An Improved Method Based on Support Vector Regression with Application Independent Training for State of Charge Estimation

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

PLEASE NOTE: We cannot accept new source files as corrections for your article. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

Carefully check the page proofs (and coordinate with all authors); additional changes or updates WILL NOT be accepted after the article is published online/print in its final form. Please check author names and affiliations, funding, as well as the overall article for any errors prior to sending in your author proof corrections.

AQ:1 = According to our records, Mattia Stighezza is listed as a Member, IEEE. However, the files provided list him as a Student Member, IEEE. Please verify.

AQ:2 = Please confirm or add details for any funding or financial support for the research of this article.

AQ:3 = Please provide the expansion of the acronym FIL for your funding agency. Providing the correct acknowledgment will ensure proper credit to the funder.

AQ:4 = Please provide the appropriate section number for the phrase "previous section."

AQ:5 = This term "master-slave" may be considered non-inclusive. Please consider replacing with "primary/secondary" instead.

AQ:6 = Please provide the report no. for Ref. [5].

AQ:7 = Please provide the volume no., issue no. or month and page range for Ref. [24].

An Improved Method Based on Support Vector Regression With Application Independent Training for State of Charge Estimation

AQ:1 Valentina Bianchi, *Senior Member, IEEE*, Mattia Stighezza, *Member, IEEE*, Andrea Toscani, Giovanni Chiorboli, and Ilaria De Munari, *Senior Member, IEEE*

***Abstract*— Nowadays, lithium-ion (Li-ion) is among the most used chemistry for batteries and shows an increasing market growth rate; however, to reduce failure or safety risks, the battery state-of-charge (SoC) must be accurately monitored and predicted by a suitable battery management system (BMS). Artificial intelligence (AI) techniques have been extensively applied to this field with good results. Typically, AIs are trained on dynamic profile data, emulating battery charging and discharging cycles related to the application under test. In this article, a novel approach is presented: application-independent constant current profiles are used to train a support vector regression (SVR) algorithm. To enhance the estimation accuracy, the output of the obtained SVR model was postprocessed. Finally, an error correction algorithm was applied to further reduce the estimation error. The system is validated over test cycles, representing different application scenarios for the battery cell operations. For the development of the proposed approach, a total of 105 constant current discharge profiles for the training and 20 realistic test cycles for the validation have been considered, including standard automotive cycles and a generic battery-powered power tool. The performance in the SoC estimation resulted in a root-mean-square error (RMSE) of 0.94% and a mean absolute error (MAE) of 0.75% over all the test cycles. Error metrics are comparable to those obtained for SoC estimation AI algorithms based on traditional approaches using application-dependent battery profiles for the training phase.**

***Index Terms*— Batteries, battery management system (BMS), lithium-ion (Li-ion) batteries, state of charge (SoC), support vector machines (SVMs).**

I. Introduction

THE global lithium-ion (Li-ion) battery market is expected to reach U.S. $182.53 billion by 2030, with a compound annual growth rate of 18.1% from 2022 to 2030 [1]. Many portable electronic device products based on Li-ion batteries have been developed, ranging from mobile phones, laptops, medical instruments, and wearable devices to high-power applications, including cordless power tools [2], [3]. Nowadays, Li-ion-based chemistry is among the most commonly used, a success that can be ascribed to several advantages [4] such as high energy density, low self-discharge rate, high voltage of about 3.6 V, lightweight, good safety, and excellent cycling performance in addition to a sharply decreasing trend in the cost over the next years [5].

Battery management system (BMS) plays a key role in an efficient, safe, and reliable operation of a battery pack. The BMS guarantees continuous control and monitoring, acquiring voltage, current, temperature, and evaluating the state-of-charge (SoC) and the state-of-health (SoH) indicators to prevent the cell to work outside the safe operating area and avoid overstress situations leading to premature failure and safety risks [6]. Indeed, a battery pack is usually composed of modules consisting of several cells connected in series and/or in parallel to obtain the desired pack voltage and capacity; however, due to manufacturing tolerances, they do not work exactly in the same way, and the battery pack does not show its maximum capacity. Different cell balancing methods have been proposed in the literature [7], [8], and different topologies can be employed for a battery pack, and therefore, for the BMS design [9], [10], [11]. But, for all of them, it is always crucial to know the SoC of the cells to ensure the optimum functioning of the battery pack [12]. The battery SoC gives information on the available battery energy and represents the foundation of the BMS. It provides the percentage of the remaining capacity with respect to the maximum available one [13]; however, since batteries are complex electrochemical devices with a behavior depending on various conditions, accurate SoC estimation is a challenging task. SoC cannot be directly measured and can only be predicted from measurable parameters such as cell voltage, current, and temperature. For this reason, many battery SoC evaluation approaches with different levels of accuracy and implementation complexity have been studied and reported in the literature. To assess the goodness of each approach, along with the prediction accuracy, some other parameters can be taken into account, such as the capability of an online/offline computation, the suitability to be applied to in situ measurements or only in a laboratory environment and the need to determine the battery model parameters with prior experiments [13].

Artificial intelligence (AI) techniques have recently gained significant importance in battery SoC estimation [14]. The inferred models, i.e., the data-driven models (DDMs), do not require the implementation of an equivalent battery circuit

AQ:2 Manuscript received 7 June 2023; revised 18 July 2023; accepted 1 August 2023. This work was supported in part by the Project "Advanced Approaches for Batteries State of Charge (SoC) Evaluation," funded in 2021 by Programme AQ:3 "FIL-Quota Incentivante" of University of Parma; and in part by Fondazione Cariparma. The Associate Editor coordinating the review process was Dr. Ziqiang Cui. *(Corresponding author: Valentina Bianchi.)*

The authors are with the Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze, 43124 Parma, Italy (e-mail: valentina.bianchi@unipr.it; mattia.stighezza@unipr.it; andrea.toscani@unipr.it; giovanni.chiorboli@unipr.it; ilaria.demunari@unipr.it).

Digital Object Identifier 10.1109/TIM.2023.3306816

(like in the electrical circuit model approach [15]) nor deep knowledge of the electrochemical process inside the battery (electrochemical model [16]). The DDM has the substantial advantage of describing the behavior of the battery without prior knowledge of its internal structure. Various DDMs, including artificial neural networks (ANNs), Adaptive Neuro-Fuzzy Inference Systems, deep neural networks (DNN), and support vector machines (SVM), have been presented in the literature [14], [17], [18]. Among the presented models, SVM has been recognized as one of the most interesting to predict SoC for its capability to deal with complex nonlinear functions [19]; moreover, it is suitable for implementation on embedded systems based on field programmable gate arrays (FPGAs) or microcontrollers, facilitating the integration in a BMS [20].

Usually, the DDM is trained, validated, and tested on current, voltage, and temperature measurements acquired, emulating the battery load use. In [14], a review of AI-based approaches to SoC estimation for Li-ion batteries is presented. SVMs, Gradient boosting, and ANN techniques are reported and compared. The application field is the electric vehicles (EVs), and all the models refer to driving cycles for both training and testing. The same approach is reported in [21] and [22], where battery SoC estimation models based on deep learning and machine learning techniques are discussed. Also, for evaluating the SoC for energy storage systems for grid services, the DDM is trained and tested on application-specific data as acquired in the field [23], [24]; therefore, the system validation is based on data that are similar to those used in the training phase. Actually, this allows obtaining good accuracy in SoC prediction but makes the developed system dependent on the particular application and on discharge profiles used in the training. For example, in the field of EVs, the existing SoC estimation is not suitable for vehicles running with velocity profiles that are different from those used to define the model [6]. Changing the application context, therefore, implies the reexecution of the training procedure, which is usually very long and requires the recording of a new large number of measurements; moreover, if the system is implemented on an embedded solution for in situ measurements, a redesign would be needed.

