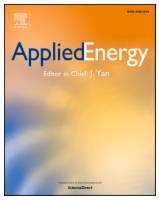
Applied Energy 389 (2025) 125783

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy



Real-time estimation of battery SoC through neural networks trained with model-based datasets: Experimental implementation and performance comparison

Giovanni Chianese [a], Luigi Iannucci [a,b], Ottorino Veneri [a], Clemente Capasso [a,*]

[a] National Research Council of Italy, Institute of Sciences and Technologies for Sustainable Energy and Mobility, Via Marconi, 4 – 80125, Naples, Italy
[b] University of Naples Federico II, Fraunhofer J-Lab IDEAS, Department of Industrial Engineering, P. le V. Tecchio, 80–80,125, Naples, Italy.

HIGHLIGHTS

- An electro-thermal battery model was calibrated and validated to generate training data for NN-based SoC estimators.
- Different NNs were trained with model-based data, tested with experimental data reproducing real operations, and compared.
- The trained NNs were verified for BMS applications, with implementation on a microcontroller and real-time SoC estimation.

ABSTRACT

Data-driven methods have been widely investigated to estimate battery SoC due to their great potential in solving regression problems. However, expensive experimental campaigns are generally required to collect large training datasets. To address this need, this paper demonstrates the advantages of using a validated battery simulation model to easily generate data for training neural networks (NNs) estimating SoC. Such a procedure drastically reduces the number of experiments, which are only required to calibrate/validate a physics-based battery model and to test the NNs in real driving operative conditions. A Li-NMC storage cell for automotive applications was considered as case study to verify the presented methodology. The analysis was performed in a wide range of operative conditions in terms of temperatures and load dynamics. Offline tests, based on data collected during experiments, showed that the trained NNs were able to predict the SoC with an accuracy comparable to NNs trained with standard experimental-based procedures. In the end, the trained NNs were implemented on a microcontroller to prove their real-time applicability in BMS boards.

1. Introduction

Due to the transition from fossil fuels to electric mobility, the automotive industry is facing challenges in supplying an increasing share of electric vehicles [1] with the related onboard intelligent battery management systems (BMSs). The BMS monitors and controls the state of batteries at cell, module, and pack levels. In this regard, the proper knowledge of state-of-charge (SoC) plays a key role in the secure use of the battery pack within optimal operating conditions [2]. In fact, the SoC is an internal state of battery and cannot be accurately measured directly during operation onboard of EVs. This poses challenges that are addressed by using different approaches to estimate it.

Direct measurements of the SoC include ampere-hour/Coulombic counting and look-up tables' based methods, whose implementation onboard of EVs during operation poses challenges. Indeed, the ampere-hour counting requires prior knowledge of the initial value of SoC when the operation starts and the updated value of the battery capacity. Additionally, it is prone to environmental perturbations which can lead to error cumulation in the SoC estimation during the integrative procedure [3,4]. Look-up tables require knowledge of open-circuit voltage (OCV), which cannot be measured in real-time during vehicle operations.

Model-based methods employed for battery cells SoC estimation involve either electro-chemical models (EMs) or equivalent circuit models (ECMs). EMs implement partial-derivative differential equations governing electrical and chemical phenomena occurring in the computational domain, enabling accurate deterministic solutions [5]. They require significant computational resources, which make their use impractical for real-time applications on microcontrollers. In ECMs, the battery cell is modelled through an electric circuit. Its use holds a good balance between the accuracy of the results and the computational effort required, providing a powerful tool in the design stage of the battery

* Corresponding author.
E-mail address: clemente.capasso@stems.cnr.it (C. Capasso).

https://doi.org/10.1016/j.apenergy.2025.125783
Received 10 December 2024; Received in revised form 21 February 2025; Accepted 18 March 2025
Available online 28 March 2025

pack [6]. Model-based methods are generally used in combination with adaptive filters such as the Kalman filter and its variants [7].

Data-driven approaches include machine learning (ML) algorithms that can automatically learn relationships between a response variable and input data. Many studies explored the application of ML algorithms, such as decision trees, random forests, support vector machine, neural networks (NNs), and hybrid methods, for SoC estimation [3,7]. In particular, a variety of typologies of NNs received great interest due to the various potentials they offer. Chemali et al. investigated feed-forward neural networks (FFNNs) for SoC estimation and reported that, on average, it was almost an order of magnitude faster than a methodology based on an extended Kalman filter [8]. Vidal et al. compared the performances of a long short-term memory recurrent neural network (LSTM-RNN) and FFNNs [9], by using large dataset consisting of current, voltage and temperature measured during electric tests on batteries reflecting standardized drive cycles at temperatures ranging between −10 and 25 °C. In [10], Guo and Ma compared the prediction accuracy, robustness to noise, and computational burden of state-of-the-art NNs, such as a FFNN, a LSTM, a gate recurrent unit (GRU) and a convolutional neural network (CNN). Jian et al. investigated mechanical stress measurements along with voltage and current to improve SOC estimation performances of a LSTM [11]. Use of NNs in hybrid approaches involving physics models and filter-based algorithms was also analyzed in several studies. Tian et al. proposed deep learning approach to estimate SoC in $LiFePO_4$ batteries and explored incorporation of the DNN in a Kalman filter to increase estimation robustness [12]. Tang et al. analyzed fusion of NNs with ECM to increase SoC estimation accuracy in wide temperature range [13]. With respect to the importance of data, in their review on the use of ML to estimate the SoC [14], Vidal et al. remarked the importance of using datasets that are representative of all the possible operative conditions. Indeed, a large set of data, comprehensive of all the plausible environmental and load conditions, is required for training SOC estimators and to achieve a proper prediction accuracy and reliability. In this regard, Do Reis et al. [15] listed and reviewed different battery datasets existing in the literature, concluding that their distribution is still sporadic despite their critical role in the advancement of data-driven methods. This gap can be explained considering that the experimental testing procedure necessary to generate a comprehensive dataset is a resource and time-consuming task, which cannot be completely automated as it would require human supervision and data organization.

The need for adequate datasets reflects data hungriness of data-driven approaches, especially those which employ NNs [7]. Recognized scarcity of experimental data led researchers such as Channegowda et al. to investigate synthetic data augmentation and generation [16]. To efficiently cope with this requirement, Li et al. investigated synthetic training data generation with a Pseudo-2-Dimensional EM as an efficient method to obtain a training dataset of a suitable size [17]. Conveniently defined random current loads were considered to simulate voltage signals and SoC conditions to obtain a concise training dataset for various ML algorithms. However, no reference to experimental data for the test of the trained models was reported. Hattouti et al. tested the capability of a second-order ECM to generate reliable training data for LSTM-based SoC estimators [18], but further developments are needed as their study did not consider neither charging with regenerative breaking nor temperatures different than 25 °C. A similar approach was developed also in the context of industrial research [19].For all these reasons, generation of data by means of physics-based simulation is a promising topic with further investigations needed. Research interest in such a topic stems from the opposing need to have an exhaustive dataset for training of the SoC estimator and the valuable advantage of having it in a significantly reduced time.

Starting from the context outlined above, this work aims to bridge the research gaps highlighted in the literature analysis. In the first place, the proposed research covers the lack of experimental knowledge related to the full demonstration and validation activities on 1:1 scale automotive applications. In addition, the innovation and the contribution introduced in this paper are both related to a comprehensive investigation on the use of physics-based simulated data from experimentally validated simulation models to train neural networks. Such procedure leads to a significant and valuable reduction in time and effort required for experimental measurement campaigns. Indeed, the proposed methodology leverages physics-based data simulation and leads to the advantage of not requiring such an amount of data for training, with only characterization tests needed. It is worth recalling that typical procedures for the implementation of NNs require large sets of data. Those sets of data generally consist of more than 100,000 datapoints based on the specific application and accuracy required, with the majority of them (about 70–80 %) used for training. For instance, El Fallah et al. in [20] used a dataset with about 300,000 data-points to implement NNs for SoC estimation. Implementation of the proposed methodology would leverage characterization tests instead, which can be generally carried out in different stages of development of battery systems.

This study encompasses all the steps from data simulation to on-field final deployment of the estimators on a microcontroller to predict the SoC of a Li-ion battery cell. The performances of three NNs in SoC estimations were evaluated and compared on real data collected during a reduced experimental campaign on single battery cells. The implementation of the trained estimators on a BMS board for Li-NMC vehicle battery packs demonstrated the methodology as a case study to evaluate its real-time performance under operative conditions representative of driving cycles. The advantage of the proposed procedure also consists in its high level of modularity since it can be easily re-adapted to batteries with different size and applications by the simple recalibration of the battery simulation models able to generate new training datasets.

The resume of this paper is organized as it follows. Section 2 gives some background on the most used NN typologies and their application in SoC estimation; Section 3 describes the considered case study along with materials and methods, including the proposed procedure, the modelling approach, and the NNs trained; Section 4 analyses and discusses the obtained results for the case study; Section 5 resumes the main results and conclusions.

2. Background on NNs for battery status estimation

In this section, a description of the working principles and advancements in battery SoC estimation is provided for some of the most comprehensively explored NNs, such as FFNNs, LSTM-RNNs, and CNNs.

2.1. Feed-forward neural networks

The Feed Forward Neural Network (FFNN) is the simplest form of NN [21] and enables mapping of non-linear relations between input and response variables through forward sequential processing of the data by layers of artificial neurons, as shown in Fig. 1 (B). The artificial neuron,

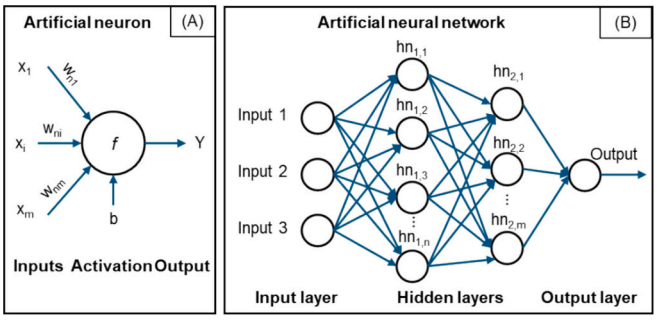


Fig. 1. Schematic representation of the artificial neuron (A), and of the multilayer NN (B).

schematized in Fig. 1 (A), processes numeric inputs coming from the previous neurons and returns an output according to eq. (1), where x_i is the input from the *i-th* neurons within the previous layer, w_i is the weight associated with the *i-th* neuron, b is an adjusting parameter, termed bias, and f is the activation function. Values of biases and weights are tuned during the training process to map the inputs with the reference output of each observation in the training dataset, therefore, they are also indicated as learnable parameters.

$$Y = f(b + \Sigma_i(w_i \cdot x_i)) \quad (1)$$

One of the greatest advantages of using FFNNs consists of its inherent simplicity [9], which results in lower computational time/effort required to perform inferences [21]. For this reason, their potential has been widely investigated for a variety of tasks including SoC estimation for electric vehicle battery packs. In particular, in [8], Chemali et al. trained deep NNs to estimate the SoC of a small cylindrical 18,650 Li-ion battery cell, using as input voltage, current, and temperature raw measures in combination with averaged voltage and current measures. The use of averaged measures as additional input was motivated considering that FFNNs do not have inherent memory on data processed previously. The same group, in another study [9], compared the effects of moving-window averaging operation and Butterworth filtering. Based on similar motivations, Zhao et al. employed voltage variation as additional input for the NN to estimate the SoC of lithium-ion ESS [22]. Hannan et al. focused on the impact of hyper-parameters on the performances by demonstrating the feasibility of a backtracking search algorithm in the optimization of the number of hidden layers and neurons [23]. To demonstrate scalability of their methods to real-case applications, Chen et al. reported on deployment of an FFNN-based method for real-time SoC estimation, with a hardware-in-the-loop validation procedure, under various temperature conditions, obtaining an average error below 2 % [24].

