2024 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE) | 979-8-3503-7800-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/METROXRAINE62247.2024.10795955

Temperature-Dependent State of Charge Estimation for Electric Vehicles Based on a Machine Learning Approach

Simone Barcellona*, Loris Cannelli†, Lorenzo Codecasa*, Silvia Colnago‡, Loredana Cristaldi*, Christian Laurano* and Gabriele Maroni†
*DEIB, Politecnico di Milano, Milano, Italy
†IDSIA, SUPSI-USI, Lugano, Switzerland
‡TGM, Ricerca sul Sistema Energetico S.p.A., Milano, Italy

***Abstract*—Lithium-ion batteries (LiBs) are extensively used in numerous applications, with electric vehicles being one of the most important. Consequently, significant research efforts have been directed towards developing battery models that can predict battery behavior, increase efficiency, enhance safety, and reduce degradation. One of the key state parameters for this purpose is the state of charge (SOC). Many estimation methods in the literature rely on understanding the open circuit voltage (OCV)-SOC relationship, which is affected by battery temperature and can be modeled using different approaches, such as table-based, analytical, physical-based, and machine learning (ML). ML approaches are gaining increasing popularity and interest, although they require extensive experimental data and the identification of the most informative features. Specifically, given the nature of the problem, we proposed an ML algorithm based on deep neural networks capable of estimating the SOC of an LiB for electric vehicles using only a single measurement of the actual OCV and battery temperature. Finally, the proposed algorithm was validated through an extensive experimental campaign.**

I. Introduction

In the latest years, the introduction of batteries in the everyday life has grown dramatically, as they find their use in various fields, from biomedical to power system and new mobility appliances [1], [2].

Modeling batteries has been a crucial task since their first employment. In particular, research has been focused on the state of charge (SOC) and state of health (SOH) to determine the remaining useful charge in the present cycle and predict the remaining endurance. Different modelling approaches have been developed in literature, and focusing on the electrical aspect, they can be roughly divided into three main categories: mathematical, electro-chemical, and equivalent circuit models [3]. Among them, equivalent circuit models, which tries to represent the characteristic of the battery using lumped parameters, represents a good trade-off between complexity and accuracy.

Focusing on SOC estimation alongside the estimation of remaining capacity is crucial for ensuring that the battery operates within optimal and safe limits. Numerous studies in the literature have proposed various SOC estimation methods, most of which require knowledge of the open circuit voltage (OCV) curve. This curve varies with factors such as degradation and temperature. Therefore, estimating the SOC of a battery, which is related to its actual capacity, requires normalization with that capacity.

In this work, we analyze the effect of temperature on the OCV curve changes, with the goal of estimating the actual battery capacity and correcting SOC estimation accordingly. Several studies have focused on modeling the OCV as a funcgtion of SOC or absolute state of discharge, q, examining how this relationship varies with temperature. These studies generally use data-driven approaches, through which the OCV-SOC relationship can be modeled using table-based methods, analytical methods, and machine learning (ML) techniques. For instance, in [4], the impact of temperature on the OCV was assessed using lookup tables. In [5], the authors employed a temperature factor to correct the OCV curve, while in [6], a function was used to compensate for capacity variation, reflecting changes in the OCV curve. On the other hand, these methods are not easily adaptable to different battery chemistries. In recent years, data-driven techniques based on ML have increasingly taken the spotlight in research activities. This shift is driven by advancements in measurement methods, cloud storage, and computational speed. Additionally, the vast amount of data available from the field makes ML techniques particularly appealing for battery modeling [7]–[11]. Moreover, the optimal model can be determined by appropriately combining experimental data with training techniques. Therefore, a unified modeling procedure can implemented to establish the SOC-OCV relationship for various battery chemistries. To develop diverse ML strategies, various features have been extracted, typically from voltage, current, and temperature data collected during charging or discharging cycles. In [7], features were derived from these parameters during the constant current and constant voltage charging process. In [8], the authors incorporated these quantities along with the elapsed time of the discharge cycle. Additionally, other studies, [9], [11], have constructed unique features considering the discharge capacity

The work of Silvia Colnago has been financed by the Research Fund for the Italian Electrical System under the Three-Year Research Plan 2022-2024 (DM MITE n. 337, 15.09.2022), in compliance with the Decree of April 16th, 2018

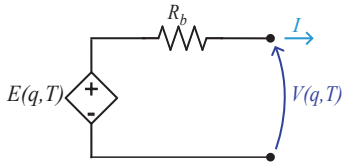

Fig. 1. Battery equivalent electric circuit.

profiles and the integral of battery temperature.