In this article, a novel approach is proposed. A DDM model, based on the SVM regression support vector regression (SVR), is designed by exploiting a series of constant current cell battery discharge curves. Then, the trained model is applied to different cycles, emulating the discharge profile of generic load-absorbing currents in the same range of the training set. The aim is to train a model independently of the specific application: this phase could be easily carried out by battery manufacturers or research laboratories, freeing the BMS designers from this process. A very first attempt to implement this approach has been presented in [25]; however, in that work, only five discharge cycles have been collected for three different constant currents. Four of them have been used for the training and the remaining for the test. The validation of the approach was then very limited and performed over the same cycles exploited in the training. Instead, in this proposed work, the SVR trained with the constant current profiles is tested with a set of random currents in the same range but with nominal values that can also be different from the train currents; moreover, as a case of study, the proposed approach is used to estimate the SoC during the operation of a battery-powered drill. The discharge profile of such a tool has been acquired following the discharge profile reported in [26]. Finally, the approach was verified considering a US06 supplemental federal test procedure (SFTP) driving cycle [27].

The accuracy of the SoC estimation was, moreover, improved by adding a digital signal processing phase at the output of the trained model, consisting of a low-pass filter (LPF) and a calibration algorithm. This stage was conceived to be performed on board with calibration parameters computed during the first discharge cycles according to the selected application.

Hence, the contribution of this article can be summarized as follows.

1) A new training approach based on constant current discharge profiles, aiming at minimizing the burden of training a specific SVR model for each different application field.
2) A new method for the SoC assessment that improves the estimate obtained from a conventional SVR through a calibration algorithm.

This article is organized as follows: in Section II, the related works are discussed; in Section III, a description of the proposed method is presented; then in Section IV, results are presented; finally, in Section V, conclusions are drawn.

II. Related Works

The SVM technique belongs to the supervised learning algorithms. Unlike unsupervised learning, all the supervised learning algorithms require input–output pairs for the training phase, in which the output represents the expected value for the specific input vector. In the SoC estimation application, a set of battery-related features is the input vector x_i to be mapped to the corresponding battery cell SoC (i.e., the output $\widehat{\text{SoC}}$). In the regression form of the SVM algorithm, from a new input vector x, the expected output $\widehat{\text{SoC}}$ can be inferred by applying the regression function in the following equation:

$$\widehat{\text{SoC}} = \sum_{i=1}^{\text{NSV}} (\alpha_i - \alpha_i^*) K(x, x_i) + b \tag{1}$$

where α_i and α_i* are the Lagrange multipliers, $K(x, x_i)$ is the kernel function, NSV is the number of Support Vectors, and b is a bias term. In (1), the kernel function has the highest impact on the algorithm performance. Different choices for the kernel function are available (e.g., linear, polynomial, radial basis function (RBF), etc.) [19], depending on the exploited input features and the expected model output. Once the input features and the kernel function have been defined, the other parameters are obtained after solving the SVM quadratic problem [14]. Different parameter sets result in different solutions to the quadratic problem, hence in different model performance. An iterative optimization process is usually applied to the training process to find the parameter set that allows achieving the minimum estimation error.

For each iteration, the obtained model is validated with the k-fold cross-validation technique [28], [29]. This process can be time-consuming. As the algorithm is, moreover, usually trained and then tested on the same application-specific profiles, the training process must be repeated if the context changes.

In the literature, root mean square error (RMSE) and mean absolute error (MAE) metrics are commonly adopted to evaluate the performance of the reported methods. They are defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum\nolimits_{k=1}^{m}\left(\widehat{\mathrm{SoC}}_k - y_k\right)^2} \tag{2}$$

$$\mathrm{MAE} = \frac{1}{m}\sum_{k=1}^{m}\left|\widehat{\mathrm{SoC}}_k - y_k\right| \tag{3}$$

where $\widehat{\mathrm{SoC}}_k$ represents the predicted value of SoC at the observation k while y_k is the observed value and m is the number of observations.

In recent years, several works have been presented on the evaluation of the SoC of Li-ion batteries by model-based approaches. Most of them are focused on EV applications, and only a few works report studies on different fields [23], [24].

In [30], a hybrid method is proposed, combining an improved bidirectional gated recurrent unit (IBGRU) network and unscented Kalman filtering (UKF). The method is trained and validated on Li-NMC data emulating UDDS and US06 driving cycles, obtaining an RMSE of 1.12% and an MAE of 0.83%. A joint approach of adaptive extended Kalman particle filter (aEKPF) with Bayesian belief network (BBN) is presented in [15]. A model was trained under UDDS driving cycle conditions and another model for ECE conditions. For each model, the tests are performed over the same profiles used for the training with a maximum RMSE of 0.38% and a maximum MAE of 0.52%.

A Bayesian-optimized bidirectional long- and short-term memory neural network (NN) is introduced in [31] with results in terms of RMSE and MAE of 0.89% and 0.60%, respectively. In [32] a long short-term memory (LSTM) DNN is developed, and to enhance SoC estimation, the input of the DNN is augmented, introducing information from an equivalent circuit model. Differently from the other works, here, real-world driving profiles were used to train the model. An RMSE and an MAE of 1.85% and 3.35% were computed.

In [33], a physics-constrained NN was trained on data collected by applying four different driving cycle current profiles to the battery cell and then tested with a single hybrid driving cycle case consisting of the same four profiles, obtaining an RMSE of 0.36% and an MAE of 0.83%; moreover, this model has a maximum RMSE in validation of 0.67%.

The least squares SVM (LSVM) model was trained in [34] on battery data emulating FUDS driving cycles and tested on DST cycles, obtaining an RMSE of 1.14% and an MAE equal to 7.78% at 25 °C. Great improvement in the algorithm is obtained by introducing models, including sliding window methods with feedback vectors (RMSE = 0.20%, MAE = 0.21%). This strategy, however, complicates the model, especially for implementation on embedded devices.

SVR has been successfully applied to various nonlinear classification and regression problems [35]. It works on the principle of SVM, but it predicts the value of output instead of classifying the output into categories. In [36] several kernel functions of SVR were investigated and applied to the SoC estimation problem. The RBF kernel has been demonstrated as the best solution. It has also been proved that the use of only three features (voltage, current, and temperature) reduces model complexity with a minimal impact on performance. An RMSE of 1.18% and an MAE of 0.94% have been obtained. In [37], an online SVR machine able to update the model in real time is presented. The RMSE obtained during the test of the system is 1.72%, while the MAE is 6.58%.

SoC estimation for applications different from EV is reported in [23], where, with an ANN, an RMSE of 1.6% and an MAE of 1.2% are reached.

In all the aforementioned articles, experimental results are carried out on tests performed using the same application-related profiles also exploited in the training phase.

Few works so far have investigated the more general case in which training and testing are not so related to each other. In [22], an SVR approach is adopted, employing a low-complexity linear kernel combined with an ant colony optimization (ACO) algorithm. In this work, the algorithm is trained with different EV profiles than those used in the test, obtaining an RMSE and MAE of 1.4% and 1.2%, respectively.

In [38], an RBF-NN has been exploited. The training set was composed of constant current discharge cycles mixed with EV discharge cycles; thus, the training set was not completely application dependent. When tested with application-specific profiles, the authors found an RMSE of 4.45%. If the test set is composed of both specific profiles and constant current profiles, as the training set, the error drops to less than 2%. In [39], the problem of obtaining a general model is addressed with transfer learning. Here, the focus is not to extend the estimation capability of the model to other discharge profiles but to other battery types. The training set was composed of different automotive-related driving cycles, and the test set was a mixed drive cycle. The authors achieved an RMSE of 2.47% by applying a DCNN to different scenarios. A similar approach was investigated in [30], where to demonstrate the validity of the results obtained, the performance has been also estimated after transfer learning on US06 data coming from a different battery, obtaining an MAE and RMSE of 1.32% and 1.51%, respectively. Finally, in [40], the same problem is approached with a CNN combined with transfer learning, obtaining an RMSE of 1.37% and an MAE of 1.12%.