Despite the advantages deriving from their simplicity, FFNNs can neither track information gathered in previous time steps nor recognize features in the signals that were learned during the training process, leading to lower values of accuracy compared to more complex types of NNs, as reported in [10,14]. For these reasons, applications based on LSTM-RNNs and CNNs have been also investigated for SoC estimation in the scientific literature [7].

2.2. Long-short term memory recurrent neural networks

Recurrent Neural Networks (RNNs) can recognize and process dependencies within sequential inputs. Indeed, the current output of a recurrent neuron depends on the current input and on its status at the previous time-step, as schematized in Fig. 2 (A).

Among recurrent neural networks, LSTM-RNNs are one of the most used for SoC evaluation because they are characterized by a memory cell

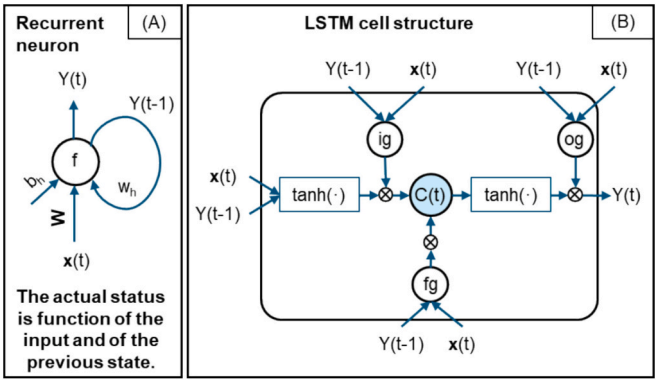


Fig. 2. Schematic representation of a recurrent neuron (A), and of an LSTM cell (B).

that can hold information across multiple timesteps, providing a solution to issues related to vanishing/exploding gradients during the training process. As shown in Fig. 2 (B), information in the LSTM cell flows based on the activation of input (*ig*), output (*og*), and forget gates (*fg*), depending on the current input, $\boldsymbol{x}(t)$, and the neuron state, $Y(t-1)$, at the previous timestep, according to eqs. (3–7).

$$ig = \sigma\left(W_h^i \cdot Y(t-1) + \boldsymbol{W}^i \cdot \boldsymbol{x}(t) + b^i\right) \quad (3)$$

$$og = \sigma\left(W_h^o \cdot Y(t-1) + \boldsymbol{W}^o \cdot \boldsymbol{x}(t) + b^o\right) \quad (4)$$

$$fg = \sigma\left(W_h^f \cdot Y(t-1) + \boldsymbol{W}^f \cdot \boldsymbol{x}(t) + b^f\right) \quad (5)$$

$$C(t) = fg \odot C(t-1) + ig \odot tanh\left(\boldsymbol{W}^c \cdot \boldsymbol{x}(t) + W_h^c \cdot Y(t-1) + b^c\right) \quad (6)$$

$$Y(t) = og \odot tanh(C(t)) \quad (7)$$

Chemali et al. analyzed the impact of the number of hidden units in the LSTM layer on the accuracy of SoC estimation, at fixed and variable temperatures [25]. They observed that when the state of the hidden layer is not properly initialized, the stateful prediction of the SoC to converges to the correct value after a certain amount of time depending on number of LSTM hidden units. Based on similar evaluations, Tian et al. [26] combined the adaptive cubature Kalman filter with LSTM to smooth oscillations and improve initial convergence in the SoC estimation task. Ren et al. focused on the automatic optimization of the number of hidden layer neurons and learning algorithm parameters for a LSTM-RNN by using the particle swarm optimization [27]. Ma et al. focused on the simultaneous estimation of SoC and SoE using LSTM-RNNs in different temperatures and driving operations [28]. They evaluated the accuracy and the computational effort, in terms of computing time, for different numbers of hidden units.

2.3. Convolutional neural networks

During the last decades, CNNs have become the de facto standard in various computer vision and image processing tasks [29]. In principle, they consist of two parts: one dedicated to feature extraction, and a second part, processing the information extracted from the input to compute the output. Features extraction is performed thorough stacks of convolutional and subsampling layers [3]. In each of these stacks, a specified number of filters, also known as *kernels*, slide over the input and perform convolutions. Size of extracted feature map is reduced with pooling operations. The output of these sequential operations is processed by fully connected layers to compute the output of the CNN.

CNNs have been also used to process time series with both 1-D convolutional and pooling layers [29], which operate according to the working principles described above, as schematized in Fig. 3. Many papers presented in the scientific literature focused on the estimation of battery status through CNNs. In particular, Bhattacharjee et al. trained CNNs with data available from two open-access datasets and explored transfer learning as a method to improve the generalization capability of the considered NN [30]. Chen et al. proposed the use of a self-attention layer in combination with two convolutional stacks to predict battery SoC using voltage, current, and temperature measurements. The performance of their 1D-CNN-Attention network was validated on available datasets composed of dynamic driving cycles at various temperature values [31]. Liu et al. trained a particular type of 1-D CNN, which is indicated as temporal CNN (TCN) [32]. It ensures causality and enables recognition of longer dependencies in time during convolution of the input without a significant increase in the number of learning parameters. Kim et al. [33] adopted an even different approach to estimate SoC with voltage, current, and temperature measured signals as input. They trained 2-D CNNs with time-frequency spectrograms instead of automatically extracting features with 1-D convolutional layers.

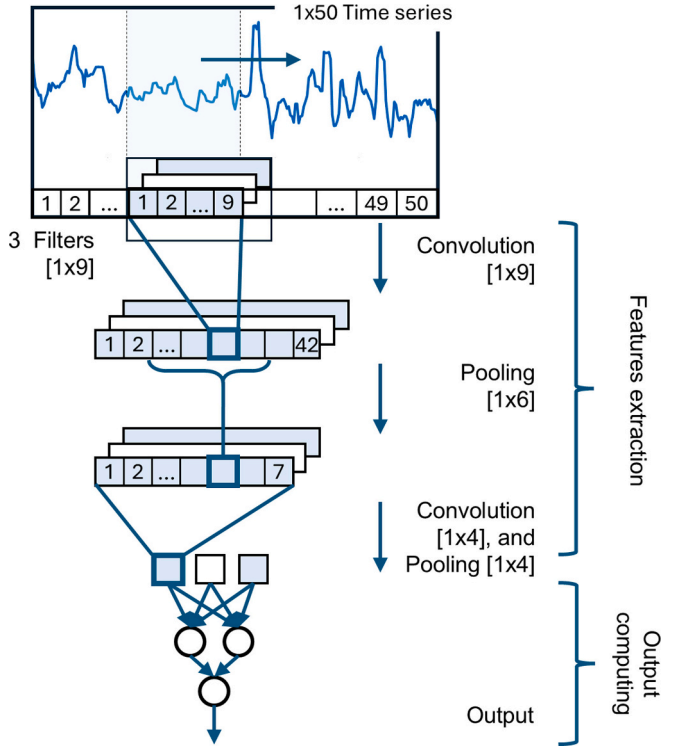


Fig. 3. Feature extraction and output computing operations in a conceptual representation of a 1-D CNN (particular case of specified dimensions of the 1-D input, filters and pooling window).

3. Materials and methods

3.1. Proposed methodology

Training of NN generally requires a large number of experimental data, coming from long and expensive laboratory measurement campaigns. To limit the data collection effort and simplify the overall training procedure, this paper investigated the feasibility of training NN-based SoC estimators by using data generated with an experimentally validated simulation model of the battery cell under study at different temperatures and dynamic load profiles. The workflow followed to address this purpose is shown in Fig. 4.

The experimental campaign is limited to the tests needed to perform the calibration/validation of electro-thermal model parameters and to create testing datasets for the SoC estimator. In particular, experiments used for testing reflected operating conditions with standardized driving cycles at various environmental temperatures. Once the simulation model was completely validated, training datasets were generated by taking advantage of the computational capability of the high-performance computer processors. Then, different typologies of deep NNs were trained using the model-based training datasets. The NNs were tested by using the experimental test dataset. Performances of the NNs were compared in terms of accuracy, robustness to noise in the signals, and computational cost, before being implemented on a microcontroller to test and demonstrate their validity for applications in BMS of electric vehicles.

It is important to highlight that model-based and NNs are both valuable approaches for the evaluations of SoC, as also confirmed by many papers available in the literature on both subjects [4]. In terms of cost-efficiency, the only use of the physics-based model cannot be applied, since the initial conditions of SoC are not known. In this case, the proper design and implementation of an observer/fitting procedure, after battery model parametrization, would be required. The overall estimation procedure is then required to run in real time on the onboard

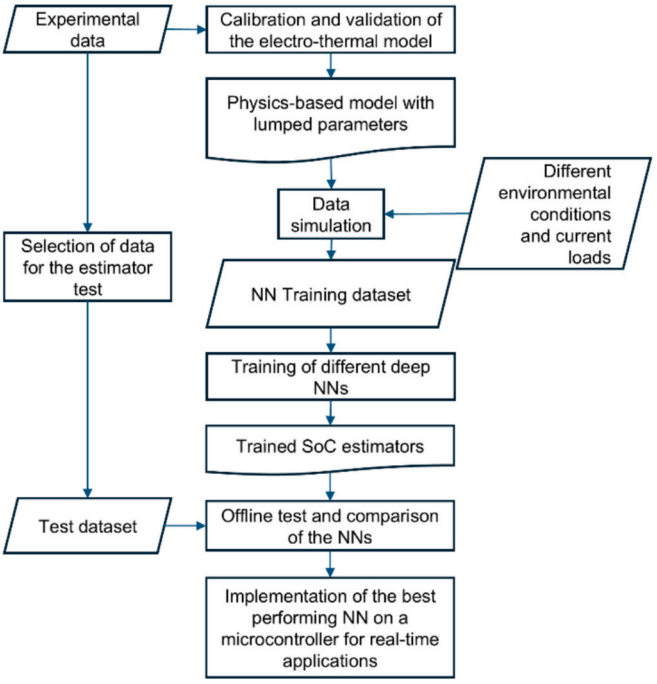


Fig. 4. A workflow for the training and validation of NNs by leveraging reduced experimental campaigns and simulations.

microcontroller. On the other hand, the real effort of the proposed approach consists of automated procedures to be performed offline just one time on dedicated hardware to obtain dataset generation and training of NNs. The computational effort required once neural networks are trained is particularly low, as also highlighted by the computational time evaluated for the on-board system.