Based on these achievements, this paper employs an ML technique based on deep neural networks (NNs) to estimate the actual SOC of a battery for electric vehicle (EV) applications using just a single measurement of the OCV and battery temperature.

II. Electrical Model and Experimental Activity

The methodology and electrical circuit model used to generate and elaborate the OCV-q experimental curves are thoroughly described in [12], with a concise summary provided here. The tests were conducted on a 10 Ah LiCoO2 pouch cell (model 8773160K). This cell, manufactured by General Electronics Battery Co. Ltd, operates within a voltage range of 2.75 V to 4.2 V and can handle a maximum discharge current of 100 A (10C).

A. Battery model

Figure 1 illustrates the equivalent electric circuit of the battery, which includes a controlled voltage source that represents the battery open circuit voltage, E. This voltage source varies according to the absolute state of discharge, q, and battery temperature, T. Additionally, the model features a series resistor, R_b, that represents the total internal resistance of the battery. The mathematical equation governing the model, which allows for deriving the output battery voltage, V, as a function of the battery current, I, is as follows:

$$V(q,T) = E(q,T) - R_b \cdot I. \tag{1}$$

The absolute state of discharge is obtained by integrating the battery current with respect to an initial value, $q(0)$, as follows:

$$q = \int_0^t I \cdot d\tau + q(0). \tag{2}$$

Finally, the SOC of the battery can be evaluated by the knowledge of the absolute state of discharge, q, and of the actual battery capacity, C_a, as

$$SOC = \left(1 - \frac{q}{C_a}\right) \cdot 100. \tag{3}$$

In [12], the OCV-q curves were derived by appropriately adjusting the discharge voltage curves obtained at a current rate of 1C. This current rate was selected to expedite the tests and minimize the battery exposure to aging mechanisms, particularly during characterization tests conducted at extreme temperatures. On the other hand, the output battery voltage, V, which is the physical electric quantity that can be measured, differs from the OCV-q curves due to the voltage drop across the internal resistance of the battery. Based on the assumption in [13] that the battery resistance remains relatively constant along the OCV-q curve, this resistance was estimated for various temperatures, and the corresponding voltage drop was calculated and subtracted to obtain the OCV-q curves.

Following the method outlined in [14], the battery resistance was estimated in the time domain. When the fully charged battery begins discharging at 1C, an electric transient occurs. By selecting an appropriate time interval during which this transient can be considered extinguished, the internal resistance was determined as the ratio of the current to the voltage steps. The time interval was chosen by setting a threshold for the voltage derivative across all tests, below which the voltage variation could be ignored. In this case, it was 1 mV/ s. Finally, the segment of the OCV-q curves corresponding to this time interval was excluded.

B. Experimental Setup and Test Procedure

The experimental setup used to test the battery included a potentiostat (SP-150) and a 100 A booster (VMP3B-100) from Biologic Science Instrument, both managed via EC Lab software on a PC connected to the SP-150 through an Ethernet cable. To regulate the battery temperature, three Peltier cells were positioned beneath the battery. These cells were mounted on a heatsink equipped with two fans. The Peltier cells, connected in series, had their current regulated by a Texas Instrument DRV8303 inverter, which was controlled by an F28069M controller board utilizing a proportional-integral regulator. The inverter was powered by a DC voltage source, and a PT100 sensor measured the battery temperature, feeding this data to the F28069M controller board. Although the Peltier cells were only on one side of the battery, the temperature gradient across the battery was minimal due to its thin profile.

The output battery voltage was recorded at fixed reference temperatures ranging from 10 °C to 50 °C in 5 °C increments. For each temperature setting, the battery was fully charged using the constant current-constant voltage (CC-CV) protocol and then discharged using the CC protocol. During the charging process, a constant current of 10 A (1C) was applied until the voltage reached 4.2 V. At this point, the voltage was held constant at 4.2 V until the current dropped below 200 mA (0.02C). For the discharging process, the battery was discharged at a constant current of 10 A (1C) until it reached the lower cut-off voltage of 2.75 V.