III. Materials and Methods

A. Proposed SoC Estimation Method

In this work, a novel approach based on an SVR model has been used for the evaluation of the battery SoC. Instead of training the model with discharge profiles compliant with a specific application, the model was trained using constant discharge currents. The goal is to demonstrate that this model is then applicable to different contexts without requiring different model training for each one. The MATLAB environment and

Fig. 1. Schematic of the proposed SoC estimation method.

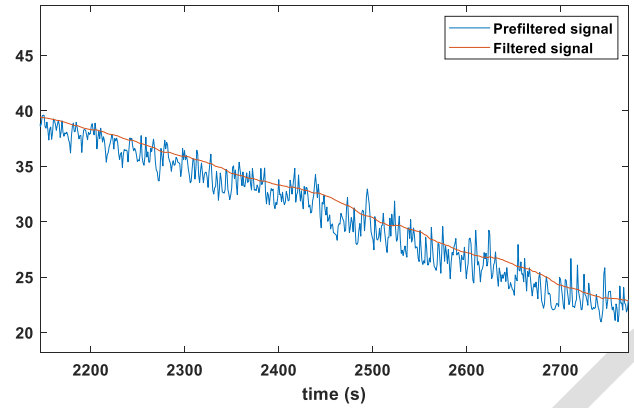


Fig. 2. Comparison between prefiltered estimated SoC signal and filtered estimated SoC signal in the time domain.

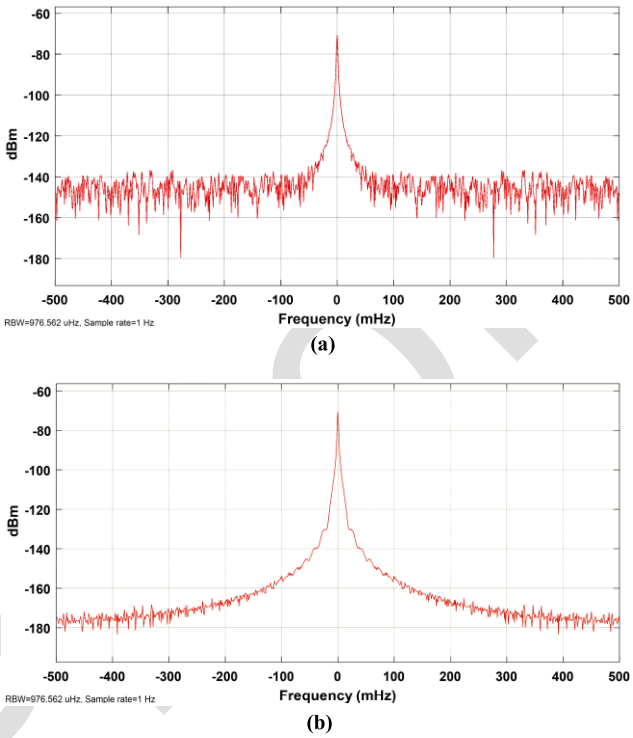


Fig. 3. (a) Example of the prefilter signal spectrum. (b) Example of filtered signal spectrum.

the Bayesian algorithm have been used to train and optimize an SVR model with an RBF kernel. This kernel was selected since it has been demonstrated as the best approach to the SoC estimation [36].

Once trained, the SVR model has to be applied to different application contexts. To improve the estimation accuracy of the trained SVR, data postprocessing stages have been applied to the model output, consisting of an LPF and a calibration algorithm. In Fig. 1, a schematic of the applied algorithm in the test phase is shown.

AI techniques are prone to produce abrupt changes in the predicted value of the SoC [33], which can worsen the prediction accuracy. To limit this effect, the variation of the SoC has been limited to discard unfeasible values (i.e., values beyond the physical thresholds of 100%–0%); moreover, a 64-tap FIR filter has been applied to smooth the spikes in the SoC estimation, considering a normalized cut-off frequency of 0.01 Hz. In Fig. 2, an example of estimated SoC before and after filtering is shown, while in Fig. 3, the corresponding spectra are reported.

Finally, a calibration stage has been applied. The error ε in the estimation of the SoC can be expressed as

$$\varepsilon = \widehat{\mathrm{SoC}} - y. \tag{4}$$

A calibration line can be computed by linear fitting, and then it can be used to correct the estimated value of the SoC, i.e., $\hat{\mathrm{SoC}}$. In particular, the corrected SoC (SoC_c) can be computed as

$$\mathrm{SoC}_c = \frac{\widehat{\mathrm{SoC}} - q}{1+m} \tag{5}$$

where m and q are the slope and the intercept, respectively, of the calibration line.

To consider possible nonlinearities occurring in the whole SoC range (i.e., from 100% to 0%) due to the intrinsically nonlinear behavior of the SoC, the full range has been divided into smaller 10% width ranges. For each of them, a calibration line has been evaluated.

The calibration correction is intended to be applied onboard to the application-specific data. The simple linear regression (SLR) algorithm can be considered for the onboard implementation [41], [42]. The SLR fits the data with the best-fit line in least squares terms, i.e., minimizes the residual errors between each data point and the best first-order polynomial line. The cost function to be minimized is then

$$J(m,q) = \sum_i (mx_i + q - y_i)^2 \tag{6}$$

where m and q are the slope and the intercept of the line, respectively, and (x_i, y_i) are the coordinates of each data point to be fit. Equating the gradients of J to zero allows to solve for m and q [41], [42]. The solutions can be simplified by substituting with the sum of squares, defined as

$$S_x = \sum_i x_i \tag{7}$$

$$S_y = \sum_i y_i \tag{8}$$

$$S_{xx} = \sum_i x_i^2 \tag{9}$$

$$S_{xy} = \sum_i x_i y_i. \tag{10}$$

Then, the best-fit line can be computed with low effort through the following pseudo-code.

As can be seen, the computational complexity of the fitting algorithm is $O(n)$. The code above is a mathematically

Algorithm 1 Best Fit Line Computation

$S_x \leftarrow 0$
$S_y \leftarrow 0$
$S_{xx} \leftarrow 0$
$S_{xy} \leftarrow 0$
define n = number of data points
for $i = 1{:}n$
$S_x \leftarrow S_x + x(i)$
$S_y \leftarrow S_y + y(i)$
$S_{xy} \leftarrow S_{xy} + x(i) * y(i)$
$S_{xx} \leftarrow S_{xx} + x(i) * x(i)$
end
m$\leftarrow$ (n$*$ S_{xy} - S_x $*$ S_y) / (n$*$ S_{xx} - S_x $*$ S_x)
q$\leftarrow$ (S_{xx} $*$ S_y - S_x $*$ S_{xy})/ (n$*$ S_{xx} - S_x $*$ S_x)

return m and q

efficient formulation of the general least square algorithm employed by the MATLAB polyfit function. Considering a 32-bit Arm[1] Cortex-M4[1] microcontroller, the averaged error computed when estimating the parameters m and q with the algorithm described above with respect to the MATLAB polyfit function are 3.5×10^{-5} and 2.5×10^{-5}, respectively.

