the route as separated segments because of their significant impacts on the whole driving time. A location is identified as an intersection or a T-junction with stop/yield signs if it is detected with a complete stop or low velocity (i.e., 10 km/h) more than twice. Once the intersections are located, the routes in between will also be defined as independent cycle segments. It is worth mentioning that there are two major reasons why traffic signal identification is necessary rather than using the labeled data from public map sources directly: (i) the vehicles have a high probability of moving through some labeled intersections or T-junctions without any interruption and (ii) the vehicles need to wait for two rounds at some traffic lights due to the heavy traffic conditions, resulting in another indirect hidden stop.

As can be seen from Fig. 4, the final location of a vehicle is scattered at an intersection, depending on the traffic volume and its arrival time. The furthest point downstream among all is treated as the location of the intersection. The primary step here is to organize the stop points that belong to the same intersection or potential stop into a group. General density or centroid-based machine learning-based clustering algorithms like K-means have been tested however with unsatisfied performance, since it is challenging to determine the appropriate number of clusters. In this study, we attempt to cluster the stop points using a connectivity-based single linkage hierarchical clustering algorithm, in which only connections with a coordinate distance smaller than the maximum threshold will be considered to form a same group, i.e., 40 m for this DRD dataset. Given the specific velocity profile, the single linkage approach has a great superiority regarding the accuracy

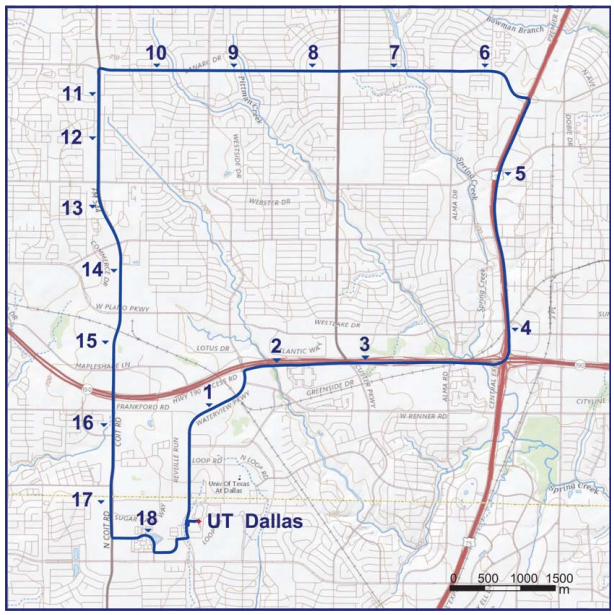

Fig. 1 The testing route of DRD repeated driving cycles, which is close to the University of Texas at Dallas (the numbers indicate the positions for traffic congestion analyses)

[7] https://github.com/UTD-DOES/Dallas-Repeated-Driving-Cycle-Dataset

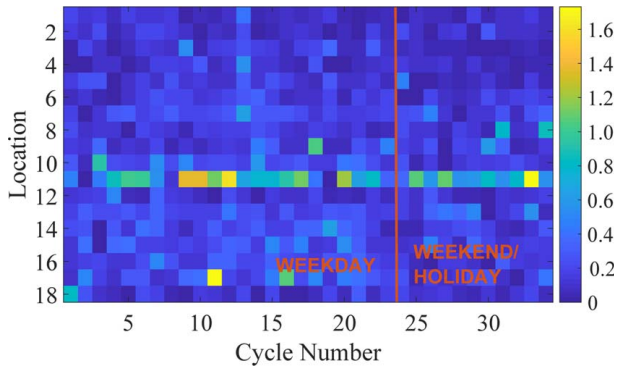

Fig. 2 A comparison of the congestion index between weekdays and weekends/holidays

over other clustering methods for a relatively small dataset, as it determines a group in a more straight-forward manner by merely using a hard distance threshold. The clustering results are also highlighted in Fig. 4. A group consisting of fewer than three points is ignored and will not be treated as an intersection or a stop, given a low stopping probability of 5.8% (2/34).

However, the computational complexity may increase considerably when applying the method to a large repeated driving cycle dataset. Given its temporal and spatial nature, a semi-supervised iterative approach is employed aiming to improve the computation efficiency. The clustering process begins with a single driving cycle, which is then clustered using a maximum distance threshold as discussed above. By adding new cycles, the new clustering outcomes get updated iteratively using a divide and conquer method. The proposed method seeks to avoid unnecessary coordinate distance calculations, as illustrated in Fig. 5.

2.4 Road Segment. To better model the velocity trajectory passing through an intersection, besides the final stopping location, it is also crucial to identify the deceleration and reacceleration processes and divide the routes into varying segments, as illustrated in Fig. 6. An intersection segment consists of a deceleration, a waiting, and a reacceleration process, while a normal road segment refers to a continuous move at a steady speed. When it comes to the local street with a lower speed limit and a smaller traffic volume, i.e., a stop sign, the deceleration and acceleration prepossess are captured with a very similar pattern, i.e., same stopping locations and almost equal waiting time.

Following the principle discussed earlier, the whole cycle is divided into 42 segments using location coordinates, and a portion of the segments are presented in Figs. 7 and 8. It is seen from the figures that within a same road segment, the velocity

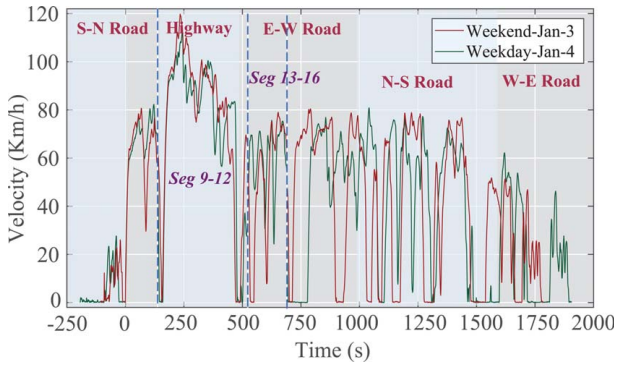

Fig. 3 The velocity trajectories of typical driving cycles on weekdays and weekends/holidays (the location annotations are based on the weekday cycle)

patterns may still differ, especially for intersection segments. However, it is observed from Fig. 8 that the trends of velocity trajectories versus location for intersection segments are more uniformed than that versus time in Fig. 7. The possible discrepancies are mainly due to the waiting time for traffic lights. Thus, no further clustering is implemented in this study. We also assume that the final stopping location may have considerable influences on the waiting time at an intersection, which is crucial to vehicle energy management and will be further investigated in the following section.

3 Base Forecasting Methods and Results

From the perspective of energy management with a specific driving route, it is expected that accurate traffic forecasting, including the averaged velocity, the deceleration and acceleration processes could significantly enhance the efficiency via energy scheduling and planning. In this study, we construct a forecasting model pool that consists of a collection of stochastic and deterministic methods based on their popularity and performance, i.e., LSTM, HMM, ANN, SVR, and a similarity-based method, with varying kernels, training algorithms, and hyper parameters. However, due to a very short prediction and action time interval (e.g., 5 s for most of the predictive energy management implementations [17,18,23,43]), it remains challenging to include all the submodels simultaneously for short-term velocity forecasting. The best subset of models are preselected in the training-validation stage using two evaluation metrics, i.e., the mean absolute error (MAE) and the root-mean-square error (RMSE), expressed as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{3}$$

where $\hat{y}_i$ and y_i are the forecasted and actual value of sample index i, respectively.

Via the model preselection, only five models are employed to forecast the velocity for each road segment. In this process, a total of 23 randomly selected driving cycles (i.e., renamed as Cycle 1-23) are utilized for training, and another five cycles (i.e., Cycle 24-28) are used for validation.