3.2. Case study

This study considers a battery cell typically employed in battery packs for automotive applications, equipped with a BMS board implementing the SoC estimator. That cell has a nominal capacity of 20 Ah, rated voltage of 4.2 V, and geometric dimensions equal to 220 mm × 150 mm × 7 mm. Full specifications are reported in Table 1. As a brand-new battery cell was employed for the experimental campaign, SoH was assumed equal to 100 %.

The cell is equipped with an STM32-Nucleo-F401RE, which is used as a reference embedded micro-board to demonstrate the suitability of the proposed methodology for a real BMS board. Its main characteristics are reported in Table 2.

In particular, the board is equipped with 64 pins, which can be configured as either analog or digital input/output ports, to interface the system with the battery voltage, current, and temperature real-time measurement signals. In addition, this board can be directly interface with Matlab/Simulink environment through specific toolboxes

Table 1
Specifications of the tested Li[NiCoMn]O_2 battery cell.

Cathode chemistry	NMC (4:4:2)
Electrolyte chemistry	BASF (LP50) mixed with EC and DMC (1:1)
Nominal voltage	3.6 V
Nominal capacity	20.0 Ah
Specific energy	180 Wh/kg
Maximum charge voltage	4.2 V
Maximum charge current	1C
Lower voltage limit	2.5 V
Maximum discharge current	5C
Maximum peak discharge current	10C

Table 2
Specifications of the STM32-Nucleo-F401RE board.

Clock	32.768 kHz crystal oscillator
Flash memory size	512 kbytes
Number of pins	64
Connectors for the ST-LINK	USB Type-C®, Micro-B, or Mini-B
ADC Resolution	12 bits

dedicated to *ST Microelectronics* devices. Details about the implementation of the estimator on the microcontroller are reported in a dedicated subsection in the following.

3.3. Experimental setup

Following the workflow described in Fig. 4, an experimental campaign was designed to obtain datasets addressing two different tasks. The first task is focused on the complete calibration of a high-fidelity electro-thermal model of the battery cell under test, operating in various temperature conditions. The second task is focused on the construction of a small-size experimental dataset, which was just used for the validation of the electro-thermal model and the testing of NN-based SoC estimators.

Considering the above tasks, the test bench for the electric characterization/validation experiments was based on controlled bidirectional power suppliers, able to perform charging/discharging operations on the storage cell under test with a maximum current of 240 A. The power suppliers are controlled with the ITECH-ITS5300 remote software. This software allows the implementation of charging/discharging cycles representative of either standard model calibration or real EV load profiles. All the experimental tests were performed in an ACS Discovery Climate Chamber to ensure constant ambient temperature equal to 283 K, 298 K, and 313 K. Picture and functional schemes of the above experimental test bench are reported in Fig. 5.

The acquisition and control systems were based on a National Instruments Compact RIO NI 9056 platform, operating, for the tests reported in this paper, at a 5 Hz sampling frequency. The NI cRio was equipped with a thermocouple NI 9210 module and an IO analogue voltage NI 9207 module with a resolution of 24 bits. Battery cell voltage has been measured through direct connection of the battery terminals to the NI 9207 module, through fuse protections; a LEM LA 125 P/SP3 Hall effect current transducer has been used for current measurements; 5 thermocouples were fixed on different points of the battery surface and 1 thermocouple was used for the environmental temperature measurement. Further details on the considered test bench are reported in [6].

3.4. Calibration and validation of the battery multi-physics model

In this work, the lumped parameter electro-thermal model presented in [6] is employed to simulate data related to the electro-thermal behavior of the cell under study. For the proposed data-driven application, a high level of simulation accuracy is required to ensure reliable patterns within training data for regressions. Therefore, the model parameters were completely re-calibrated by taking advantage of the high-resolution 24 bit voltage acquisition module. Parameters' tuning also considered a wider range of temperatures spanning from 283 to 313 K.

The simulation battery cell model consists of a first-order Thevenin circuit coupled with a one-state lumped thermal model, schematized in Fig. 6 (A) and (B). The cell characterization process started from the evaluation of actual cell capacity at 283 K, 298 K, and 313 K for a complete 1C constant current discharge. The proper calibration of the following pulse discharge tests was carried out on the basis of the evaluated actual capacity values. At this stage, reparameterization of the ECM was carried out using pulse discharge tests at the considered temperature values to tune the open-circuit voltage (OCV), the internal resistance (R_0), the polarization resistance (R_1), and the polarization capacity (C_1). As an example, a pulse discharge test performed at 298 K is reported in Fig. 7 in terms of battery voltage, current, and SoC vs Time.

In this case, a higher number of levels of SoC, compared to [6], spanning from 0 % to 100 % with a step of 5 % were considered.

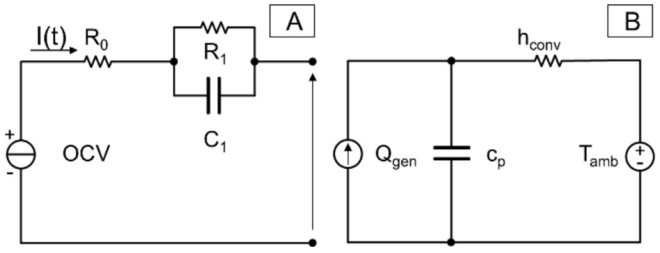


Fig. 6. First-order ECM (A) and a one-state lumped thermal cell model (B).

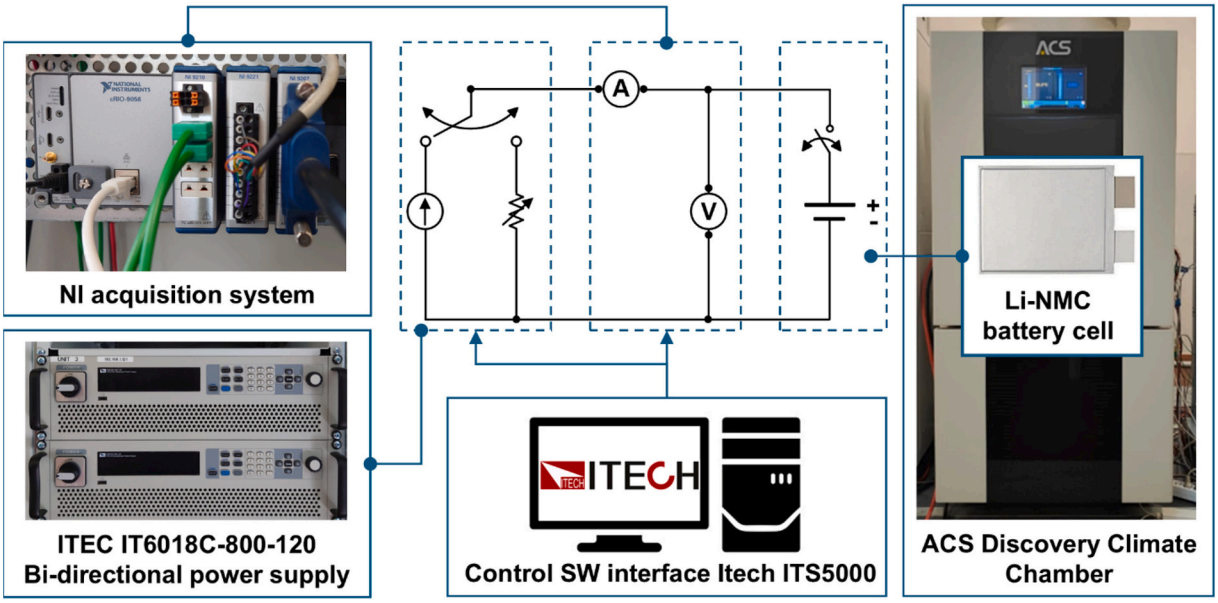


Fig. 5. Schematic representation of the experimental setup for electro-thermal tests.

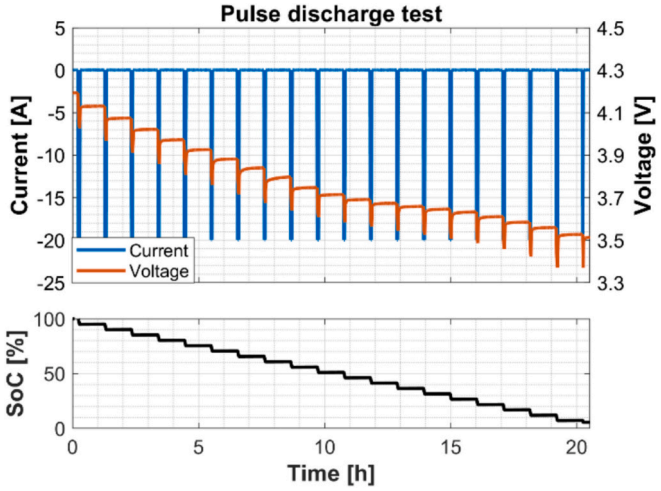


Fig. 7. Current, Voltage and SoC of the battery cell during the pulse discharge test.

Duration of the 20 A pulses, based on the actual capacity evaluations, was set to 158, 185, and 200 s to ensure 5 % SoC drops at 283 K, 298 K, and 313 K. After each pulse, a rest time of 1 h enabled evaluation of the OCV. R_0 was evaluated as the ratio between instantaneous voltage drop/step due to discharge current pulse start/end. R_1, and C_1 were calibrated via numerical optimization with the Parameter Estimator tool embedded in MATLAB SIMULINK ® so that the simulated terminal voltage would fit the measured voltage during the pulse test. About parameters accuracy of one-state lumped thermal model, evaluated above, it is considered enough for the applications reported in this paper, as demonstrated in the following.

The validation procedure of the calibrated cell model was performed by comparing the simulated cell voltage and temperature with their actual values measured during experimental tests. These tests were performed, at the considered temperature levels, with current profiles reflecting the loads due to Highway Fuel Economy Test (HWFET) and the Urban Dynamometer Drive Schedule (UDDS) [34]. These current profiles were obtained from [35] by scaling the current values based on the ratio between the rated capacity of the battery cell used in this study (20 Ah) and the cell used to generate the dataset (2.9 Ah). The obtained profiles are reported in Fig. 8, along with the one reflecting the LA92 drive cycle, which was included as additional scenario for the SoC estimator during the real-time test.