III. Methods

A. Learning framework

In this work, we formulated the problem of estimating the SOC of a battery, given individual observations of pairs of OCV and battery temperature measurements, as a supervised ML problem. In such problems, the objective is to estimate a mapping f from input data $x \in \mathcal{X}$ to output data $y \in \mathcal{Y}$. The inputs x, also known as features, are typically represented by vectors of a fixed dimension D (corresponding to the number of features), i.e., $\mathcal{X} = \mathbb{R}^D$. For instance, in this study, the features are the measurements of the OCV, E, and battery

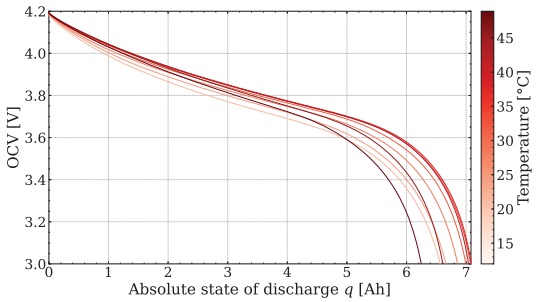

Fig. 2. Measured OCV-q curves.

temperature T, thus $D = 2$. The output y is often called the target, and in this case, it is the absolute state of discharge, q, of the battery. Since q is a continuous quantity, i.e., $q \in \mathbb{R}$, the problem of estimating SOC is specifically a supervised regression problem.

To address these types of problems, we assumed the availability of a set of N input-output observation pairs $\mathcal{D} = \{(x^{(i)}, y^{(i)}\}_{i=1}^{N}$, called the training dataset. Utilizing this dataset, we selected a function f_θ from a set $\mathcal{F} = \{f_\theta\}_{\theta \in \Theta}$ of candidate functions, parameterized by the parameter vector θ, such that each value assumed by the vector θ corresponds to a different function in the set $\mathcal{F}$. Specifically, we aimed to select the values of θ (and consequently the function f_θ in $\mathcal{F}$) that minimizes the empirical risk $L(\theta)$ of some scalar cost function $\ell(y, f_\theta)$, the latter representing a measure of discrepancy between the target and the model predictions. $L(\theta)$ is defined as the average loss computed over the dataset $\mathcal{D}$, i.e., $L(\theta) \triangleq \frac{1}{N} \sum_{i=1}^{N} \ell\left(y^{(i)}, f_\theta\left(x^{(i)}\right)\right)$ (a common choice for L is, e.g. the Mean Squared Error).

B. Dataset

The available dataset consists of the OCV-q curves shown in Figure 2. As mentioned in Section II, a total of 9 curves were measured for 9 different battery temperature values T, ranging approximately from 10 °C to 50 °C in uniform increments of 5 °C. Each curve is composed of approximately 245 samples on average for a total of 2212 samples. Since each curve had a different minimum value of E on the y-axis compared to the others, we decided to consider 3 V as the lower limit, which is reached by all the measured curves and is close to the actual cut-off value of 2.75 V.

C. Deep Neural Networks

Choosing the right model class $\mathcal{F}$ is a crucial design decision closely linked to the problem at hand. Estimating the SOC of a battery presents specific challenges: i) the mapping f between inputs and the target variable is non-linear and difficult to express in a closed form; ii) there is a need to extrapolate beyond the training distribution, such as predicting values at extreme temperatures within the dataset, like 10 °C and 50 °C; and iii) the limited number of features hinders the effectiveness of manual feature engineering. Points i) and iii) discourage the use of overly simplistic models like linear models, while point ii) rules out the use of ensemble tree-based methods (such as Random Forest and Gradient Boosting models) because these models generally fail to extrapolate well due to their training methods. For these reasons, in this work, we considered the parametric model class $\mathcal{F}$ of deep NNs. This choice is theoretically motivated by the universality of these models, the fact that the functional form of f does not need to be precisely specified, and the core concept of deep NNs, which is to lift the input features into a higher-dimensional space to automatically extract relevant latent features before projecting them onto the output space.

Mathematically, a NN consists of a composition of functions known as layers, $f_l : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$ for $l = 1, \ldots, L$, where $d_0 = D$ is the dimensionality of the input space, and d_L is the dimensionality of the output space [15]. For simple regression problems, for instance, $d_L = 1$. A NN is considered deep if it comprises more than two layers ($L > 2$); otherwise, it is referred to as a shallow NN.

