Electric Vehicle Energy Consumption Modeling using Real World Driving Data: System Identification vs. Machine Learning

Mohamad Bilal Baluch, Suleiman M. Sharkh *Senior Member, IEEE*, and Anil Madhusudhanan

***Abstract*—Implementing energy-saving practices requires accurately modeling electric vehicle (EV) energy consumption. This research proposes a data-centric approach to predict the energy consumption of EVs using the nonlinear autoregressive with exogenous inputs (NARX) model structure with the motor torque and vehicle speed as inputs. The model was identified and validated with on-road driving data from four trips between Southampton and various UK cities, achieving at least 99.2% accuracy in total energy consumption predictions. The NARX model achieved good accuracy in both simulation and prediction configurations, while reducing the mean squared error (MSE) by 30% compared to the Long Short-Term Memory (LSTM) network, which is a machine learning model structure with feedback and feedforward components. Although a feedforward neural network (NN) performed comparable to the NARX model when using past inputs and outputs, the NARX model proved more computationally efficient due to its more straightforward structure with fewer parameters. These results also demonstrate potential benefits of integrating system identification techniques to enhance machine learning models' performance for dynamic systems.**

***Index Terms*—Electric vehicle, energy consumption, NARX, system identification, LSTM, artificial neural network, machine learning, deep learning, real-world driving data.**

I. Introduction

SWITCHING to electric vehicles (EVs) can help in the global endeavor to lower carbon emissions and curb the effects of climate change [1]. Given the increasing number of EVs, there is a growing focus on optimizing powertrain energy management, improving the charging infrastructure and storage solutions for renewable energy sources, and integrating autonomous vehicle technology to enhance eco-driving practices [2]. These strategies are essential to address the anticipated energy demand caused by EVs, making accurate estimation of EV energy consumption a crucial research topic [3].

A variety of methodologies for modeling the energy consumption of EVs have been investigated in the literature, including physics-based modeling, data-driven modeling, and hybrid approaches that combine both elements. Fundamental physical modeling equations describing the vehicle's powertrain and longitudinal dynamics, and incorporating physical parameters of different powertrain components are employed in physics-based models. As discussed in [4], each component is individually modeled based on manufacturer specifications to capture their unique efficiency characteristics in transmitting energy. This approach uses a backward calculation framework to calculate losses incurred in components at different operating points based on the required traction force at the wheel. It integrates multiple subsystems such as the high-voltage battery, electric machine with power electronics, transmission, and auxiliary devices.

In [5] and [6], the EV model presents an approach to modeling subsystems by scaling powertrain components based on their technical specifications, such as the electric motor and inverter, and by assuming a constant transmission efficiency. Work done in [7] estimates the EV's instantaneous power by simplifying road load equations, with parameters determined from the vehicle's specifications. It validates estimated energy consumption with on-road measurements, achieving a mean absolute percentage error (MAPE) of 15.6% for all trips. Validation is done by manually gathering gradient information and measuring vehicle velocity and acceleration at each step. Another study proposes a computationally effective modeling framework that breaks down the road load equation into drive cycle and vehicle parameter vectors to predict and evaluate energy consumption against real-world driving data [8]. Physics-based models are highly interpretable and can assess the impacts of vehicle subsystems on overall energy requirements. However, they require substantial work to extract vehicle-specific parameters [9]. Additionally, real-time measurements of external factors such as road slope and wind speed are often necessary to estimate energy usage. These factors significantly influence energy consumption, but measuring these inputs is complex and error-prone, potentially reducing the model's accuracy [10], [11]. Also, parameters like drag coefficient, rolling resistance coefficient, and efficiency of powertrain components are typically considered constant but vary in real-world scenarios, further affecting the prediction accuracy of these models [12].

Gray box modeling combines physics-based and data-driven modeling. The model structure in this approach is based on physical laws that are either entirely or partially known, and model parameters are estimated based on the collected data [13–16]. In [13], the powertrain configuration was established

Manuscript created 24 May 2024. This work was supported by The Royal Society under Grant RGS\R1\231218: Experimental Research On Energy Efficiency Of Autonomous Electric Vehicles Considering Powertrain Characteristics, Traffic And V2X Communication. (*Corresponding author: Mohamad Bilal Baluch.*)

Mohamad Bilal Baluch, Suleiman M. Sharkh, and Anil Madhusudhanan are with the Mechatronics Research Group, Faculty of Engineering and the Physical Sciences, University of Southampton, SO17 1BJ Southampton, U.K. (e-mail: bilal[dot]baluch[at]soton[dot]ac[dot]uk; s[dot]m[dot]sharkh[at]soton[dot]ac[dot]uk; anil[at]soton[dot]ac[dot]uk).

in the vehicle system simulation tool Autonomie. It analyzes input and output signals of all components involved with dynamometer testing data and estimates the physical properties of the battery, inverter, and motor. The total energy needed for multiple driving cycles is calculated once the parameters of each component are known, and the differences between the simulated outcomes and test results vary between 0.5% and 4.2%. Zhang et al. [14] and Fiori et al. [15] established a relationship between energy consumption and influencing factors such as instantaneous speed, acceleration, road gradient, and State of Charge (SOC). This relationship is based on vehicle dynamic equations with coefficients estimated using regression analysis of the measured data. In [16], a model is established with five key components: the road load model, powertrain loss model, regenerative braking model, auxiliary system model, and the battery model. Parameters for these models were derived from extensive real-world and lab testing, demonstrating the calculation of energy usage across various driving speeds and conditions with a maximum error of 5%. The structure of models used in gray box modeling strongly relies on first principles and qualitative knowledge, and the measured data determines the model parameters. However, this reliance creates a framework that limits the extent to which parameters can be changed [17].

Conversely, black box modeling is a purely data-driven approach that does not depend on the system's physical nature and allows flexibility in adjusting the model parameters. Model structure and parameters are derived based on measured data, providing an input-output relationship between different vehicle variables without explicitly adhering to basic principles, making them difficult to interpret [3]. This methodology has recently attracted much attention, resulting in various studies on machine learning (ML) techniques for predicting energy consumption [18–22]. Work done in [18] estimates fuel consumption using separate steady-state and transient modules, trained using support vector regression (SVR). The steady-state module predicts fuel consumption using engine speed and torque as inputs. Meanwhile, the transient module inputs the vehicle's acceleration and speed. The combination of outputs from these two modules provides transient fuel consumption data, and findings show that the MAPE of this model is below 14%. The approach in [19] uses a three-level driving fragment split of real-world training data. Features are then taken from the data and utilized as input for an ensemble learning technique known as XGBoost to estimate energy usage. In another research [20], Long Short-Term Memory (LSTM) and Deep Neural Network (DNN) models are combined. Time-series profiles are created using vehicle speed and acceleration data to train the LSTM network. The LSTM output is then integrated with SOC, outdoor temperature, and auxiliary electric load to provide additional inputs for a DNN that predicts energy consumption. The model's prediction accuracy is measured in normalized root mean square error (NRMSE), with a value of 0.055 achieved for a one-year dataset.

In [21], an artificial neural network (ANN) is trained using the Levenberg-Marquardt (LM) backpropagation algorithm and synthetic training data from a virtual chassis dynamometer. This ANN uses vehicle speed, acceleration, motor torque, and revolutions per minute (RPM) to predict energy consumption with an MAPE of 1.25%. However, the model is validated with data generated by a virtual chassis dynamometer instead of on-road data. It is worth noting that this model may not be robust to noise commonly found in real-world driving data. In [22], the model is based on Deep Convolutional Neural Network (D-CNN) layers utilizing vehicle speed, road gradient, and tractive force as inputs. It investigates the effects of the number of layers on the accuracy of energy consumption estimation. This model was trained using data generated in the Future Automotive Systems Technology Simulator (FASTSim) and tested against real-world driving datasets obtained from the Argonne National Laboratory (ANL), showing an average percentage deviation of 5.6% in total energy consumption predictions. Although these ML techniques exhibit reliable prediction accuracy, training them generally requires significant processing power, which raises computation costs and training times.

Alternatively, the work done in [23] employs a linear Box-Jenkins (BJ) model with the prediction error method for system identification (SI) to estimate the energy consumption of internal combustion engine vehicles using real-world driving data. Despite their inability to capture engine nonlinearities, these linear models demonstrate a 4.3% variation from the measured data. In contrast, nonlinear autoregressive with exogenous input (NARX) models have been successfully identified to model complex real-world systems, effectively addressing the challenges posed by nonlinearities [24], [25]. This makes NARX model suitable for applications requiring nonlinear dynamic behavior modeling. Although less popular, SI models are a valuable substitute for more sophisticated ML techniques since they require less computing power and provide comparable accuracy on CPU-based processing [26].

This article proposes an approach to overcome the challenges discussed earlier. The primary contributions of this article are:

1) A modeling approach is proposed for estimating energy consumption of electric vehicles using on-road driving data and a NARX model structure with the vehicle speed and motor torque as the inputs, and electric power as the output. This approach does not require the knowledge of parameters related to the powertrain or longitudinal dynamics. It simplifies data collection by negating the need for complex and error-prone external measurements or estimates such as road slope and wind speed, while achieving high prediction accuracy.
2) The energy consumption model was identified and validated using four on-road driving datasets. The model, including parameters, and datasets are made publicly available on GitHub[1]. It is control system design friendly and can model other vehicles if their driving data is available.
3) Several ML models, including feedforward and recurrent neural networks (RNN), were also trained and validated

[1]GitHub repository for this work: https://github.com/FMLabSouthampton/EV_EnergyConsumptionModel_v1_2024.

with the same on-road datasets and compared with the proposed NARX model, which demonstrated higher prediction accuracy with fewer parameters.

4) The NARX model provides improved computational efficiency compared to ML models, making it particularly suitable for real-time applications and hardware with limited computing power.

This article is structured as follows: Section II describes the instrumentation of the test vehicle, the data collection method involved in different road trips, and the dataset preparation for training and validation prior to identifying the energy consumption model. Section III describes the proposed modeling approach. Section IV presents a comparative validation results analysis with ML models. Section V summarizes the findings and discusses future work.