3.1 Stochastic Approach. Hidden Markov chain is a discrete-time stochastic memoryless process to model a sequence of events, in which the future state or action only depends on the current state. The chaining process is characterized by a set of implicit hidden states S and its transition probabilities matrix A, and each a_{ij} represents the probability of moving from a state i to a new state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1, \ \forall i$, which is also referred to as the Markov assumption, expressed as follows:

$$P(s_i|s_{i-1}, s_{i-2}, \cdot, \cdot, \cdot, s_1) = P(s_i|s_{i-1}) \tag{4}$$

Another fundamental assumption of HMM is that the explicit observation O only relies on the state that generates the observation with a probability of $B = [b_j(O_t)]$. Starting with an initial probability distribution over states π, an HMM process can be modeled as $\lambda = (A, B, \pi)$.

To apply HMM for velocity prediction, it is required to transform the historical speed records into a set of observations indexed by integers. If the numbers of states and observations are known and set to be equal, the probability matrix A and B can be solved by calculating the frequency counts of a labeled state transition among all the transitions or a specific observation among the observations. In this study, the DRD dataset contains a very limited number of data

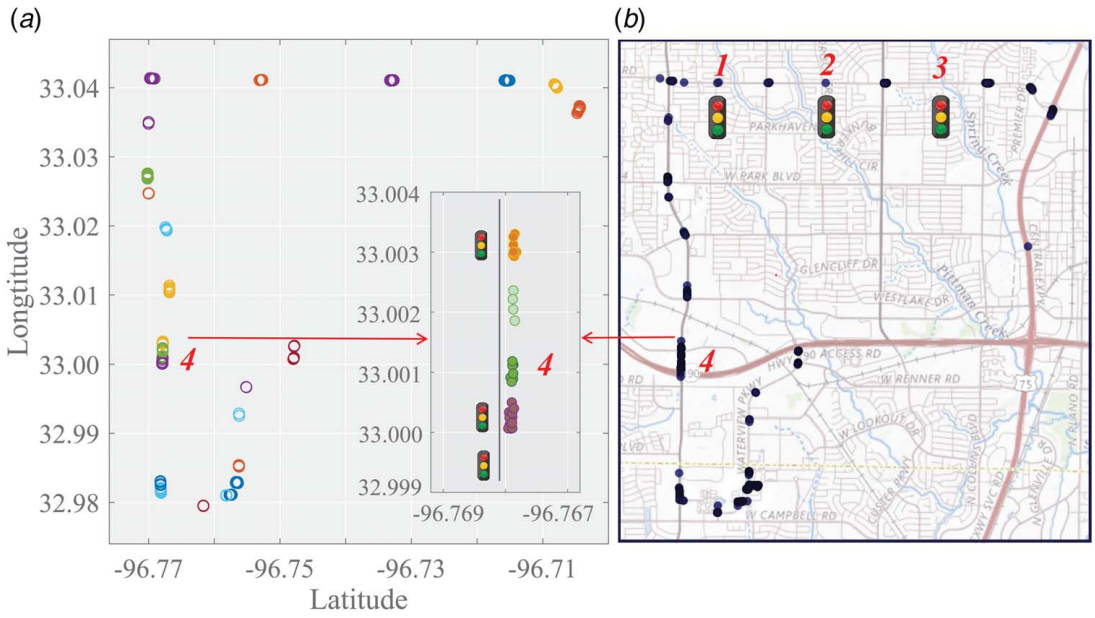

Fig. 4 Intersection detection using the velocity profile and its location on a map: (*a*) the locations of intersections after identification. The onward route before the first stop sign has been removed, so has the return route. (*b*) Annotations 1 and 2 illustrate that these two intersections are equipped with traffic lights but will be ignored as a normal straight road due to a low stopping probability. Annotation 3 shows no stops have been ever detected even though there is a traffic light. Annotation 4 indicates that vehicles are very likely to stop moving somewhere between two intersections because of heavy traffic, which may be regarded as a stop.

points, posing challenges in directly solving this problem, as the matrix might be singular.

To enhance the accuracy, two data augmentation techniques for time series are leveraged to expand the existing dataset, including the weighted dynamic time warping barycenter averaging (w-DBA) and Gaussian noise injection. The w-DBA is accomplished via assigning varying weight factors to a set of similar temporal sequences measured using the dynamic time warping (DTW) [44], as given by:

$$T_{new}(n) = \omega_b T_{base}(n) + \sum_{i=1}^{4} \omega_i \mathbb{D}[T_{base}(n), T_i(\overline{m})] \tag{5}$$

where T_{base} and T_i are the basic sequence to be augmented by its selected i_{th} sequence with similarity. $[\omega_b, \omega_i]$ denotes the assigned weight factor consistently with the DTW normalized distances, s.t. $\sum \omega_j = 1$. Here, it is predefined as [0.4, 0.18, 0.16, 0.14, 0.12] through trials for simplification. $\mathbb{D}[\cdot]$ represents the DTW

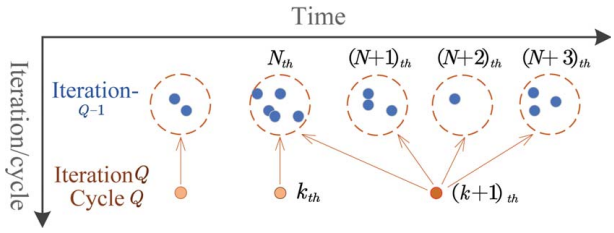

Fig. 5 The sketch of an iterative clustering method for a large dataset. (Detailed processes: after clustering the k_{th} point in *Cycle* $-$ *Q* into the existing N_{th} group, the $(k+1)_{th}$ point only needs to calculate its distance with groups starting from N_{th}. Once it is clustered, i.e., into the $(N+i)_{th}$ group, an additional distance with the $(N+i+1)_{th}$ group needs to be conducted to confirm that the new point does not belong to the next group, otherwise, the $(N+i)_{th}$ group will be combined with the $(N+i+1)_{th}$ group to form another group.)

alignments between two time sequences, i.e., the n_{th} element of the base sequence may align with the m_{th} element or several continuous $m_{th} - p_{th}$ elements of a similar sequence, where the value of $\mathbb{D}[\cdot]$ equals to $T_i(m)$ or the average of sequence $T_i(m) - T_i(p)$. The main motivation of adopting the w-DBA approach is that the new augmented sequence keeps the same length as the original base sequence but with reasonable variations.

Gaussian noise injection is another popular and alternative augmentation approach for time sequences, defined as follows:

$$\begin{aligned} &T_{new} = T_{base} + \mathcal{N}(\mu, \sigma^2) \\ &s.t. \left| \sum_{i=1}^{N} \mathcal{N}(\mu, \sigma^2) \right| \le 0.02\mathcal{L} \end{aligned} \tag{6}$$

where $\mathcal{N}(\mu, \sigma^2)$ represents a zero-mean normal distribution with a variance of 0.8 ($\mu = 0$, $\sigma^2 = 0.8$) and $\mathcal{L}$ denotes the total length of the sequence. The motivation why a variance of 0.8 is chosen is to generate a series of noises, 98% of which lie on an appropriate range of $[-2, 2]$. The constraint here is to confine the total length of augmented noises within an acceptable range, making the total distance remains unchanged for each segment after data augmentation. Figure 9 illustrates the comparison between the original sequence and the augmented sequences. It is observed that adding the Gaussian noise may bring in a larger fluctuation than the w-DBA method, but overall, the augmented series stays in step with the general trend. Through these processes, the training dataset has tripled to 69 sequences compared to the original dataset.

Regarding the HMM training, we obtain a model by employing the Baum–Welch algorithm that treats the hidden states as implicit variables using an expectation maximization algorithm, and the detailed derivation and explanation can be found in Ref. [45]. After the model is trained, the prediction procedure uses a sequence of velocity as the observation input, and a dynamic programming (i.e., Viterbi) algorithm as the solver to acquire the most probable state path. The prediction results are achieved one step ahead along the Markov chain within the variant constraints.