It is worth noticing that the use of data from the specific application in the calibration stage does not jeopardize the generality of the approach. Indeed, unlike the training phase, whose computational complexity is $O(n^3)$ [43], and it is scarcely executable on devices with limited computing resources like microcontrollers, the calibration process can be performed when the battery is being used by the final users; moreover, differently from the training phase, the calibration needs less data that can be collected during the in-field estimation. Hence, the proposed approach allows programming of the firmware of the device only once with the trained model and then modifying the calibration parameters during the first cycles of operation. Then, if the device should be applied to a new scenario, a new calibration process will be required without the need to retrain the model and reprogram the firmware of the device.

B. Measurement Setup

To validate the proposed approach, data from a brand-new 18 650 Li-ion 2.75 Ah Panasonic NCR18650PF cylindrical battery cell with nickel–manganese–cobalt oxide (NMC) chemistry have been acquired. This type of battery has a large diffusion, and it is used in different contexts, i.e., EVs, electric bicycles, energy storage systems, power tools, and medical instruments [4]. Voltage, current, and temperature data have been gathered across several discharge cycles, repeated at different constant current values. To avoid damaging or early aging of the cell, a standard constant current–constant voltage (CC-CV) procedure has been exploited for the battery charging phase, accomplishing the specific battery datasheet [44], reported in Table I. A programmable 200 W ITECH IT-M3412 bidirectional power supply can handle both charging and discharging phases, acting as an electronic load. It is rated for 60 V and ±30 A; hence, it is perfectly suitable for cell-level operations. When working in constant voltage mode, the accuracy is better than 0.1% of the maximum voltage, whilst the programming accuracy in constant current mode is better than 0.1% plus 0.1% of the full scale.

TABLE I
PANASONIC NCR18650PF ELECTRICAL CHARACTERISTICS

Chemistry	LiNiCoMnO2
Nominal capacity at 25 °C after standard CC-CV charge [Ah]	2.75
Nominal Voltage [V]	3.6
Voltage cut-off [V] (lower-upper)	2.5 – 4.2
Max. discharge current [A]	10
Standard CC-CV cut-off current [A]	0.10

TABLE II
DMM SETTINGS AND SPECIFICATIONS

	Instrument		
	DMM 1	***DMM 2***	***DMM 3***
Measured Cell Feature	Voltage	Current	Temperature
Configuration	Voltage	Voltage	4-wire resistance
Range	10 V	0.1 V	100 kΩ
Resolution	100 µV	1 µV	1 Ω
Accuracy ± (% reading + % range)	0.0035 + 0.0005	0.0035+ 0.0005	0.01 + 0.001

The workbench setup, shown in Fig. 4, aims to acquire cell voltage, current, and temperature. For the measurements, three Agilent 34401A digital multimeters (DMMs) have been used; all DMMs operate with auto-zero instrument function enabled to avoid any drift, while the auto-range function has been disabled so as not to slow down the readings/s, held to 10. The selected reading/s is a common trade-off between monitoring accuracy and computational demand [20]. Detailed considerations on the instrument configuration, resolution, and accuracy are presented in Table II.

The cell voltage has been measured by DMM 1 in a 10 V range configuration with 100 μV resolution (Table II), while cell current has been estimated by measuring, with the DMM 2, the voltage across a 10 mΩ shunt resistor, specified for 1% tolerance and less than 50 ppm/K temperature coefficient. The voltage resolution of 1 μV in the 0.1 V range corresponds to a 100 μA current resolution.

Finally, the cell temperature has been measured through a 10 kΩ ± 1% NTC thermistor, longitudinally applied on the cell's surface as shown in Fig. 5. The thermistor has

[1]Registered trademark.

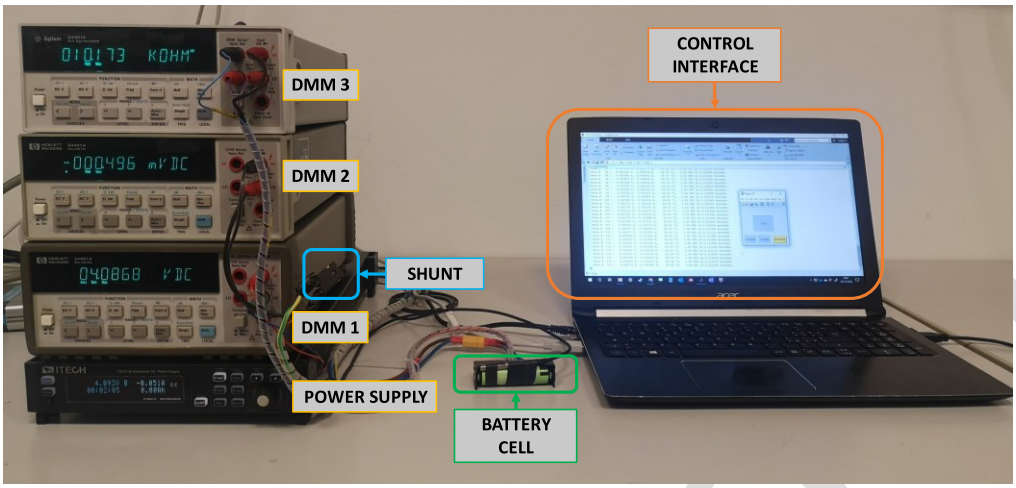


Fig. 4. Workbench setup for battery cell measurements.

Fig. 5. Placement of the NTC thermistor into the battery holder.

a resistance value at 25 °C of 10 kΩ and a characteristic temperature of the material of 3435 K so that the resistance has been measured in the 100 kΩ range with a resolution of 1 Ω (i.e., DMM 3 in Table II) not considering temperatures below −20 °C. The estimated model accuracy of the NTC is below 1 °C.

Each DMM communicates with a desktop PC through an Agilent 82357B USB/GPIB interface in a daisy-chain configuration, allowing a plug-and-play connection. A MATLAB script controls DMMs by sending IEEE 488 standard SCPI commands for initial configuration and memory data fetching. The same script also interfaces with the IT-M3412 power supply by sending SCPI commands through an ethernet connection to its fixed IP address, allowing switching between charge/discharge operations. The whole system has been wired using AWG 14 copper wires for power delivery, compliant with the wire section range advised in the IT-M3412 datasheet and allowing low parasitic voltage drops.

C. Acquired Dataset

The training dataset consists of current, voltage, and temperature measurements acquired during discharge cycles at constant currents. All the cycles have been started at the ambient temperature.

At the beginning, the range from 3 to 4 A was considered, with steps of 0.1 A. This current range was chosen as it is compatible with those of some commonly used battery-powered tools. For each current value, a set of five discharge cycles has been collected. The recorded dataset, with a reading every 100 ms, resulted in 55 discharge profiles for a total of more than one million of data points.

After the complete discharge of the cell, a CC-CV charge cycle has been applied to restore the SoC of the cell to 100%. Each cell charge/discharge phase has been followed by 30 min resting period before starting the next phase, to allow for the disappearance of polarization effects [45]. This is required to ensure that each charge/discharge phase is independent of the previous battery state.

To test model estimation performance, several test cycles have also been acquired. In each test cycle different random current values in the range 3–4 A were exploited. In this case, a minimum variation of 0.1 mA was set, different from the 0.1 A steps used for the constant current acquired to train the model. This guarantees the test set, also including discharge current values different from those considered in the training.

To test the generalization capability of the approach and to prove that extending to a different current range is feasible, the whole process has been repeated in the current range from 2 to 3 A. The current step was the same as the 3–4 A range, and for each step, five current profiles were acquired. A new test set corresponding to the new range was also acquired.

Finally, training and test sets covering the full range of currents (i.e., from 2 to 4 A) were also considered. This allows the evaluation of the flexibility of the approach, allowing the battery manufacturers to develop and make available different models, according to the needs of the final application.

Moreover, as an additional approach validation, some realistic battery-powered drill current profiles [26] have also been acquired, appropriately scaled to the considered current ranges. The proposed approach was also tested over a discharge current compliant with the US06 driving cycle.