II. Test Vehicle Instrumentation and Data Acquisition

The second-generation Kia Niro EV, part of the Future Mobility Lab at the University of Southampton, was used as the test vehicle (Fig. 1). This front-wheel drive vehicle has a permanent magnet synchronous motor (PMSM), powered by a lithium-ion battery pack. The vehicle specifications are in Table I. It is instrumented with a CANedge1 datalogger to capture necessary Controller Area Network (CAN) messages via the Unified Diagnostic Service (UDS) protocol [27], [28]. Before the data collection, vehicle variables such as accelerator pedal position, battery current, battery voltage, brake pedal position, longitudinal acceleration, SOC, motor RPM, motor torque, and vehicle speed were decoded from raw CAN messages using reverse engineering tools. The CANedge1 device stores specific CAN messages in its configuration file on a Secure Digital (SD) card and is plugged into the vehicle's On-Board Diagnostic (OBD) port. These stored messages are in the request format as per the UDS protocol, which facilitates responses from specific Electronic Control Units (ECUs). As shown in Fig. 2, the CANedge1 transmits these stored messages to the Central Gateway Module through the OBD port, which relays them to all ECUs on the CAN bus network, including the Motor Control Unit (MCU), Vehicle Control Unit (VCU), Battery Management System (BMS), and Anti-Lock Braking System (ABS). These ECUs respond with the required CAN messages, which are logged by the CANedge1 and are stored on the SD card. This request-response cycle continues throughout the datalogging process. These logged messages are extracted from the SD card and preprocessed using MATLAB. The CAN messages are received at different sampling times, for example, the messages for motor torque at 0.22 s, vehicle speed at 0.07 s, and battery current and battery voltage at 0.14 s, due to their varying lengths and protocols used by the ECUs. These messages are decoded into the physical values of variables and then resampled, resulting in the logged dataset with a sampling interval of 0.22 s, constrained by the motor torque sampling time.

Data was collected over four days on different routes covering Southampton to Chilworth, Alderbury, Basingstoke, Portsmouth, Beaulieu, Christchurch, and Bournemouth. The

Fig. 1. Test vehicle used in this study, a Kia Niro EV, instrumented for on-road data collection.

TABLE I
Test Vehicle Specifications.

Parameters	Value
Vehicle model	MY23 SG2EV
Drive type	Front-wheel drive
Kerb weight	1739 kg
Maximum speed	103 mph
Battery type	Lithium-ion
Battery capacity	64.8 kWh/180.9 Ah
Motor type	PMSM
Maximum power/at RPM	150 kW/6000-9000
Maximum torque/at RPM	255 Nm/0-6000
Tire size	215/55 R17

trips were carried out on typical weekdays (5th, 7th, 8th, and 14th March 2024) during regular daytime working hours (13:30-16:30, 12:30-14:45, 09:30-11:45, and 09:00-11:30, respectively) to capture usual weekday traffic conditions, including varying levels of congestion across urban, primary, and motorway sections, from stop-and-go segments to higher-speed stretches, and to avoid unusually low-traffic periods such as late night or very early morning. The test vehicle was driven through a combination of urban, motorway, and rural scenarios, as shown in Fig. 3. Trip 1 primarily consists of urban driving, with some rural driving. In contrast, trips 2 and 3 focus heavily on motorway driving, and the rest of the drive is mainly in the urban setting. Trip 4 offers a balanced combination of urban and rural driving. During these trips, air conditioning in the passenger cabin was not used. The trips began and ended at the University of Southampton's

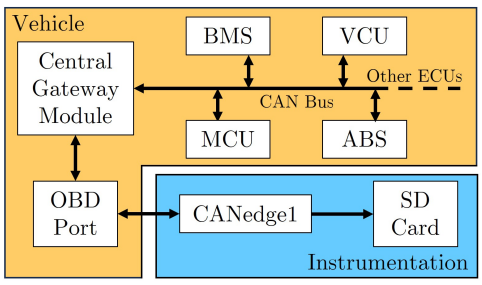


Fig. 2. An illustration of the data logging process in the test vehicle, showing acquisition of vehicle variables from the Controller Area Network (CAN) bus.

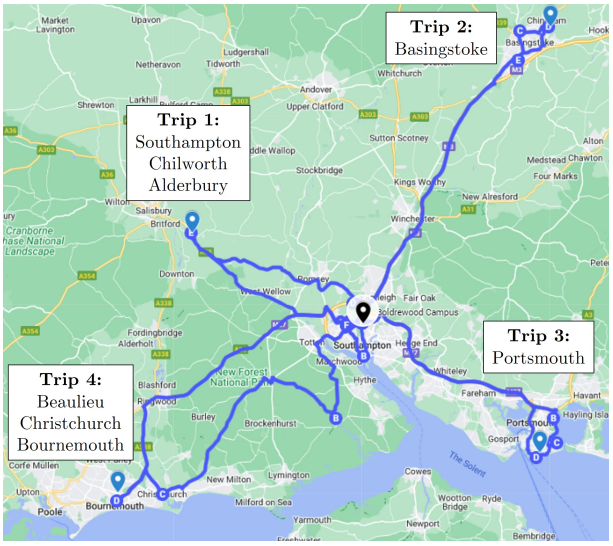


Fig. 3. Routes covered during the four data collection trips from Southampton to different UK cities.

Boldrewood Innovation Campus. Each trip's characteristics are mentioned in Table II, and the distribution of road categories in Table III, both describing distinct driving conditions. Average speeds vary from 20 to 32 mph. The road slope was computed offline every 20 meters using the digital surface model extracted from the Department for Environment Food & Rural Affairs (DEFRA) LIDAR dataset [29]. Box plot analysis in Fig. 4 reveals the interquartile range (IQR) difference, i.e the middle 50% of the dataset, indicating varying speed levels across all trips despite similar overall ranges. These diverse input conditions across trip datasets are useful to validate the model's robustness.

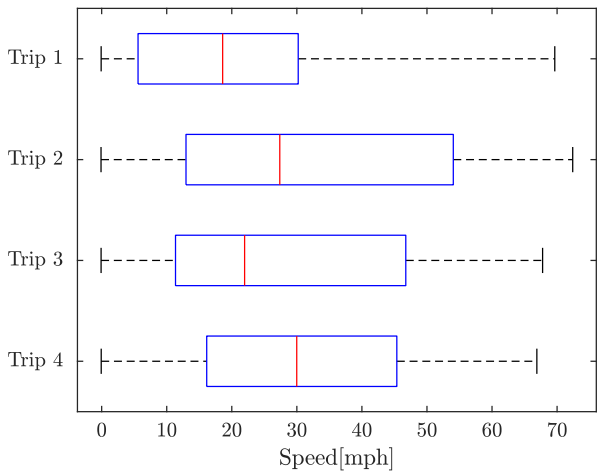


Fig. 4. Box plot of vehicle speed for each trip, highlighting different interquartile ranges (IQR) and varying speed levels across trips.

The gathered datasets consisting of inputs and output across all the trips were preprocessed before modeling. Instances in which a vehicle was driven in the reverse direction for parking or route-changing situations were removed. The electric power, i.e. the model output, is calculated by multiplying the battery's voltage and current reading for every data point. The model was identified using data from trip 1 and subsequently validated using trip 2, 3 and 4 data. The preprocessed data from trip 1 is hereafter referred to as the identification dataset, and the preprocessed data from trips 2-4 as the validation datasets. Fig. 5, 6, 7, and 8 show preprocessed datasets of trips 1, 2, 3 and 4, respectively.

TABLE II
SUMMARY OF TRIP CHARACTERISTICS.

Variable	Trip 1	Trip 2	Trip 3	Trip 4	Unit
Trip duration	153	125	108	139	minutes
Average speed	20	32	28	31	mph
Distance covered	51	66	51	71	miles
Max acceleration	2.6	2.3	2.4	2.5	m/s^2
Max deceleration	-3.8	-3.1	-2.8	-3.5	m/s^2
Max jerk	11.2	9.0	11.6	10.6	m/s^3
Min jerk	-8.1	-8.3	-13.7	-11.8	m/s^3
Max road slope	11	7	6	11	%
Min road slope	-10	-8	-6	-13	%
Max power	107	97	76	69	kW
Min power	-63	-57	-87	-95	kW
Energy consumed	11.824	15.768	11.633	15.784	kWh
Wind speed	0-10	8-16	10-14	0-11	mph
Outdoor temperature	7-11	9-12	7-10	9-14	°C

TABLE III
ROAD CATEGORY DISTRIBUTION IN DIFFERENT TRIPS.

Category	Trip 1	Trip 2	Trip 3	Trip 4
Motorway (M roads)	6%	66.2%	70.4%	8.8%
Primary (A roads)	74.5%	27.3%	25.3%	67.2%
Secondary (B or minor roads)	19.5%	6.5%	4.3%	24%

III. PROPOSED NARX MODEL

As the energy consumption characteristics of electric vehicles are nonlinear, the proposed approach uses a nonlinear autoregressive with exogenous input (NARX) model, which is frequently used for modeling various nonlinear engineering systems [17].

A. Choice of inputs

Various combinations of vehicle speed, acceleration, road slope, SOC, tractive force, motor torque, and RPM as inputs have been used by other models in the literature to predict energy consumption [14], [15], [18], [20–22]. Selecting suitable inputs is an important modeling choice. If enough inputs are not selected, the model might fail to capture the system's behavior. On the other hand, if an excessive number of inputs are selected, the model may become complicated, which may cause overfitting and reduce its robustness when dealing with new data. The proposed model uses only motor torque and vehicle speed as inputs. They can represent the power requirements at the wheels. The model's objective is to predict energy consumption, which depends on the vehicle's power requirements. It can be described by the fundamental longitudinal vehicle dynamics equations [30].

$$F_w(t) = F_r(t) + F_s(t) + F_d(t) + F_a(t) \quad (1)$$

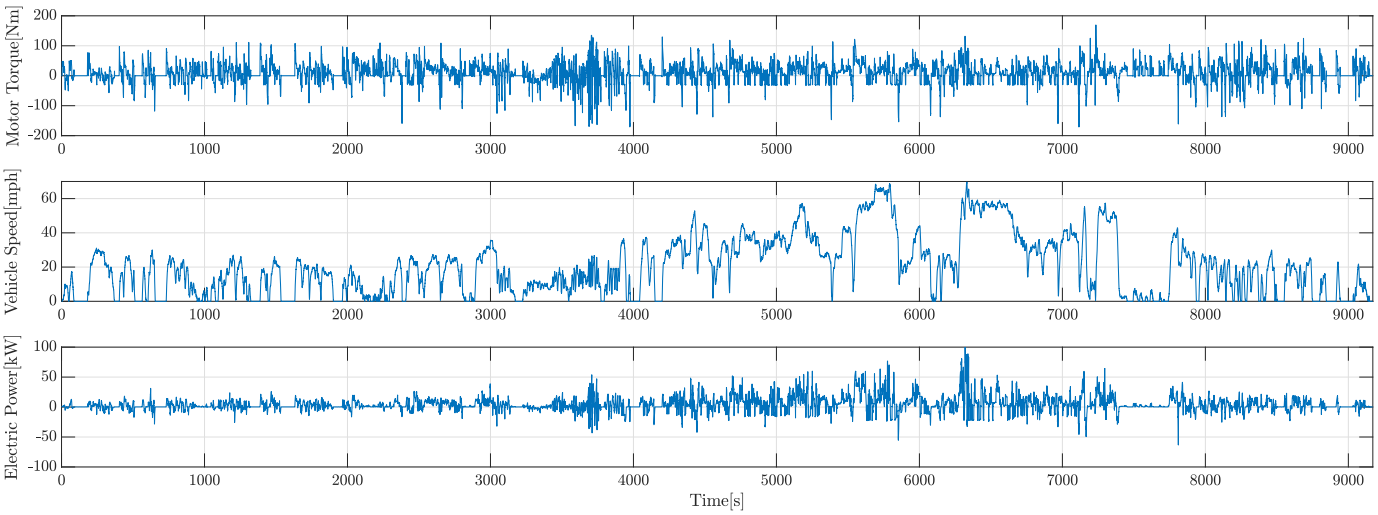


Fig. 5. Identification dataset from Trip 1 (5 March 2024), showing motor torque, vehicle speed, and electric power used to identify the models.