The forecasting task in this study is to predict the vehicle velocity 10 s ahead, where approaches with different step windows can be

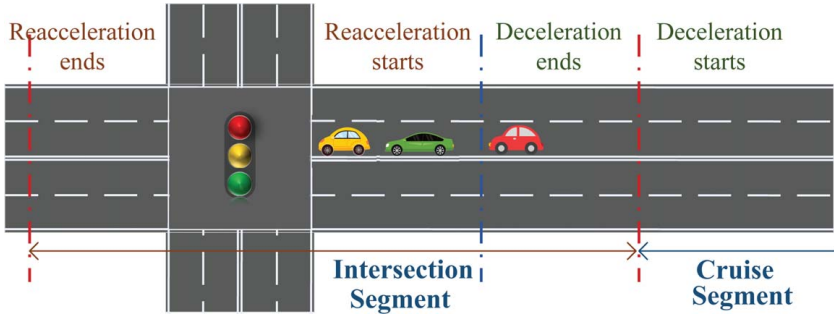

Fig. 6 The schematic diagram for road segment division

used, e.g., five steps ahead with a 2-s interval or two steps ahead with a 5-s interval. The reason to adopt these recursive multi-step approaches is that most of the decelerations and accelerations in the DRD dataset occur within 12 s, and it is challenging for a one-step 10-s ahead direct prediction to capture these processes using HMM.

Table 1 tabulates the validation performances across the original and the augmented datasets. The recursive two-step approach achieves an approximately 3% higher accuracy than the five-step method using these two datasets. The accuracy discrepancies are mainly due to error accumulations, where a double recursion performs better. Moreover, considerable enhancements are also observed after leveraging the data augmentation approaches. However, there still exist drastic fluctuations in the HMM forecasting as shown in Fig. 10, especially during the deceleration and stop states, which may lead to undesired disturbances when applying to the energy management system. The possible reasons for causing these drastic fluctuations and discrepancies are threefold: (i) the transformation of original data into discrete positive integers may potentially lead to non-ignorable accuracy loss; (ii) both the transition and observation matrices are sparse in nature, which can be mitigated by accumulating a much larger number of driving cycles as the implementation of data augmentation indicates; and (iii) for the two-step method, only three previous observations/velocities (i.e., $[O_{t-10}, O_{t-5}, O_t]$) are used as the model input. Additionally, the single HMM model built on the whole cycle is investigated, yielding an undesirable outcome with an MAE of 3.49 m/s and an RMSE of 5.38 m/s based on the original dataset, which is approximately 30% worse than the segment-based approach. The results also further validate the necessities and effectiveness of the segment analysis discussed in Sec. 2. It is worth noting that there are multiple ways to implement HMM in velocity forecasting, e.g., treating the acceleration/deceleration speed as the observable states, real-time updated transition matrix, higher-order Markov chain model, and subdivided HMM models for varying driving conditions [25–27]. However, due to the huge diversity of driving datasets, no comparative study has been reported yet regarding the performance of these different approaches in the literature.

3.2 Deterministic Approaches. Long short-term memory model (LSTM) is an improvement over the recurrent neural network (RNN) with feedback connections designed to address the long-term dependency challenge when modeling sequential events by propagating through time. In addition to the existing structures of RNN like hidden states, LSTM employs a novel layer, named the cell states, to selectively store the previous event information, making it capable of alleviating the vanishing/exploding gradient issue. In this study, a typical LSTM structure with a forget gate is employed to forecast velocity 10 steps/seconds ahead, which takes approximately 300 epochs to converge. Given the sequence length of the dataset, the LSTM model adopts a deep learning structure with one hidden layer after random search. The number of hidden units is narrowed down within the range of 180–260, varying from segment to segment. To avoid potential overfitting, the model also employs the settings of bidirectional layer and dropout layer with a dropout rate of 0.5. Three popular training algorithms are considered here, i.e., stochastic gradient descent (SGD), adaptive moment estimation (Adam), and root-mean-square propagation (RMSprop), which provides important properties of spatial and temporal locality.

Compared to LSTM, feed-forward artificial neural network with back-propagation and SVR are other two popular machine learning-based estimation methods that mainly consider spatial properties. For temporal forecasting, we utilize the past ten sequential steps/seconds record as the input to estimate the velocity 10 s ahead. We take into account three training algorithms for ANN, i.e., the standard Levenberg–Marquardt (LM), variable learning rate

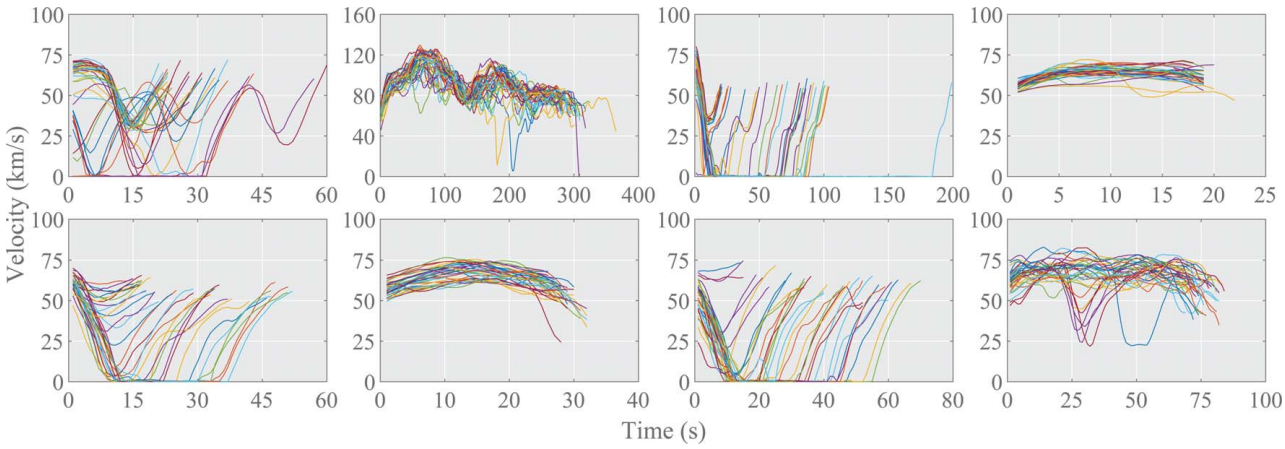

Fig. 7 Segment velocity trajectory *versus* time (top: Segments 9–12; bottom, Segments 13–16 (see Fig. 3), cycle size: 34)

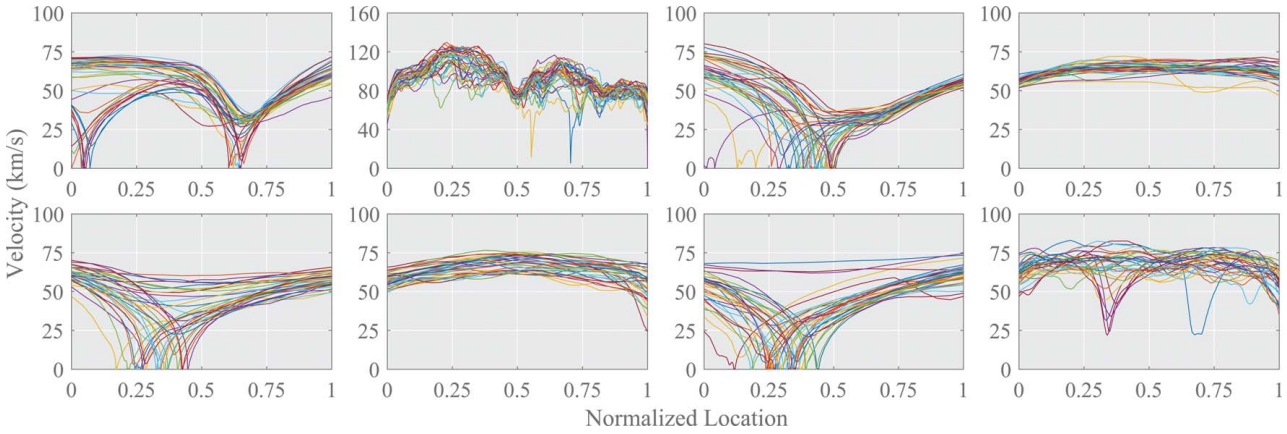

Fig. 8 Segment velocity trajectory *versus* location (top: Segments 9–12, bottom: Segments 13–16 (see Fig. 3), cycle size: 34)

gradient descent (GDX), and resilient back-propagation (RP), in which a fully connected two-layer structure is empirically predefined with a maximum size of 30 neurons for each layer. Similar to the ANN settings, we diversify and examine the model by utilizing three different kernels for SVR based on their popularity, i.e., linear, polynomial, and Gaussian kernels. Other hyperparameters like the penalty weight factor and intensive parameter are optimized and determined based on the validation dataset.

