Received: 23 March 2022 | Revised: 6 July 2022 | Accepted: 1 September 2022

IET Intelligent Transport Systems

DOI: 10.1049/itr2.12278

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

A comparative study of driver torque demand prediction methods

Luca Cavanini[1] | **Lucio Ciabattoni**[2] | **Francesco Ferracuti**[2] | **Enrico Marchegiani**[2] | **Andrea Monteriù**[2]

[1]Industrial Systems and Control Lt, Glasgow, UK

[2]Department of Information Engineering (DII), Università Politecnica delle Marche, Via Brecce Bianche 1, Ancona 60131, Italy

Correspondence
Francesco Ferracuti, Università Politecnica delle Marche, Via Brecce Bianche 12, Ancona, 60131, Italy.
Email: f.ferracuti@univpm.it

Abstract

The performances of energy management systems or electric vehicles and hybrid electric vehicles are highly dependent on the forecast of future driver torque/power request sequence that affects vehicle efficiency and economy. Since the behaviour of the driver is challenging to model/predict by first-principles models, modern artificial intelligence algorithms would represent feasible methods for approaching this problem in real-world automotive systems. This work provides a comparative study and analysis of performances of different data-driven torque prediction strategies. The studied and compared torque demand prediction techniques are exponentially varying model, linear regression, shallow and deep neural networks, and least square support vector machine-based approaches. The prediction performance and computational cost of these techniques are evaluated and reported, and the possibility of exploiting these techniques in real-world scenarios is also discussed.

1 | INTRODUCTION

In the last years, the climate crisis has oriented worldwide research labs and companies to investigate technologies to reach substantial decarbonisation and the consequent emission reduction. In particular, the automotive field has been heavily involved in this process, with resulting massive production of hybrid electric vehicles (HEVs) and pure electric vehicles (EVs). These vehicles are featured by higher efficiency with respect to internal combustion engine (ICE) vehicles, achieved by recent advances in energy management systems (EMSs) [1]. Modern EMSs are designed according to optimisation-based techniques and aim to manage the vehicle power flow, following an optimal approach that leads the powertrain to operate at the best operating point of its characteristics. Common optimisation-based EMSs are designed according to Pontryagin's minimum principle (PMP), equivalent consumption minimisation strategy (ECMS), and its adaptive version termed A-ECMS [2]. Furthermore, model-based techniques have been recently studied and applied to hybrid/electric vehicles (H/EVs). In particular, model predictive control (MPC) paradigm has been recently considered due to its capability to evaluate the H/EV performances over a certain future horizon [3–5] directly exploiting physical and logical constraints featuring the controlled plant [6, 7]. Mentioned EMSs are led by the driver's action that is mapped on the demanded torque control, and the use of predicted driver control signal is considered a standard approach in order to increase the H/EV performances. Driving cycle pattern, vehicle velocity and torque demand have been considered as the main factors to be studied and analysed to optimize the H/EV powertrain [8].

Driving cycle pattern prediction consists of the forecast of aggregated features of the whole cycle segment, leading to an approximate knowledge of next route characteristics. In [9], driving cycles have been clustered in six different groups by means of a k-shape algorithm, a technique that employs a shape-based distance as a clustering metric. Then, each driving series has been divided into several shorter segments on which features such as maximum velocity, average velocity, minimum acceleration, maximum acceleration and maximum deceleration have been computed. All this information has been used as input for a convolutional neural network (CNN) to predict the type of driving cycle. Niu et al. [10] have combined two neural networks to predict both driving trends and synthetic driving cycles with a time horizon of 1 s. Driving trends have been grouped in a set of five possible types, whereas

synthetic driving cycles in a set of 11. Features used in both neural networks have been built on signals related to velocity, acceleration, and time spent at a specific velocity. The authors aimed to develop a fuzzy-based controller to manage energy consumption. Furthermore, the use of learning vector quantisation neural network (LVQNN) has been exploited to recognize real-time driving patterns and improve velocity prediction in [11].

Driving cycle patterns could be considered for EMS integration and optimisation, but their use within mentioned EMS frameworks would be limited due to the hard customisation required to exploit this information. On the other side, velocity predictors have been widely studied due to the possibility of directly exploiting vehicle speed prediction within EMSs to achieve superior H/EV performances. Due to the stochastic nature of driving profiles, many studies have employed Bayesian network and Markov process in their predictive algorithms. Zhang et al. [12] have developed an algorithm to predict velocity based upon a Bayesian network that uses information about driving characteristics of succeeding vehicles, geographic information system (GIS) and global positioning system (GPS) data. Several studies have employed Markov processes, another kind of stochastic process, as in [11] mentioned before. In [13], a discrete Markov process has been built on velocity and acceleration: similar values of velocity and acceleration compose a state of transition matrix, each representing similar driving conditions. Then, combining the Markov chain and Monte Carlo theory, an adaptive-horizon prediction method has been proposed and used to anticipate future moments. Furthermore, the combination of Markov process with backpropagation neural network (BPNN) has been explored [14]. A vehicle speed prediction has been proposed using both approaches to forecast velocity in a horizon of 5 s. Another study [15] has involved velocity and acceleration data from highways and urban driving cycles and has compared forecasting results obtained by several predictors: a Markov chain, an exponential-based predictive model and the outcomes from three types of neural networks (NNs), a BPNN, a layer recurrent neural network (LRNN), and a radial basis function neural network (RBFNN). In [16], authors have proposed a driving profile prediction approach employing and comparing a Markov chain and a feedforward neural network (FNN) for a horizon up to 30 s. A dataset composed of four different test driving profiles in terms of velocity, acceleration, and road slope has been used, reaching the best performance with the FNN approach. Thus, NN-based models have widely been employed to predict velocity, as reported before ([14, 15]). Moreover, Rezaei et al. [17] have employed an autoregressive model to predict the desired driver's velocity with a horizon of 10 s. Data from three driving cycles have been used, and GPS and GIS information have been involved in the predictive model to improve forecasting, depending on upcoming events.

The principal drawback of speed prediction models is that velocity does not exactly reflect the driver's guide style because it depends on torque demanded by the driver, which instead represents directly driver's reaction. Furthermore, the desired velocity contains an intrinsic delay in its information about the driver's will, owed to engine time response and time elapsed to reach the desired velocity. The main approaches explored in literature for torque prediction have been neural networks, stochastic processes, autoregressive models, and support vector machine (SVM). In [18], a three-layer NN has been used to predict torque demand and vehicle velocity at 2.5 s horizon. Six driving cycles have been recorded from a vehicle conducted by the same driver on different days. Five driving cycles have been used for training and the other one for testing. In the study conducted by Zeng et al. [19], a single-hidden layer feedforward neural network (SLFN) has been presented to predict gasoline engine output torque with high accuracy but using features computed on engine characteristics rather than driving profiles, such as engine speed, intake manifold pressure, barometric pressure, intake air. Also, stochastic approaches have been employed to predict torque demand. Shi et al. [20] have used a one-step Markov chain for an MPC model with a time horizon of 1 s. Positive driving torque has been discretized between the maximum and minimum values to create a transition probability map based on two real drive cycles and CTBCDC (China Transit Bus City Driving Cycle) profile. In [21], a fixed gain algorithm method has been developed for an online, multi-step and real-time prediction for the demanding power of an electro-mechanical transmission based on an autoregressive model with external inputs. Both desired power demand and actual power demand of the vehicle have been used as input. The prediction horizon has been set at 0.3 s. Meng et al. [22] have applied an autoregressive model for torque demand prediction as well. Simulated data have been generated with 0.01 s sampling interval and ten steps, and this interval has been set as horizon prediction; features have been computed by means of torque and vehicle-to-vehicle (V2V) information, such as distance, speed and acceleration of preceding vehicle. In [23], a second order polynomial regressor and a Volterra model have been compared using engine velocity and throttle signal. Dataset has been collected from 300 samples of real vehicle data and prediction step has been set at 1 s. At last, a different approach has been proposed by Vong et al. [24], using a least squares SVM (LS-SVM) based on engine characteristics. 200 different engine setups have been acquired from a Honda B16A DOHC to predict output torque.