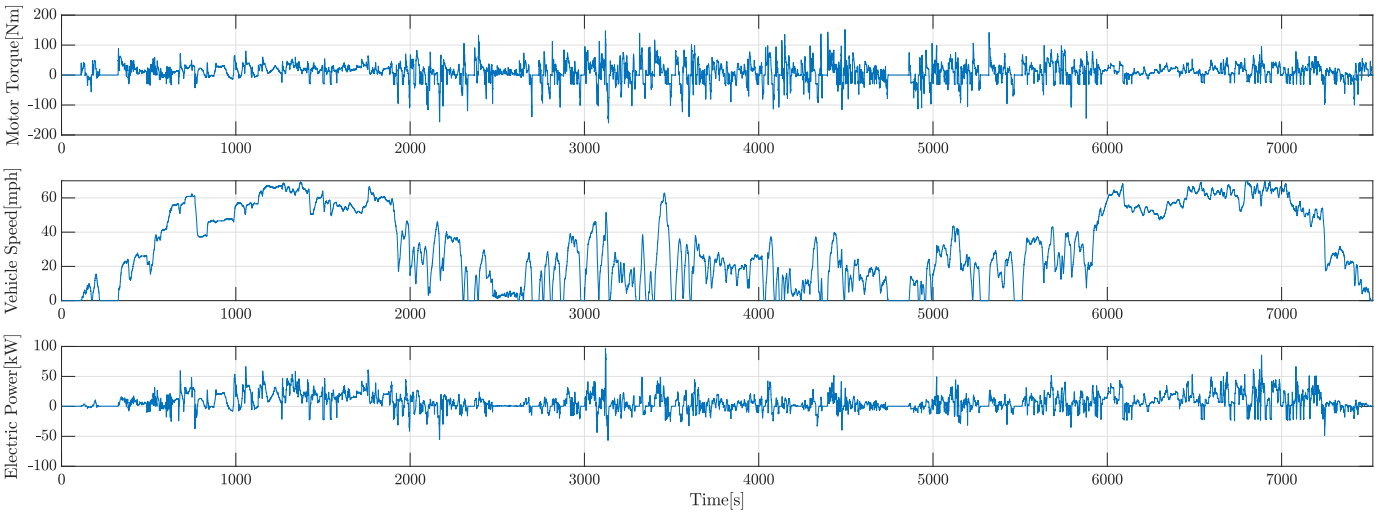


Fig. 6. Validation dataset from Trip 2 (7 March 2024), showing the measured signals used to assess the models' performance.

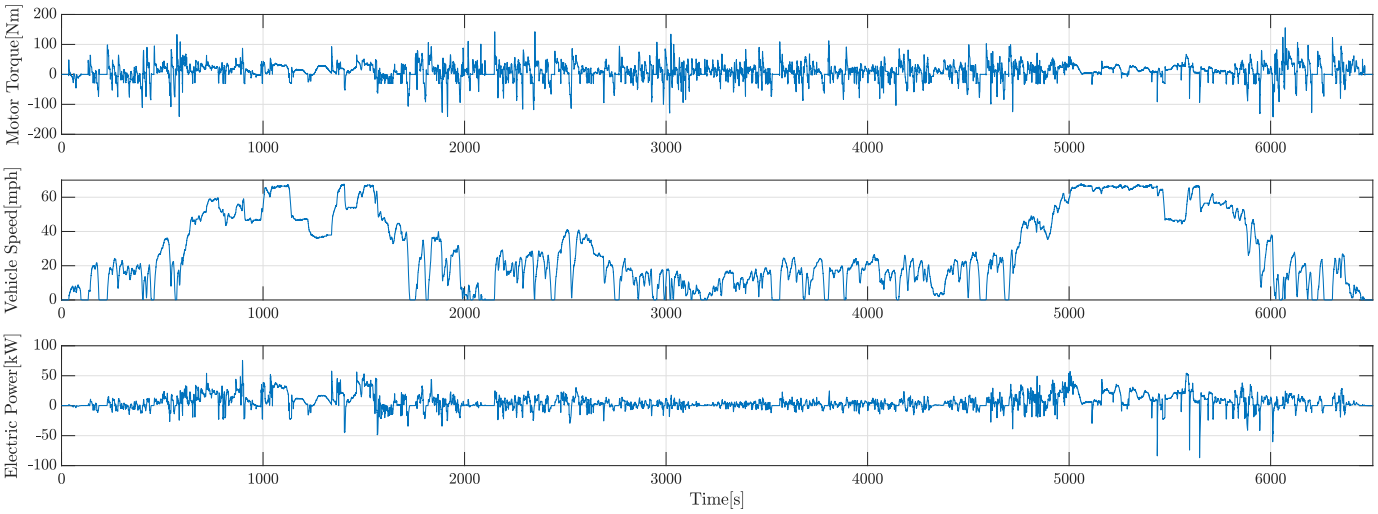


Fig. 7. Validation dataset from Trip 3 (8 March 2024), used to test the models under various driving conditions.

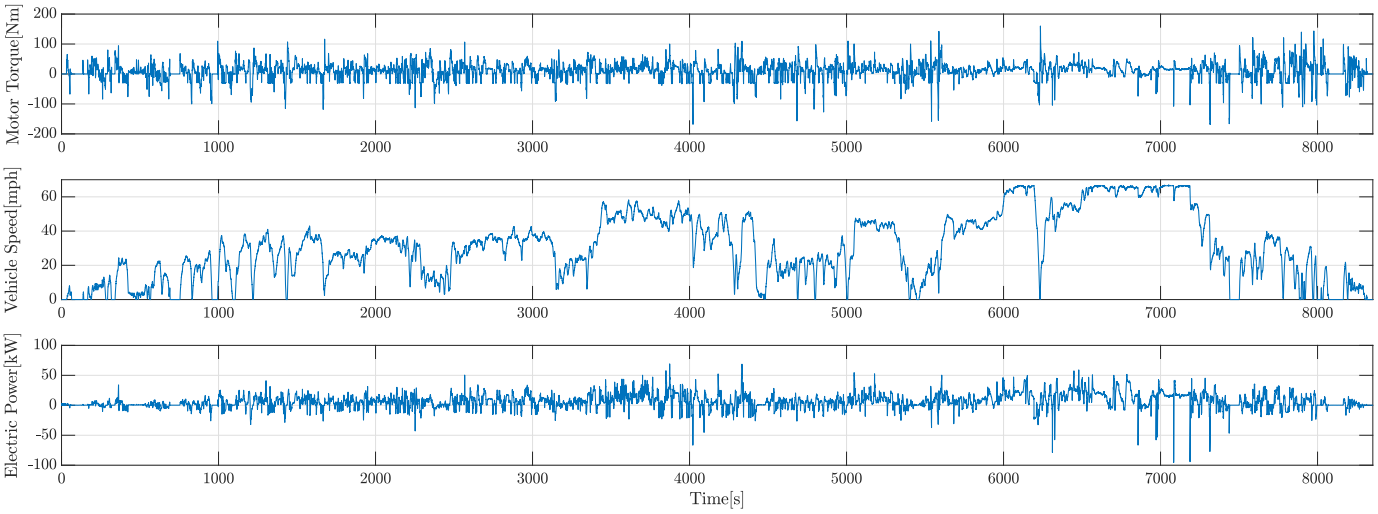


Fig. 8. Validation dataset (14 March 2024), used for further evaluation of the models on another long-duration trip.

$$F_w(t) = mgC_r \cos\alpha(t) + mg\sin\alpha(t) + \frac{1}{2}\rho C_d A(v(t) + v_{wn}(t))^2 + \lambda ma(t) \quad (2)$$

$$P_w(t) = F_w(t)v(t) \quad (3)$$

Here, $F_w(t)$ denotes the traction force required at the wheels. $F_r(t)$, $F_s(t)$, $F_d(t)$, and $F_a(t)$ are the forces needed to overcome tire rolling resistance, road slope, aerodynamic drag and to provide acceleration. m denotes the vehicle mass, g is the acceleration due to gravity, C_r denotes the coefficient of rolling resistance, $\alpha(t)$ denotes the road slope, ρ is the air density, C_d denotes the coefficient of aerodynamic drag, A is the vehicle's frontal area, $v(t)$ denotes the longitudinal speed, $v_{wn}(t)$ is the wind speed component in the longitudinal axis direction of the vehicle, λ is a rotational inertia coefficient, which is higher than one and indicates the proportion of total mass that is rotational, $a(t)$ is the longitudinal acceleration, and $P_w(t)$ is mechanical power required at wheels. The electric motor draws current from the battery pack through the inverter assembly. The motor converts electrical power into mechanical by generating torque based on the tractive force requirement, which is transmitted to the wheels via the drivetrain and can be represented using the following set of equations [31].