In the simplest case, the layers f_l are affine transformations followed by the application of an activation function. Given an input vector $z_{l-1} \in \mathbb{R}^{d_{l-1}}$:

$$f_l(z_{l-1}) = \sigma_l(W_l z_{l-1} + b_l), \tag{4}$$

where $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and $b_l \in \mathbb{R}^{d_l}$ are learnable parameters known as weights and bias terms respectively, and σ_l is a non-linear activation function applied element-wise.

NNs structured as a composition of layers, $f = f_L \circ \cdots \circ f_1$, where each layer f_l is of the form (4), have been demonstrated to be universal approximators for the class of continuous functions [16]. This means they can approximate any continuous function to any desired degree of accuracy, provided the network is sufficiently large. Additionally, research has highlighted the representational advantages of using deeper networks. It has been shown that deep networks can represent certain functions with a polynomial number of parameters, which would require exponentially more parameters if attempted with shallow networks [17].

D. Deep Neural Network Training

Once the model class $\mathcal{F}$ is selected, as discussed in section III-A, our goal is to identify the best approximating candidate function $f_\theta \approx f$ using a training dataset $\mathcal{D}$. This involves solving the following optimization problem:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, L(\theta) \tag{5}$$

In the case of a deep NN, θ contains all trainable parameters, including weight matrices and bias terms for all layers. When the model f_θ is a deep NN, the number of parameter can be very large and the loss landscape $L(\theta)$ is generally highly non-convex, making the task of finding a good minimizer challenging. The most widely used approach to solve this optimization problem is to employ iterative algorithms in the form of some variant of first-order gradient descent. In practice, a variant known as mini-batch gradient descent

is often used, where parameter values are updated based on gradient estimates $\nabla_\theta L\left(\theta(k)\right)$ calculated using randomly selected subsets of the training dataset, known as batches. Each complete pass through the dataset is called an epoch. This algorithm is shown to converge toward minimizers that generally offer better generalization compared to standard gradient descent [18]. Furthermore, there are more advanced first-order optimizer variants based on gradient descent that incorporate features like momentum and adaptive learning rates. In our experiments, we specifically used mini-batch gradient descent and the Adam optimizer [19], which updates parameter values based on adaptive estimates of the first and second-order moments.

E. Validation strategy

In the context of ML, this work is situated within a small data regime, where the available data consists of 9 OCV-q curves, each curve approximately comprising 246 observations. Under these conditions, a well-thought-out cross-validation strategy is essential to achieve both a reliable estimate of generalization performance and optimal use of the available data.

An ML model, to be employed as a predictor, necessitates the tuning of both its hyperparameters and model coefficients. In the case of a feed-forward NN, the coefficients include the weights and biases of the network, while hyperparameters may comprise the number of hidden layers, number of neurons per layer, type of activation functions, learning rate, batch size, and the number of epochs selected through early stopping. Consequently, at least three independent datasets are required: a training dataset to determine the model coefficients, a validation dataset to optimize hyperparameter values, and a test dataset to assess the generalization performance of the optimized model.

In a practical application context, effective cross-validation must be designed with the specific use of the model in mind. For this project, the goal is to predict the SOC of a battery based on a given OCV value at a previously unseen discharge temperature. Therefore, randomly splitting the available data is not feasible for cross-validation in this scenario, as it would include all observed charge and discharge temperatures across all sets, leading to potential information leakage from the test dataset to the training dataset. This could ultimately result in an overly optimistic estimate of the model generalization performance.

In this setting, a reliable evaluation of performance requires data to be split twice according to a nested leave-one-curve-out cross-validation scheme [20]. Specifically, in the outer cross-validation loop, data from one of the n curves are held out. The remaining data from the other $n-1$ curves undergo a further inner cross-validation loop. In this inner loop, data from one of the available $n-1$ curves are held out at a time for validation, while the data from the remaining $n-2$ curves are used to train a model. This model then predicts the held-out curve at each iteration. The overall procedure is depicted in Figure 3.

Hyperparameters for the model were selected within this inner cross-validation loop by testing all possible combinations and choosing the one that yielded the best average validation performance across the folds. Finally, all $n-1$ curves were used to determine the model coefficients (training the model), and the resultant model was then used to predict the performance on the held-out curve of the outer loop (testing the model). This procedure was repeated for all n curves, thereby obtaining n independent performance evaluations, which were averaged to provide a final assessment of the model generalization performance.