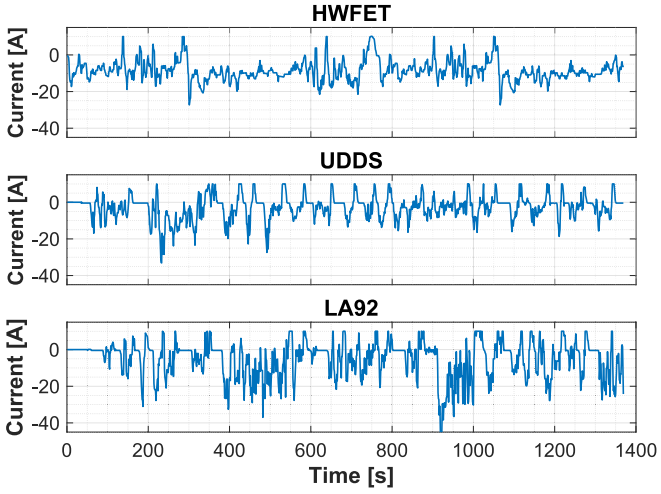


Fig. 8. Current profiles reflecting HWFET, UDDS and LA92, that have been used to test the electrothermal model and the NNs.

3.5. Training, test, and comparison of NNs for SoC estimation

The proper prediction of the storage cell SoC was performed through three typologies of NNs, namely the FFNN, the CNN, and the LSTM-RNN, described in Section 2. Prediction accuracies were evaluated based on comparison with target response sequences that were calculated with off-line SoC Coulombic counts.

The model-based training dataset generation, representing the core of the proposed methodology, was based on simulations, whose input current loads were collected onboard during real drives and are publicly available in [36]. Simulations were carried out for each of the considered temperatures. The load profiles were repeated until the complete discharge condition was achieved (SoC = 0 %). To augment the dataset, also in-time flipped current profiles were used, by taking advantage of the versatility of the simulation environment in generating new training data. Training and comparison of NNs were performed in MATLAB environments using the *Machine Learning and Statistics* and the *Deep Learning* toolboxes.

Simulated voltage, current and temperature reads with a sampling frequency of 1 Hz were employed as input sequences during training whereas the SoC was the response sequence. The input sequences were split in 516 measurements-long time series. This value was determined after sensitivity analysis. The training dataset did not include any of the standardized cycles that were used during the test. Filtered value of voltage and current were used as additional inputs for the FFNN only, due to their positive effect as reported in [9].

The architecture of the NNs was defined trough hyperparameter optimization. The number of hidden layers and the number of neurons in each hidden layer were the hyper-parameters optimized for the FFNN. Similarly, the number of hidden units was optimized in the LSTM-RNN, and the number of filters and their size were optimized for a CNN with two convolutional layers. The estimated SoC sequence was smoothed by means of a gaussian filter to avoid possible spikes and discontinuities. The Gaussian filter computes weighted averages for each value of the sequence with a moving window. Weights are determined based on a Gaussian distribution that is defined by the average and standard deviation of the values within the moving window. For the evaluations reported in this work, the moving window is assumed equal to 60 s, even though no sensible variations have been observed in the range 40–80 s. Hyperparameters to be optimized are resumed, for each NN, in Table 3. Their values were obtained by minimizing the RMSE over a validation dataset made of two real-drive cycles shared in [36] at 283, 298 and 313 K, which were, therefore, excluded from the training data. The current profiles, used to simulate the voltage and temperature of the battery for building the validation dataset, were obtained by scaling the original data on the considered battery cell and are reported in Fig. 9. It is worth remarking that in ML, the validation is mutually distinct from both the training and the test dataset, and it is used to optimize the hyperparameters, to iteratively check the generalization capability of the model and eventually trigger an early stop during the training. On the other hand, the test dataset is intended to definitively evaluate the performances of the NNs after the training over a set of data that was not used in any previous stage of the NN development, so that unbiased estimation of the final capability of the prediction accuracy is possible.

The inherent versatility of simulations enabled the generation of a training dataset with a suitable size.

To prevent overfitting during the training, dropout layers were used

Table 3
List of the hyper-parameters to be optimized.

Type of NN	Hyper-parameters to be optimized
FFNN	Number of layers, neurons in the hidden layers
LSTM-RNN	Number of hidden units
CNN	Number and size of the convolutional filters

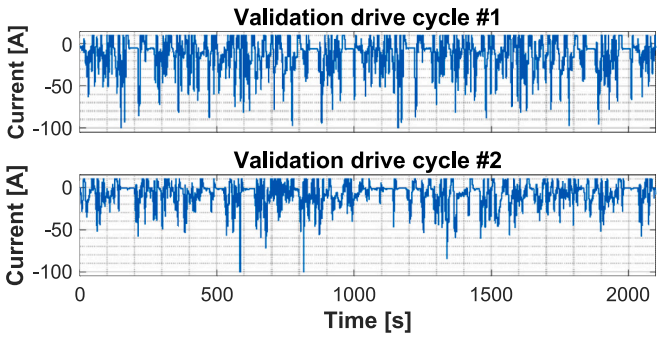


Fig. 9. Current profiles used for generation of the validation datasets.

in all of the NNs considered. During the training, the minibatch size was set equal to 64 observations. Values of the learning rate, learning rate drop factor and period, maximum number of epochs are reported in Table 4 for each of the considered type of NN.

Once their structure was defined with hyper-parameters optimization, the NNs were tested with data gathered during experiments with the NI acquisition systems and loads reflecting UDDS and HWFET. Then, real-time experimental tests were carried out to fully demonstrate the effectiveness of the proposed methodology for applications in BMS.

The evaluation and comparison of performances of NNs were obtained on the basis of the following metrics: root mean squared error (RMSE), mean average error (MAE), maximum absolute error (MaxAE), and mean average error difference (MAEd). Those metrics are defined in eq. (8–11), and their values were calculated based on predictions of NNs over the test dataset. MAE was used to evaluate the global accuracy of the NNs, whereas MaxAE was used to quantify and localize the maximum deviations in the SoC predictions from the actual value. MAEd was introduced to estimate the robustness of the NNs to the noise, which is very likely to occur in real applications. In particular, MAEd evaluates the error in the SoC estimation due to a random noise injected in the input sequences of measurements.

$$RMSE = \sqrt[2]{\frac{1}{N}\sum_{n=1}^{N}(Y_n - \widehat{Y}_n)^2} \tag{8}$$

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|Y_n - \widehat{Y}_n| \tag{9}$$

$$MaxAE = max(|Y_n - \widehat{Y}_n|) \tag{10}$$

$$MAEd = MAE_{noise} - MAE_{clean} \tag{11}$$

With the purpose of verifying the validity of the presented methodology in the case of batteries with different geometries, chemistries and capacities, the entire procedure described in this section was implemented to develop and offline test NNs for SoC estimation of Samsung ICR 18650-26 J battery cells. Specifications are reported in Table 5.

3.6. Experimental test of the methodology for BMS boards

Additional experimental tests in real-time were performed to demonstrate the feasibility of the entire methodology for BMS applications. In particular, the NNs were loaded on the STM32-Nucleo-F401RE development board to estimate, in real-time, the battery SoC during dynamic and constant current discharge tests. For this test, the current profile in the dynamic test was implemented to reflect the expected equivalent load of a LA92 drive schedule. The algorithm that implemented the NNs running on the microcontroller board was developed in MATLAB Simulink environment and transferred on the STM board by means of the *Simulink Coder Support Package for STMicroelectronics Nucleo Boards* toolbox. The complete procedure for the onboard development of the NNs is functionally schematized in Fig. 10.

Finally, the computational times were experimentally evaluated for each of the NNs running onboard of the STM32-Nucleo. These evaluations were performed by acquiring, through the use of a high-resolution oscilloscope, the difference in time between the input measurement signals and the board output signal related to the evaluated SoC.

Table 4
Training option for piecewise learning rate scheduling.

Training option	FFNN	LSTM-RNN	CNN
Max. epochs	250	90	500
Initial learning rate	0.001	0.01	0.01
Learning rate drop period	–	20	200
Learning rate drop factor	–	0.5	0.5

Table 5
Specifications of the Samsung ICR18650-26 J battery cell.

Nominal voltage	3.7 V
Nominal capacity	2.6 Ah
Maximum charge voltage	4.2 V
Lower voltage limit	2.7 V
Maximum discharge current	5C
Dimensions: height, diameter	67 mm, 18 mm
Weight	0.045 kg

4. Results and discussion

4.1. Validation of the electro-thermal physical model

Calibration of the parameters of the electro-thermal model led to values of OCV, R_0, R_1 and C_1 that are reported in Fig. 11 for each of the considered temperatures. Values of these parameters are made available for the benefit of the research and industrial operators that may carry on activities on this type of battery cell.

As shown in Fig. 12, the model fits the experimental data with a good accuracy. The simulated voltage is plotted with the values measured during the experiments. The absolute errors is also plotted on a dedicated y axis to provide the scale of the deviations of the simulations from the experimental ground truth. It can be observed that maximum error peaks do not exceed 0.1 V. Similarly, the temperature is predicted with maximum error of 1.0 K.

4.2. Hyper-parameters optimization and training of the NNs

As detailed in the previous section, the definitive structure of the NNs was determined with the optimization of the hyper-parameters reported

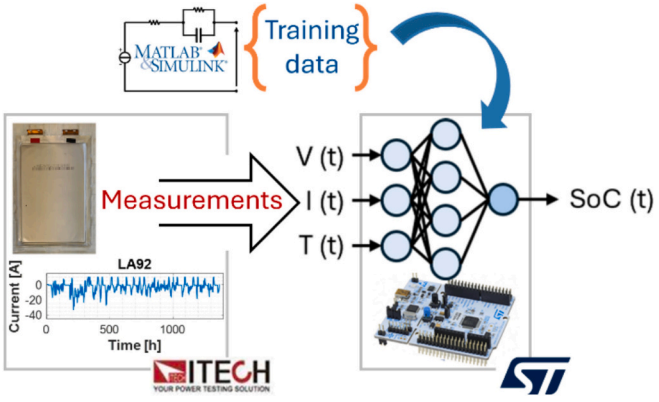


Fig. 10. Schematization of procedure to estimate battery SoC, implemented on ST board for testing NN in real-time.