$$T_m(t) = \frac{F_w(t)r}{o_{dr}\eta_{dr}} \quad (4)$$

$$P_w(t) = P_e(t)\eta_m\eta_i\eta_{dr} \quad (5)$$

Here, $T_m(t)$ is the motor torque, r is the tire rolling radius, o_{dr} denotes the drivetrain gear ratio, η_{dr} denotes the overall drivetrain efficiency, $P_e(t)$ is the electrical power required, η_m is the efficiency of the motor, and η_i represents the efficiency of inverter in converting direct current into alternating current. η_m and η_i are usually nonlinear functions of the motor speed and torque [5]. From (3), (4), and (5), we can rewrite $P_e(t)$ as

$$P_e(t) = \frac{o_{dr}}{\eta_m\eta_i r}T_m(t)v(t) \quad (6)$$

In EVs, kinetic energy while braking can be recovered with the help of regenerative braking. This usually occurs when there is braking torque at the wheels, leading to negative motor torque. As a result, the motor acts as a generator and feeds regenerated electric power to the battery pack, which can be expressed in the following equation.

$$P_{w_b}(t) = F_{w_b}(t)v(t) \quad (7)$$

$$P_{rg}(t) = P_{w_b}(t)\eta_{rg}\eta_m\eta_i\eta_{dr} \quad (8)$$

$$T_m(t) = \frac{\eta_{rg}F_{w_b}(t)\eta_{dr}r}{o_{dr}} \quad (9)$$

Here, $P_{w_b}(t)$ and $F_{w_b}(t)$ are the braking power and force at wheels, $P_{rg}(t)$ is the regenerative electric power, and η_{rg} denotes the regenerative braking efficiency, which is the percentage of recoverable braking power. From (7), (8), and (9), we can rewrite $P_{rg}(t)$ as

$$P_{rg}(t) = \frac{\eta_m\eta_i o_{dr}}{r}T_m(t)v(t) \quad (10)$$

In the case of $P_e(t)$, it is inversely proportional to η_m and η_i, whereas $P_{rg}(t)$ is directly proportional to η_m and η_i because the flow of energy is reversed in regenerative braking. From (6) and (10), the electric power associated with the vehicle can be represented as

$$P(t) = \begin{cases} P_e(t) & \text{if } T_m(t) \geq 0 \\ P_{rg}(t) & \text{if } T_m(t) < 0 \end{cases} \quad (11)$$

$$P(t) = f(T_m(t), v(t)) \quad (12)$$

$$E(t) = \int_0^t P(t)\,dt \quad (13)$$

Here, $P(t)$ denotes the electric power flowing in or out of the battery pack. f is a nonlinear function of the motor torque and vehicle speed, and $E(t)$ denotes the overall energy

consumption. In equation (13), non-traction loads such as auxiliary and air conditioning are assumed to be negligible, so they are not considered in the modeling process and are outside the scope of this work. Equation (12) expresses $P(t)$ as a function of $T_m(t)$ and $v(t)$. By choosing $T_m(t)$ and $v(t)$ as the model inputs, we are modeling the combined efficiency map of the motor and inverter.

Motor torque is generated to propel the vehicle to overcome the combined effects of factors like wheel rolling resistance, road slope, aerodynamic drag, and vehicle rotational inertia, as shown in equations (2) and (4). Therefore, the motor torque, one of the model inputs, takes into account the impacts of these factors, making the model independent of their measurements or estimates. This flexibility benefits all on-road driving scenarios, where operating conditions such as road slope and wind speed can change considerably and significantly influence energy usage. On the other hand, applications which need a motor torque estimate at a given time may require the consideration of these factors. Another model is needed to determine the required motor torque, but it is outside the scope of this work.

B. Model structure

NARX is an extension of linear autoregressive with exogenous input (ARX), commonly used in many time-series applications. Following is the general equation of a linear single input and single output (SISO) ARX model in the discrete-time domain [32].

$$y(k) + a_1 y(k-1) + \cdots + a_{n_a} y(k-n_a) = b_1 u(k-n_g) + \cdots + b_{n_b} u(k-n_g-n_b+1) + h(k) \quad (14)$$

Here, k is the discrete time sampling number. $u(k)$, $y(k)$, and $h(k)$ are input, output and white noise, respectively. This structure implies that the current output $y(k)$ is predicted as a weighted sum of past output values, and past and current input values. n_a is the number of past output terms, n_b is number of past input terms, and n_g is input delay [33]. Putting (14) in a product form to compute the model output results in:

$$\hat{y}(k) = [-a_1, \cdots, -a_{n_a}, b_1, \cdots, b_{n_b}]. [y(k-1), \cdots, y(k-n_a), u(k-n_g), \cdots, u(k-n_g-n_b+1)]^T \quad (15)$$

Here, $[y(k-1), \cdots, y(k-n_a)]$ are delayed outputs and $[u(k-n_g), \cdots, u(k-n_g-n_b+1)]$ are delayed inputs, called regressors, and $[-a_1, \cdots, -a_{n_a}, b_1, \cdots, b_{n_b}]$ are coefficients applied to the regressors to estimate $\hat{y}(k)$. This equation represents linear mapping. In NARX, it is replaced by a nonlinear function, as shown below.

$$\hat{y}(k) = f_{nl}(y(k-1), y(k-2), \cdots, y(k-n_a), u(k-n_g), u(k-n_g-1), \cdots, u(k-n_g-n_b+1)) \quad (16)$$

Here, f_{nl} is a nonlinear mapping function. The proposed model in this article uses a sigmoid network as the nonlinear mapping function. It calculates the output using a combination of linear weights, an offset, and a sigmoid nonlinear function,

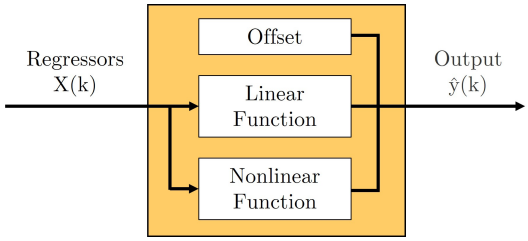


Fig. 9. Structure of the nonlinear mapping function in the NARX model, mapping regressors $X(k)$ to the estimated output $\hat{y}(k)$ using a combination of an offset, linear weights, and a sigmoid nonlinear function.

as shown in Fig. 9. This function maps all the regressors in $X(k)$ to estimated output $\hat{y}(k)$ with the following equation.

$$\hat{y}(k) = y_0 + X(k)^T ML + S(X(k)) \quad (17)$$

$$X(k) = [y(k-1), y(k-2), \cdots, y(k-n_a), u(k-n_g), u(k-n_g-1), \cdots, u(k-n_g-n_b+1)]^T \quad (18)$$

Here, $X(k)$ is an $l \times 1$ vector of regressors, where $l = n_a + n_b$ is the total number of regressors, y_0 is the scalar output offset, M is an $l \times p$ matrix with p denoting the number of linear weights, and L is a $p \times 1$ vector of weights. $S(X(k))$, the sigmoid nonlinear function, is a sum of dilated and translated sigmoid functions and is defined by:

$$S(X(k)) = \sum_{i=1}^{n} s_i \sigma(X(k)^T Q b_i + c_i) \quad (19)$$

Here, Q is an $l \times q$ projection matrix, s_i is the i^{th} output coefficient, $b_i \in R^{q \times 1}$ is the i^{th} dilation coefficient, c_i is the i^{th} translation and $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function, also called a unit function of the sigmoid network. The total number of sigmoid functions is the number of units, n, of the network [34]. As shown in Fig. 10, the $X(k)$ vector is normalized before being input into the nonlinear mapping function. Hence, a normalized output is computed from the nonlinear mapping function, which is then restored to its original range, producing model output $\hat{y}(k)$. This flow of information via the NARX structure is depicted in Fig. 10.

The NARX model can be used in two configurations: prediction and simulation. In prediction, the model forecasts the output one or more steps into the future based on the past inputs, $u(k-n_g), \cdots, u(k-n_g-n_b+1)$, and past measured outputs, $y(k-1), \cdots, y(k-n_a)$. A prerequisite for prediction is that past measured output is available. On the other hand, in the simulation configuration, the model computes output based on past inputs, $u(k-n_g), \cdots, u(k-n_g-n_b+1)$, and past simulated outputs, $\hat{y}_s(k-1), \cdots, \hat{y}_s(k-n_a)$, i.e. it does not need past output measurements. Both configurations are shown in Fig. 11 with input delay $n_g = 0$, $h_c(k)$ as noise and time shift operator $q^{-1}(u(k)) = u(k-1)$.

In order to estimate the model parameters mentioned in (17) and (19), an optimization problem is solved to minimize the sum of squares of error between measured and model output, expressed in (20).

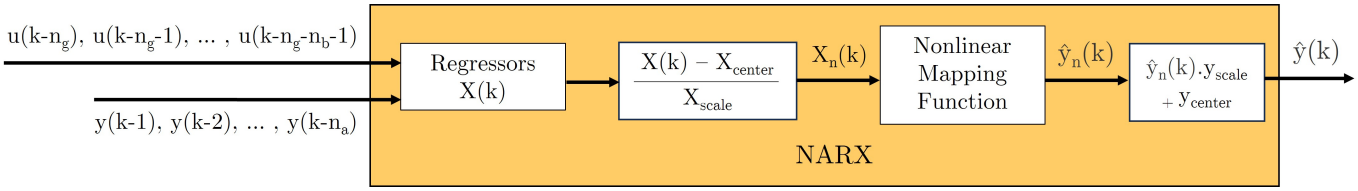


Fig. 10. Information flow in the NARX structure: the regressor vector $X(k)$ is normalized, passed through the nonlinear mapping function, and then denormalized to produce the estimated output $\hat{y}(k)$.

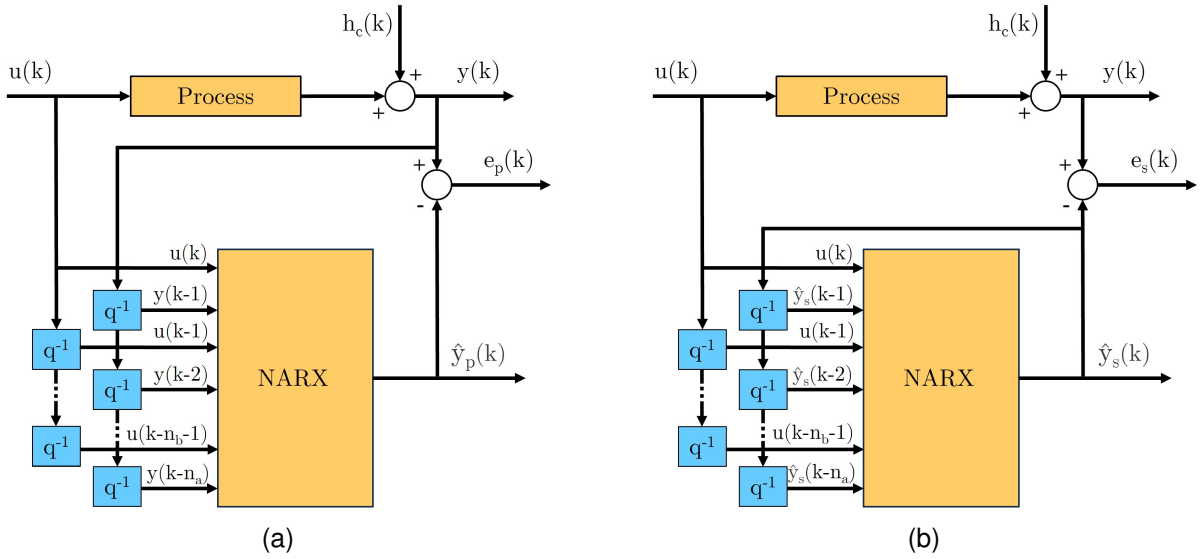


Fig. 11. NARX model configurations: (a) prediction, using past inputs and measured outputs and (b) simulation, using past inputs and simulated outputs.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_{p_i})^2 \tag{20}$$

Here, $\hat{\theta}$ represents the model parameter vector, N is the number of data samples, and $\hat{y}_{p_i}$ denotes the model output in prediction mode.