Following this protocol, n distinct models were trained within the outer loop, with one model dedicated to each of the n curves. As a result, each iteration of the outer loop may yield different hyperparameters and model coefficients. However, in a real testing scenario, all n curves would be used to train the model. This step followed the determination of optimal hyperparameters using a single round of leave-one-curve-out cross-validation, ensuring that the final model is robustly trained on the complete dataset for maximum generalization capability.

IV. Numerical results

All computations were performed on a server with two 64-core AMD EPYC 7742 processors, 256 GB of RAM, and 4 Nvidia RTX 3090 GPUs. The hardware resources of the server were limited to 16 CPU threads and 1 GPU. Python was used as a programming language, with PyTorch framework for building and training the deep NNs. The code to reproduce the results is available at the GitHub repository: https://github.com/gabribg88/Deep_SOC_estimation.

The network size was set to $L = 10$ layers, each with 100 neurons, using the ReLU activation function $\sigma_l(x) = \text{ReLU}(x) = \max(0, x)$. Each network was trained using mini-batch gradient descent with a batch size of 16 and the Adam optimizer, with a learning rate $\lambda = 0.0001$, and with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-08$. We employed the PyTorch `ReduceLROnPlateau` learning rate scheduler with hyperparameters set to `factor` $= 0.9$, `patience` $= 10$, and `threshold` $= 0.01$, for a maximum of 1000 epochs. The actual number of training epochs for each training run was determined by early stopping, monitoring the Mean Absolute Error (MAE) in the validation set with a patience value of 20 epochs. Before providing the data to the NN, it was normalized using `StandardScaler` from Python open-source library `scikit-learn`.

A. Prediction results

As described in Section III, we used our model trained on the available dataset to predict values of q from pairs (V, T). Figure 4 shows the predictions achieved in reconstructing the available curves across various temperatures T. With temperature T fixed, we make predictions for each available measured value of E within the approximate range of $[3\text{V}, 4.2\text{V}]$. This approach, allow us to reconstruct the entire OCV-q curve. It is important to note, as previously explained, that each time we

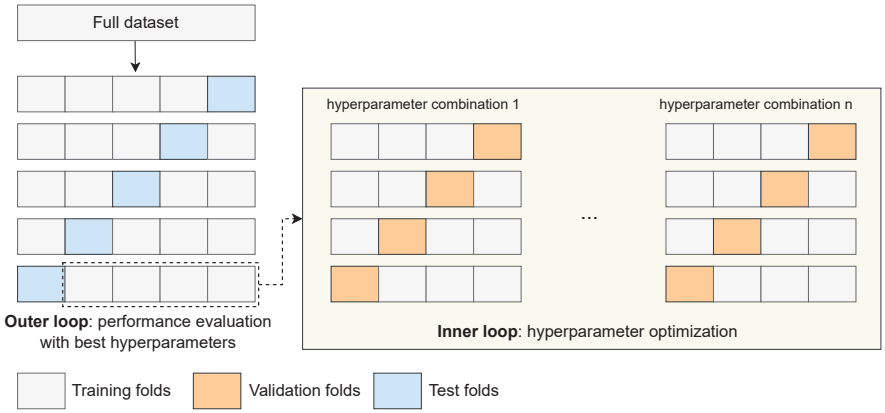

Fig. 3. Nested leave-one-curve-out cross-validation.

reconstruct a given curve, the model has been trained on all the other curves, meaning it does not have prior knowledge of the points it is estimating on that curve. Figure 4 also shows the MAE measured for each curve on all predicted q values; we emphasize that the obtained MAE is generally very low, indicating the accuracy of our estimation model. More specifically, the MAE is defined as: $\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} \text{AE}_i$, with $\text{AE}_i = |q(E_i) - \hat{q}(E_i)|$, and N being the number of samples available for each OCV-q curve. From a practical point of view, as explained in Section I, it is important for a user to be able to determine the SOC of their battery by measuring a voltage value $\bar{E}$ and the current battery temperature value $\bar{T}$. Since we showed in Figure 4 that our model can achieve good predictions of q, knowing the pair (E, T), we can calculate the SOC of a battery using formula (3), where q is the model prediction for the input pair $(\bar{E}, \bar{T})$, and C_a is the model prediction for the input pair $(3, \bar{T})$ ($E_{\text{min}} = 3$ V is the cut-off voltage value measured in practice, as explained in section III). The results demonstrating the accuracy of our model in predicting the SOC of a battery are shown in the boxplots of Figure 5. Each boxplot in the figure corresponds to a different curve in the dataset (thus, a different temperature value), and for each curve, the SOC was predicted for every available measured value of E. Each box shows the quartiles of the distribution of the prediction errors, while the whiskers extend to show the rest of the distribution, except for points that are determined to be "outliers" using a method that is a function of the inter-quartile range. In this case as well, the results are satisfactory since the prediction error on average remains below a single percentage point.