A segment of intelligent transport literature has also dealt with driving style recognition in order to provide significant information and assist automotive-related contexts, such as insurances, near-misses, theft prevention and others. For instance, Martinelli et al. [25] have proposed an analysis of five decision tree-based techniques to classify different driving styles considering a feature set related to the vehicle in the attempt of recognizing if the driver is the car owner or an impostor. In [26], employing telematic data, such as velocity, GPS information and acceleration, the study has been focused on two approaches: first, a driver classification and thus, driving style, using a CNN has been proposed; at last, both aggregated time series data and single trips have been statistically modelled in order to predict claims frequency. De Rango et al. [27] have developed a fuzzy inference system able to recognize driver behaviour by taking into account the environment, speed and jerk. Another driving style detection approach based on a combination of a Bayesian

model with a Euclidean distance has been proposed by Han et al. [28]. The dataset has been collected through a driving simulator and included driver-dependent, vehicle-dependent and driving environment-dependent information.

Another emerging scenario that has been addressed in the literature is the vehicle-to-everything (V2X) context, where the vehicle is connected to the surrounding environment and it is able to exchange information with any entity that can affect its performance and energy consumption. For example, Deng et al. [29, 30] have proposed a model-free approach and a multi-layer architecture to predict long-distance traffic velocity in real-world V2X considering the future dynamics of V2X information. The prediction has been based on the most similar past trajectories to the given route and this information has been exploited for forecasting up to 20 min; then, traffic velocity predicted has been adopted for an MPC control approach. In V2X context, the study proposed in [31] and consistent with the attempt of our work, has developed a novel real-time EMS based on an improved torque demand predictor. A Gaussian process (GP)-based predictor is employed aiming to forecast torque demand at 2-s and 4-s horizon. The dataset has been generated using a co-simulation platform for traffic and powertrain models; 12 real-time traffic information have been included as input features, related to vehicle torque demand, brake, throttle, speed data and information about preceding vehicle and distance to the successive intersection.

In this work, a comparative study of different strategies for torque prediction requested by the driver for optimal H/EVs energy management is presented. Exponentially varying torque predictor, linear regressor, shallow and deep NN and LS-SVM-based strategies have been considered [32, 33]. The choice fell on the above techniques since they allow developing multiple-input and multiple-output (MIMO) models and direct models for accurate long-term prediction. MIMO and direct models permit to obtain better results in terms of forecasting accuracy with respect to iterative one-step-ahead models [34]. Mentioned predictors are systematically compared in terms of prediction capability and computation burden. This work adds two original contributions to the related literature. First, an extensive analysis of data-driven based torque demand predictors is conducted, for the first time, for HEV energy management. Second, the LS-SVM-based torque predictor is investigated to fully explore its potential in the model adaptation when the driving style changes over time. Finally, the predictors are tested with real data over two benchmarks in order to model the driving cycles in real-world scenarios with driver dynamic/stochastic behaviours.

The work is organized as follows. The paper presents the prediction techniques and the online updating procedure in Section 2, datasets used in this study are described in Section 3 and results and comparison of the different predictors are reported in Section 4. Conclusions are provided in Section 5.

2 | METHODOLOGY

In this section, the considered data-driven estimation policies for driver torque demand prediction are presented. The included methods belong to the supervised learning class, as they are trained offline on data containing known outputs or labels. Then, these methodologies can be employed online to provide the required torque forecast.

2.1 | Exponential predictor

Exponential torque demand predictor assumes that power request increases or decreases exponentially over the prediction horizon. In each prediction horizon, the exponential torque prediction model is formulated as

$$T_{k+n} = T_k\,(1+\epsilon)^n, \quad n = 1, 2, \dots, p \tag{1}$$

where p is the prediction horizon, T_k is the initial torque at time step k, and ϵ is the exponential coefficient. Different ϵ values are considered to evaluate the sensitivity of the model. The exponential model is included due to its ease of integration in a real-time application [15].

2.2 | Multivariate linear regression

Multivariate linear regression (MLR) considers the following model:

$$y_i = W^T x_i + e_i \tag{2}$$

where y_i is the l-dimensional output vector, x_i is a design d-dimensional vector of predictor variables, $W \in \mathbb{R}^{d\times l}$ is the matrix of regression coefficients and $e_i \approx \mathcal{N}(0, \Sigma)$ is the l-dimensional vector of error terms with multivariate normal distribution. The prediction is computed as $\hat{y}_i = W^T x_i$, where regression coefficients matrix W is obtained by means of ECM algorithm [35].

2.3 | Cascade forward neural network

The cascade forward neural network is a class of NNs similar to feedforward networks, characterised by forward connection between the input and every following layer. The advantage of this network is given by the ability to emulate input–output relationships by maintaining a linear mapping between the layers. Figure 1 shows the general structure of a cascade forward neural network.

2.4 | Deep neural network

With the advancement of computational hardware resources and algorithms, deep learning methods such as the long short-term memory (LSTM) and sequence-to-sequence (seq2seq) model have shown a good deal of promise in dealing with time series forecasting by considering long-term dependencies and multiple outputs [36]. An LSTM network is a type of recurrent

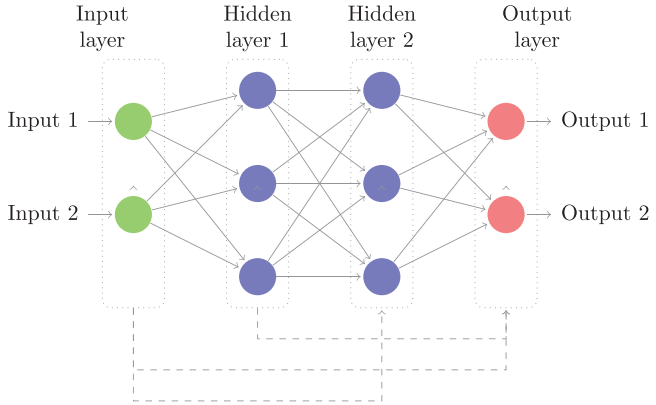

FIGURE 1 Cascade neural network

neural network (RNN) used in the field of deep learning that can learn long-term dependencies between time steps of sequence data; it is used for several tasks such as anomaly detection, speech recognition and time series forecasting as well. This study presents a regression model based on LSTM and the seq2seq structure to predict the driver torque demand.

2.5 | Least squares support vector machine

The LS-SVM solves the regression equation according to the following optimal problem:

$$\min_{w;e;b} J(w;e) = \frac{1}{2}w^T w + \frac{1}{2}\gamma \sum_{i=1}^{n} e_i^2 \tag{3}$$

$$\text{subject to } \; y_i = w^T \phi(x_i) + b + e_i, \quad i = 1, \dots, n \tag{4}$$

where $w, b \in \mathbb{R}$, γ is a regularisation constant, $\phi(x_i)$ is the feature map to the high dimensional future space and e_i denotes the prediction error term for the ith data point. The Lagrangian function is

$$L(w;b;e;\alpha) = J(w;b) + \sum_{i=1}^{n} \alpha_i \left(y_i - w^T \phi(x_i) - b - e_i \right) \tag{5}$$

where α_i are the Lagrange multipliers. Thus, by using Lagrange multipliers, the solution can be obtained by considering the Karush–Kuhn–Tucker (KKT) conditions for optimality and solving:

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i \phi(x_i) \tag{6}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{n} \alpha_i = 0 \tag{7}$$

$$\frac{\partial L}{\partial e_i} = 0 \rightarrow e_i = \frac{\alpha_i}{\gamma} \tag{8}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow y_i = w^T \phi(x_i) + b + e_i \tag{9}$$

Standard LS-SVM framework is based on a primal-dual formulation and the solution in α, b is given by the following linear system

$$\left[\begin{array}{c|c} \Omega + I/\gamma & 1_n^T \\ \hline 1_n & 0 \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline b \end{array}\right] = \left[\begin{array}{c} Y \\ \hline 0 \end{array}\right], \tag{10}$$

with $Y = [y_1, \dots, y_n]^T$, $\alpha = [\alpha_1, \dots, \alpha_n]^T$, $1_n = [1, \dots, 1]$, $\Omega_{ij} = \phi(x_i)^T \phi(x_j) = K(x_i, x_j)$ and with $K(x_i, x_j)$ a positive definite kernel. The previous linear system can be rearranged as

$$\Theta_n \hat{\alpha}_n = Y_n \tag{11}$$

where $\hat{\alpha}_n = [\alpha, b]^T$, $Y_n = [Y, 0]^T$ and the Lagrangian multiplier can be estimated by inverting the matrix Θ_n. According to Mercer's theorem, the resulting LS-SVM model for function estimation becomes $\hat{y}(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i) + b$. When working with large datasets, it is important to emphasize that the use of the entire training sample of size n to compute kernel matrix and the solution of Equation (10) can be prohibitive. Thus, in [37, 38], the authors proposed an explicit approximation for the feature map ϕ when working in the primal space. The method is based on the Nyström method [39] that determines an approximation of ϕ. This finite dimensional approximation $\hat{\phi}(x)$ can be used in the primal problem to estimate w, b and authors provided an algorithm to properly find the support vectors and a methodology to choose a working set of fixed size $m \ll n$. In order to make a more suitable selection of the support vectors instead of a random selection, one can relate the Nyström method to kernel principal component analysis, density estimation and entropy criteria to select the m support vectors which maximize the quadratic Rényi's entropy approximated by

$$\int \hat{p}(x)^2 dx = \frac{1}{N^2} 1_n^T \Omega 1_n \tag{12}$$

Once the fixed size m is set, points from the pool of training data have to be actively selected as candidate support vectors. An algorithm that finds the points from the training dataset that iteratively improves the entropy criterion was proposed in [37, 38] and included in this paper. Fixed-size LS-SVM model is considered for its ability to handle large datasets [38].