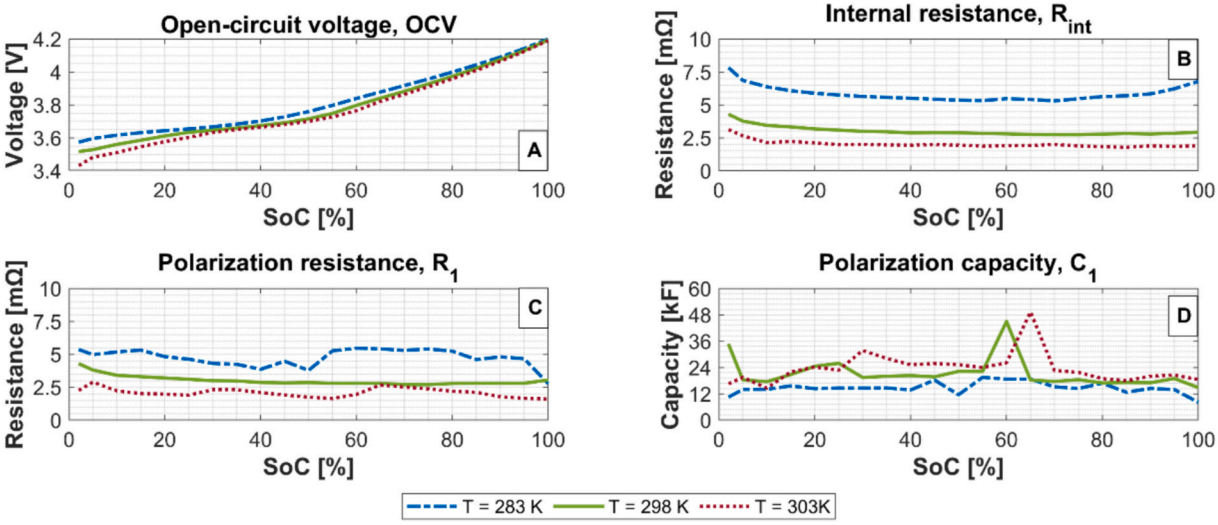


Fig. 11. Calibrated parameters of the ECM model of the battery.

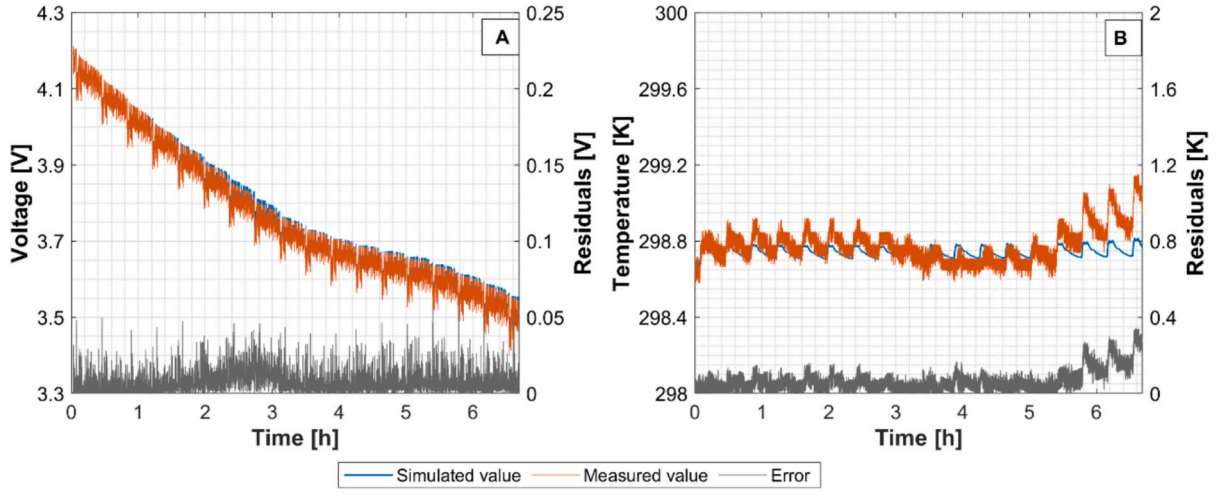


Fig. 12. Comparative plots of the simulated and experimentally measured voltage (A) and temperature (B) of the battery cell under current load reproducing HWFET drive schedule.

in Table 3. The results of this procedure are reported in Table 6 with the final NN structures employed and the optimized hyper-parameters values indicated in bold. The resulting number of learnable parameters is also reported to provide a scale of the size and complexity for the NNs structures. The LSTM-RNN has the lowest number of learnable parameters, indicating its more compact size in comparison with the other NNs to obtain similar performance.

It is worth noting that the optimized numbers of hidden units in the LSTM-RNN and of the filter size in the first convolutional layer of the CNN led to the coincident value of 32 indicating a consistency in the scales of the observed dependencies within the signals leveraged by these NNs. Once their structures were defined, NNs were trained according to the procedure described in Section 3.5. Due to random initialization of the learnable parameters at the beginning of the training, the final accuracy of the NNs can vary with a certain scatter, as shown in Fig. 13. Bar plots show the distribution of final RMSEs after 30 training processes, indicating a variability that needs to be considered to avoid biased trends and draw sound conclusions. In this study, this issue is addressed by averaging the values of the performance metrics introduced above over 30 repetitions. In this way, the impact of the degree of randomness inherent in each training process is accounted in compliance with several literature studies [14].

4.3. *Offline testing of the NNs and comparison of their performances*

In this subsection, NNs are offline tested on data collected during experiments reproducing loads related to the UDDS and HWFET drive schedules at 283, 298 and 313 K. Their performances are compared in terms of accuracy, noise robustness, and computational time.

Averaged values of MAE, MaxAE, and MAEd are reported in Table 7 for each one of the experiments. Focusing on accuracy, the values of the MAE and MaxAE indicate that the predictions of the LSTM-RNNs and the CNNs were significantly more accurate than those with the FFNNs. This is clearly visible in Fig. 14, in which the predicted SoC curves are plotted as representative examples for one of the trained NNs of each type. The absolute errors are plotted separately with a dedicated scale and their trends indicate that the prediction with the FFNN started to deviate from the reference value at half of the test, with an increasing drift. Lower accuracy in their predictions can be interpreted as a result of the fact that the FFNNs are non-dynamic NNs. Indeed, the FFNN processes only actual raw measurements and averaged voltage/current measurements, whilst LSTM-RNN processes the current input and information coming from previous states, and the CNN processes information with a depth in time depending on the filter-size (causal convolutional layers). It is clear that, for the procedures reported in this study, the use of averaged

Table 6
Architectures of the considered NNs with hyperparameters.

FFNN	LSTM-RNN	CNN
Num. of learnable parameters: 51,000	**Num. of Learnable parameters: 4600**	**Num. of Learnable parameters: 71,900**
Input layer (5 sequences, normalization: z-scores)	Input layer (3 sequences, normalization: z-scores)	Input layer (3 sequences, normalization: z-scores)
Fully connected layer **N. 1** (**256 neurons**)	LSTM layer (**32 hidden units**)	1-D Convolution layer N. 1 (**64 filters, size of filters: 32**)
• Drop-out (25 %) • Relu-activation	• Drop-out (12.5 %)	• Padding: causal • Drop-out (25 %) • Relu-activation • Layer Normalization
Fully connected layer **N. 2** (**128 neurons**)	Fully connected layer (1 response neuron)	1-D Convolution layer N. 2 (**16 filters, size of filters: 64**)
• Drop-out (25 %) • Relu-activation	• Relu-activation • Output layer	• Padding: causal • Relu-activation • Layer Normalization
Fully connected layer **N. 3** (**128 neurons**)		Fully connected layer (1 response neuron)
• Drop-out (25 %) • Relu-activation		• Relu-activation • Output layer
Fully connected layer (1 response neuron)		
• Relu-activation • Output layer		

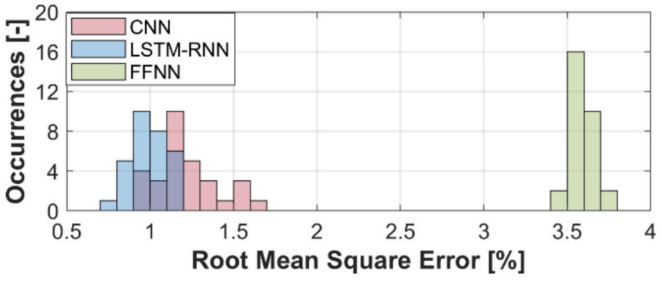


Fig. 13. Occurrence of final RMSE after training of FFNNs, LSTM-RNNs, and CNNs without early stop.

Table 7
Average values of MAE, MaxAE and MAEd [%] for FFNN, LSTM and CNN.

NN	Metric	HWFET			UDDS			Average
		283 K	298 K	313 K	283 K	298 K	313 K	
FFNN	MAE	2.03	2.61	1.97	2.95	4.06	2.03	2.61
	MaxAE	12.4	10.7	6.34	9.46	10.8	6.07	9.31
	MAEd	0.76	0.98	0.42	0.98	0.30	0.45	0.65
LSTM-RNN	MAE	0.90	0.67	0.62	0.66	0.98	0.97	0.80
	MaxAE	2.99	2.20	2.06	2.47	2.88	3.34	2.66
	MAEd	4.47	4.43	3.09	3.94	3.67	1.91	3.58
CNN	MAE	1.45	0.80	0.78	0.74	1.00	1.14	1.01
	MaxAE	4.91	2.40	2.48	3.81	3.21	3.55	3.40
	MAEd	0.52	0.73	0.40	0.39	0.50	0.31	0.47

measurements as additional inputs did not lead to accuracy metrics comparable with LSTM-RNNs as reported in [9]. On the other hand, the accuracy of the predictions of the LSTM-RNN and the CNN were comparable. However, the former performs slightly better than the latter, as also indicated by MAE values in Table 7, even if the number of learnable

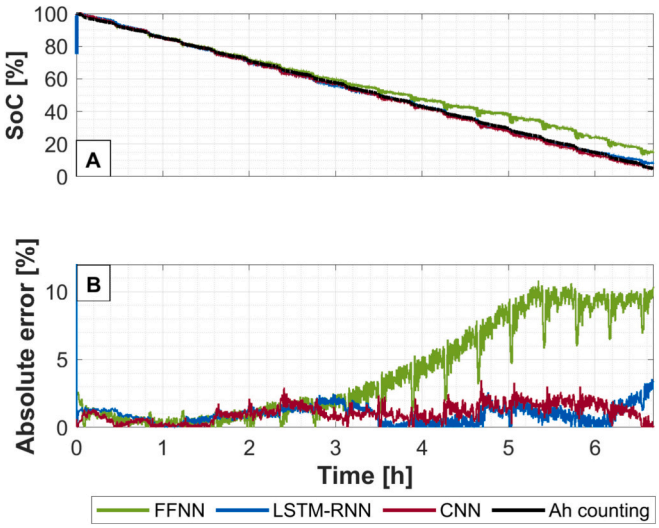


Fig. 14. Offline validation of the NNs. Data collected during test reflecting loads due to UDDS at 298 K: (A) SoC predictions, and (B) absolute errors.