C. Identification and Validation

This article minimizes the error in the prediction configuration, as measured output is available. The optimization problem with the identification dataset (Trip 1, Fig. 5) was solved with the *nlarx* command using the Levenberg-Marquardt (LM) [35] least squares search method for 1000 iterations with the System Identification Toolbox in MATLAB [36]. The candidate model has an order of n_a = 3, n_b = [3 3], and n_g = [0 0]. These orders were determined by gradually increasing them from the lowest order until there was no significant improvement or deterioration due to overfitting in the average mean squared error (MSE) for the validation datasets.

Fig. 12 illustrates the identified NARX prediction model's performance for the identification dataset. The model achieved a 96.9% fit, calculated using the normalized root mean square defined as:

$$\text{Fit} = 100\% \left(1 - \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2}} \right) \tag{21}$$

and the MSE of 0.1261 defined as:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{22}$$

between the predicted output from the model and measured data. Here, $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} \hat{y}_i$ denotes the mean value of $\hat{y}_i$. These results are promising, considering the different driving conditions encountered during the journey and the nonlinearities involved in the powertrain dynamics. Fig. 13 shows a zoomed-in version of Fig. 12 from 3705 to 3740 seconds, where the most significant error is observed. An abrupt change in the accelerator pedal position between 3718 and 3719 seconds when the pedal is quickly released and pressed again coincides with a significant spike in prediction error, which implies that quick variations in motor torque may not be captured well at the current sampling rate.

Further observations include rapid releases of the accelerator around 3710 and 3729 seconds, leading to spikes in pedal rate and the engagement of regenerative braking. Energy regeneration can be seen in the plot, with electric power shifting to negative values indicating that energy is being fed back into the battery. This quick engagement of regenerative braking and the model's slow responsiveness in such situations probably contribute to the prediction errors observed here. Minor errors could also be due to not considering auxiliary power consumption. Table IV demonstrates that the model's overall performance remains robust despite these limitations.

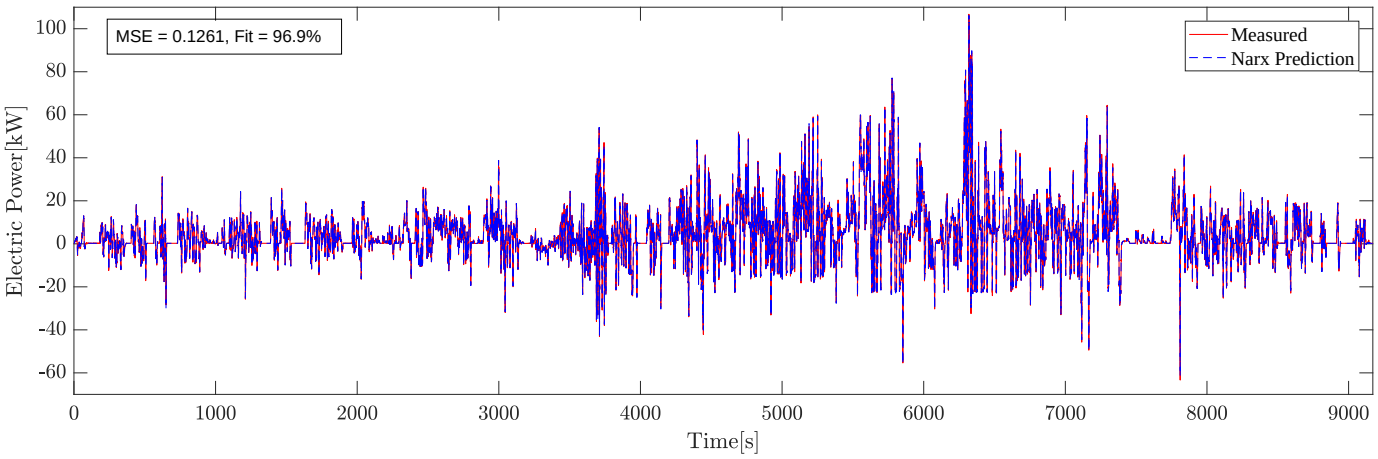


Fig. 12. Performance of the NARX prediction model on the identification dataset, showing measured and predicted electric power with 96.9% fit.

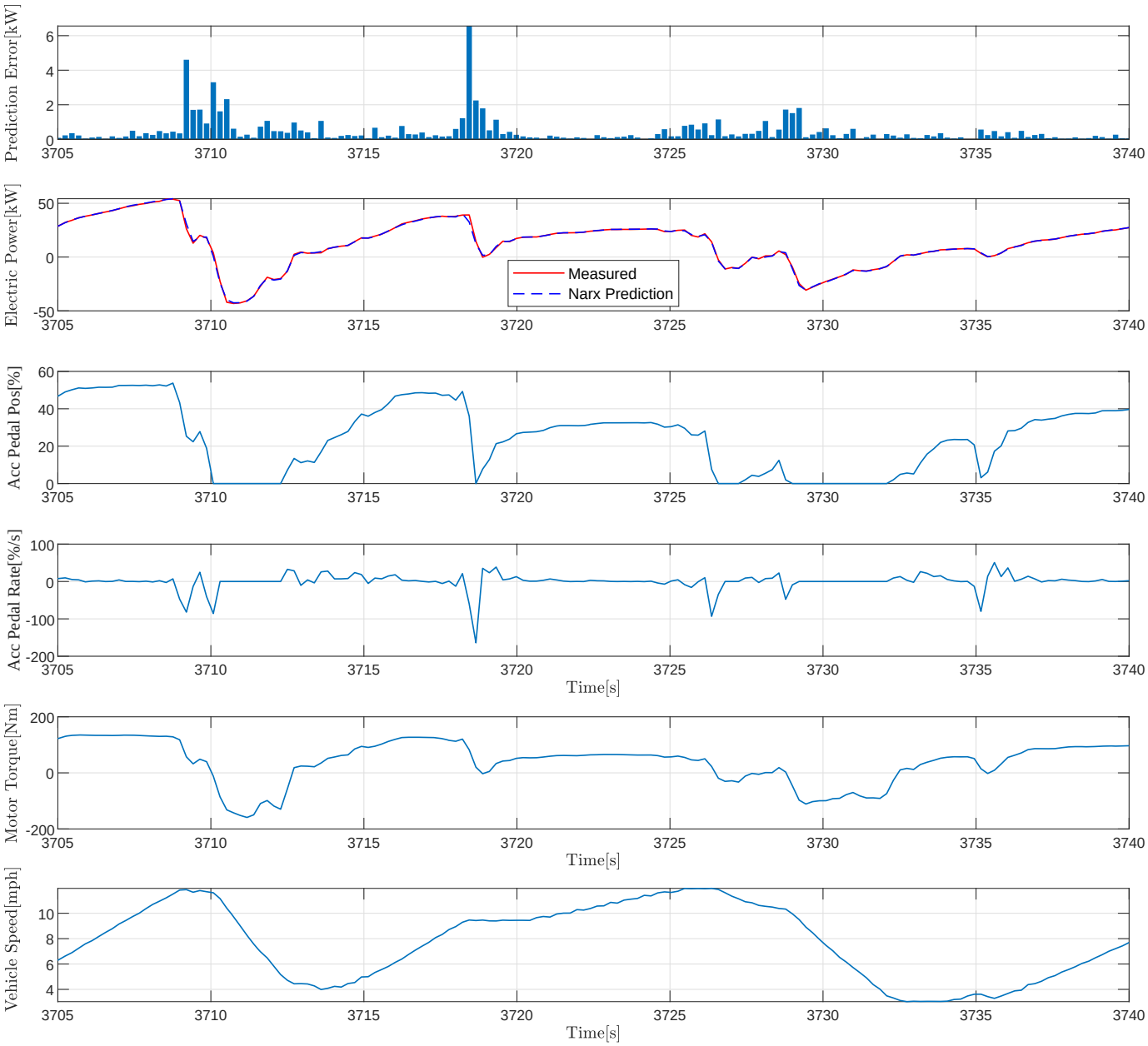


Fig. 13. Zoomed-in segment of the identification dataset from t = 3705 s to 3740 s, highlighting the most significant prediction error of the NARX model.

TABLE IV
NARX PREDICTION MODEL PERFORMANCE IN MSE.

Dataset				Average Validation MSE
Identification	Validation			
Trip 1	Trip 2	Trip 3	Trip 4	
0.1261	0.1556	0.1567	0.1755	0.1626

It illustrates the model's consistent performance with all the trip datasets. The closer the MSE is to zero, the better the model's performance and vice-versa for the fit performance.

Furthermore, an effort was made to enhance the model by including the accelerator pedal position as an additional input in the NARX prediction model. This structure modification reduced MSE by 12%. However, it did not enhance the total energy consumption estimation accuracy for the validation datasets and resulted in a 16% increase in the time needed for model identification. As a result, the accelerator pedal position was not included as an input in the model. In system identification (SI), the standard practice is to use the minimum number of inputs possible to satisfy the validation criteria, e.g., minimizing the average validation MSE.

The prediction performance for total energy consumption is shown in Table V. The most significant percentage error is 0.8%, making the minimum prediction accuracy of the model 99.2%. This table also illustrates how accurately the model captures system behaviors in both positive and negative energy consumption scenarios, which is important for an eco-driving controller to manage braking and acceleration while estimating and optimizing energy consumption.