2.6 | Fixed-size LS-SVM with online updating procedure

The online updating procedure for fixed-size LS-SVM based predictor is described in this section. In literature, an incremental procedure for standard LS-SVM has been described in [40]. The same approach has been proposed in [41, 42] for an application in the automotive context. In [43], the authors have proposed an online learning algorithm based on an incremental chunk for LS-SVM. In this work, conversely to literature, the entropy value is adopted for a fixed-size LS-SVM model to evaluate the amount of novelty of information introduced by a new sample with respect to the training dataset D_N which

is composed of predictor variables. The incremental algorithm updates the trained LS-SVM model whenever the entropy of a new sample (x_{n+1}, y_{n+1}) overcomes a fixed threshold, enhancing the LS-SVM model generalisation and boosting model accuracy. Since the solution is given by the set of linear equations $\Theta_n \hat{\alpha}_n = Y_n$, in order to compute the online procedure in the dual space the new updated model is given by

$$\Theta_{n+1}\hat{\alpha}_{n+1} = Y_{n+1} \tag{13}$$

where Y_{n+1} is the vector of samples $[Y_n, y_{n+1}, 0]$. In order to efficiently update Θ_{n+1} whenever a new sample is included without explicit computation of the matrix inverse, the matrix inverse Θ_{n+1} can be evaluated from Θ_n with the bordering method as in the following

$$\Theta_{n+1} = \begin{bmatrix} \Theta_n & u \\ u^T & a \end{bmatrix} \tag{14}$$

$$\Theta_{n+1}^{-1} = \begin{bmatrix} \Theta_n^{-1} + \frac{\Theta_n^{-1} u u^T \Theta_n^{-1}}{q} & -\frac{\Theta_n^{-1} u}{q} \\ -\frac{u^T \Theta_n^{-1}}{q} & \frac{1}{q} \end{bmatrix} \tag{15}$$

where $q = a - u^T \Theta_n^{-1} u$, $a = \gamma^{-1} + K(x_{n+1}, x_{n+1})$, and $u = [K(x_{n+1}, x_1), \ldots, K(x_{n+1}, x_n), 1]$. The above incremental procedure can update, reboost and improve the built LS-SVM model continually. When a new sample occurs, the training dataset and the data vectors are incremented as follows

$$\mathcal{D}_n \rightarrow \mathcal{D}_{n+1} \tag{16}$$

$$x_{train} = \left[x_{train}, x_{n+1}\right]^T \tag{17}$$

$$Y_{train} = \left[Y_{train}, y_{n+1}\right]^T \tag{18}$$

and the Lagrangian multiplier becomes

$$\hat{\alpha} = \Theta_{n+1}^{-1} Y_{train} \tag{19}$$

The incremental procedure continuously increases the memory length, resulting in both increase in model complexity and reduction in computational speed; thus, a first in first out (FIFO) decremental procedure is therefore employed after every incremental step by removing the earliest trained data in the training dataset. Similar to the case of the incremental procedure, in order to avoid the computation of the matrix inverse, Θ_n is updated from Θ_{n+1}, where Θ_{n+1} is the matrix without the first row and the first column. Following the decremental procedure, the resulting matrix is obtained as follows:

$$\Theta_n(i-1, j-1) = \Theta_{n+1}(i, j) - \frac{\Theta_{n+1}(i, 1)\Theta_{n+1}(1, j)}{\Theta_{n+1}(1, 1)} \tag{20}$$

where $i, j = 2, \ldots, n+1$ and the training dataset and the data vectors are decremented as follows:

$$\mathcal{D}_{n+1} \rightarrow \mathcal{D}_n \tag{21}$$

$$x_{train}(1) \rightarrow \emptyset \tag{22}$$

$$Y_{train}(1) \rightarrow \emptyset \tag{23}$$

3 | DATASETS DESCRIPTION

Three datasets are used in this study. The first dataset is based on synthetic data of standard driving cycles, that is, NEDC, WLTP (class 1, 2, and 3 termed as WLTP1, WLTP2, and WLTP3, respectively), FTP75 and the US06. The Mathworks MATLAB/Simulink HEV P3 reference application simulation model has been considered to generate the datasets evaluated in this study. The second dataset consists of collected signals in real driving cycles [44]. 70 real driving trips with a BMW i3 were recorded in different weather conditions and different seasons (winter, summer) for a total of 1340 km and 1824 min of route. The third dataset is a collection of streamed data from real vehicles collected by adapter OpenXC which is a combination of open-source hardware and software that lets to extend the vehicle with custom applications and pluggable modules [45]. In this study, nine trips are considered for a total of 156 min of route. The considered scenarios are New York City, U.S.A. (Downtown Crosstown, East Downtown, West Downtown, West Downtown 2, Uptown Crosstown, Uptown Crosstown 2, West Uptown, West Uptown 2) and aggressive driving with hard acceleration and braking.

Algorithms evaluated in this study have been developed by considering the last ten samples of the torque demand signal, the last five samples of vehicle speed and acceleration as inputs. These signals compose the buffer whose size represents the algorithm memory load and that is iteratively updated at each sampling time with the last measured values. It has been empirically established that using longer historical series does not significantly improve the performances in this specific experimentation. The telemetry signals of the third dataset have been synchronized considering linear interpolation. A sampling time T_s of 0.1 s is considered for all datasets. The prediction horizon is 20 steps ahead in this study.

The working strategy for modelling the synthetic driving cycles in terms of training, validation and testing is:

- Training and validation samples: models are estimated using 50% of data for training and validation. Bayesian optimisation algorithm and tenfold CV are used to set the hyperparameters. The BFR on the validation set is computed for hyperparameter selection.
- Testing samples: the models are tested considering the last 50% of data. After defining the optimal model (using the validation BFR), the prediction for the test set is done.

The working strategy for modelling the second real dataset in terms of training, validation and testing is:

- Training and validation samples: a total of 48712 datapoints related to the first five trips. Bayesian optimisation algorithm

and tenfold CV are used to set the hyperparameters. The BFR on the validation set is computed for hyperparameter selection.
- Testing samples: a total of 1043 812 datapoints. The dataset presents the last 65 trips. After defining the optimal model (using the validation BFR), the prediction for the test set is done.

The working strategy for modelling the third real dataset in terms of training, validation and testing is:

- Training and validation samples: a total of 27 542 datapoints related to the first two trips. Bayesian optimisation algorithm and tenfold CV are used to set the hyperparameters. The BFR on the validation set is computed for hyperparameter selection.
- Testing samples: a total of 66 386 datapoints. The dataset presents the seven trips. After defining the optimal model (using the validation BFR), the prediction for the test set is done.

4 | RESULTS

In this section, the prediction results of synthetic and real driving cycles are reported. The achieved prediction results have been evaluated according to the best fit rate (BFR) index, adjusted root mean square error (RMSE), and adjusted mean absolute error (aMAE), computed as

$$\mathrm{BFR} = 100 \times \max\left(0, 1 - \frac{\|y - \hat{y}\|_2}{\|y - \frac{1}{N}\sum_{k=1}^{N} y(k)\|_2}\right), \tag{24}$$

$$\mathrm{aRMSE} = \frac{\left(\frac{1}{N}\sum_{k=1}^{N}\left(y(k) - \hat{y}(k)\right)^2\right)^{0.5}}{\max(y) - \min(y)}, \tag{25}$$

$$\mathrm{aMAE} = \frac{\frac{1}{N}\sum_{k=1}^{N}\left(y(k) - \hat{y}(k)\right)}{\max(y) - \min(y)}. \tag{26}$$

where y and $\hat{y}$ are the real and the predicted torque, respectively, and $y(k)$ is the real torque at the time instance k.