parameters is significantly lower. Indeed, the chosen values for the optimized hyperparameters selected in the CNN led to an architecture with about 71,900 learnable parameters (Table 6), which is significantly higher than those of a LSTM-RNN (about 4600) with 32 hidden units. This result indicates that this type of RNN may have a greater potential in processing time series, due to its inherent dynamicity. It is worth noting that, since LSTM-RNN is a dynamic NN, the highest deviation in the prediction (of about 27 %) from the actual SoC value occurs just for the initial values of its output sequence. This is due to random initialization of the network status at an unknown value. Anyway, this error converged to values below 2.5 % in a small time of about 15 s. As this systematic error in the SoC prediction is quickly recovered, in Table 7, for LSTM-RNN, MaxAE evaluated after 60s of the tests are reported to localize further misprediction and discuss their cause. In the prediction with the LSTM-RNN, deviations from actual values occurred at the end of the test, whereas it was less localized for the CNN. This behavior of the LSTM-RNN is in line with the results shown in the literature [18]. In particular, in Fig. 14, the MAE of the LSTM-RNN was below 2.5 % until the battery approached the fully discharged condition (up to about 10 %), then the error increased. The MAE of the CNN was generally slightly higher than the LSTM-RNN, and more distributed. These qualitative trends observed in representative examples plotted in Fig. 14, were consistently observed during the analysis of most of the NNs that resulted from the repeated training procedures for the three NNs.

For a deeper analysis of the trends and isolation of the influencing factors, the distributions of the estimation errors were evaluated, plotted in histograms of Fig. 15, and analyzed for the considered NNs over all the test data. In particular, subfigure (A) reports the error distributions during complete tests, whereas subfigure (B) isolated error distributions when the battery cell approached the fully discharged conditions (SoC < 30 %). The two subfigures show that the error distributions of LSTM-RNN and the CNN for both above conditions are comparable, highlighting no sensible performance reductions. Additionally, absolute error at the 5th and 95th percentiles below 2.3 %, revealed that the largest error values (the tails in the distribution) were sporadic and had limited impact on the overall prediction for these two NNs. This generally results in slight underestimations of the SoC. On the other hand, error distributions for the FFNN revealed, for low charge conditions, an increase of the mean value and higher scatters. These increases are in good agreement with the discussion of the results above, and in particular, with over-estimations clearly visible in Fig. 14 (A), and the increasing noise in Fig. 14 (B). One influencing factor that can explain this performance decrease is the intrinsic non-linearities of the knee

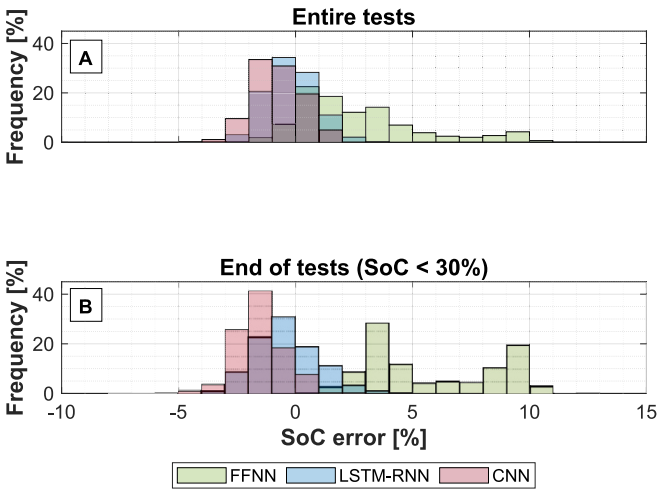


Fig. 15. Errors in the estimation for the FFNN, LSTM-RNN, CNN over the entire data-sequencies of all of the test cases (A), and over the end of the data-sequencies (B).

trend battery cell voltage at low SoC values. In these conditions, given its inherent simplicity, the FFNN captures less information and dependencies in the data. A promising corrective action for potential improvement of this methodology with respect to FFNN use is the characterization of the electric behavior of the battery cell with tighter sampling for SoC values below 20 %.

To highlight the effectiveness of the proposed methodology, MAE and MaxAE listed in Table 7 were compared with those summarized in three of the most comprehensive and recent review paper [3,37,38]. Indeed, these review papers report values of either MAE or RMSE achieved by different NN-based estimators that were trained with experimental data. These values were consulted and matched to define reference values or ranges of accuracy for NNs similar to those analyzed in this work. Values and ranges of accuracies obtained with were reported in Table 8 to enable comparison with the performances of the estimators trained in this work. Comparison revealed that NNs proposed in this study resulted in values of error metrics comparable to similar NNs trained with experimental data in other studies, with the only exception of the FFNN, whose MAEs are greater than those indicated in reference papers. This discrepancy can be explained as the consequence of the variety of results achieved by the many studies that used FFNNs, what were expressed with one or the other error metric. Average values of MAE and RMSE achieved by LSTM-RNNs fit perfectly in the reference range, whereas those of CNNs are slightly larger, but still comparable. Additionally, different types of advanced estimation techniques were included in Table 8 for comparison such as filter-based EKF and UKF. Once again the comparison revealed that the values of the accuracy metrics of the estimators proposed in this work have values comparable or below UKF and EKF.

As observed during the discussion of the results related to LSTM-RNNs, another important aspect to consider, in comparing performances of the NNs, is the eventual occurrence of SoC estimation errors at the initial times of the output sequence due to unknown initial states. Indeed, non-continuous operation of the battery can lead to stop and resume of the estimation task with different initial SoC values that need to be correctly predicted. To address these issues, test data sequences were split into 10 sequences, each spanning about 10 % of the SoC, and prediction errors were analyzed. The analysis accounted for all of the 30 NNs obtained from repetitive training processes. Maximum absolute errors at the beginning of each segment due to unknown initial value of the SoC have been plotted in Fig. 16 (A), for all of the actual initial states. Averaged values resulted equal to 4.5, 14.3, 5.2 % approximatively for FFNN, LSTM-RNN, and CNN, respectively. A global trend can be observed: FFNNs resulted insensitive to unknown values of the status at previous time, as it is a static NN; the LSTM-RNN was the most impacted, as expected from the previous analysis of the results shown in Fig. 14. In particular, this behavior occurred starting from initial SoC values higher than 30 %. Slightly higher error in initial predictions of CNNs can be explained by considering lower amount of useful information extracted with convolution involving padding values. Similar values for the prediction errors with all the NNs at the beginning of the output sequences when SoC is below 20 % can be explained by considering the above-mentioned non-linearities occurring when the battery approaches the complete discharge conditions.

On the other hand, it is important to highlight that the LSTM-RNN quickly recovers the initial deviations from actual values. This is clearly shown in Fig. 16 (B), which reports estimation in the first 30 s of a HWFET test sequences at 313 K with an initial SoC value equal to 50 %.

The computational effort required for the SoC estimation with each of the considered NNs was evaluated in terms of computational time on a commercial computer with 13th Gen Intel(R) Core (TM) i9-13900H 2.60 GHz processor, and then compared each other. The time required for SoC estimation of a data-point corresponding to a sequence time-step was measured equal to: $1.7e^{-3}$, $1.9e^{-3}$, $3.1e^{-3}$ s for FFNN, LSTM-RNN, and CNN, respectively. For sake of comparison, estimated computational times were normalized with reference to the slowest NN, i.e. the CNN. LSTM-RNNs required 61.3 % of the time required by the CNNs, whereas FFNNs had the fastest estimations with the 54.8 % of the time required by the CNNs. One reason for the higher time required by CNNs to process a single data-point can be that estimation at each time-step

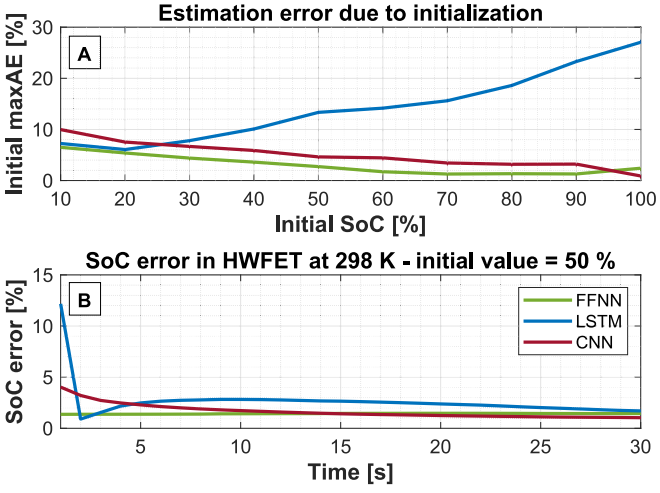


Fig. 16. SoC estimation error due unknown initial SoC value: (A) estimation error due to unknown initial values in sequences with different initial SoC values; (B) case of HWFET drive cycle at 298 K and initial SoC = 50 %.

Table 8
Averaged error metrics reported in review papers for different NNs and filter based SoC estimators.

Estimator	FFNN		LSTM		CNN		UKF	EKF	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	MAE	RMSE
This work	2.61	3.45	0.80	1.03	1.01	1.23	–	–	–
Integration of data in [3,37,38]	0.6–1.0	1.4–3.7	0.7	1.7–2.7	0.8	0.6	< 1.25	< 3.0	3.0

involves convolution between kernels and previously gathered or padded sequences.

Table 7 reports also the values of MAEd, calculated to evaluate the robustness of the NNs, in case of random noise injected in the signals. The injected signal was randomly generated to obtain a signal-to-noise ratio of about 16 dB. Average values of the MAEd were equal to about 0.65 %, 3 %, and 1 % for the FFNN, LSTM-RNNs, and the CNNs, respectively. The averaged value of the MAEd for the LSTM-RNNs was the highest. This could be explained by considering the fact that the injected noise could have altered the recurrent dependencies within the signals, affecting the information leveraged by the LSTM-RNNs. MAEd of the CNN and FFNN indicated superior robustness to the noise. These results can be interpreted by considering that 1D filters tuned during the training of the CNNs captured trends at different scales. On the other hand, the FFNN received in input also averaged measurements of voltage and current. This feature mitigated the effect of the noise injected into the input signal. These results are in line with those shown by Guo et al. for similar values of perturbances [10].

Finally, a completely different battery cell with cylindric geometry and capacity of 2.6 Ah was considered as additional case study to verify generalization capability of the proposed methodology to different materials. Therefore, training and validation data were generated via simulation with current loads obtained by scaling the same profiles used for the previous battery to the new one. In the same way testing procedures were performed on experimental data. Fig. 17 shows SoC predictions during offline test reproducing loads due to UDDS at room temperature.