B. Sensitivity analysis

Finally, we also found it interesting to conduct a sensitivity analysis to see how our model can generally tell us how the temperature value influences the battery capacity. Indeed, as mentioned above, we can estimate the battery capacity value, C_a, as the model prediction $q(E_{\text{min}})$, obtained for a fixed T value and for the minimum voltage value in the dataset $E_{\text{min}} = 3$ V, which represents the cut-off voltage. The result is shown in Figure 6. Thanks to this analysis, we can attempt to predict the behavior of the capacity for temperature values that are entirely absent from the dataset. Naturally, the reliability of these predictions is higher the closer we remain to the interpolation zone. We want to emphasize that the results obtained for extrapolation at temperatures not present in the dataset would not have been possible with some ML models, such as Decision Trees and all algorithms derived from them. NNs are a more suitable ML model for extrapolation compared to others, as already discussed in section III.

V. Conclusion

In the present work, an ML approach based on deep NNs was used to develop a SOC estimation algorithm for EVs equipped with LiBs. Specifically, the effect of battery temperature on the OCV-q relationship was considered. Through the proposed algorithm, it is sufficient to measure just one experimental OCV point and the related battery temperature to properly estimate the remaining autonomy of the EV. The results show that the predicted SOC exhibited a deviation error, with an interquartile range below 2% and a median lower than 1% across all temperatures.

References

[1] C. Liu, F. Li, L.-P. Ma, and H.-M. Cheng, "Advanced materials for energy storage," *Advanced Materials*, vol. 22, no. 8, pp. E28–E62, 2010.
[2] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, "A review on the key issues for lithium-ion battery management in electric vehicles," *Journal of Power Sources*, vol. 226, pp. 272–288, 2013.
[3] S. Barcellona and L. Piegari, "Lithium Ion Battery Models and Parameter Identification Techniques," *Energies*, vol. 10, p. 2007, dec 2017.
[4] Y. Xing, W. He, M. Pecht, and K. L. Tsui, "State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures," *Applied Energy*, vol. 113, pp. 106–115, 2014.
[5] A. Farmann and D. U. Sauer, "A study on the dependency of the open-circuit voltage on temperature and actual aging state of lithium-ion batteries," *Journal of Power Sources*, vol. 347, pp. 1–13, 2017.
[6] A. Gismero, E. Schaltz, and D.-I. Stroe, "Recursive state of charge and state of health estimation method for lithium-ion batteries based on coulomb counting and open circuit voltage," *Energies*, vol. 13, no. 7, 2020.