A total of 105 training cycles and 20 test cycles were acquired.

IV. Results and Discussion

A. SVR Model Training

The training input features for the SVR algorithm are current, voltage, and temperature data acquired by the DMMs, as explained in the previous section. To compute the expected SoC, the Coulomb counting (CC) technique has been exploited [33]. This method involves current integration over time and is prone to accumulate errors. The CC

TABLE III
DATA DOWNSAMPLING EFFECT ON TRAINING TIME

Training Data Readings/s	Training Time	RMSE (%)
2	15 days	0.9
1	6 days	0.6
0.1	7 hours	1.1
0.01	5 minutes	1.3

algorithm is, therefore, suitable only in the case the input data are acquired through high-accuracy instrumentation, and the initial state of the battery is known a priori, as in laboratory workbenches [46]. This method is not advisable for real-time on-field estimation.

The SVR model has been designed in the MATLAB environment through the embedded regression learner app. An RBF kernel has been exploited, which is a popular kernel choice when dealing with nonlinear problems such as battery state estimation [6]. The SVR parameters have been optimized through the Bayesian algorithm, set to run for 100 iterations. In each iteration, the model parameters were validated by performing a ten–fold cross validation on the training set. As a result, the cross-validated model with the minor RMSE and the related optimum parameter set have been determined.

Since in [47], it has been demonstrated that a BMS sampling data at 2 Hz successfully runs and controls the algorithm performed, as a first attempt, the training dataset has been downsampled at this frequency.

Considering the training dataset ranging from 3 to 4 A, down sampled to 2 Hz, and a personal computer with an 11th gen. Intel i9-11900 core at 2.5 GHz and 32 GB RAM, a trained model was obtained in about two weeks, reaching a validation RMSE of 1.4%. This time is significantly long, especially when considering a real scenario in which, possibly, a multitude of SVR models must be trained on different ranges. To test the feasibility of maintaining the same accuracy in evaluating the actual SoC while shortening the time required to obtain models, more severe downsampling rates have also been evaluated, resulting in datasets sampled every 1, 10, and 100 s. The required training timing and the related validation accuracy are reported in Table III.

As it can be seen from the results, the training time is subjected to a highly nonlinear drop with a higher down sampling. This is probably due to the fact that the training time depends not only on the dataset size but also on other parameters like the computing load of the processor. As far as the validation accuracy is concerned, the results show that there is not a straightforward dependency of accuracy on data sampling. It is noticeable that in validation, the same kind of data is exploited for both training and testing (i.e., constant discharge currents). The great improvements in the training time made the 100 s dataset the most suitable choice for further investigation in this article from now on.

B. Testing Phase

To prove the feasibility of a model trained with constant current profiles in a different application scenario, discharge profiles with random currents were considered for the test. For each considered current range, five test cycles were acquired.

To apply the calibration algorithm, one of the test cycles was used to compute the linearization parameters through the linear fit of the error in the estimation of the SoC with respect to the expected SoC value, as illustrated in Section III-A. Then, the calculated parameters were applied to the remaining four tests, and the resulting RMSE was calculated as the average of the RMSEs obtained on the individual tests. To avoid the choice of the calibration test influencing the results, this approach was repeated on each of the five tests. So, the final RMSE is the result of the average of all RMSEs obtained during the process. This resulted in a total of 80 tests.

To analyze the effectiveness of the postprocessing stages shown in Fig. 1, Table IV shows the average RMSE and MAE values after each stage, applying random discharge profiles in the range from 3 to 4 A. As it can be seen the SVR estimation accuracy is greatly affected by the fact that training and test include different profiles and different steps of currents. The subsequent filtering, however, reduces the RMSE error by 28% and the error correction stage by a further 58% with respect to the filtering stage. Considering the total data postprocessing phase, the improvement in RMSE is found to be about 70%.

To further prove the effectiveness of the proposed algorithm, SVR models with different kernels (i.e., linear, quadratic, and cubic) were also evaluated. In Table V, results obtained applying only the SVR stage were compared with the performance of the proposed method, including the filter and the error correction stages. Results demonstrate that the proposed algorithm is effective in improving the performance of a conventional SVR model with all the kernels considered. In particular, the

TABLE IV
TEST RESULTS WITH RANDOM DISCHARGE CYCLES IN 3–4 A RANGE

Stage Output	RMSE (%)	MAE (%)
SVR model	3.98	3.57
Fir Filter	2.87	2.49
Error Correction	1.2	1

TABLE V
TEST RESULTS WITH DIFFERENT SVR KERNELS IN 3–4 A RANGE

SVR Kernel	SVR Model output		SVR Model + Filter +Error Correction Output	
	RMSE (%)	MAE (%)	RMSE (%)	MAE (%)
Linear	4.98	4.22	2.58	2.38
Quadratic	5.12	4.29	1.31	1.14
Cubic	4.24	3.78	1.2	1.05
RBF	3.98	3.57	1.2	1

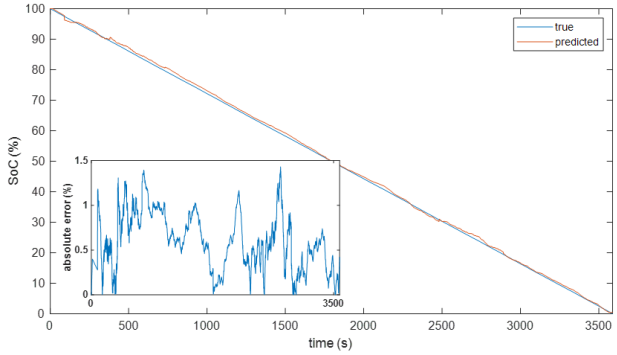


Fig. 6. Comparison between the estimated and the expected SoC signals. In the box, the absolute error is shown.

TABLE VI
TEST RESULTS WITH RANDOM DISCHARGE CYCLES ON DIFFERENT RANGES

Current Range	RMSE (%)	MAE (%)
2 A – 3 A	1	0.89
3 A – 4 A	1.2	1
2 A-4 A (one SVR model)	0.8	0.65
2 A – 4 A (multiple models)	0.77	0.61

proposed method improves the RMSE of 48%, 74%, and 72% in the case of linear, quadratic, and cubic kernels, respectively. The RMSE and MAE after error correction in cubic and RBF kernels are approximatively the same, but the RBF has the lowest RMSE and MAE after the SVR model output, as also reported in the literature [36]. For this reason, the following tests were carried out, considering only the RBF kernel.

In Fig. 6, an example of the estimated SoC compared with the expected SoC is shown. The estimated SoC trend is similar to the expected one. In Table VI, the results of the tests in each current range are summarized. As it can be seen from the results, with discharge cycles, including random currents in the range from 3 to 4 A, an RMSE of 1.2% and an MAE of 1% can be obtained. The results are similar if the proposed algorithm is applied to other ranges, confirming the generalization capability of the method. It is worth noting that when the full range from 2 to 4 A is considered, two different approaches were applied and compared to each other. In the first one, a unique SVR model was trained for the whole 2–4 A range, while in the other one, the two models already trained for the current ranges from 2 to 3 A and from 3 to 4 A have been used together, applying an input decision logic to direct the input vector (i.e., current, voltage and temperature values) toward the most suitable machine according to the measured current. This can also be useful to demonstrate the flexibility of the proposed approach. The training over a larger dataset can be a longer and heavier task, as data in Table III has shown. The possibility of training machines with fewer points, can therefore, be interesting, especially for the possibility of evaluating to configure the implemented algorithms differently according to the current interval required by the particular application.