As it can be clearly observed from the above figure, SoC followability continued to be reliable also for this different battery technology for all the considered NNs.

As brand new batteries were employed for the characterization, modelling and experimental collection of test data, the methodology mainly focused on the SoC evaluation at SoH = 100 %. On the other hand, the proposed methodology is designed to incorporate information related to the contribution of SoH. Indeed, this contribution can be taken into account, in the development of battery simulation model, by considering the effects of calendar and cycling aging on the battery cell parameters on the basis of evaluations reported in the scientific literature [39,40].

4.4. Real-time SoC estimation with the NNs running onboard of the STM32-Nucleo microcontroller

To fully demonstrate the feasibility of the proposed methodology on battery management systems, two tests were carried out with different current profiles. The first one consisted of current profiles reflecting loads of an additional drive schedule, the LA92. The second one aims to evaluate the performance of the considered NNs on a static profile, intrinsically different from the training ones, related to a constant current discharge. The results are respectively reported in Fig. 18 and Fig. 19.

In Fig. 18 (A), the SoC estimation was performed in real-time, based on measurements gathered by the STM32-Nucleo board, which is

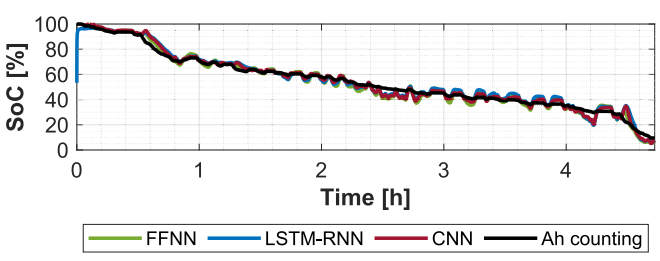


Fig. 17. Offline test of NNs on UDDS at temperatures ranging between 295 and 298 K.

equipped with a 12 bits analog/digital converter (ADC). It is clearly visible that the estimations had a greater scatter compared to offline tests, as also shown in Fig. 18. For a better understanding of these results, the same data were also gathered with the NI acquisition system for further offline estimations in MATLAB-PC environment, which are reported in Fig. 18 (B). In particular, the shift toward noisier predictions can be explained considering that a commercial development board, with low resolution ADC, allowed the less precise acquisition of signals than a more sophisticated system, such as the NI compact RIO. Alteration due to measurement noise impacted the SoC estimation stability especially when the battery cell approached low SoC conditions. This because the OCV is dependent on temperature and SoC. For lower values of the SoC, battery parameters calibration is more difficult because the OCV varies with a knee trend and higher non-linearity, which may require tighter sampling during the battery cell characterization. On the other hand, during real usage of the battery pack, operative conditions with SoC lower than 15 % are generally avoided to not affect its durability. For this reason, higher fidelity in the electric characterization of the battery cell, when their SoC decreases, can improve this particular aspect.

Since the evaluations of this paper are referred to NNs trained with simulated data with dynamic current loads, an additional demonstrative test was carried out with a constant curve discharge to test their robustness to data with different patterns. The results are plotted in Fig. 19. Once again, the FFNN had the poorest performances, whereas the error of the LSTM-RNNs and CNNs stayed below 2 % for most of the test with an increase to 6 % and 4 %, respectively once the battery approached the discharged condition.

The computational time required by the NNs for SoC evaluation was evaluated also when they were running on the STM32-Nucleo board. Estimation times were evaluated approximatively equal to $1.10e^{-2}$, $3.11e^{-2}$, and $4.14e^{-2}$ s, for FFNNs, LSTM-RNNs, and CNNs, respectively.

Finally, by considering also the results obtained and resumed in Table 8 and Table 9, the feasibility of the proposed methodology has been demonstrated in all the operative conditions for both off-line and on-line SoC estimation procedures. The results obtained using model-based datasets are comparable to the performance indexes reported in the scientific literature based on experimentally collected training datasets. Additionally, the expected decrease in performances due to implementation on commercial boards with lower resolution was verified and considered in an acceptable range.

5. Conclusions

In this work, the use of model-based datasets for training SoC estimators was proposed in a structured methodology. A lumped-parameters electro-thermal model was calibrated and validated to simulate the behavior of a Li-NMC battery cell, between 283 and 313 K, with voltage and temperature errors within 0.1 V and 1.0 K. The validated model was used to generate data via simulations, reproducing the operation of the battery, with load profiles reflecting about 190 real drive cycles.

Simulated data were used to train three different NNs, namely: FFNN, LSTM-RNN, CNN. Offline testing and comparison of those NNs were performed on experimental data, reflecting the HWFET and the UDDS standardized drive schedules at various temperatures. In particular, the testing was carried out with the estimator running on a commercial laptop to compare the performances of NNs in terms of MAE, MaxAE, computational time, sensitivity to unknown initial SoC value, and noise robustness. The LSTM-RNNs estimated the SoC with the lowest MEA and MaxAE, which were respectively equal to 0.80 % and 2.66 %. On the other hand, those NNs showed higher sensitivity to measurement noise and initialization at unknown SoC value. FFNNs showed the lowest computational cost, with significantly lower performances, in terms of prediction accuracy (MAE = 2.6 % and MaxAE = 9.3 %), compared with the LSTM-RNNs and CNNs. Indeed, the inherent

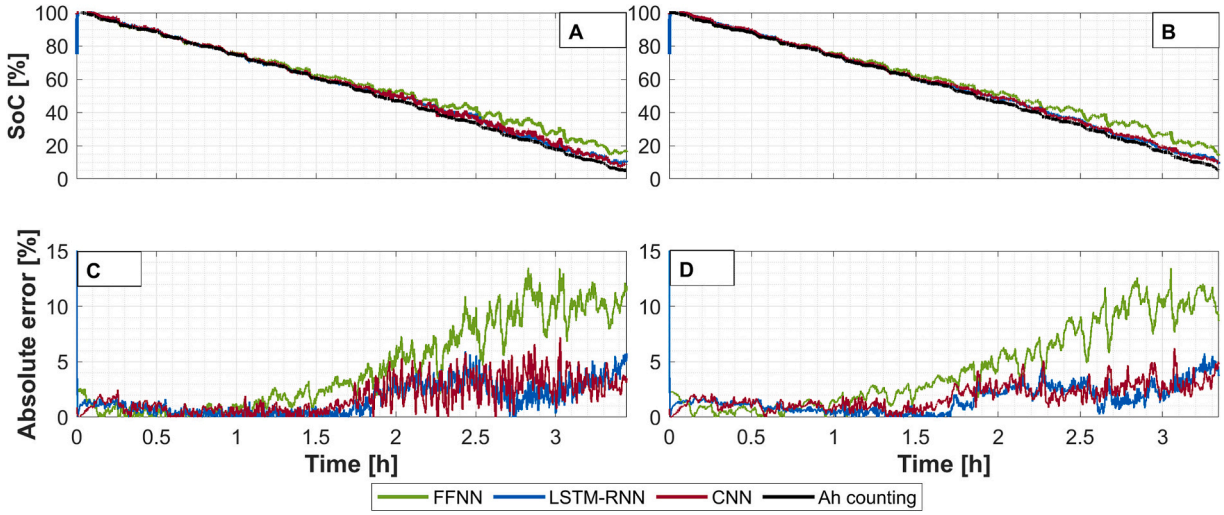


Fig. 18. Test of the NNs on the LA92 standardized drive schedule: (A, C) Real-time estimation with the NNs running on the STM32-Nucleo microcontroller, and (B, D) offline processing of data gathered with NI acquisition system.

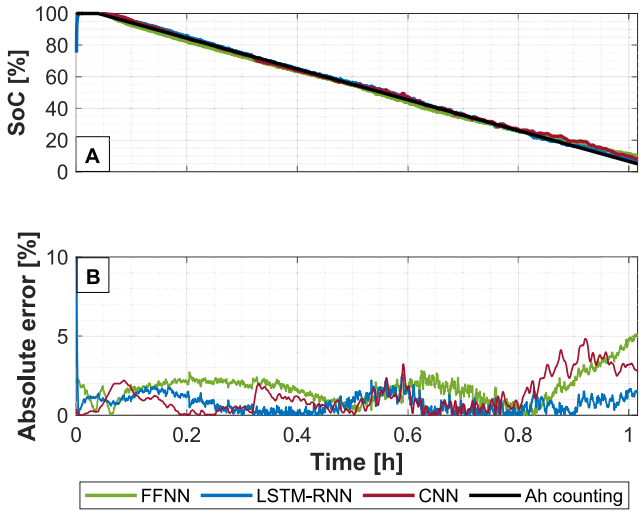


Fig. 19. Test of the NNs with a constant current discharge experiment and the SoC estimated in real-time with the NNs running onboard of the STM32-Nucleo microcontroller: (A) SoC predictions, and (B) absolute errors.

Table 9
Average values of MAE and MaxAE [%] for FFNN, LSTM and CNN evaluated over constant current discharge and an experiment reproducing loads due to LA92.

NN	Metric	LA92 on-line	LA92 off-line	CC disch. on-line
FFNN	MAE	4.45	4.47	1.63
	MaxAE	13.5	13.4	5.19
LSTM-RNN	MAE	1.61	1.60	0.84
	MaxAE	5.92	5.73	2.50
CNN	MAE	1.82	1.84	1.25
	MaxAE	7.2	6.23	4.89

simplicity of FFNNs enables faster calculations, without taking advantage of recurrent dependencies within training data. Additionally, predictions with FFNN were not affected by initialization to unknown SoC value. CNNs, with their capability to learn dependencies in the signals on different scales with convolutions, showed a good balance between the simplicity of FFNNs and the high accuracy of LSTM-RNNs, however, required larger computational times to process one datapoint, compared with FFNNs and CNNs.

The effectiveness of the proposed methodology for BMS development applications was verified by implementing NNs on an STM32-Nucleo microcontroller for real-time SoC estimation. In this case, despite the lower measurement accuracy of the board, the experimental results confirmed the differences among the analyzed NNs, and proved the feasibility of the methodology for real onboard applications. Estimation times with of the NNs running on the STM32-Nucleo board were approximatively equal to 0.011, 0.031, and 0.0414 s, for FFNNs, LSTM-RNNs, and CNNs, respectively.

Finally, the scalability of the proposed methodology to the case of a battery with different geometry and capacity, such as Samsung ICR18650-26 J, was demonstrated.

It is worth noting that such a methodology can be scaled for the estimation of SoC in batteries with different applications, reducing the number of experiments, needed to generate data to the sole validation and test stages. The proposed procedure enables significant savings in terms of time and resources to generate the training dataset, which accounts for about 70 % of the data needed to implement data-driven approaches.