4.1 | Synthetic driving cycles

This section shows the torque prediction results obtained with synthetic data. Results are reported comparing the performance of the baseline predictor with the considered supervised learning methods. Then, the proposed policy based on the online model updating technique is validated. The working strategy for modelling the synthetic driving cycles in terms of training, validation and testing has been replicated with 20 Monte Carlo simulations, and the results shown in this paragraph are the average and standard deviation of the performance indexes.

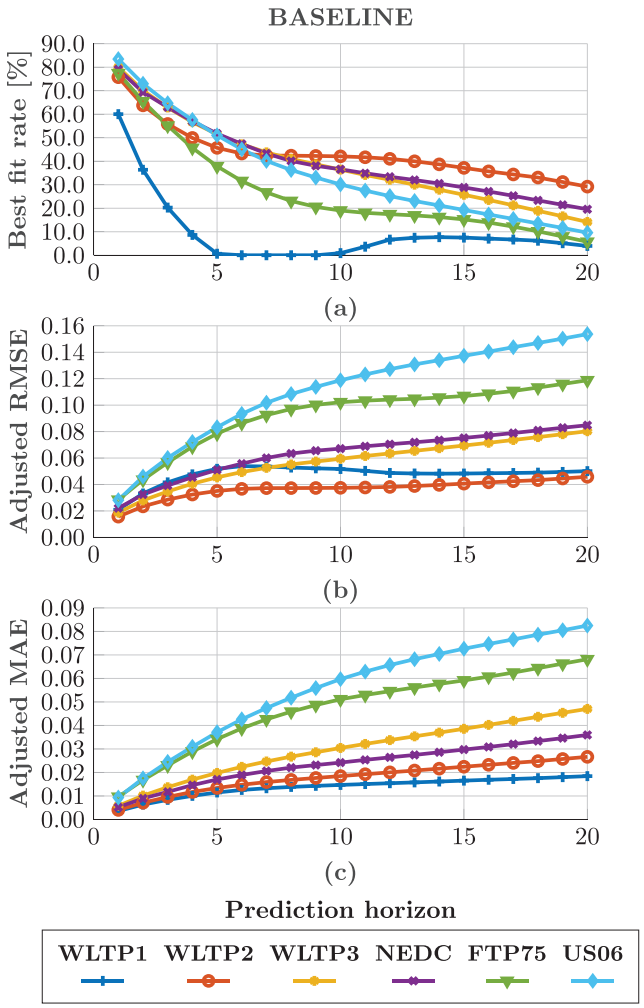

FIGURE 2 Baseline: (a) BFR, (b) adjusted RMSE and (c) adjusted MAE

4.1.1 | Baseline predictor performance

The considered baseline predictor maintains constant the measured torque value at instant k to predict the value to the instants $k+1, \ldots, k+p$, where p is the prediction horizon. Figure 2 shows the indexes BFR, adjusted RMSE, and adjusted MAE of the baseline, in which the driver torque demand is considered constant from k to $k+p$. Figure 2a–c shows that the WLTP1 in terms of BFR and the US06 according to the adjusted RMSE and adjusted MAE represent the worst cases. The 1-step ahead prediction performs reasonably well, whereas the performance worsens as prediction step increases for some type of driving cycles.

4.1.2 | Exponential predictor performance

The exponential predictor gives results highly close to the baseline, as shown in Figure 3. Figure 3a–c shows the BFR, adjusted RMSE and adjusted MAE of the exponential predictor. The parameter ϵ belongs to the set of values $[-0.1, -0.09, \ldots, 0.1]$.

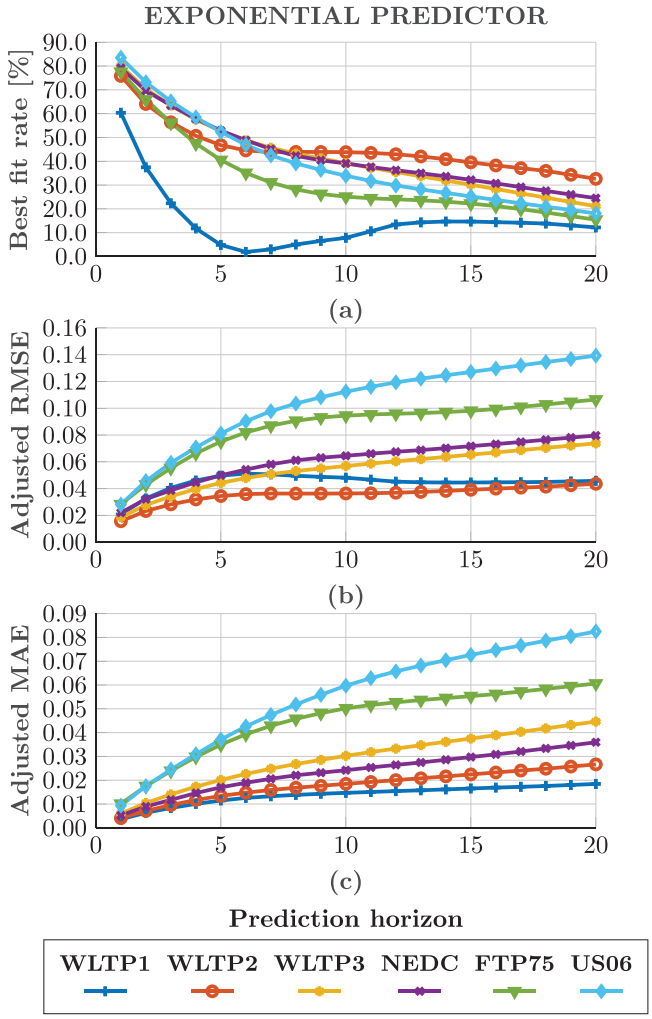

FIGURE 3 Exponential predictor: (a) BFR, (b) adjusted RMSE and (c) adjusted MAE

The average BFR, RMSE and MAE reach their minimum for each driving cycle when $\epsilon \in \{-0.03, -0.02, -0.01, 0\}$.

Exponential predictor performances are similar to the baseline and the tuning of the parameter ϵ is relevant to obtain successful results.

4.1.3 | MLR-based predictor performance

Concerning multivariate linear regression, first-order, second-order and third-order polynomial models are trained. Figure 4 shows the indexes BFR, adjusted RMSE, and adjusted MAE for the case MLR. In particular, Figure 4a,d,g depicts the indices for the linear model, Figure 4b,e,h shows the indices for the second-order polynomial model and, finally, Figure 4c,f,i presents the indices for the third order polynomial model.

The best BFR value is reported by the third-order model (i.e. 64 predictors), but the improvement with respect to the second-order (i.e. 43 predictors) is slight in spite of a greater increase in the computational burden for model training. Thus, a second-order polynomial model is preferable.

4.1.4 | Cascade neural network-based predictor performance

Concerning cascade neural network, Figure 5 depicts the indexes BFR, adjusted RMSE, and adjusted MAE. In particular, Figure 5a,c,e shows the indices for the cascade neural network trained with scaled conjugate gradient (SCG) training algorithm and Figure 5b,d,f evidences the indices for the cascade neural network trained with Levenberg–Marquardt (LM) training algorithm. After hyperparameter tuning, the optimal neural network is composed of three hidden layers with 20, 10, and 20 neurons, respectively. A regularisation term of 0.01 is added to the training algorithm.

The results show as LM training algorithm gives better performance indexes than SCG training algorithm at the expense of higher computation time for offline modelling.

4.1.5 | Seq2seq-based predictor performance

Concerning deep neural network, a structure composed of six layers has been included comprising an input layer, an LSTM layer of 500 neurons, a fully connected layer of 100 neurons, a dropout layer with probability 0.5, a fully connected layer of 20 neurons and a regression layer. Figure 6a–c depicts the performance indices for the sequence-to-sequence regression model using LTSM model.

The results show, as sequence-to-sequence regression model gives worse performance indexes, in particular, lower average and higher variance than cascade neural network-based predictor.

4.1.6 | Fixed-size LS-SVM-based predictor performance

Figure 7a,d,g depicts the performance indices for the fixed-size LS-SVM model with $m = 500$ samples, Figure 7b,e,h shows the indices for the fixed-size model with $m = 200$ samples and as last Figure 7c,f,i shows the indices for the standard LS-SVM model.

In this study, fixed-size LS-SVM returns the best results with respect to the other predictors and it shows to be able to handle large datasets with a low computational burden with respect to neural network-based predictors.