In addition, given the temporal and spatial repetition nature of the DRD dataset as discussed in Sec. 2, a similarity-based estimation approach is proposed here by comparing the similarities. This method takes the previous multiple steps and the real-time position (i.e., the GPS longitude and latitude, and the elevation data as supplementary) as the inputs to retrieve the most similar historical sequences near the location. In contrast to the DTW algorithm, since all the sequences are extracted in the same length, the sequence similarity can be attained directly by calculating the Euclidean distance expressed as follows:

$$Dist = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \alpha_i (S_i - T_i)^2} \tag{7}$$

where S and T are the test sequence and the ranked historical sequences, respectively. n denotes the total length of the two sequences, and here, n is set to be 10. α is a set of unequal weighted factor assigned to different steps, where the latest steps have a larger weight. According to the similarity ranking, the most similar sequences are selected, and the new forecast can be integrated as follows:

$$T_{fore}(t+10) = \sum_{i=1}^{3} \beta_i T_i(t+10) \tag{8}$$

where β is an equal weight factor for generalization, s.t. $\sum \beta_j = 1$. The selected cycle number is narrowed down to 3 with the best performance.

The global performance using the original dataset is compared and tabulated in Table 2. Prior to outcome analyses, it is worth mentioning that RMSE is more sensitive than MAE to large bias or outliers due to the squaring operation. When it comes to this driving cycle dataset, the cruising segments tend to yield relatively smaller absolute differences, while larger biases/residuals normally occur at intersections. Accurate forecasting at intersections is more likely to produce a smaller RMSE. To evaluate the model performance, it is very necessary to report both metrics.

As a piece-wise approach, the segment-based approach dominates the whole cycle-based method for all machine learning models. It is also seen that the diversified submodels with different kernels or training algorithms vary in global accuracy. Given the limited processing time, only the LSTM-sgd, ANN-lm, SVR-Gaussian, and the similarity-based method are selected to perform velocity forecasting. It is worth noting that LSTM models have relatively lower accuracies comparing to the ANN approaches, especially in the piece-wise segment-based forecasting, while LSTM dominates other algorithms in similar studies [27,28]. There are two possible reasons that may account for this discrepancy: (i) the majority of the sequence lengths in this study merely range from 15 to 60 after segmentation, while the sequence lengths in Ref. [27] are around 700 and (ii) the LSTM models are pre-trained offline based on the historical records, which will not be updated during the cycle velocity forecasting given the on-board computational limitations. The detailed forecast results of the best submodels are shown in Fig. 11. Less fluctuations are observed using deterministic methods compared to the stochastic HMM method in Fig. 10.

It needs to be noted that all the deterministic models are based on the original dataset, since no significant improvements are observed by using the augmented dataset. For example, the MAE of ANN-lm and SVR-Gaussian methods using augmented dataset is 2.19 m/s and 2.21 m/s, respectively, which are slightly worse than those using the original dataset. Considering the unique nature of the DRD dataset, one of the possible reasons is that the data augmentation techniques employed in this study is not forecasting-oriented [46].

4 Localized Model Selection and Ensemble Approach

As illustrated in Figs. 10–12, we have observed that the accuracies of the forecasting models differ from segment to segment. To further improve the forecast performance based on these individual models, ensemble models and dynamic model selections are two widely-used second-stage approaches. As a higher-level forecast algorithm, an ensemble model aggregates the predictions of certain diverse base models and results in a final output using averaged or weighted methods, while an online dynamic model selection could deploy an optimal individual model via methods such as reinforcement learning or Bayesian updating [47]. This section will comparatively investigate ensemble and online model selection methods, aiming to develop an enhanced forecasting model with a higher generalization and accuracy. Due to the lack of available data, we reuse the validation dataset (Cycle 24–28) together with a set of new data (Cycle 29–31) for the model training if needed, and the remaining portion of the DRD dataset (Cycle 32–34) is used for testing.

4.1 Single Model Selection. The segment-dependent probability approach is an offline individual model selection method that only considers the prior probabilities for different segments

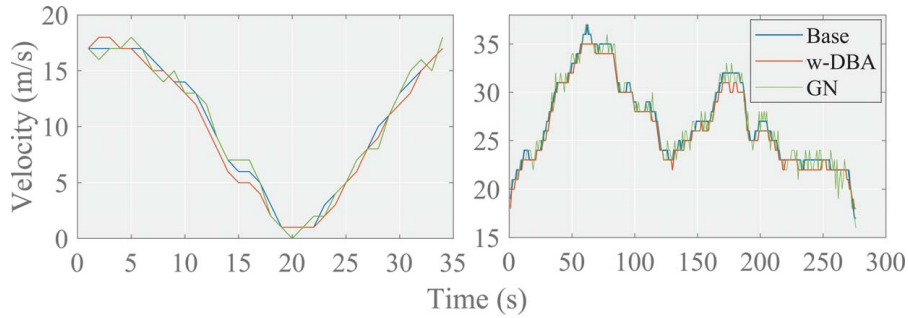

Fig. 9 A comparison among the base, the w-DBA augmented, and the Gaussian noise (GN) augmented sequences (left: Segment 9 and right: Segment 10)

Table 1 Segment-based HMM accuracy analysis

	Original Dataset		Augmented Dataset	
HMM Model	5-step	2-step	5-step	2-step
MAE (m/s)	2.68	2.55	2.61	**2.47**
RMSE (m/s)	4.39	4.22	4.30	**4.07**

Note: Bold values indicate the best MAE or RMSE within each category.

and directly utilizes the base model with the maximum likelihood for prediction. One of the prominent advantages of this method is that no online training is required and the model can be updated offline. Only the selected base model is implemented for prediction, which saves computing time. The probability distribution calculated via frequency count is illustrated in Fig. 13, which results in an improved MAE of 1.99 m/s and a deteriorated RMSE of 3.29 m/s, compared to the best single base model (i.e., ANN) with an MAE of 2.07 m/s and an RMSE of 3.11 m/s for this DRD dataset. The results indicate that this offline model selection method may not be an appropriate solution for this problem.

In contrast to the offline probability-aided method, a persistence approach directly utilizes the dominating model of the last segment as the forecasting model for the current segment. A training process is unnecessary but it entails the implementations of all the submodels pre-selected in Sec. 3. As shown in Fig. 12, the accuracy ranking differs in both cycles and segments, the intersection segments in particular. However, the performance of a prior dominating model is not likely to change dramatically in its next step prediction. The base model ranking and the dynamic model selection are illustrated in Fig. 14, which successfully chooses the top two models with a ratio of 59% (23/39). Considerable improvements are observed with an averaged MAE of 2.02 m/s and an RMSE of 3.07 m/s.