CRediT authorship contribution statement

Giovanni Chianese: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Luigi Iannucci:** Software, Conceptualization. **Ottorino Veneri:** Writing – review & editing, Visualization, Supervision, Funding acquisition, Formal analysis, Conceptualization. **Clemente Capasso:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research has been supported by the European Union-NextGenerationEU - National Sustainable Mobility Center

CN00000023, Italian Ministry of University and Research Decree n. 1033—17/06/2022, Spoke 12, CUP B43C22000440001.

Data availability

Data will be made available on request.

References

[1] Demartini M, Ferrari M, Govindan K, Tonelli F. The transition to electric vehicles and a net zero economy: a model based on circular economy, stakeholder theory, and system thinking approach. J Clean Prod 2023;410:137031. https://doi.org/10.1016/j.jclepro.2023.137031.

[2] Markus L, Braun T, Knips M, Nordmann H, Ringbeck F, Zappen H, et al. Battery management system hardware concepts: an overview. Appl Sci 2018;8:534. https://doi.org/10.3390/app8040534.

[3] Ren Z, Du C. A review of machine learning state-of-charge and state-of-health estimation algorithms for lithium-ion batteries. Energy Rep 2023;9:2993–3021. https://doi.org/10.1016/j.egyr.2023.01.108.

[4] Hannan MA, Lipu MSH, Hussain A, Mohamed A. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: challenges and recommendations. Renew Sust Energ Rev 2017;78:834–54. https://doi.org/10.1016/j.rser.2017.05.001.

[5] Wu L, Lyu Z, Huang Z, Zhang C, Wei C. Physics-based battery SOC estimation methods: recent advances and future perspectives. J Energy Chem 2024;89:27–40. https://doi.org/10.1016/j.jechem.2023.09.045.

[6] Capasso C, Iannucci L, Patalano S, Veneri O, Vitolo F. Design approach for electric vehicle battery packs based on experimentally tested multi-domain models. J Energy Storage 2024;77:109971. https://doi.org/10.1016/j.est.2023.109971.

[7] Sesidhar DVSR, Badachi C, Green Robert C. A review on data-driven SOC estimation with Li-ion batteries: implementation methods & future aspirations. J Energy Storage 2023;72C:108420. https://doi.org/10.1016/j.est.2023.108420.

[8] Chemali E, Kollmeyer PJ, Preindl M, Emadi A. State-of-charge estimation of Li-ion batteries using deep neural networks: a machine learning approach. J Power Sources 2018;2018(400):242–55. https://doi.org/10.1016/j.jpowsour.2018.06.104.

[9] Vidal C, Malysz P, Naguib M, Emadi A, Kollmeyer PJ. Estimating battery state of charge using recurrent and non-recurrent neural networks. J Energy Storage 2022;47:103660. https://doi.org/10.1016/j.est.2021.103660.

[10] Guo S, Ma L. A comparative study of different deep learning algorithms for lithium-ion batteries on state-of-charge estimation. Energy 2023;263C:125872. https://doi.org/10.1016/j.energy.2022.125872.

[11] Jiang B, Tao S, Wang X, Zhu J, Wei X, Dai H. Mechanics-based state of charge estimation for lithium-ion pouch battery using deep learning technique. Energy 2023;278(A):127890. https://doi.org/10.1016/j.energy.2023.127890.

[12] Tian J, Xiong R, Shen W, Lu J. State-of-charge estimation of LiFePO4 batteries in electric vehicles: a deep-learning enabled approach. Appl Energy 2021;291:116812. https://doi.org/10.1016/j.apenergy.2021.116812.

[13] Tang A, Huang Y, Liu Yu Q, Shen W, Xiong R. A novel lithium-ion battery state of charge estimation method based on the fusion of neural network and equivalent circuit models. Appl Energy 2023;348:121578. https://doi.org/10.1016/j.apenergy.2023.121578.

[14] Vidal C, Malysz P, Kollmeyer P, Emadi A. Machine learning applied to electrified vehicle battery state of charge and state of health estimation: state-of-the-art. IEEE Access 2020;8:52796–814. https://doi.org/10.1109/ACCESS.2020.2980961.

[15] Reis G, Strange C, Yadav M, Li S. Lithium-ion battery data and where to find it. Energy and AI 2021;5:100081. https://doi.org/10.1016/j.egyai.2021.100081.

[16] Channegowda J, Maiya V, Joshi N, Urs VR, Lingaraj C. An attention-based synthetic battery data augmentation technique to overcome limited dataset challenges. Energy Storage 2022;4(5):e354. https://doi.org/10.1002/est2.354.

[17] Li J, Ziehm W, Kimball J, Landers R, Park J. Physical-based training data collection approach for data-driven lithium-ion battery state-of-charge prediction. Energy and AI 2021;5:100094. https://doi.org/10.1016/j.egyai.2021.100094.

[18] Hattouti LA, Di Rienzo R, Nicodemo N, Verani A, Baronti F, Roncella R, et al. Comparison of Lithium-ion battery SoC estimation accuracy of LSTM neural network trained with experimental and synthetic datasets. In: Applications in electronics pervading industry, environment and society (ApplePies 2023). Lecture notes in electrical engineering. Cham: Springer; 2025. p. 1110. https://doi.org/10.1007/978-3-031-48121-5_58.

[19] Di Dio R, Aurilio G, Di Rienzo R, Saletti R. Machine learning for SOC estimation in Li-ion batteries. In: Bellotti F, et al., editors. Applications in electronics pervading industry, environment and society (ApplePies 2023). Lecture notes in electrical engineering. Cham: Springer; 2025. p. 1110. https://doi.org/10.1007/978-3-031-48121-5_55.

[20] El Fallah S, Kharbach J, Vanagas J, Vilkelyt Z, TolvaišienS Gudžius S, et al. Advanced state of charge estimation using deep neural network, gated recurrent unit, and long short-term memory models for Lithium-ion batteries under aging and temperature conditions. Appl Sci 2024;14:6648. https://doi.org/10.3390/app14156648.

[21] Jain AK, Mao J, Mohiuddin KM. Artificial neural networks: a tutorial. Computer 1996;29-3:31–44. https://doi.org/10.1109/2.485891.

[22] Zhao B, et al. Estimation of the SOC of energy-storage Lithium batteries based on the voltage increment. IEEE Access 2020;8:198706–13. https://doi.org/10.1109/ACCESS.2020.3031327.

[23] Hannan MA, Lipu MSH, Hussain A, Saad MH, Ayob A. Neural network approach for estimating state of charge of Lithium-ion battery using backtracking search algorithm. IEEE Access 2018;6:10069–79. https://doi.org/10.1109/ACCESS.2018.2797976.

[24] Chen C, Xiong R, Yang R, Shen W, Sun F. State-of-charge estimation of lithium-ion battery using an improved neural network model and extended Kalman filter. J Clean Prod 2019;234:1153–64. https://doi.org/10.1016/j.jclepro.2019.06.273.

[25] Chemali E, Kollmeyer PJ, Preindl M, Ahmed R, Emadi A. Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries. IEEE Trans Ind Electron 2018;65(8):6730–9. https://doi.org/10.1109/TIE.2017.2787586.

[26] Tian Y, Lai R, Li X, Xiang L, Tian J. A combined method for state-of-charge estimation for lithium-ion batteries using a long short-term memory network and an adaptive cubature Kalman filter. Appl Energy 2020;265:114789. https://doi.org/10.1016/j.apenergy.2020.114789.

[27] Ren X, Liu S, Yu X, Dong X. A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM. Energy 2021;234:121236. https://doi.org/10.1016/j.energy.2021.121236.

[28] Ma L, Hu C, Cheng F. State of charge and state of energy estimation for Lithium-ion batteries based on a long short-term memory neural network. J Energy Storage 2021;37:102440. https://doi.org/10.1016/j.est.2021.102440.

[29] Kiranyaz S, Avci O, Abdeljaber O, Ince T, Gabbouj M, Inman DJ. 1D convolutional neural networks and applications: a survey. Mech Syst Signal Process 2021;151:107398. https://doi.org/10.1016/j.ymssp.2020.107398.

[30] Bhattacharjee A, Verma A, Mishra S, Saha TK. Estimating state of charge for xEV batteries using 1D convolutional neural networks and transfer learning. IEEE Trans Veh Technol 2021;70(4):3123–35. 10.1109/TVT.2021.3064287.

[31] Chen J, Zhang C, Chen C, Lu C, Xuan D. State-of-charge estimation of Lithium-ion batteries using convolutional neural network with self-attention mechanism. J Electrochem Energy Convers Storage 2023;20(3):031010. https://doi.org/10.1115/1.4055985.

[32] Liu Y, Li J, Zhang G, Hua B, Xiong N. State of charge estimation of Lithium-ion batteries based on temporal convolutional network and transfer learning. IEEE Access 2021;2021(9):34177–87. https://doi.org/10.1109/ACCESS.2021.3057371.

[33] Kim KH, Oh KH, Ahn HS, Choi HD. Time–Frequency Domain Deep Convolutional Neural Network for Li-Ion Battery SoC Estimation. IEEE Trans Power Electron 2024;39(1):125–34. https://doi.org/10.1109/TPEL.2023.3309934.

[34] Dynamometer Drive Schedules | US EPA. https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules;; 2025. accessed [05 December 2024].

[35] Kollmeyer P. Panasonic 18650PF Li-ion battery data. Mendeley Data 2018. https://doi.org/10.17632/wykht8y7tg.1.

[36] Steinstraeter M, Buberger J, Trifonov D. Battery and heating data in real driving cycles. IEEE Dataport 2020. https://doi.org/10.21227/6jr9-5235.

[37] Pandey SV, Parikh N, Prochowicz D, Akin S, Satapathi S, Tavakoli MM, et al. Predicting the state parameters of lithium ion batteries: the race between filter-based and data driven approaches. Sustainable Energy Fuels 2023;7:598–628. https://doi.org/10.1039/D2SE01209J.

[38] Barik S, Saravan B. Recent developments and challenges in state-of-charge estimation techniques for electric vehicle batteries: a review. J Energy Storage 2024;100B:113623. https://doi.org/10.1016/j.est.2024.113623.

[39] Wang J, Purewal J, Liu P, Hicks-Garner J., Soukazian S, Sherman E, Verbrugge M. W., Degradation of lithium-ion batteries employing graphite negatives and nickel–cobalt–manganese oxide+ spinel manganese oxide positives: part 1, ageing mechanisms and life estimation. J Power Sources 2014; 269:937–948.

[40] Irujo E, Berrueta A, Sanchis P, Ursúa A. Methodology for comparative assessment of battery technologies: experimental design, modeling, performance indicators and validation with four technologies. Appl En 2025;378:124757.