TABLE VII
COMPARISON TO THE STATE-OF-THE-ART

Work	Application field	Same profiles for Training and test	RMSE (%)	MAE (%)
[36]	Automotive	YES	1.18	0.94
[38]	Automotive	Partially	2	--
[22]	Automotive	NO	1.4	1.2
[39]	Automotive	NO	2.47	--
[30]	Automotive	NO	1.51	1.32
[40]	Automotive	NO	1.37	1.12
Proposed	**Miscellaneous**	**NO**	**0.94**	**0.75**

To further validate the proposed technique, the RMSE and the MAE were evaluated in the case of two different and more realistic application fields: some discharge cycles following the behavior of a common battery-powered drill and other driving cycles acquired according to the US06 SFTP. The first has been obtained by measuring the current consumption of a drill during a screwing process in [26]. At the motor startup, a spike in the current profile is experienced. Then, the current falls and rises again as the friction of the screw increases. Finally, a higher current value is due to the conclusive tightening.

The US06 SFTP is defined as vehicle speed over time to obtain a worldwide standardized estimate of vehicle fuel consumption under different road scenarios [48]. In particular, the US06 SFTP includes rapid speed fluctuations, from high-speed (e.g., highways) to low-speed (e.g., city traffic) driving. The cycle represents a 12.8 km route with an average speed of 77.9 km/h, a maximum speed of 129.2 km/h, and a duration of 596 s. The speed set points have been converted into current set points to be applied to the battery by defining the vehicle specifications (i.e., weight, motor efficiency, battery, etc.) affecting the vehicle acceleration, and hence the power consumption. The reference vehicle was a 2012 Tesla Model S 75.

For testing purposes, both the vehicle and drill current profiles have been scaled to the required current range, still preserving the shape.

In the case of the power tool, an RMSE of 0.9% and an MAE of 0.6% were obtained, while in the case of the US06 driving cycle, an RMSE of 0.96% and an MAE of 0.76% were achieved. Considering all the tests, an average RMSE of 0.94% and an average MAE of 0.75% were obtained.

In Table VII, these results are compared with those of other works already published in the literature and previously discussed in Section II. In this regard, it is important to consider that the majority of the works use the same profiles for the training and the test: among these, in [36], an extended set of features has been used as input of an SVM regression

TABLE VIII
COMPARISON OVER THE SAME DATASET

Dataset	Metrics	[36]	Proposed
2A 3A	RMSE (%)	9.95	1.04
	MAE (%)	5.86	0.89
	cost (ms)	21.2	18.8
3A 4A	RMSE (%)	12.4	1.21
	MAE (%)	8.90	1.03
	cost (ms)	18.2	15.2

model with an RBF kernel. Along with current, temperature and voltage, the authors added filtered current and voltage measurements. Two different Butterworth filters were applied for this aim, one with a 0.5 mHz cut-off frequency and the other with 5 mHz. Compared to this case, the proposed method, although allows a more general approach, reaches similar performance. Some works [15], [33] also exploiting the same profiles in training and testing, achieved even considerably lower errors but using more complicated methods and a higher number of input features.

Instead, considering articles that tried to exploit different training and testing sets, the proposed work performs well with a reduction of both RMSE and MAE. These results confirm the feasibility of the method based on SVR training with constant currents, which allows a more general approach to the problem of SoC estimation.

The performance of the proposed approach in terms of accuracy and computational cost was, moreover, compared to the method presented in [36] (i.e., the work featuring the lowest errors in Table VII) considering the same dataset. For comparison, an SVR model was trained with the extended set of input features used by the authors in [36] over a set of constant current profiles, and then the trained model was tested over five different random discharge profiles. During the training phase, the same optimization method used for the proposed algorithm was applied. In this experiment, both the 2–3 A and the 3–4 A datasets were used.

The results of this analysis are reported in Table VIII.

As can be seen, performance in terms of accuracy worsens compared with the proposed method for both datasets. Computational cost was also evaluated, showing lower times for the proposed method. This is also very important when considering the actual implementation of the algorithm in a BMS, which accomplishes the battery monitoring task, avoids the battery operation outside safe conditions, and performs cell balancing. Indeed, a common BMS design strategy involves a master-slave approach [10], [11], [49]. Several slave units perform measurements on multiple cells, sending data to the master processor unit, which performs computations. In this case, the proposed method could then be applied once for each monitored cell. Employing master/slave topologies could allow to parallelize the SoC estimation on multiple slave boards (i.e., the proposed approach could be replicated on each slave board) [49]; moreover, if the BMS has been designed to also include an FPGA, the algorithm could be replicated multiple times in hardware, and multiple cell SoC could be evaluated at the same time on the same board [50]. It is worth considering that if the implementation relies on a microcontroller device, the operating frequency is a limit to the maximum number of cells that the proposed method can evaluate since each cell SoC must be estimated sequentially. The possibility to run an algorithm with a low computational cost like the one proposed is an advantage because it allows the monitoring of several cells at the same time.

V. CONCLUSION

In this article, an application-independent approach for the SoC estimation based on SVR and a postprocessing phase is presented. Typically, AI algorithms for SoC estimation are trained on a set of data that mimics the current profile of the real application. In this work, an application-specific approach exploiting constant current profiles for the SVR training phase has been validated over different test sets. This approach has the advantage of separating the battery use phase from the training phase, which typically requires a lot of time and computing power and can hardly be performed by the BMS designer. Data from a Panasonic NCR18650PF battery cell have been acquired and then fed to the training algorithm. Data covering different ranges of current have been exploited, demonstrating the generalization capability of the approach. For the test set, three different scenarios have been considered: random discharge profiles with no relevance for a specific application, battery-powered drill discharge profiles, and US06 automotive driving cycles. A maximum RMSE of 0.94% and an MAE of 0.75% were observed, which is comparable with the results already reported in the literature obtained by exploiting the same application-specific scenarios for the training and the test. In the future, this approach can be further investigated. The approach will have to be further characterized by considering its operation at different temperatures. Furthermore, the results using different models and types of batteries have to be evaluated. It worthy of remark that the presented approach can also be applied to other ML models different from the SVR to further verify the generality of the approach and if the combination with other algorithms can lead to a further improvement in the SoC estimation. In this context, other input features can also be taken into account as well (e.g., battery impedance spectroscopy [51]). Finally, the approach should be implemented on an embedded device to verify the real-time operation and the possibility to exploit an FPGA device to apply it to parallel cells.

REFERENCES

[1] *Lithium-ion Battery Market Size, Share & Trends Analysis Report by Product (LCO, LFP, NCA, LMO, LTO, NMC), by Application (Consumer Electronics, Energy Storage Systems, Industrial), by Region, and Segment Forecasts, 2022–2030*, Accessed: Feb. 9, 2023. [Online]. Available: https://www.researchandmarkets.com/reports/4396452/lithium-ion-battery-market-size-share-and-trends

[2] G. E. Blomgren, "The development and future of lithium ion batteries," *J. Electrochem. Soc.*, vol. 164, no. 1, pp. A5019–A5025, 2017, doi: 10.1149/2.0251701jes.

[3] Y. Liang et al., "A review of rechargeable batteries for portable electronic devices," *InfoMat*, vol. 1, no. 1, pp. 6–32, Mar. 2019, doi: 10.1002/INF2.12000.

[4] P. H. Camargos, P. H. J. D. Santos, I. R. D. Santos, G. S. Ribeiro, and R. E. Caetano, "Perspectives on Li-ion battery categories for electric vehicle applications: A review of state of the art," *Int. J. Energy Res.*, vol. 46, no. 13, pp. 19258–19268, Oct. 2022, doi: 10.1002/ER.7993.

[5] N. Lebedeva, D. Tarvydas, and I. Tsiropoulos, "Li-ion batteries for mobility and stationary storage applications," European Commission, Joint Research Centre., Brussels, Belgium, Tech. Rep., 2018, doi: 10.2760/87175.