An enhanced online model selection is accomplished via extending the evaluation interval several steps backward rather than only a single step. The optimal base model is selected and updated dynamically and continuously in a certain rolling window. Reinforcement learning-based algorithms like Q-learning are one of the widely used solutions that have been successfully developed in the literature to determine the optimal base model for wind and solar forecasting [48]. However, one of the barriers is that the Q-learning based model selection requires a large amount of dataset and intensive online model updating. Another implementable alternative is possibility-based algorithms like Bayesian model selection [49]. Similarly, in this study, given the piece-wise nature of the segment-based velocity forecasting, we develop a probability-based second-order Markov chain model that takes into account the two previous states to determine the optimal base model, as expressed by

$$P(s_i|s_{i-1}, s_{i-2}, \cdot, \cdot, \cdot, s_1) = P(s_i|[s_{i-1}, s_{i-2}]) \tag{9}$$

where the state s_i represents the best base model. In contrast to the HMM model discussed in Sec. 3.1, the transition matrix is achieved

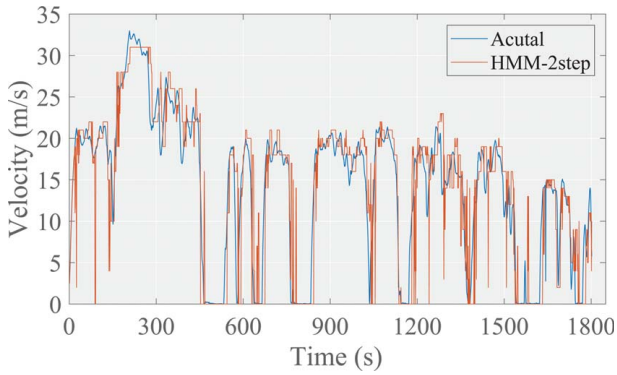

Fig. 10 Two steps ahead HMM forecasting based on a 5-s interval for Cycle-28

Table 2 Deterministic forecasting model accuracy

	Segment-based		Cycle-based	
Metrics (m/s)	MAE	RMSE	MAE	RMSE
LSTM-adam	2.36	4.01	2.79	3.89
LSTM-sgd	**2.26**	**3.23**	2.91	4.18
LSTM-rmsprop	2.29	3.74	2.78	3.83
ANN-lm	***2.15***	3.17	2.69	3.78
ANN-gdx	2.35	3.31	3.15	4.34
ANN-rp	2.19	***3.14***	2.79	3.89
SVR-linear	2.21	**3.38**	2.74	4.32
SVR-poly	2.49	3.73	3.14	4.87
SVR-Gaussian	**2.16**	3.42	2.65	3.91
Similarity	**2.25**	**3.71**	2.25	3.71

Note: Bold values indicate the best MAE or RMSE within each category; Bold italic values indicate the best MAE or RMSE among all models.

explicitly via frequency counting. Only the best-performing model of each segment is considered, and if the transition probability is a null set, it can be replaced with the best model of the last step as described previously. Similar to the probability approach, this method witnesses a significant enhancement on MAE (1.93 m/s), but a worse RMSE (3.16 m/s), meaning more larger variations have been brought into the forecasting.

4.2 Ensemble Model. Instead of utilizing a single prediction, ensemble forecasting combines a set of diverse models to mitigate the forecasting fluctuations and improve the robustness, and a linear ensemble model is given by

$$M_{ensemble} = \sum_{i=1}^{N} \omega_i M_i \tag{10}$$

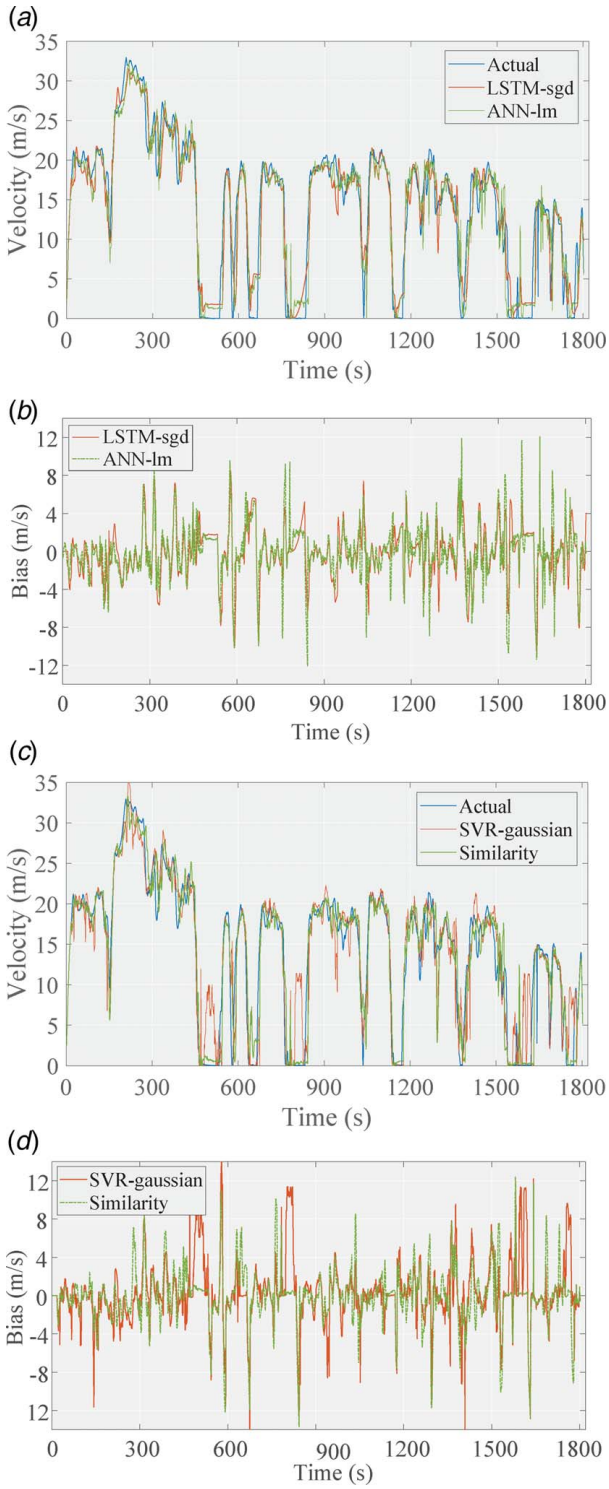

Fig. 11 Forecasting results of Cycle-28 using deterministic methods: (*a*) cycle-based forecasting results using LSTM and ANN, (*b*) forecasting biases using LSTM and ANN, (*c*) cycle-based forecasting results using SVR and similarity, and (*d*) forecasting biases using SVR and similarity

where ω_i denotes the weighting factor, s.t. $\sum \omega_i = 1$. Note that the weighting factor can be determined via optimization, and this study adopts an equal-weighting scheme for a generalization purpose. Combining all the five base models, the achieved MAE and RMSE are 1.95 m/s and 2.89 m/s, respectively, and a remarkable improvement has been observed in terms of RMSE compared to base models.

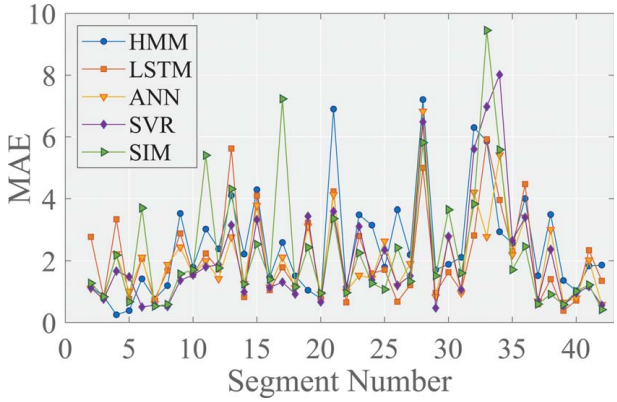

Fig. 12 The segment-based model error distribution of Cycle-32

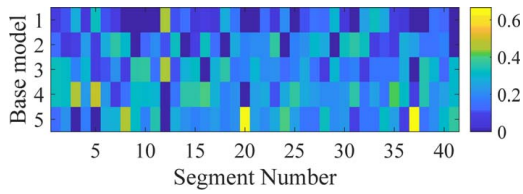

Fig. 13 The normalized discrete prior probability distribution of base models for Cycle-32 (The model numbers 1–5 indicate the HMM, LSTM, ANN, SVR, and SIM model, respectively. The models with an MAE difference threshold of 0.2 m/s are counted as the top models.)