4.1.7 | Predictors comparison

Table 1 reports mean and standard deviation among all the driving cycles and prediction horizons from 1 to 20 steps ahead. Only the performances of the CNN trained with Levenberg–Marquardt are reported because the scaled conjugate gradient training algorithm returns worse results. Outcomes reported in Table 1 highlight as the exponential predictor performs really close to the baseline. The maximum average BFR and minimum average RMSE and MAE are reached by the fixed-size

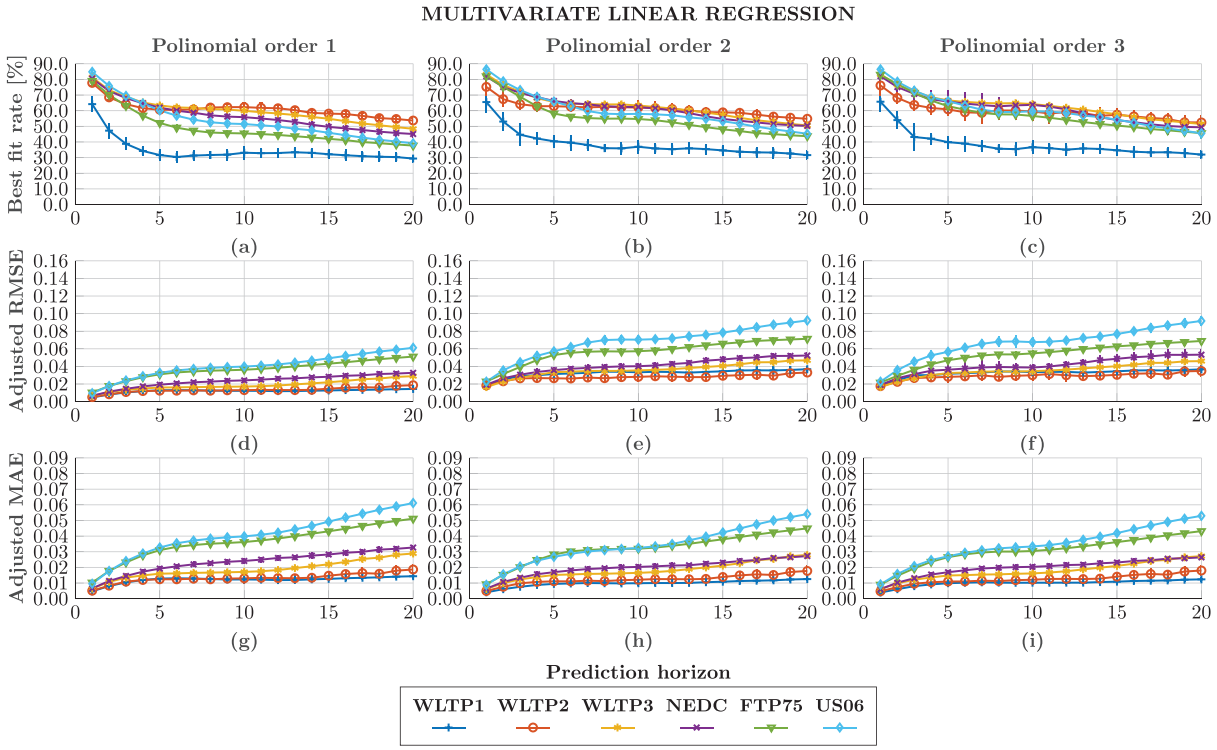

FIGURE 4 Multivariate linear regression based predictor: (a,d,g) first order polynomial model, (b,e,h) second order polynomial model and (c,f,i) third order polynomial model

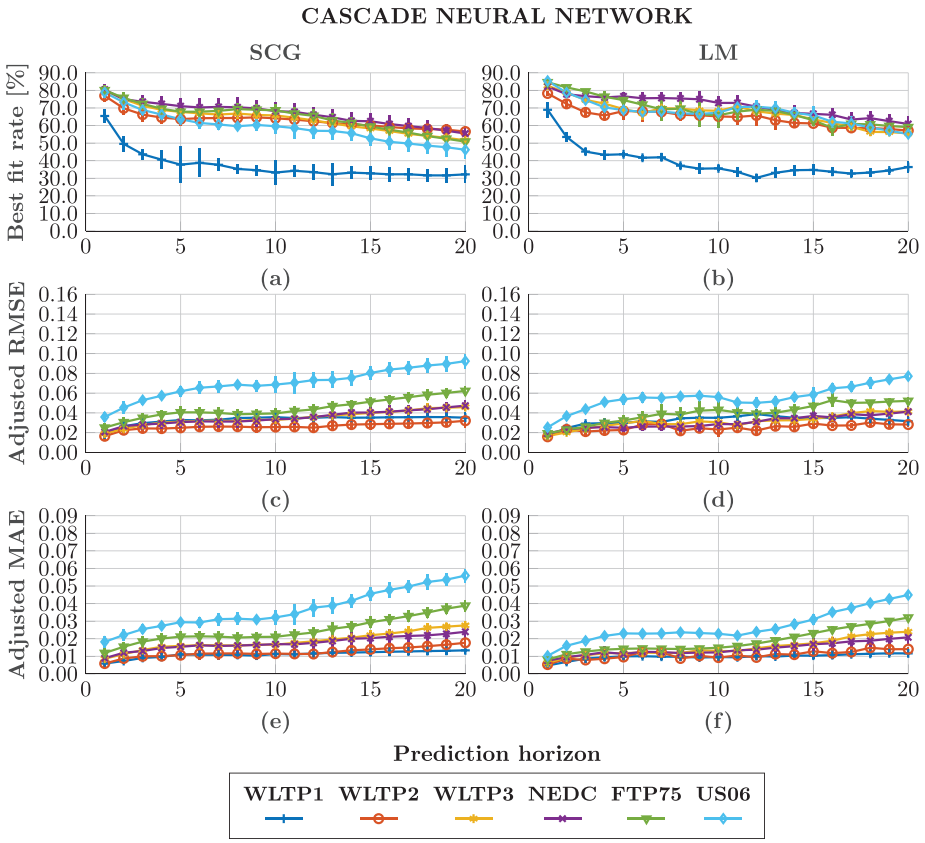

FIGURE 5 Cascade neural network based predictor: (a,c,e) cascade NN trained with SCG training algorithm and (b,d,f) cascade NN trained with LM training algorithm

TABLE 1 Predictors Results

	BFR	adjRMSE	adjMAE	Tcomputation (microseconds)
Baseline	32.2 ± 20.42	0.0680 ± 0.0330	0.0313 ± 0.0200	
Eponential predictor	35.9 ± 18.24	0.0641 ± 0.0299	0.0308 ± 0.0194	0.0027
Fixed-size LS-SVM m=500	69.4 ± 0.29	0.0519 ± 0.0006	0.0240 ± 0.0002	1.1516
Fixed-size LS-SVM m=200	67.0 ± 0.47	0.0555 ± 0.0008	0.0263 ± 0.0003	0.3423
Cascade neural network (Levenberg–Marquardt)	67.7 ± 1.03	0.0556 ± 0.0013	0.0268 ± 0.0005	0.2853
MVN (first order)	53.6 ± 0.26	0.0778 ± 0.0004	0.0405 ± 0.0003	0.0019
MVN (second order)	59.5 ± 0.39	0.0679 ± 0.0006	0.0343 ± 0.0002	0.0148
MVN (third order)	60.2 ± 0.56	0.0668 ± 0.0010	0.0345 ± 0.0003	0.0734
Sequence-to-sequence (LSTM)	49.2 ± 1.57	0.0929 ± 0.0026	0.0440 ± 0.0004	3.4254