Generally, models identified by SI techniques are validated using autocorrelation and cross-correlation criteria. However, these criteria are most appropriate for linear models. Model validation should address the objective of the intended application, especially with nonlinear models [37]. Cross-validation is a frequently used method for validating a model. It involves testing the model on new datasets that cover a broad range of operating conditions to check whether it performs well under real-world scenarios. The validation of the NARX prediction model over three different trips with distinct driving conditions demonstrates its robustness and ability to accurately capture the system's behavior in real-world driving scenarios.

IV. PERFORMANCE COMPARISON WITH ML MODELS

A neural network (NN) is a fundamental form of an artificial neural network (ANN) inspired by the workings of neurons in the brain. It is also known as a single-layer perceptron. As depicted in Fig. 14, it consists of an input layer, a single hidden layer of neurons, and an output layer [38]. Every input is connected to every neuron in the hidden layer, and all these neurons are connected to the output neuron. Each connection has a weight.

The algebraic calculations involved in an NN structure are as follows:

1) Input to hidden layer: The inputs are represented in vector form as $u = [u_1, u_2, \cdots, u_{n_n}]$, where n_n is the number of the inputs. The input given to each hidden neuron j receives a weighted sum of the inputs with a bias term b_j:

$$z_j = \sum_{i=1}^{n_n} w_{ij} u_i + b_j$$

Here, z_j is the weighted sum for the j^{th} hidden neuron, and w_{ij} is the weight that connects the i^{th} input to the j^{th} hidden neuron.

2) Activation function: The weighted sum z_j is passed through an activation function ϕ, which introduces nonlinearity in the neural network. ReLU is considered for this case, which is defined as:

$$a_j = \phi(z_j) = \max(0, z_j)$$

Here, a_j is the activation of the j^{th} hidden neuron.

3) Hidden to output layer: The weighted sum of activations from the hidden layer is then combined with output bias term c to calculate the final output. For a single output neuron, the output $\hat{y}$ is given by:

$$\hat{y} = \sum_{j=1}^{m_n} v_j a_j + c$$

Here, m_n is the number of neurons in the hidden layer, and v_j is the weight connecting hidden neuron j to the output.

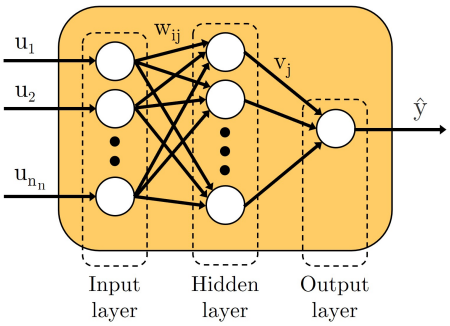


Fig. 14. Structure of a single-layer feedforward neural network (NN).

Performance of the proposed model is compared with several machine learning techniques, which utilize the same inputs, motor torque and vehicle speed, as those employed in the proposed model. This ensures a consistent basis for comparing the performance of all methods, including an NN, as demonstrated in Table VI and Table VII. Regression Tree (RT) [39], Ensemble Learning using Least-Square Boosting (LSBoost) [40], Robust Linear Regression (RLR) [41], Support Vector Regression (SVR) [42], and Single-Layer Feedforward Neural Network (NN) [43] were implemented utilizing the Statistics and Machine Learning Toolbox in MATLAB [44]. These models' training hyperparameters were first optimized by Bayesian optimization [45], followed by additional iterations to minimize the average validation MSE. For simplicity, from Section IV, "Training" will refer to identification for SI models and the process of training for ML models, as both involve optimizing model parameters to fit the data from Trip 1. The results in Table VI reveal that the NN, which has a single layer of

TABLE V
TOTAL ENERGY CONSUMPTION (EC) ESTIMATES USING THE NARX PREDICTION MODEL.

Dataset	Trip	Total EC [kWh]		Difference	Accuracy	Positive EC [kWh]		Difference	Negative EC [kWh]		Difference
		Actual	Model	[%]	[%]	Actual	Model	[%]	Actual	Model	[%]
Identification	1	11.82389	11.82400	0.0009	99.9991	16.1310	16.1253	0.04	-4.307	-4.301	0.1
Validation	2	15.768	15.823	0.3	**99.7**	19.268	19.332	0.3	-3.500	-3.508	0.2
	3	11.633	11.671	0.3	**99.7**	14.230	14.270	0.3	-2.597	-2.598	0.05
	4	15.784	15.915	0.8	**99.2**	19.329	19.461	0.7	-3.54505	-3.54529	0.007

TABLE VI
COMPARISON OF TRADITIONAL MACHINE LEARNING METHODS.

Method	Dataset				Average Validation MSE
	Training	Validation			
	Trip 1	Trip 2	Trip 3	Trip 4	
RT	0.3314	1.4139	1.9693	1.9250	1.7694
LSBoost	0.4086	0.7836	1.0859	1.1415	1.0037
RLR	0.4955	0.5114	0.4347	0.4835	0.4765
SVR	0.4524	0.4813	0.4136	0.4807	0.4585
NN	**0.4041**	**0.4702**	**0.4177**	**0.4843**	**0.4574**

TABLE VII
COMPARISON OF RECURRENT DEEP LEARNING METHODS.

Method	Dataset				Average Validation MSE
	Training	Validation			
	Trip 1	Trip 2	Trip 3	Trip 4	
GRU	0.1397	0.1632	0.2826	0.3981	0.2813
LSTM	**0.1437**	**0.1670**	**0.2173**	**0.3121**	**0.2322**

280 neurons, uses the rectified linear unit (ReLU) activation function [46] and is trained using limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) solver [47], outperformed the other machine learning methods by obtaining the lowest average validation MSE. This superior performance could be due to the large number of parameters in the model structure, which enables it to effectively capture the nonlinear relationship between the inputs and output from the measured data.

Comparison is also made with recurrent deep learning (DL) methods [48], specifically Gated Recurrent Unit (GRU) [49] and Long Short-Term Memory (LSTM) [50] were implemented using the Deep Learning Toolbox in MATLAB [51].

LSTM network was designed to handle the vanishing gradient problem of recurrent neural networks (RNN), enabling the model to capture long-term dependencies between inputs and outputs over time. As shown in Fig. 15, LSTM units contain a set of gates that regulate the flow of information [52].

The mathematical equations involved in a single LSTM unit are as follows:

1) Forget Gate F_k decides which information to discard from the cell state:

$$F_k = \sigma(W_F \cdot [H_{k-1}, U_k] + B_F)$$

2) Input Gate I_k decides which new information to add to the cell state:

$$I_k = \sigma(W_I \cdot [H_{k-1}, U_k] + B_I)$$

3) Candidate Memory Gate $\tilde{C}_k$ creates new candidate values that could be added to the cell state:

$$\tilde{C}_k = \tanh(W_C \cdot [H_{k-1}, U_k] + B_C)$$

4) Output Gate O_k decides which part of the cell state to output:

$$O_k = \sigma(W_O \cdot [H_{k-1}, U_k] + B_O)$$

5) Cell State C_k Update:

$$C_k = F_k \cdot C_{k-1} + I_k \cdot \tilde{C}_k$$

6) Hidden State H_k Update:

$$H_k = O_k \cdot \tanh(C_k)$$

Here, $\sigma(x) = 1/(1 + e^{-x})$ denotes the sigmoid function, $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$ denotes the hyperbolic tangent function, W_F, W_I, W_C and W_O are the weight matrices, and B_F, B_I, B_C and B_O are the bias vectors. U_k is the input and $\hat{Y}_k$ is the output at time step k, and H_{k-1} is the hidden state and C_{k-1} is the cell state from the previous time step.

The models in Table VII were trained with Adam Optimizer [53] using parameters ranging from 500 to 4000 epochs, learning rates of 0.1 to 0.01, and 10 to 50 hidden units, and the model with the lowest average validation MSE was selected. The results in Table VII show that a single layer of LSTM with ten hidden units exhibited higher performance compared to the traditional ML models in Table VI and the GRU model. This can be attributed to its structure, which consists of memory state, feedback and feedforward components, and enables the capture of temporal dependencies.

Performance of the LSTM model is compared with NARX models in Table VIII, and it is evident that the proposed NARX prediction model, denoted as NARXP, outperforms the LSTM model with a 30% reduction in MSE. The NARXP uses the past measured output as one of its inputs, a standard practice in SI, whereas the ML models do not use it. So, the table also includes the NARX simulation model, denoted as NARXS, which does not use the past output measurements to ensure a fair comparison. Both the NARX models outperform the ML models. The NARXS model takes longer training time as identifying the simulation model requires an extra step. The model is first identified in prediction mode, and the identified parameters are utilized as initial parameters in the second step, in which the model is again identified in simulation mode, i.e. without the past output measurements as input [37]. This extra step ensures that the performance of the simulation model is

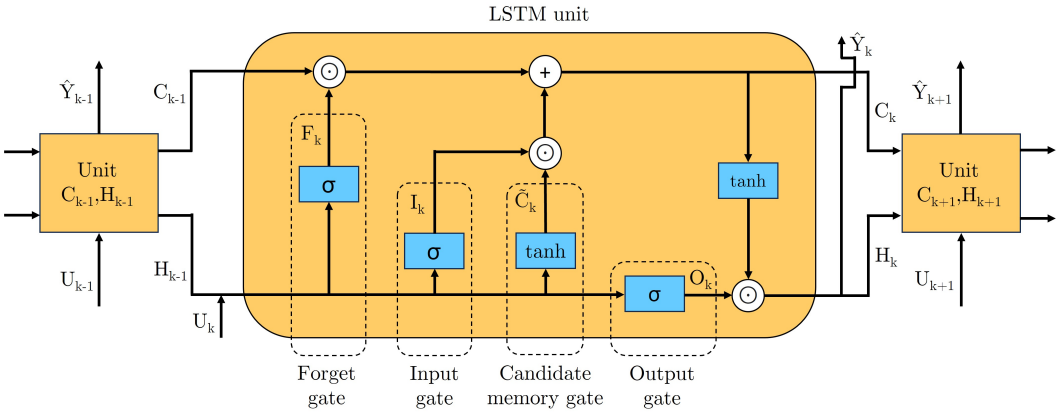


Fig. 15. LSTM unit structure, showing the gates that regulate information flow and enable learning long-term temporal dependencies.

TABLE VIII
PERFORMANCE COMPARISON OF LSTM AND NARX MODELS.