[6] J. N. Hu et al., "State-of-charge estimation for battery management system using optimized support vector machine for regression," *J. Power Sources*, vol. 269, pp. 682–693, Dec. 2014, doi: 10.1016/J.JPOWSOUR.2014.07.016.

[7] M. Kumar, V. K. Yadav, K. Mathuriya, and A. K. Verma, "A brief review on cell balancing for Li-ion battery pack (BMS)," in *Proc. IEEE 10th Power India Int. Conf. (PIICON)*, Nov. 2022, pp. 1–6, doi: 10.1109/PIICON56320.2022.10045109.

[8] N. Ghaeminezhad, Q. Ouyang, X. Hu, G. Xu, and Z. Wang, "Active cell equalization topologies analysis for battery packs: A systematic review," *IEEE Trans. Power Electron.*, vol. 36, no. 8, pp. 9119–9135, Aug. 2021, doi: 10.1109/TPEL.2021.3052163.

[9] G. Plett, *Battery Management Systems, Volume II: Equivalent-Circuit Methods*. Norwood, MA, USA: Artech House, 2015.

[10] L. Schärtel, B. Reick, M. Pfeil, and R. Stetter, "Analysis and synthesis of architectures for automotive battery management systems," *Appl. Sci.*, vol. 12, no. 21, p. 10756, Oct. 2022, doi: 10.3390/app122110756.

[11] *D6.1—Analysis of the State of the Art on BMS*. Accessed: May 19, 2023. [Online]. Available: https://everlasting-project.eu/2017/03/

[12] P. Takyi-Aninakwa, S. Wang, H. Zhang, E. Appiah, E. D. Bobobee, and C. Fernandez, "A strong tracking adaptive fading-extended Kalman filter for the state of charge estimation of lithium-ion batteries," *Int. J. Energy Res.*, vol. 46, no. 12, pp. 16427–16444, Oct. 2022, doi: 10.1002/er.8307.

[13] R. Xiong, J. Cao, Q. Yu, H. He, and F. Sun, "Critical review on the battery state of charge estimation methods for electric vehicles," *IEEE Access*, vol. 6, pp. 1832–1843, 2018, doi: 10.1109/ACCESS.2017.2780258.

[14] A. Manoharan, K. M. Begam, V. R. Aparow, and D. Sooriamoorthy, "Artificial neural networks, gradient boosting and support vector machines for electric vehicle battery state estimation: A review," *J. Energy Storage*, vol. 55, Nov. 2022, Art. no. 105384, doi: 10.1016/J.EST.2022.105384.

[15] Q. Shi, Z. Jiang, Z. Wang, X. Shao, and L. He, "State of charge estimation by joint approach with model-based and data-driven algorithm for lithium-ion battery," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, Aug. 2022, doi: 10.1109/TIM.2022.3199253.

[16] S. J. Moura, N. A. Chaturvedi, and M. Krstić, "Adaptive partial differential equation observer for battery state-of-charge/state-of-health estimation via an electrochemical model," *J. Dyn. Syst., Meas., Control*, vol. 136, no. 1, pp. 1–11, Jan. 2014, doi: 10.1115/1.4024801.

[17] S. Wang, P. Ren, P. Takyi-Aninakwa, S. Jin, and C. Fernandez, "A critical review of improved deep convolutional neural network for multi-timescale state prediction of lithium-ion batteries," *Energies*, vol. 15, no. 14, p. 5053, Jul. 2022, doi: 10.3390/EN15145053.

[18] Y. Liu, Y. He, H. Bian, W. Guo, and X. Zhang, "A review of lithium-ion battery state of charge estimation based on deep learning: Directions for improvement and future trends," *J. Energy Storage*, vol. 52, Aug. 2022, Art. no. 104664, doi: 10.1016/J.EST.2022.104664.

[19] M. Aftowicz, K. Lehniger, and P. Langendoerfer, "Scalable FPGA hardware accelerator for SVM inference," in *Proc. 11th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2022, pp. 1–4, doi: 10.1109/MECO55406.2022.9797110.

[20] D. Anguita, A. Ghio, S. Pischiutta, and S. Ridella, "A hardware-friendly support vector machine for embedded automotive applications," in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 2007, pp. 1360–1364, doi: 10.1109/IJCNN.2007.4371156.

[21] C. Vidal, P. Malysz, P. Kollmeyer, and A. Emadi, "Machine learning applied to electrified vehicle battery state of charge and state of health estimation: State-of-the-art," *IEEE Access*, vol. 8, pp. 52796–52814, 2020, doi: 10.1109/ACCESS.2020.2980961.

[22] M. Stighezza, V. Bianchi, and I. De Munari, "FPGA implementation of an ant colony optimization based SVM algorithm for state of charge estimation in Li-ion batteries," *Energies*, vol. 14, no. 21, p. 7064, Oct. 2021, doi: 10.3390/EN14217064.

[23] S. Boulmrharj et al., "Online battery state-of-charge estimation methods in micro-grid systems," *J. Energy Storage*, vol. 30, Aug. 2020, Art. no. 101518, doi: 10.1016/J.EST.2020.101518.

[24] D. H. C. Lam, Y. S. Lim, L. C. Hau, and J. Wong, "Long short-term memory recurrent neural network for estimating state of charge of energy storage system for grid services," *SSRN Electron. J.*, 2022, doi: 10.2139/SSRN.4120303.

[25] M. Stighezza, V. Bianchi, A. Toscani, and I. De Munari, "A flexible machine learning based framework for state of charge evaluation," in *Proc. IEEE Int. Workshop Metrology Automot. (MetroAutomotive)*, Jul. 2022, pp. 111–115, doi: 10.1109/METROAUTOMOTIVE54295.2022.9855050.

[26] *Duplicating Real Life Load Profiles in the Laboratory*. Accessed: Jan. 27, 2023. [Online]. Available: http://basytec.de/applications/reallife.pdf

[27] *US06 Drive Cycle*. Accessed Jan. 30, 2023. [Online]. Available: https://dieselnet.com/standards/cycles/ftp_us06.php

[28] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *Proc. 6th Int. Adv. Comput. Conf. (IACC)*, Feb. 2016, pp. 78–83, doi: 10.1109/IACC.2016.25.

[29] MathWorks. *Select Data for Regression or Open Saved App Session*. Accessed Sep. 16, 2022. [Online]. Available: https://it.mathworks.com/help/stats/select-data-and-validation-for-regression-problem.html

[30] Z. Cui, L. Kang, L. Li, L. Wang, and K. Wang, "A combined state-of-charge estimation method for lithium-ion battery using an improved BGRU network and UKF," *Energy*, vol. 259, Nov. 2022, Art. no. 124933, doi: 10.1016/J.ENERGY.2022.124933.

[31] B. Yang, Y. Wang, and Y. Zhan, "Lithium battery state-of-charge estimation based on a Bayesian optimization bidirectional long short-term memory neural network," *Energies*, vol. 15, no. 13, p. 4670, Jun. 2022, doi: 10.3390/EN15134670.

[32] J. Tian, R. Xiong, J. Lu, C. Chen, and W. Shen, "Battery state-of-charge estimation amid dynamic usage with physics-informed deep learning," *Energy Storage Mater.*, vol. 50, pp. 718–729, Sep. 2022, doi: 10.1016/J.ENSM.2022.06.007.

[33] Z. Ni and Y. Yang, "A combined data-model method for state-of-charge estimation of lithium-ion batteries," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022, doi: 10.1109/TIM.2021.3137550.

[34] J. Li, M. Ye, W. Meng, X. Xu, and S. Jiao, "A novel state of charge approach of lithium ion battery using least squares support vector machine," *IEEE Access*, vol. 8, pp. 195398–195410, 2020, doi: 10.1109/access.2020.3033451.