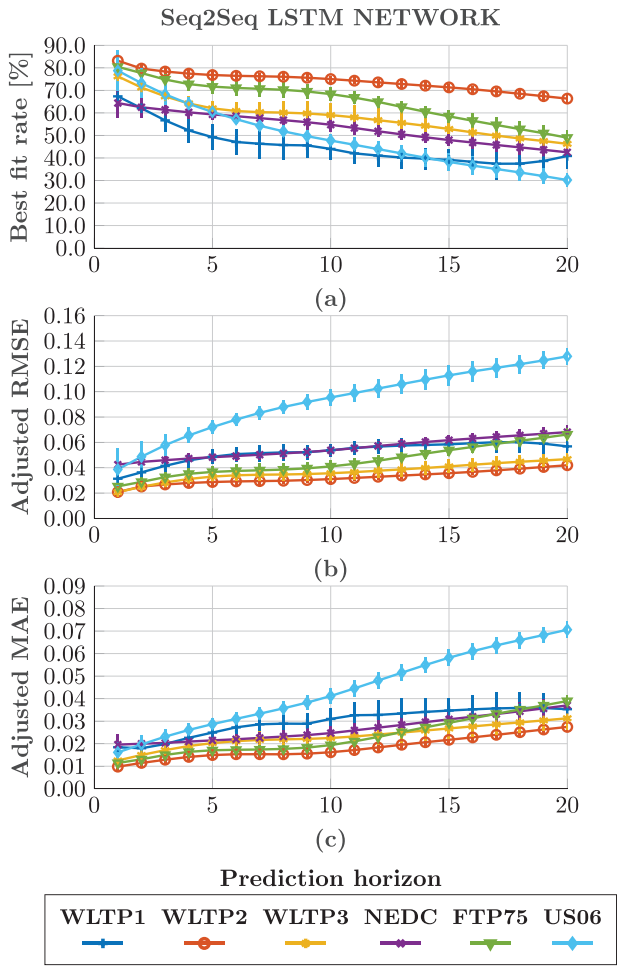

FIGURE 6 Sequence-to-sequence regression based predictor: (a) BFR, (b) adjusted RMSE and (c) adjusted MAE

LS-SVM ($m = 500$ samples). CNN shows similar average results to fixed-size LS-SVM but with a higher standard deviation. MLR presents the worst results compared to fixed-size LS-SVM and cascade neural network, but it is characterised with smaller standard deviation. Finally, the sequence-to-sequence model based on LSTM shows the worst outcomes among the machine learning based predictors.

Simulation has been performed on a personal computer with an Intel Core i7-7700HQ CPU at 2.8 GHz CPU. Computational time required by each prediction algorithm is reported in Table 1 showing how sequence-to-sequence regression-based predictor is computationally heavier than CNN, MVN, and exponential based predictors, whereas fixed-size LS-SVM requires a computational burden proportional to the sample number m.

4.1.8 | Online updating procedure

To test the proposed online updating procedure, the fixed-size LS-SVM model is trained with the driving cycle FTP75, and the driving cycle US06 is concatenated with FTP75 for testing. Figure 8 depicts the results in terms of normalized real torque and normalized torque prediction at 0.5 s horizon (5 steps ahead). Figure 8 shows the cases with and without online updating procedure, and the coefficient of determination R^2 is reported as well. The online updating procedure permits to improve the results, reducing the adjusted RMSE from 0.183 for the case without the online updating procedure to 0.0943 for the opposite case.

Figure 9 presents the US06 driving cycle and the prediction at 0.5 s horizon with and without the online updating procedure, showing how the proposed method permits improving the model performance in the case of driving style changes.

4.2 | Real driving cycles

In this section, the torque prediction results obtained with two real datasets are reported. Only fixed-size LS-SVM algorithm is considered for torque prediction modelling since it achieved the best performances in synthetic data and offers the advantage of dealing easily with prediction problem size up to millions of samples. Concerning the dataset related to BMW i3, we

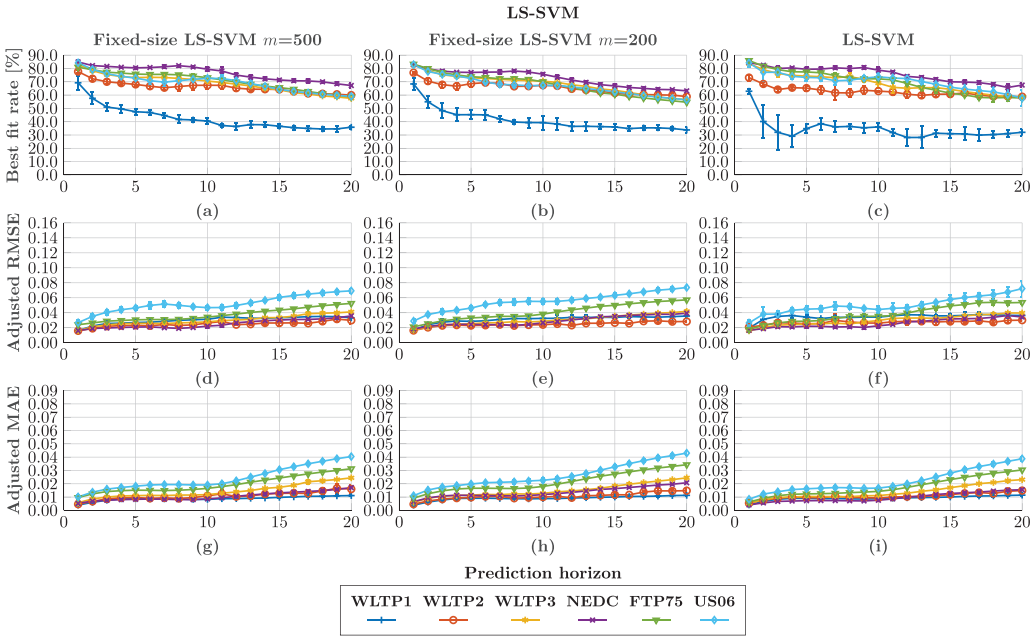

FIGURE 7 LS-SVM based predictor: (a,d,g) fixed-size model, $m = 500$, (b,e,h) fixed-size model, $m = 200$, and (c,f,i) full model

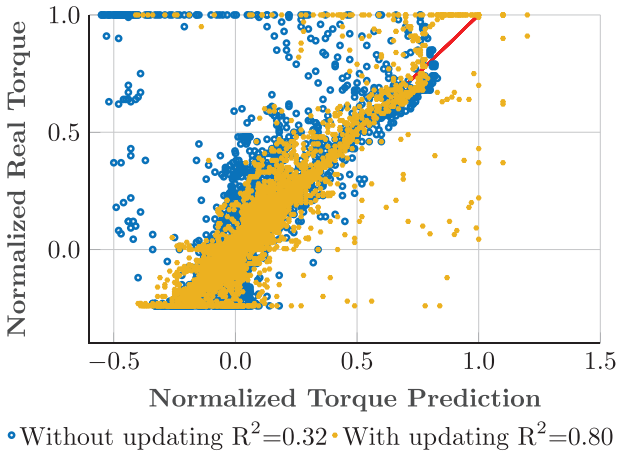

FIGURE 8 Fixed-size LS-SVM versus fixed-size LS-SVM with on-line updating: R^2 plot

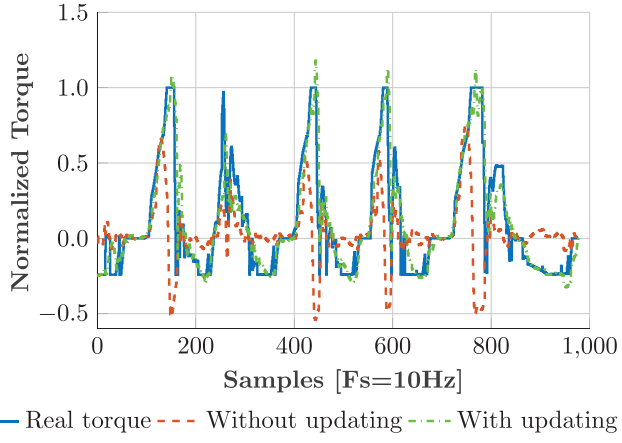

FIGURE 9 Fixed-size LS-SVM versus fixed-size LS-SVM with on-line updating: performances over US06 driving cycle

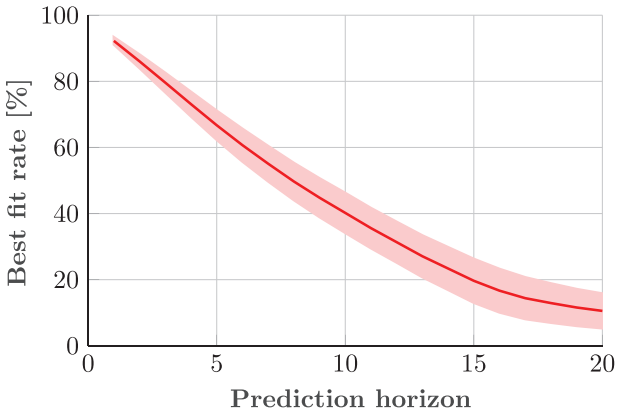

FIGURE 10 BFR of driving cycle prediction of BMW i3 through fixed-size LS-SVM

considered the first five trips for training and validation and the last 65 trips for testing in order to predict the driving cycle behaviour in a real-world scenario and test the fixed-size LS-SVM prediction model with driver dynamic/stochastic behaviours. Figure 10 shows the average of BFR over the testing trips and the shaded area refers to the ± standard deviation. The prediction model is built considering $m = 500$ samples.