Method	Dataset				Average Validation MSE	Δ [%]
	Training	Validation				
	Trip 1	Trip 2	Trip 3	Trip 4		
LSTM	0.1437	0.1670	0.2173	0.3121	0.2322	
NARXS	**0.1310**	**0.1590**	**0.1418**	**0.1804**	**0.1604**	**-31**
NARXP	**0.1261**	**0.1556**	**0.1567**	**0.1755**	**0.1626**	**-30**

close to the prediction model, which is typically more accurate and stable due to the use of past output measurements. This SI practice is similar to the ML concept called transfer learning [54].

Since its inception, SI has aimed to develop mathematical models of dynamic systems from measured data. SI originated from the control systems community, which aims to understand, predict, and control these systems. In contrast, the machine learning community has focused on pattern recognition and prediction in various data types, which has broader applicability [55]. Still, ML models have a close relationship to the field of SI when they are utilized to estimate dynamic systems [56]. A critical difference between SI and ML techniques is the analysis of nonlinearities in dynamic systems. When temporal dependencies are considered, there are two types of nonlinearities: static and dynamic [37]. Static nonlinearities depend only on the current inputs. In contrast, dynamic nonlinearities depend on current and past inputs, exhibiting memory effects. This indicates that a system with dynamic nonlinearities responds based on the history of the previous inputs. The NARX model's superior performance could be due to past inputs impacting the current output, and therefore it accurately captures dynamic nonlinearities. ML models usually concentrate on current inputs, especially those not explicitly designed for time series data, making it challenging for them to assess dynamic nonlinearities. An exception like RNN models, such as LSTM networks, uses gates and memory cells to handle past information instead of directly using previous inputs. This approach is practical in handling long-term temporal dependencies. However, it may be less effective in dynamic systems where past inputs significantly influence the current output.

Additionally, ML models were trained with past inputs and outputs as additional inputs, similar to the NARX prediction configuration, to determine if they could capture the dynamic nonlinearities more effectively. This iteration aimed to facilitate a fair comparison between the structure of the NARX model and the ML models, given that the NARX model utilizes both past inputs and outputs. However, the LSTM model's performance did not improve significantly, presumably because the LSTM unit already attempts to grasp long-term and short-term dependencies through its memory and feedback components. With the addition of past inputs and outputs, memory cells may not have gained new information, making them redundant. NNs with past inputs and outputs were also trained in prediction configuration, denoted as NNP, and in simulation configuration, denoted as NNS, as depicted in Fig. 16 [57]. Both resulted in at least a 60% reduction in MSE compared to the NN model, which only used current inputs, as shown in Table IX.

From the results in Table VIII and Table IX, the performance of NN in prediction configuration, NNP, is comparable to NARX models and superior to LSTM model, which demonstrates that for this application, higher accuracy can be obtained by directly feeding previous inputs and outputs into the model. However, the number of trainable parameters of NNP is almost 11 times that of NARXP as shown in Table X, leading to a higher computation time, which was measured on an Intel Core i5-1235U CPU running at 1.30 GHz with 16 GB of RAM. This article uses the R2023b version of MATLAB. Despite both models performing similarly in prediction accuracy, the NARXP model stands out due to its more straightforward structure and higher computational efficiency. These results also highlight the potential benefits of integrating SI community practices into ML model training for dynamic systems.

V. CONCLUSIONS AND FUTURE WORK

This study proposes an approach for predicting the energy consumption of electric vehicles using the NARX model structure. The model uses motor torque and vehicle speed as inputs to estimate electric power as the output. Model

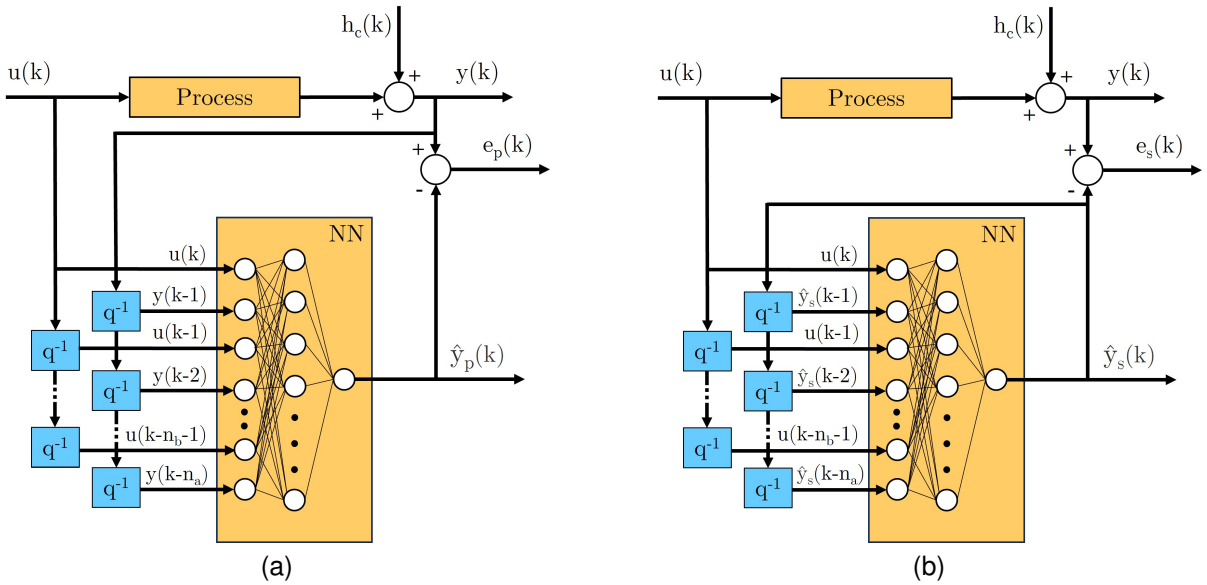


Fig. 16. NN model in system identification (SI) configurations: (a) prediction (NNP), using past inputs and measured outputs and (b) simulation (NNS), using past inputs and simulated outputs.

TABLE IX
NN PERFORMANCE IN PREDICTION AND SIMULATION MODE.

Method	Dataset				Average Validation MSE	Δ [%]
	Training	Validation				
	Trip 1	Trip 2	Trip 3	Trip 4		
NN	0.4041	0.4702	0.4177	0.4843	0.4574	
NNS	0.1235	0.1754	0.1635	0.2090	0.1827	-60
NNP	**0.1147**	**0.1615**	**0.1506**	**0.1847**	**0.1656**	**-64**

TABLE X
NARX AND NN COMPARISON IN PREDICTION CONFIGURATION.

Method	Average Validation MSE	No. of Model Parameters	Computation Time [ms]
NNP	0.1656	3081	3.4
NARXP	**0.1626**	**282**	**0.007**

parameters were identified and validated using real-world driving data from four trips from Southampton to different UK cities. The identified model was able to predict total energy consumption with at least 99.2% accuracy when validated across three validation trip datasets with a total duration of approximately 6 hours. The NARX model achieves this accuracy with significantly fewer parameters compared to ML methods, reducing complexity and the likelihood of overfitting. This modeling approach can be applied for modeling other vehicles, and adapted for energy management in powertrains.

The comparison of the NARX model with ML techniques included LSTM, which has memory state, feedback and feedforward components. The NARX model showed a 30% reduction in MSE compared to the LSTM model. The NARX model can be used in both configurations: prediction, which uses past measured output as an input, and simulation, which does not use past measured output as an input, achieving high accuracy in both settings. The NN with lower accuracy than the LSTM model achieved accuracy comparable to the NARX model when used in a prediction configuration. However, the NARX model proved more computationally efficient due to the number of parameters being an order lower than with NN in prediction configuration. This study also suggests that using insights from SI techniques may enhance the capabilities of an ML model in capturing dynamic nonlinearities. The models and results discussed in this article can be reproduced using the code available on GitHub [58].

Future work will incorporate the impact of air conditioning and auxiliary power on energy consumption, and explore the use of the NARX model in eco-driving applications and the online recursive system identification approach to account for changes in the model parameters as the vehicle ages.

REFERENCES

[1] B. Bilgin et al., "Making the Case for Electrified Transportation," *IEEE Transactions on Transportation Electrification*, vol. 1, no. 1, pp. 4–17, 2015. DOI: 10.1109/TTE.2015.2437338.

[2] M. Haghani, F. Sprei, K. Kazemzadeh, Z. Shahhoseini, and J. Aghaei, "Trends in electric vehicles research," *Transportation Research Part D: Transport and Environment*, vol. 123, p. 103 881, 2023. DOI: https://doi.org/10.1016/j.trd.2023.103881.

[3] Y. Chen, G. Wu, R. Sun, A. Dubey, A. Laszka, and P. Pugliese, "A Review and Outlook on Energy Consumption Estimation Models for Electric Vehicles," *SAE J. STEEP 2(1):79-96*, 2021. DOI: 10.4271/13-02-01-0005.

[4] I. Miri, A. Fotouhi, and N. Ewin, "Electric vehicle energy consumption modelling and estimation—A case study," *International Journal of Energy Research*, vol. 45, no. 1, pp. 501–520, 2021. DOI: https://doi.org/10.1002/er.5700.

[5] A. K. Madhusudhanan and X. Na, "Effect of a traffic speed based cruise control on an electric vehicle s performance and an energy consumption model of an electric vehicle," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 386–394, 2020. DOI: 10.1109/JAS.2020.1003030.

[6] E. Grunditz and T. Thiringer, "Modelling and Scaling Procedure of a Vehicle Electric Drive System," *Report, Chalmers University of Technology, Sweden*, 2017, Accessed: 26 Aug. 2024. [Online]. Available: http://publications.lib.chalmers.se/records/fulltext/253488/local_253488.pdf.

[7] X. Wu, D. Freese, A. Cabrera, and W. A. Kitch, "Electric vehicles' energy consumption measurement and estimation," *Transportation Research Part D: Transport and Environment*, vol. 34, pp. 52–67, 2015. DOI: https://doi.org/10.1016/j.trd.2014.10.007.

[8] A. K. Madhusudhanan, X. Na, and D. Cebon, "A Computationally Efficient Framework for Modelling Energy Consumption of ICE and Electric Vehicles," *Energies*, vol. 14, no. 7, 2021. DOI: 10.3390/en14072031.

[9] A. Di Martino, S. M. Miraftabzadeh, and M. Longo, "Strategies for the Modelisation of Electric Vehicle Energy Consumption: A Review," *Energies*, vol. 15, no. 21, 2022. DOI: 10.3390/en15218115.

[10] Z. Gao et al., "Evaluation of electric vehicle component performance over eco-driving cycles," *Energy*, vol. 172, pp. 823–839, 2019. DOI: https://doi.org/10.1016/j.energy.2019.02.017.