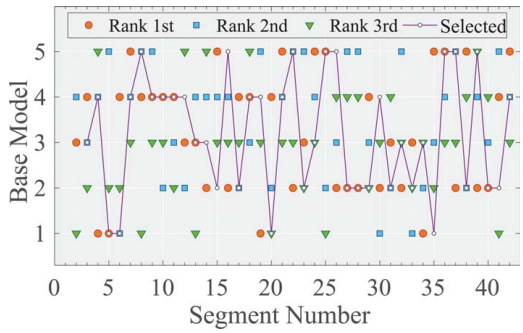

Fig. 14 The base model ranking and selected models using the persistence method for Cycle-32 (Base models labeled 1–5 are HMM, LSTM, ANN, SVR, SIM, respectively.)

4.3 Hybrid Approach. From the aforementioned comparative studies, it is found that the online and offline model selection methods tend to improve the MAE but deteriorate the RMSE, while the ensemble approach tends to enhance the RMSE and decrease the fluctuations. As illustrated in Fig. 15, in this subsection, we integrate the offline probability-aided and online Markov chain model (MM)-based model selection methods with the equal-weighted ensemble approach, aiming to further improve the performance of velocity forecasting.

Based on the probability-based model, the top three base models rather than the best single one are integrated together using an ensemble method. The combined offline model yields an MAE of 1.89 m/s and an RMSE of 2.93 m/s. It is also noticed that there are non-negligible discrepancies for the performance among different cycles. The reason is that the offline probability model heavily relies on the historical data, making it challenging to adapt itself to the real-time driving conditions.

Regarding the Markov chain-based model, we further modify this approach by taking into account the top two models in each state, $s_t = \{M_{ti}, M_{tj}\}$, $i, j \in [1, 2, 3, 4, 5]$. For instance, there are eight

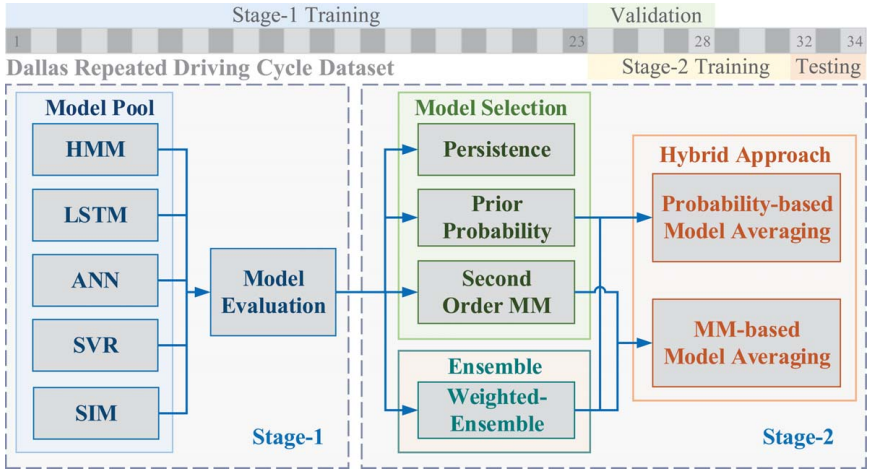

Fig. 15 The prediction framework for short-term velocity forecasting

scenarios for training and four scenarios for testing (only two states are considered for testing, and the third state is the one to be predicted) in forming a Markov chain $[s_{t-2}, s_{t-1}, s_t]$. Especially, a larger factor is assigned to emphasize the sequences with top models. The final forecasts are aggregated by averaging the outputs of the four testing scenarios, which produces dominating results with an MAE of 1.87 m/s and an RMSE of 2.92 m/s. This is also the best forecasting output obtained. Compared to the single ANN model, both the MAE and RMSE have been significantly improved by 9.7% and 6.1%, respectively.

By comparison, it is found that the hybrid approaches combining the individual model selection and ensemble methods perform better than other approaches, as illustrated in Table 3. Although the Markov chain-based model averaging algorithm tends to produce more desirable results, given its training and updating expenses, it is more reasonable to implement the offline probability-based model averaging method in practice as a trade-off between the forecasting accuracy and computational efficiency within a control interval of 5 s. However, its feasibility in practical applications still requires further on-board testing and validation, when the control interval is shorten to 1 or 2 s, aiming to achieve more accurate and sensitive controls. Moreover, due to the very limited volume of the DRD dataset, its generalization also needs to be further verified with more accumulated cycles.

5 Discussions of Future Research Prospects

5.1 Estimation of Intersection Waiting Time. The most challenging forecast task comes from the intersection segments, where the traffic conditions as well as the operations of individual vehicles are complicated, and it is challenging to fully capture the stochastic natures. The forecasting accuracy can be significantly improved if the waiting time could be accurately predicted. According to the discussions in Sec. 2.4, it is assumed that the waiting time at an intersection is highly related to the final stopping location at roughly the same period of a day.

Given this consideration, all the intersections as well as the waiting time are extracted from the DRD dataset, and ANN models are established to further estimate the waiting time. This approach utilizes 21 cycles for training, seven cycles for validation, and the rest six cycles for testing. The hyperparameters like the number of layers and neurons are determined via grid search. The actual and estimated waiting times of Cycle 31–34 are compared in Fig. 16, yielding an averaged MAE of 19.08 s and an RMSE of 27.22 s. This is an acceptable outcome with such a limited set of data samples, given the purpose of waiting time estimation is to optimize the short-time scheduling of onboard systems, such as the air conditioning system and battery cooling/heating system, to avoid overlapping with the energy demand from motor start and reacceleration.

The waiting time highly depends on the arrival time or the remaining time in a traffic light cycle. The assumption behind the ANN approach is that the arrival time can be estimated based on the vehicle's location under a constant traffic volume condition. However, the vehicle locations are heavily affected by the actual accumulated in-between space, making the waiting time challenging to be accurately estimated. Without the implementation of any connectivity devices, An alternative way to further improve the estimation accuracy is to analyze the traffic signal via image detection embedded in the auto-driving module, which is actually mounted available in most newly-released electric vehicles. For a straight road with no large slopes, accurate traffic light recognition can be achieved as far as 130 m away via image-based deep learning

Table 3 A performance comparison among the individual model selection, ensemble, and hybrid approaches

Data source / Model	Cycle-32		Cycle-34		Averaged	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
ANN-base	1.94	2.87	2.13	3.37	2.07	3.11
LSTM-base	1.97	2.80	2.16	3.17	2.09	3.00
Persistence	1.86	2.79	2.03	3.19	2.02	3.07
Probability	1.67	2.60	2.04	3.44	1.99	3.29
2nd HMM	1.71	2.76	2.10	3.49	1.93	3.16
Ensemble	1.81	2.61	1.94	**2.97**	1.95	**2.89**
Prob-averaging	1.75	2.59	**1.90**	3.05	1.89	2.93
MM-averaging	**1.67**	**2.56**	1.99	3.14	**1.87**	2.92

Note: Bold values indicate the best MAE or RMSE among different approaches.