The results show as the prediction accuracy is similar to the results obtained with the synthetic data for short-term prediction whereas the results are worse over five steps of prediction horizon. Concerning the dataset related to OPENXC, we considered the first two trips for training and validation and six trips for testing. Figure 11 shows the average of BFR over the testing trips and the shaded area refers to the ± standard deviation. The prediction model is built considering $m = 500$ samples. The results show as the prediction accuracy is worse than the

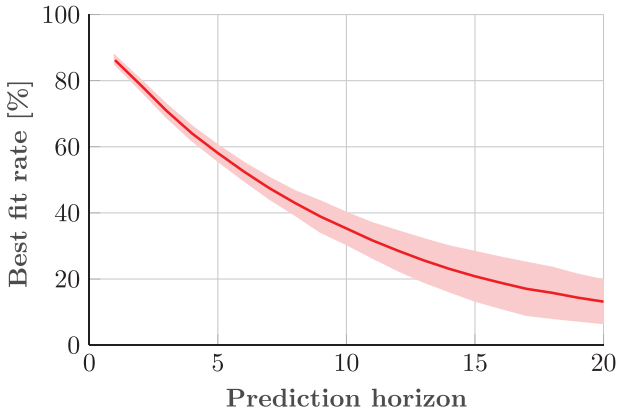

FIGURE 11 BFR of driving cycle prediction of OPENXC data through fixed-size LS-SVM

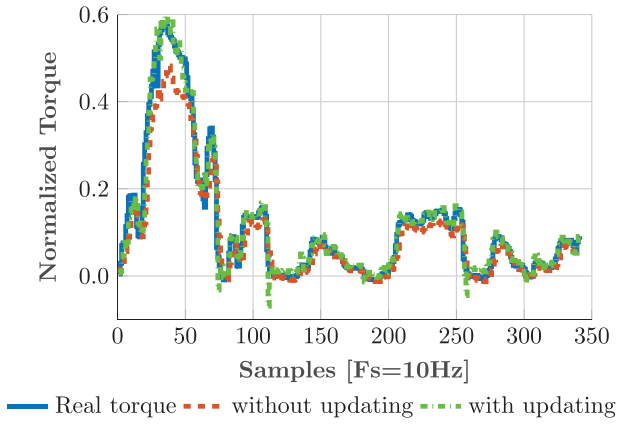

FIGURE 12 Fixed-size LS-SVM versus fixed-size LS-SVM with on-line updating: OPENXC driving cycle

results obtained with the synthetic data and the BMW i3 dataset and the shape of the curve shows how the long-term prediction is difficult to model in real scenarios. The ninth route is related to aggressive driving with hard acceleration and braking and it is used to test the updating procedure since it shows a different driving cycle with respect to trips 1 and 2 used for prediction modelling. Figure 12 presents the OPENXC driving cycle and the prediction at 0.2 s horizon with and without the online updating procedure, showing how this method permits improving the model performance in the case of driving style changes. The updating procedure permits improving BFR from 67.2% to 73.1%.

In conclusion, fixed-size LS-SVM-based torque predictor has been investigated to evaluate its capability, using the proposed on-line updating policy, to adapt the trained model to driving style changes. The results highlight improved performances in case of model update, evidencing the capability to efficiently follow the driving style characterizing a single driver or different driving environment, related to route type, for example, highway or urban routes, or traffic conditions. One drawback of this solution is related to the learning stage that is computationally demanding and it could be performed by using edge computing or cloud solutions and whereas it is challenging to implement in embedded solutions.

5 | CONCLUSION

This work presents two main contributions: a comparative study of different data-driven torque prediction algorithms and the implementation of an online model updating technique that aims to adapt to driving style changes. For the comparative analysis, the considered techniques are: exponentially varying model, linear regression, shallow and deep neural networks, and least squares support vector machine-based torque demand predictors. Prediction performances and computational burden are evaluated and their real-world application is discussed. Reported results showed that fixed-size LS-SVM-based predictor is able to provide the best performances while testing over a range of well-known and synthetic driving cycles. Moreover, this approach is able to handle large datasets, representing an optimal and alternative solution to other methodologies having higher computational costs. Thus, it is finally tested on two datasets consisting of real driving cycles. Then, the fixed-size LS-SVM-based torque predictor has been investigated to evaluate capabilities, using the proposed on-line updating policy, to adapt the trained model to driving style changes. The results highlight improved performances in case of model update, evidencing the capability to efficiently follow the driving style characterizing a single driver or different driving environment, related to route type, for example, highway or urban routes, or traffic conditions. The proposed methodology, moreover, could be integrated into a V2X context in order to improve forecasting performance and model adaption to several driving styles.

LIST OF ABBREVIATIONS

aMAE	adjusted mean absolute error
BFR	best fit rate
BPNN	Backpropagation neural network
CNN	convolutional neural network
CTBCDC	China transit bus city driving cycle
CV	cross-validation
ECM	expectation conditional maximisation
ECMS	equivalent consumption minimisation strategy
EMS	energy management systems
EV	electric vehicle
FIFO	first in first out
FTP	Federal Test Procedure
GIS	geographic information system
GPS	geographic positioning system
KKT	Karush–Kuhn–Tucker
LM	Levenberg–Marquardt
LRNN	layer recurrent neural network
LS-SVM	least squares support vector machine
LSTM	long short-term memory
LVQNN	learning vector quantisation neural network
MIMO	multi input multi output

MLR	multivariate linear regressor
MPC	model predictive control
NEDC	new European driving cycle
NN	neural network
PMP	Pontryagin's minimum principle
RBFNN	radial basis function neural network
RMSE	root mean square error
RNN	recurrent neural network
SCG	scaled conjugate gradient
Seq2seq	sequence-to-sequence
SLFN	single-hidden layer feedforward neural network
SVM	support vector machine
V2V	vehicle-to-vehicle
WLTP	Worldwide harmonized Light-Duty vehicles Test Procedure

AUTHOR CONTRIBUTIONS

Luca Cavanini: Conceptualization, investigation, supervision, validation, writing - original draft. Lucio Ciabattoni: Resources, visualization. Francesco Ferracuti: Data curation, formal analysis, investigation, methodology, software, validation, visualization, writing - original draft, writing - review and editing. Enrico Marchegiani: Data curation, investigation, validation, visualization, writing - original draft, writing - review and editing. Andrea Monteriã: Supervision, validation, visualization, writing - original draft, writing - review and editing.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in IEEE Dataport at http://doi.org/10.21227/6jr9-5235, reference number 44.

The data that support the findings of this study are available in OpenXC, reference number 45. These data were derived from the following resources available in the public domain: http://openxcplatform.com/resources/traces.html.