[11] L. Thibault, G. De Nunzio, and A. Sciarretta, "A Unified Approach for Electric Vehicles Range Maximization via Eco-Routing, Eco-Driving, and Energy Consumption Prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 463–475, 2018. DOI: 10.1109/TIV.2018.2873922.

[12] L. Koch et al., "Accurate physics-based modeling of electric vehicle energy consumption in the SUMO traffic microsimulator," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 1650–1657. DOI: 10.1109/ITSC48978.2021.9564463.

[13] D. Lee, A. Rousseau, and E. Rask, "Development and Validation of the Ford Focus Battery Electric Vehicle Model," *SAE Technical Paper 2014-01-1809*, 2014. DOI: 10.4271/2014-01-1809.

[14] R. Zhang and E. Yao, "Electric vehicles' energy consumption estimation with real driving condition data," *Transportation Research Part D: Transport and Environment*, vol. 41, pp. 177–187, 2015. DOI: https://doi.org/10.1016/j.trd.2015.10.010.

[15] C. Fiori, K. Ahn, and H. A. Rakha, "Power-based electric vehicle energy consumption model: Model development and validation," *Applied Energy*, vol. 168, pp. 257–268, 2016. DOI: https://doi.org/10.1016/j.apenergy.2016.01.097.

[16] J. Wang, I. Besselink, and H. Nijmeijer, "Electric Vehicle Energy Consumption Modelling and Prediction Based on Road Information," *World Electric Vehicle Journal*, vol. 7, no. 3, pp. 447–458, 2015. DOI: 10.3390/wevj7030447.

[17] O. Nelles, "Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes," in Cham: Springer International Publishing, 2020, pp. 1–20. DOI: 10.1007/978-3-030-47439-3_1.

[18] X. Liu and H. Jin, "High-precision transient fuel consumption model based on support vector regression," *Fuel*, vol. 338, p. 127 368, 2023. DOI: https://doi.org/10.1016/j.fuel.2022.127368.

[19] J. Zhang, Z. Wang, P. Liu, and Z. Zhang, "Energy consumption analysis and prediction of electric vehicles based on real-world driving data," *Applied Energy*, vol. 275, p. 115 408, 2020. DOI: https://doi.org/10.1016/j.apenergy.2020.115408.

[20] D. Kim, H. G. Shim, and J. S. Eo, "A Machine Learning Method for EV Range Prediction with Updates on Route Information and Traffic Conditions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, pp. 12 545–12 551, 2022. DOI: 10.1609/aaai.v36i11.21525.

[21] Z. Nie and H. Farzaneh, "Real-time dynamic predictive cruise control for enhancing eco-driving of electric vehicles, considering traffic constraints and signal phase and timing (SPaT) information, using artificial-neural-network-based energy consumption model," *Energy*, vol. 241, p. 122 888, 2022. DOI: https://doi.org/10.1016/j.energy.2021.122888.

[22] S. Modi, J. Bhattacharya, and P. Basak, "Estimation of energy consumption of electric vehicles using Deep Convolutional Neural Network to reduce driver's range anxiety," *ISA Transactions*, vol. 98, pp. 454–470, 2020. DOI: https://doi.org/10.1016/j.isatra.2019.08.055.

[23] A. K. Madhusudhanan, X. Na, D. Ainalis, and D. Cebon, "Engine Fuel Consumption Modelling Using Prediction Error Identification and On-Road Data," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1392–1402, 2023. DOI: 10.1109/TIV.2022.3167855.

[24] Y. Ma, H. Liu, Y. Zhu, F. Wang, and Z. Luo, "The NARX Model-Based System Identification on Nonlinear, Rotor-Bearing Systems," *Applied Sciences*, vol. 7, no. 9, 2017, ISSN: 2076-3417. DOI: 10.3390/app7090911.

[25] M. Alhanouti and F. Gauterin, "Predicting the Torque Demand of a Battery Electric Vehicle for Real-World Driving Maneuvers Using the NARX Technique," *World Electric Vehicle Journal*, vol. 15, no. 3, 2024, ISSN: 2032-6653. DOI: 10.3390/wevj15030103.

[26] X. Liu, D. Chen, W. Wei, X. Zhu, and W. Yu, "Interpretable Sparse System Identification: Beyond Recent Deep Learning Techniques on Time-Series Prediction," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=aFWUY3E7ws.

[27] *Road vehicles – Unified Diagnostic Services (UDS)*, ISO 14229-1:2020. [Online]. Available: https://www.iso.org/standard/72439.html.

[28] *Road vehicles – Communication between vehicle and external equipment for emissions-related diagnostics*, ISO 15031-1:2010. [Online]. Available: https://www.iso.org/standard/51828.html.

[29] Department for Environment, Food & Rural Affairs (DEFRA), *DEFRA Survey Data*, Accessed: 20 Oct. 2024. [Online]. Available: https://environment.data.gov.uk/survey.

[30] H. Naunheimer, B. Bertsche, J. Ryborz, and W. Novak, "Mediating the Power Flow," in *Automotive Transmissions: Fundamentals, Selection, Design and Application*. Berlin, Heidelberg: Springer, 2011, pp. 73–99. DOI: 10.1007/978-3-642-16214-5_3.

[31] M. Ehsani, Y. Gao, S. Longo, and K. Ebrahimi, "Electric Vehicles," in *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles*. Boca Raton, FL, USA: CRC Press, 2018, pp. 95–112. DOI: 10.1201/9780429504884.

[32] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, New Jersey, NJ, USA: PTR Prentice Hall, 1999.

[33] Mathworks, *What are Nonlinear ARX Models?* Accessed: 26 Aug. 2024. [Online]. Available: https://uk.mathworks.com/help/ident/ug/what-are-nonlinear-arx-models.html.

[34] Mathworks, *idSigmoidNetwork*, Accessed: 26 Aug. 2024. [Online]. Available: https://uk.mathworks.com/help/ident/ref/idsigmoidnetwork.html.

[35] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963. [Online]. Available: http://www.jstor.org/stable/2098941.

[36] Mathworks, *System Identification Toolbox™*. [Online]. Available: https://uk.mathworks.com/help/ident/.

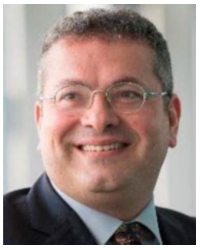
[37] J. Schoukens and L. Ljung, "Nonlinear System Identification: A User-Oriented Road Map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019. DOI: 10.1109/MCS.2019.2938121.

[38] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Heidelberg: Springer, 2006.

[39] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*.

New York, NY, USA: Springer, 2009. DOI: 10.1007/978-0-387-84858-7.
[40] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. DOI: 10.1214/aos/1013203451.
[41] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics - Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977. DOI: 10.1080/03610927708827533.
[42] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004. DOI: 10.1023/B:STCO.0000035301.49549.88.
[43] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, no. 5, pp. 551–560, 1990, ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(90)90005-6.
[44] Mathworks, *Statistics and Machine Learning Toolbox™*. [Online]. Available: https://uk.mathworks.com/help/stats/.
[45] P. I. Frazier, *A Tutorial on Bayesian Optimization*, 2018. arXiv: 1807.02811 [stat.ML].
[46] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15, 2011, pp. 315–323. [Online]. Available: https://proceedings.mlr.press/v15/glorot11a.html.
[47] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006. DOI: 10.1007/978-0-387-40065-5.
[48] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.
[49] K. Cho et al., *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, 2014. arXiv: 1406.1078 [cs.CL].
[50] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.
[51] Mathworks, *Deep Learning Toolbox™*. [Online]. Available: https://uk.mathworks.com/help/deeplearning/.
[52] C. Olah, *Understanding LSTM Networks*, Accessed: 26 Aug. 2024. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.
[53] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, 2017. arXiv: 1412.6980 [cs.LG].
[54] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. DOI: 10.1109/TKDE.2009.191.
[55] M. Schüssler, T. Münker, and O. Nelles, "Deep Recurrent Neural Networks for Nonlinear System Identification," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 448–454. DOI: 10.1109/SSCI44817.2019.9003133.
[56] L. Ljung, C. Andersson, K. Tiels, and T. B. Schön, "Deep Learning and System Identification," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1175–1181, 2020, 21st IFAC World Congress. DOI: https://doi.org/10.1016/j.ifacol.2020.12.1329.
[57] R. N. Vieira, P. Kollmeyer, M. Naguib, and A. Emadi, "Feedforward and NARX Neural Network Battery State of Charge Estimation with Robustness to Current Sensor Error," in *2023 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2023, pp. 1–6. DOI: 10.1109/ITEC55900.2023.10187084.
[58] M. B. Baluch, *NARX Model for EV Energy Consumption*, Accessed: 26 Aug. 2024. [Online]. Available: https://github.com/FMLabSouthampton/EV_EnergyConsumptionModel_v1_2024.

Mohamad Bilal Baluch received his B.Tech. Degree in Mechanical Engineering from the Sardar Vallabhbhai National Institute of Technology, Surat, India, in 2016. He is currently working toward a PhD degree in Mechanical Engineering at the University of Southampton, U.K. His research interests focus on developing real-time controllers to make Connected and Autonomous Vehicles safer and more energy efficient. Before this, Bilal worked at Maruti Suzuki India Ltd in Gurgaon, India, from 2018-2022 and at AVTEC Limited in Pithampur, India, from 2016-2018, where he was involved in designing and developing various transmission and drivetrain components to increase energy efficiency in the vehicle.



Suleiman M. Sharkh (Senior Member, IEEE) obtained his BEng and MSc from the University of Southampton in 1990 and 1994, respectively. He is Professor of Power Electronics, Machines and Drives at the University of Southampton. His research interests include electric machines, power electronics and their applications in transport, renewable energy and microgrids. He is a Senior Member of the IEEE, a member of the IET and a Chartered Engineer.

Anil Madhusudhanan received the B.Tech. degree in electronics and communication engineering from the National Institute of Technology Allahabad, India, in 2006, the MSc cum laude in Systems and Control and PhD from Delft University of Technology, The Netherlands, in 2011 and 2016, respectively. He is currently a Lecturer in future mobility systems with the University of Southampton, U.K. Before moving to Southampton, he was a Senior Research Associate (2020–2021) and a Research Associate (2017–2020) with the University of Cambridge, U.K. During 2015–2017, he was with the Netherlands Organisation for Applied Scientific Research (TNO), The Netherlands.

During 2006–2009, he was with STMicroelectronics (Greater Noida, India), Indian Institute of Science (Bengaluru, India) and Cranes Software (Bengaluru, India) for one year each.