[35] E. Vasta et al., "Models for battery health assessment: A comparative evaluation," *Energies*, vol. 16, no. 2, p. 632, Jan. 2023, doi: 10.3390/en16020632.

[36] S. Jumah, A. Elezab, O. Zayed, R. Ahmed, M. Narimani, and A. Emadi, "State of charge estimation for EV batteries using support vector regression," in *Proc. IEEE Transp. Electrific. Conf. Expo (ITEC)*, Jun. 2022, pp. 964–969, doi: 10.1109/ITEC53557.2022.9813811.

[37] W. Zhang and W. Wang, "Lithium-ion battery SoC estimation based on online support vector regression," in *Proc. 33rd Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, May 2018, pp. 564–568, doi: 10.1109/YAC.2018.8406438.

[38] B. Fu, W. Wang, Y. Li, and Q. Peng, "An improved neural network model for battery smarter state-of-charge estimation of energy-transportation system," *Green Energy Intell. Transp.*, vol. 2, no. 2, Apr. 2023, Art. no. 100067, doi: 10.1016/j.geits.2023.100067.

[39] Q. Wang, M. Ye, M. Wei, G. Lian, and Y. Li, "Deep convolutional neural network based closed-loop SOC estimation for lithium-ion batteries in hierarchical scenarios," *Energy*, vol. 263, Jan. 2023, Art. no. 125718, doi: 10.1016/j.energy.2022.125718.

[40] Y. Li, K. Li, X. Liu, Y. Wang, and L. Zhang, "Lithium-ion battery capacity estimation—A pruned convolutional neural network approach assisted with transfer learning," *Appl. Energy*, vol. 285, Mar. 2021, Art. no. 116410, doi: 10.1016/j.apenergy.2020.116410.

[41] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[42] S. M. Ross, *Regression, Introduction to Probability and Statistics for Engineers and Scientists*. Amsterdam, The Netherlands: Elsevier, 2014, pp. 357–444, doi: 10.1016/B978-0-12-394811-3.50009-5.

[43] H. Huang, X. Wei, and Y. Zhou, "An overview on twin support vector regression," *Neurocomputing*, vol. 490, pp. 80–92, Jun. 2022, doi: 10.1016/j.neucom.2021.10.125.

[44] Panasonic. *Lithium Ion Panasonic NCR18650PF Datasheet*. Accessed: Jan. 30, 2023. [Online]. Available: https://na.industrial.panasonic.com/products/batteries/rechargeable-batteries/lineup/lithium-ion/series/90729/model/90730

[45] A. Fly and R. Chen, "Rate dependency of incremental capacity analysis (dQ/dV) as a diagnostic tool for lithium-ion batteries," *J. Energy Storage*, vol. 29, Jun. 2020, Art. no. 101329, doi: 10.1016/j.est.2020.101329.

[46] S. P. Nangrani and I. S. Nangrani, "Artificial intelligence based state of charge estimation of electric vehicle battery," in *Smart Technologies for Energy, Environment and Sustainable Development*. Singapore: Springer, 2022, pp. 679–686, doi: 10.1007/978-981-16-6879-1_65.

[47] L. Zhou, L. He, Y. Zheng, X. Lai, M. Ouyang, and L. Lu, "Massive battery pack data compression and reconstruction using a frequency division model in battery management systems," *J. Energy Storage*, vol. 28, Apr. 2020, Art. no. 101252, doi: 10.1016/j.est.2020.101252.

[48] *EPA US06 or Supplemental Federal Test Procedures (SFTP)*. Accessed: May 19, 2023. [Online]. Available: https://www.epa.gov/emission-standards-reference-guide/epa-us06-or-supplemental-federal-test-procedures-sftp

[49] A. A. Chellal, J. Gonçalves, J. Lima, V. Pinto, and H. Megnafi, "Design of an embedded energy management system for Li–Po batteries based on a DCC-EKF approach for use in mobile robots," *Machines*, vol. 9, no. 12, p. 313, Nov. 2021, doi: 10.3390/machines9120313.

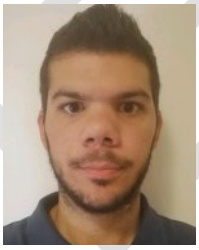
[50] A. Saday, İ. A. Ozkan, and I. Saritas, "FPGA-based battery management system for real-time monitoring and instantaneous SOC prediction," *Int. J. Appl. Math. Electron. Comput.*, vol. 11, no. 1, pp. 55–61, Mar. 2023, doi: 10.18100/ijamec.1233451.

[51] M. Stighezza, R. Ferrero, and V. Bianchi, "Machine learning and impedance spectroscopy for battery state of charge evaluation," in *Proc. IEEE Int. Workshop Metrology Automot. (MetroAutomotive)*, Jun. 2023, pp. 24–29.

Valentina Bianchi (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic engineering and the Ph.D. degree from the Department of Information Engineering, University of Parma, Parma, Italy, in 2003, 2006, and 2010, respectively.

She is currently a Research Associate with the Department of Information Engineering, University of Parma. She has participated in several national and international projects. She has authored or coauthored over 50 papers in international journals or proceedings of conferences. Her current research interests include the design and validation of sensors for human activity recognition, sensors for electrochemical applications, and digital systems implemented on field programmable gate arrays (FPGAs), with a special focus on the design of hardware for machine learning algorithms and arithmetic circuits.

Dr. Bianchi is an Associate Editor of the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

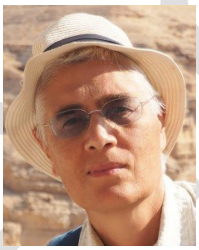


Mattia Stighezza (Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic engineering from the University of Parma, Parma, Italy, in 2017 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the Sensors and Embedded Systems Design Laboratory.

His current research interests include the evaluation of battery state of charge through machine learning algorithms and battery testing and analysis.

Andrea Toscani received the M.S. degree (cum laude) in electronic engineering and the Ph.D. degree in information technology from the University of Parma, Parma, Italy, in 2004 and 2008, respectively.

Since 2004, he has been with the Department of Information Engineering (now Department of Engineering and Architecture), University of Parma. He is currently a Research Fellow. He is the author of three patents. His research interests include power electronics, high-performance electric drives, diagnostic techniques for industrial electric systems, and power converter for audio application.



Giovanni Chiorboli received the M.S. degree (cum laude) in electronic engineering from the University of Bologna, Bologna, Italy, in 1987.

He is currently an Associate Professor of electronic measurements with the University of Parma, Parma, Italy. His research interests include electronic instruments and sensors, analog-to-digital and digital-to-analog modeling and testing, and electrical characterization of semiconductor devices.

Ilaria De Munari (Senior Member, IEEE) received the M.Sc. degree in electronic engineering and the Ph.D. degree in information technologies from the University of Parma, Parma, Italy, in 1991 and 1995, respectively.

In 1997, she joined the Department of Information Engineering (now Department of Engineering and Architecture), University of Parma, as a Research Assistant, where she has been an Associate Professor of electronics since 2004. She has authored or coauthored more than 100 papers in technical journals or proceedings of international conferences. Her past research interests include the reliability of electronic devices and the design of electronic systems for active assisted living. In this framework, she was involved in several European projects. Her current research interests include the design of digital systems based on microcontrollers and field-programmable gate arrays (FPGAs). In particular, she is dealing with the design of sensors for human activity recognition, electrochemical applications, and the evaluation of the battery state of charge through machine learning algorithms; moreover, she dealt with the FPGA implementation of arithmetic circuits and error correction code techniques.

Dr. Munari is an Associate Editor of the journal IEEE ACCESS.