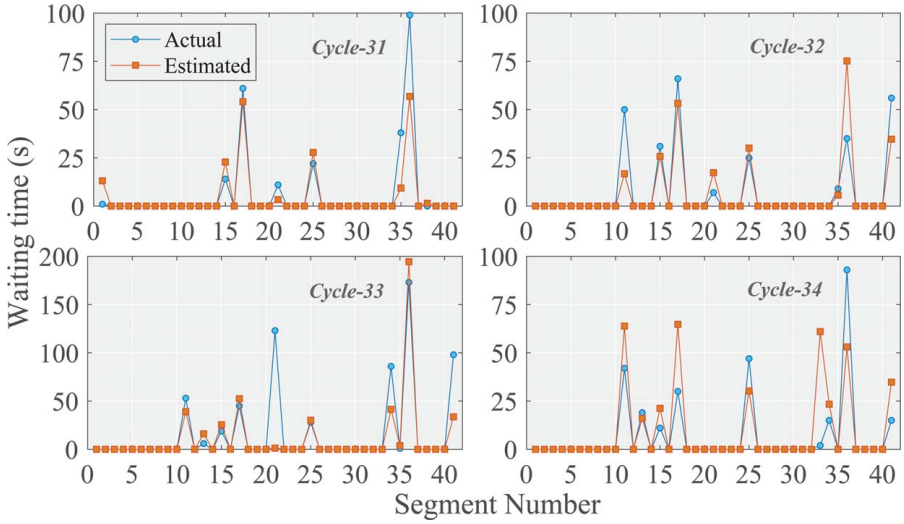

Fig. 16 The estimated waiting time *versus* the actual waiting time

object detection as reported in the literature [50], which is about 7–8 s ahead for a cruising scenario and more than 20 s ahead for a completely stop scenario. The potentials of leveraging the traffic signal detection techniques to improve the forecasting accuracy and boost the energy efficiency still need to be further investigated in the future study.

5.2 Piecewise and Meshing Approach for Local Traffic Network. For most vehicles, the daily driving routes are within the living city or a certain urban area, personal transportation vehicles in particular. Based on the accumulated historical data, an individual vehicle network as well as the private driving routes with unique probabilities can be established. However, from the perspective of personal vehicles, how to efficiently leverage the massive individual driving data remains as an emerging research topic. Based on the segment division implementations developed in this study, a more flexible piecewise segment-based cycle division algorithm needs to be developed to decompose the whole transport network using meshing techniques, which could be capable to adapt itself to different road networks. For a specific route, it can be reassembled based on the possible segments, which is expected to improve the forecasting performance over a longer time duration.

6 Conclusion

This paper generated a repeated urban driving cycle dataset at a fixed route in the Dallas area. Based on the data preprocessing and intersection identification, a cycle segmentation was conducted to provide location-dependent segmental data for improving velocity forecasting. A segment-based velocity forecasting model pool was developed to perform 10 s ahead forecasting, which takes into account the HMM, LSTM, ANN, SVR, and similarity methods. Results showed that the segment-based forecast dominated the whole cycle-based approach with great advantages. Especially, significant improvements have been observed using the ANN method with a 24% reduction for MAE and a 15% decrease for RMSE.

To further improve the forecasting accuracy, a comparative study regarding the individual model selection, ensemble approach, and a combination of them was performed. Results showed that a 9.7% improvement was obtained by leveraging the localized second-order MM-based averaging methods. However, it is more reasonable to implement the offline probability-based model averaging method in practice due to its high computational efficiency. An ANN-based intersection waiting time estimation model was also established and validated with acceptable accuracy. It is foreseeable that the improvements in both velocity forecasting and waiting time estimation will lead to better energy management, especially for electric or hybrid vehicles.

Potential future work will (i) further improve the localized model selection and averaging framework by accumulating more data or exploring reinforcement learning for model selection; (ii) detect/identify the traffic signals via CNN-based image identification techniques to further improve the waiting time estimation and velocity forecasting.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The data and information that support the findings of this article are freely available.[8] The authors attest that all data for this study are included in the paper.

References

[1] Torre-Bastida, A. I., Del Ser, J., Laña, I., Ilardia, M., Bilbao, M. N., and Campos-Cordobés, S., 2018, "Big Data for Transportation and Mobility: Recent Advances, Trends and Challenges," IET Intell. Transp. Syst., **12**(8), pp. 742–755.
[2] Vlahogianni, E. I., Karlaftis, M. G., and Golias, J. C., 2014, "Short-Term Traffic Forecasting: Where We are and Where We're Going," Transp. Res. Part C: Emerg. Technol., **43**(2014), pp. 3–19.
[3] Lana, I., Del Ser, J., Velez, M., and Vlahogianni, E. I., 2018, "Road Traffic Forecasting: Recent Advances and New Challenges," IEEE Intell. Transp. Syst. Mag., **10**(2), pp. 93–109.
[4] Guo, J., and Williams, B. M., 2010, "Real-Time Short-Term Traffic Speed Level Forecasting and Uncertainty Quantification Using Layered Kalman Filters," Transp. Res. Rec., **2175**(1), pp. 28–37.
[5] Wang, J., and Shi, Q., 2013, "Short-Term Traffic Speed Forecasting Hybrid Model Based on Chaos–Wavelet Analysis-Support Vector Machine Theory," Transp. Res. Part C: Emerg. Technol., **27**, pp. 219–232.
[6] Ye, Q., Szeto, W. Y., and Wong, S. C., 2012, "Short-Term Traffic Speed Forecasting Based on Data Recorded At Irregular Intervals," IEEE Trans. Intell. Transp. Syst., **13**(4), pp. 1727–1737.
[7] Jiang, B., and Fei, Y., 2015, "Traffic and Vehicle Speed Prediction With Neural Network and Hidden Markov Model in Vehicular Networks," 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, South Korea, IEEE, pp. 1082–1087.