ORCID

Francesco Ferracuti https://orcid.org/0000-0001-6827-6204

REFERENCES

1. Rizzoni, G., Guzzella, L., Baumann, B.M.: Unified modeling of hybrid electric vehicle drivetrains. IEEE/ASME Trans. Mechatron. 4(3), 246–257 (1999)
2. Serrao, L., Onori, S., Rizzoni, G.: ECMS as a realization of Pontryagin's minimum principle for HEV control. In: 2009 American Control Conference, pp. 3964–3969. IEEE, Piscataway, NJ (2009)
3. Di Cairano, S., Bernardini, D., Bemporad, A., Kolmanovsky, I.V.: Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management. IEEE Trans. Control Syst. Technol. 22(3), 1018–1031 (2013)
4. Cavanini, L., Majecki, P., Grimble, M.J., Uchihori, H., Tasaki, M., Yamamoto, I.: LPV-MPC path planner for autonomous underwater vehicles. IFAC-PapersOnLine 54(16), 301–306 (2021)
5. Uchihori, H., Cavanini, L., Tasaki, M., Majecki, P., Yashiro, Y., Grimble, M.J., et al.: Linear parameter-varying model predictive control of AUV for docking scenarios. Appl. Sci. 11(10), 4368 (2021)
6. Cavanini, L., Ippoliti, G., Camacho, E.F.: Model predictive control for a linear parameter varying model of an UAV. J. Intell. Rob. Syst. 101(3), 1–18 (2021)
7. Cavanini, L., Majecki, P., Grimble, M.J., Ivanovic, V., Tseng, H.E.: LPV-MPC path planning for autonomous vehicles in road junction scenarios. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 386–393. IEEE, Piscataway, NJ (2021)
8. Zhang, F., Wang, L., Coskun, S., Pang, H., Cui, Y., Xi, J.: Energy management strategies for hybrid electric vehicles: Review, classification, comparison, and outlook. Energies 13(13), 3352 (2020)
9. Chen, Z., Yang, C., Fang, S.: A convolutional neural network-based driving cycle prediction method for plug-in hybrid electric vehicles with bus route. IEEE Access 8, 3255–3264 (2019)
10. Niu, L., et al.: Intelligent HEV fuzzy logic control strategy based on identification and prediction of drive cycle and driving trend. World J. Eng. Technol. 3(03), 215 (2015)
11. Lin, X., Zhang, G., Wei, S.: Velocity prediction using Markov Chain combined with driving pattern recognition and applied to dual-motor electric vehicle energy consumption evaluation. Appl. Soft Comput. 101, 106998 (2021)
12. Zhang, S., Zhang, D., Luo, Y., Wang, J., Li, K.: Predictive energy management strategy for electric vehicles based on estimation of preceding vehicle future movements. In: 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), pp. 1436–1441. IEEE, Piscataway, NJ (2016)
13. Li, Y., He, H., Peng, J.: An adaptive online prediction method with variable prediction horizon for future driving cycle of the vehicle. IEEE Access 6, 33062–33075 (2018)
14. Zhang, L., Liu, W., Qi, B.: Energy optimization of multi-mode coupling drive plug-in hybrid electric vehicles based on speed prediction. Energy 206, 118126 (2020)
15. Sun, C., Hu, X., Moura, S.J., Sun, F.: Velocity predictors for predictive energy management in hybrid electric vehicles. IEEE Trans. Control Syst. Technol. 23(3), 1197–1204 (2015)
16. Deufel, F., Gießler, M., Gauterin, F.: A generic prediction approach for optimal control of electrified vehicles using artificial intelligence. Vehicles 4(1), 182–198 (2022)
17. Rezaei, A., Burl, J.B.: Prediction of vehicle velocity for model predictive control. IFAC-PapersOnLine 48(15), 257–262 (2015)
18. Shen, X., Shen, T.: Neural-network-based vehicle torque demand forecasting. In: 2017 17th International Conference on Control, Automation and Systems (ICCAS), pp. 21–26. IEEE, Piscataway, NJ (2017)
19. Zeng, W., Khalid, M.A., Han, X., Tjong, J.: A study on extreme learning machine for gasoline engine torque prediction. IEEE Access 8, 104762–104774 (2020)
20. Shi, D., Wang, S., Cai, Y., Chen, L.: Stochastic predictive energy management of power split hybrid electric bus for real-world driving cycles. IEEE Access 6, 61700–61713 (2018)
21. Zhao, Y., Chen, W., Wang, W., Xiang, C., Huang, K.: Research on optimal power allocation strategy based on power demand prediction for electromechanical transmission. In: 2016 35th Chinese Control Conference (CCC), pp. 8638–8643. IEEE, Piscataway, NJ (2016)
22. Meng, W., Zhang, J., Zhang, R.: A real-time prediction algorithm for driver torque demand based on vehicle-vehicle communication. In: 2019 Chinese Automation Congress (CAC), pp. 3435–3440. IEEE, Piscataway, NJ (2019)
23. Bhattacharjee, A., Saranya, S., Kuntumalla, P.: Estimation of driver demand torque using parametric and nonparametric data-driven model. In: 2021 IEEE Second International Conference on Control, Measurement and Instrumentation (CMI), pp. 87–92. IEEE, Piscataway, NJ (2021)
24. Vong, C.M., Wong, P.K., Li, Y.P.: Prediction of automotive engine power and torque using least squares support vector machines and Bayesian inference. Eng. Appl. Artif. Intell. 19(3), 277–287 (2006)
25. Martinelli, F., Mercaldo, F., Orlando, A., Nardone, V., Santone, A., Sangaiah, A.K.: Human behavior characterization for driving style recognition in vehicle system. Comput. Electr. Eng. 83, 102504 (2020)
26. Gao, G., Meng, S., Wüthrich, M.V.: What can we learn from telematics car driving data: A survey. Insur. Math. Econ. 104, 185–199 (2022)

27. De Rango, F., Tropea, M., Serianni, A., Cordeschi, N.: Fuzzy inference system design for promoting an eco-friendly driving style in IoV domain. Veh. Commun. 34, 100415 (2022)
28. Han, W., Wang, W., Li, X., Xi, J.: Statistical-based approach for driving style recognition using Bayesian probability with kernel density estimation. IET Intel. Transport Syst. 13(1), 22–30 (2019)
29. Deng, J., Adelberger, D., del Re, L.: Hybrid powertrain control with dynamic traffic prediction based on real-world V2X information. In: 2021 American Control Conference (ACC), pp. 1644–1649. IEEE, Piscataway, NJ (2021)
30. Deng, J., Adelberger, D., Del Re, L.: Communication-based predictive energy management strategy for a hybrid powertrain. Optim. Control. Appl. Methods 43(1), 86–105 (2022)
31. Zhang, B., Xu, F., Shen, T.: Receding horizon optimal control of HEVs with on-board prediction of driver's power demand. IET Intel. Transport Syst. 14(12), 1534–1545 (2020)
32. Cavanini, L., Ferracuti, F., Longhi, S., Monteriù, A.: LS-SVM for LPV-ARX identification: Efficient online update by low-rank matrix approximation. In: 2020 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1590–1595. IEEE, Piscataway, NJ (2020)
33. Cavanini, L., Ferracuti, F., Longhi, S., Marchegiani, E., Monteriù, A.: Sparse approximation of ls-svm for lpv-arx model identification: Application to a powertrain subsystem. In: 2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), pp. 1–6. IEEE, Piscataway, NJ (2020)
34. An, N.H., Anh, D.T.: Comparison of strategies for multi-step-ahead prediction of time series using neural network. In: 2015 International Conference on Advanced Computing and Applications (ACOMP), pp. 142–149. IEEE, Piscataway, NJ (2015)
35. Meng, X.L., Rubin, D.B.: Maximum likelihood estimation via the ECM algorithm: A general framework. Biometrika 80(2), 267–278 (1993)
36. Park, S.H., Kim, B., Kang, C.M., Chung, C.C., Choi, J.W.: Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder–decoder architecture. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1672–1678. IEEE, Piscataway, NJ (2018)
37. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)
38. De Brabanter, K., Dreesen, P., Karsmakers, P., Pelckmans, K., De Brabanter, J., Suykens, J.A.K., et al.: Fixed-size LS-SVM spplied to the Wiener– Hammerstein benchmark. IFAC Proc. Volumes 42(10), 826–831 (2009)
39. Williams, C., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems 13, pp. 682–688. MIT Press, Cambridge, MA (2001)
40. Tang, H.S., Xue, S.T., Chen, R., Sato, T.: Online weighted LS-SVM for hysteretic structural system identification. Eng. Struct. 28(12), 1728–1735 (2006)
41. Wong, P.K., Vong, C.M., IP, W.F.: Modelling of petrol engine power using incremental least-square support vector machines for ECU calibration. In: 2010 International Conference on Optoelectronics and Image Processing, vol. 2, pp. 12–15. IEEE, Piscataway, NJ (2010)
42. Wong, P.K., Wong, H.C., Vong, C.M., Wong, K.I.: Online wavelet least-squares support vector machine fuzzy predictive control for engine lambda regulation. Int. J. Engine Res. 17(8), 866–885 (2016)
43. Hao, Z., Yu, S., Yang, X., Zhao, F., Hu, R., Liang, Y.: Online LS-SVM learning for classification problems based on incremental chunk. In: International Symposium on Neural Networks, pp. 558–564. Springer, Berlin (2004)
44. Steinstraeter, M., Buberger, J., Trifonov, D.: Battery and heating data in real driving Cycles. IEEE Dataport. https://doi.org/10.21227/6jr9-5235 (2020). Accessed 6 July 2022
45. OpenXC: Vehicle trace files - openXC. http://openxcplatform.com/resources/traces.html (2022). Accessed 27 June 2022

How to cite this article: Cavanini, L., Ciabattoni, L., Ferracuti, F., Marchegiani, E., Monteriù, A.: A comparative study of driver torque demand prediction methods. IET Intell. Transp. Syst. 17, 534–546 (2023). https://doi.org/10.1049/itr2.12278