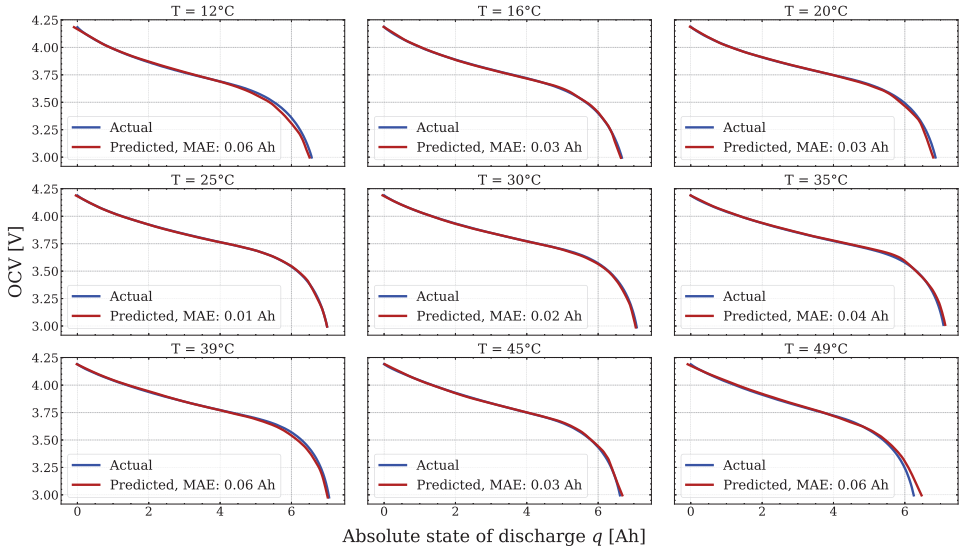

Fig. 4. Reconstruction of OCV-q curves using (E, T) pairs as input features

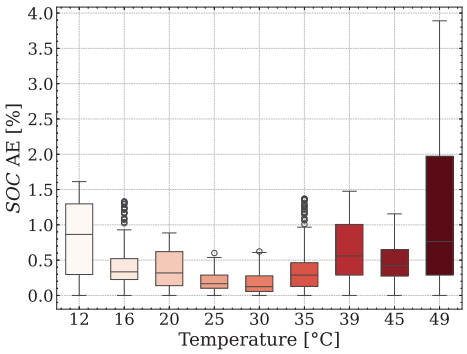

Fig. 5. Performances in terms of AE for each OCV-q curves on the estimation of SOC values using (E, T) pairs as input features

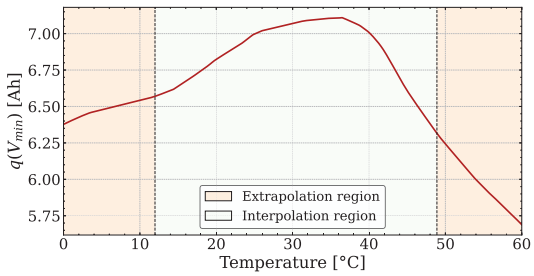

Fig. 6. Sensitivity analysis with interpolation and extrapolation: battery capacity (estimated as $q(E_{\text{min}})$ with respect to temperature.

[7] M. Cao, T. Zhang, W. Jia, and Y. Liu, "A deep belief network approach to remaining capacity estimation for lithium-ion batteries based on charging process features," *Journal of Energy Storage*, vol. 48, p. 103825, apr 2022.

[8] Z. Cui, C. Wang, X. Gao, and S. Tian, "State of health estimation for lithium-ion battery based on the coupling-loop nonlinear autoregressive with exogenous inputs neural network," *Electrochimica Acta*, vol. 393, p. 139047, oct 2021.

[9] E. Petkovski, I. Marri, L. Cristaldi, and M. Faifer, "State of health estimation procedure for lithium-ion batteries using partial discharge data and support vector regression," *Energies*, vol. 17, p. 206, dec 2023.

[10] S. Barcellona, L. Cannelli, S. Colnago, C. Laurano, and L. Piegari, "Cycle aging effect on lithium-ion battery resistance: a machine learning approach," *2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)*, pp. 389–394, 2023.

[11] K. Severson, P. Attia, N. Jin, *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, pp. 383–391, mar 2019.

[12] S. Barcellona, L. Codecasa, and S. Colnago, "Temperature effect on the open circuit voltage of lithium-ion battery," *under press*, 2024.

[13] L. Chen, M. Zhang, Y. Ding, S. Wu, Y. Li, G. Liang, H. Li, and H. Pan, "Estimation the internal resistance of lithium-ion-battery using a multi-factor dynamic internal resistance model with an error compensation strategy," *Energy Reports*, vol. 7, pp. 3050–3059, 2021.

[14] S. Barcellona, S. Grillo, and L. Piegari, "A simple battery model for ev range prediction: Theory and experimental validation," *2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC)*, pp. 1–7, 2016.

[15] L. Lucchese, M. S. Pakkanen, and A. E. Veraart, "The short-term predictability of returns in order book markets: a deep learning perspective," *International Journal of Forecasting*, 2024.

[16] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[17] M. Telgarsky, "Representation benefits of deep feedforward networks," *arXiv preprint arXiv:1509.08101*, 2015.

[18] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points—online stochastic gradient for tensor decomposition," in *Conference on learning theory*, pp. 797–842, PMLR, 2015.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] M. Meroni, F. Waldner, L. Seguini, H. Kerdiles, and F. Rembold, "Yield forecasting with machine learning and small data: What gains for grains?," *Agricultural and Forest Meteorology*, vol. 308, p. 108555, 2021.