[8] https://github.com/UTD-DOES/Dallas-Repeated-Driving-Cycle-Dataset

[8] Zhao, Z., Chen, W., Wu, X., Chen, P. C., and Liu, J., 2017, "LSTM Network: A Deep Learning Approach for Short-Term Traffic Forecast," IET Intell. Transp. Syst., **11**(2), pp. 68–75.
[9] Lee, K., Eo, M., Jung, E., Yoon, Y., and Rhee, W., 2021, "Short-Term Traffic Prediction With Deep Neural Networks: A Survey," IEEE Access, **9**, pp. 54739–54756.
[10] Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., and Wang, Y., 2017, "Learning Traffic As Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction," Sensors, **17**(4), p. 818.
[11] Wu, Y., and Tan, H., 2016, "Short-Term Traffic Flow Forecasting With Spatial-Temporal Correlation in a Hybrid Deep Learning Framework." arXiv preprint arXiv:1612.01022.
[12] Yu, H., Wu, Z., Wang, S., Wang, Y., and Ma, X., 2017, "Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks," Sensors, **17**(7), p. 1501.
[13] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., and Li, H., 2019, "T-gcn: A Temporal Graph Convolutional Network for Traffic Prediction," IEEE Trans. Intell. Transp. Syst., **21**(9), pp. 3848–3858.
[14] Lv, M., Hong, Z., Chen, L., Chen, T., Zhu, T., and Ji, S., 2020, "Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction," IEEE Trans. Intell. Transp. Syst., **22**, pp. 1–12.
[15] Mengzhang, L., and Zhanxing, Z., 2020, "Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting." arXiv preprint arXiv:2012.09641.
[16] Fan, X., Xiang, C., Gong, L., He, X., Qu, Y., Amirgholipour, S., Xi, Y., Nanda, P., and He, X., 2020, "Deep Learning for Intelligent Traffic Sensing and Prediction: Recent Advances and Future Challenges," CCF Trans. Pervasive Comput. Interact., **2**, pp. 240–260.
[17] Liu, Y., and Zhang, J., 2021, "Electric Vehicle Battery Thermal and Cabin Climate Management Based on Model Predictive Control," ASME J. Mech. Des., **143**(3), p. 031705.
[18] Amini, M. R., Wang, H., Gong, X., Liao-McPherson, D., Kolmanovsky, I., and Sun, J., 2019, "Cabin and Battery Thermal Management of Connected and Automated Hevs for Improved Energy Efficiency Using Hierarchical Model Predictive Control," IEEE Trans. Control Syst. Technol., **28**(5), pp. 1711–1726.
[19] Amini, M. R., Kolmanovsky, I., and Sun, J., 2021, "Hierarchical MPC for Robust Eco-Cooling of Connected and Automated Vehicles and Its Application to Electric Vehicle Battery Thermal Management," IEEE Trans. Control Syst. Technol., **29**(1), pp. 316–328.
[20] Zhou, Y., Ravey, A., and Péra, M.-C., 2019, "A Survey on Driving Prediction Techniques for Predictive Energy Management of Plug-in Hybrid Electric Vehicles," J. Power Sources, **412**, pp. 480–495.
[21] Wang, X., Liu, Y., Sun, W., Song, X., and Zhang, J., 2018, "Multidisciplinary and Multifidelity Design Optimization of Electric Vehicle Battery Thermal Management System," ASME J. Mech. Des., **140**(9), p. 094501.
[22] Liu, Y., and Zhang, J., 2020, "Self-Adapting J-Type Air-Based Battery Thermal Management System Via Model Predictive Control," Appl. Energy., **263**, p. 114640.
[23] Oncken, J., and Chen, B., 2020, "Real-Time Model Predictive Powertrain Control for a Connected Plug-in Hybrid Electric Vehicle," IEEE Trans. Vehicular Technol., **69**(8), pp. 8420–8432.
[24] Jing, J., Filev, D., Kurt, A., Özatay, E., Michelini, J., and Özgüner, Ü., 2017, "Vehicle Speed Prediction Using a Cooperative Method of Fuzzy Markov Model and Auto-Regressive Model," 2017 IEEE Intelligent Vehicles Symposium (IV), Redondo Beach, CA, IEEE, pp. 881–886.
[25] Zhou, Y., Ravey, A., and Pera, M.-C., 2019, "A Velocity Prediction Method Based on Self-Learning Multi-Step Markov Chain," IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, Vol. 1, IEEE, pp. 2598–2603.
[26] Sun, C., Hu, X., Moura, S. J., and Sun, F., 2014, "Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles," IEEE Trans. Control Syst. Technol., **23**(3), pp. 1197–1204.
[27] Liu, K., Asher, Z., Gong, X., Huang, M., and Kolmanovsky, I., 2019, Vehicle Velocity Prediction and Energy Management Strategy Part 1: Deterministic and Stochastic Vehicle Velocity Prediction using Machine Learning. Technical Report, SAE Technical Paper.
[28] Rabinowitz, A., Araghi, F. M., Gaikwad, T., Asher, Z. D., and Bradley, T. H., 2021, "Development and Evaluation of Velocity Predictive Optimal Energy Management Strategies in Intelligent and Connected Hybrid Electric Vehicles," Energies, **14**(18), p. 5713.
[29] Gaikwad, T., Rabinowitz, A., Motallebiaraghi, F., Bradley, T., Asher, Z., Fong, A., and Meyer, R., 2020, Vehicle Velocity Prediction using Artificial Neural Network and Effect of Real World Signals on Prediction Window. Technical Report, SAE Technical Paper.
[30] Lemieux, J., and Ma, Y., 2015, "Vehicle Speed Prediction Using Deep Learning," 2015 IEEE Vehicle Power and Propulsion Conference (VPPC), Montréal, Canada, pp. 1–5.
[31] Moser, D., Waschl, H., Schmied, R., and Efendic, H., 2015, "Short Term Prediction of a Vehicle's Velocity Trajectory Using Its," SAE Int. J. Passenger Cars-Electronic and Electr. Syst., **8**(2015-01-0295), pp. 364–370.
[32] Zhang, F., Xi, J., and Langari, R., 2016, "Real-Time Energy Management Strategy Based on Velocity Forecasts Using V2V and V2I Communications," IEEE Trans. Intell. Transp. Syst., **18**(2), pp. 416–430.
[33] Rama, N., Wang, H., Orlando, J., Robinette, D., and Chen, B., 2019, "Route-Optimized Energy Management of Connected and Automated Multi-Mode Plug-in Hybrid Electric Vehicle Using Dynamic Programming," Society of Automotive Engineers Technical Paper Series, **1**, p. 15.
[34] Chen, B., Robinette, D., Shahbakhti, M., Zhang, K., Naber, J., Worm, J., Pinnow, C., and Morgan, C., 2017, "Connected Vehicles and Powertrain Optimization," Mech. Eng., **139**(09), pp. S12–S18.
[35] Johannesson, L., Asbogard, M., and Egardt, B., 2007, "Assessing the Potential of Predictive Control for Hybrid Vehicle Powertrains Using Stochastic Dynamic Programming," IEEE Trans. Intell. Transp. Syst., **8**(1), pp. 71–83.
[36] Barik, B., Bhat, P. K., Oncken, J., Chen, B., Orlando, J., and Robinette, D., 2018, "Optimal Velocity Prediction for Fuel Economy Improvement of Connected Vehicles," IET Intell. Transp. Syst., **12**(10), pp. 1329–1335.
[37] Oh, G., Leblanc, D. J., and Peng, H., 2019, "Vehicle Energy Dataset (ved), a Large-Scale Dataset for Vehicle Energy Consumption Research." arXiv preprint arXiv:1905.02081.
[38] Bagloee, S. A., Tavana, M., Asadi, M., and Oliver, T., 2016, "Autonomous Vehicles: Challenges, Opportunities, and Future Implications for Transportation Policies," J. Modern Transp., **24**(4), pp. 284–303.
[39] NCTCG. Government Report Transportation System Management, https://www.nctcog.org/trans/manage/its/transportation-systems-management.
[40] Cheng, S., Lu, F., and Peng, P., 2020, "Short-Term Traffic Forecasting by Mining the Non-Stationarity of Spatiotemporal Patterns," IEEE Trans. Intell. Transp. Syst., **22**(10), pp. 6365–6383.
[41] Le Rhun, A., Bonnans, F., De Nunzio, G., Leroy, T., and Martinon, P., 2019, "A Bi-Level Energy Management Strategy for HEVs under Probabilistic Traffic Conditions."
[42] Nair, D. J., Gilles, F., Chand, S., Saxena, N., and Dixit, V., 2019, "Characterizing Multicity Urban Traffic Conditions Using Crowdsourced Data," PLoS. One., **14**(3), p. e0212845.
[43] Amini, M. R., Sun, J., and Kolmanovsky, I., 2018, "Two-Layer Model Predictive Battery Thermal and Energy Management Optimization for Connected and Automated Electric Vehicles," 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, IEEE, pp. 6976–6981.
[44] Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E., 2012, "Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping," Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, pp. 262–270.
[45] Blunsom, P., 2004, "Hidden Markov Models." Lecture Notes, August, 15(18–19), p. 48.
[46] Bandara, K., Hewamalage, H., Liu, Y.-H., Kang, Y., and Bergmeir, C., 2020, "Improving the Accuracy of Global Forecasting Models using Time Series Data Augmentation." arXiv preprint arXiv:2008.02663.
[47] Mark, C., Metzner, C., Lautscham, L., Strissel, P. L., Strick, R., and Fabry, B., 2018, "Bayesian Model Selection for Complex Dynamic Systems," Nat. Commun., **9**(1), pp. 1–12.
[48] Feng, C., Sun, M., and Zhang, J., 2020, "Reinforced Deterministic and Probabilistic Load Forecasting Via q -Learning Dynamic Model Selection," IEEE Trans. Smart Grid, **11**(2), pp. 1377–1386.
[49] Sato, M.-A., 2001, "Online Model Selection Based on the Variational Bayes," Neural comput., **13**(7), pp. 1649–1681.
[50] Yoneda, K., Kuramoto, A., Suganuma, N., Asaka, T., Aldibaja, M., and Yanase, R., 2020, "Robust Traffic Light and Arrow Detection Using Digital Map with Spatial Prior Information for Automated Driving," Sensors, **20**(4), p. 1181.