Contents lists available at ScienceDirect

Measurement

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

Uncertainty characterization of a CNN method for Lithium-Ion Batteries state of charge estimation using EIS data

Emanuele Buchicchio [*], Alessio De Angelis, Francesco Santoni, Paolo Carbone

Department of Engineering, University of Perugia, Via G. Duranti, 93, Perugia, 06125, PG, Italy

ARTICLE INFO

Dataset link: https://data.mendeley.com/datasets/zdsgxwksn5, https://data.mendeley.com/datasets/mbv3bx847g

Keywords:
Battery state of charge estimation
Electrochemical impedance spectroscopy
Battery impedance measurement
Uncertainty quantification
Convolutional neural network
Monte Carlo simulation

ABSTRACT

Estimating the state of charge of batteries is a critical task for every battery-powered device. In this work, we propose a machine learning approach based on electrochemical impedance spectroscopy and 2D convolutional neural networks. A case study based on Samsung ICR18650-26J lithium ion batteries is also presented and discussed in detail. An application-specific data augmentation technique is developed and applied. The proposed system achieves a classification accuracy of 93% on a test dataset of new measurements from the same battery and 88% accuracy on a different battery without prior calibration. The uncertainty of the state of charge classification provided by the proposed method is evaluated using Monte Carlo Simulations and Monte Carlo dropout methods.

1. Introduction

Many battery powered devices, such as laptops, smartphones, cameras, tablets, cordless shavers, lawnmowers, drones, and even electric cars are now part of our daily life. In most battery-operated systems, knowing the remaining charge within the battery is essential for end-users. Additionally, knowledge of the remaining battery capacity is fundamental for its management because states of extremely high or extremely low state of charge (SOC) can irreversibly damage the battery [1]. The relationship between the battery's observable signals and the estimated SOC is highly non-linear, varying with temperature and discharge/charge currents [2].

There is no practical method for SOC direct measurement outside laboratory settings [3]. Therefore, many research efforts have been conducted over the past decades to develop a secure, practical, and reliable method for SOC estimation. In this field, several studies have been based on electrochemical impedance spectroscopy (EIS) measurement data. This is due to the fact that EIS is an experimental technique that can provide a detailed characterization of the state of electrochemical cells [4]. Regarding the application of EIS to batteries, in [5], a method for determining and optimizing the parameters of an EIS measurement performed on batteries. Furthermore, [6] analyzes the problem of fitting a battery equivalent circuit model for state identification purposes. EIS data are employed for the estimation of SOC in [7], where a machine learning method is implemented.

From a methodological point of view, methods for SOC estimation can be divided into two main classes: data-driven and model-based. The data-driven SOC estimation approaches require limited knowledge about internal battery characteristics. In contrast, model-driven approaches require an in-depth understanding of the battery's internal chemical ed electrical characteristics. Model-based methods also require the assumption that the battery model is accurately established. This condition is hard to realize in real applications due to the effect of measurement noise, model parameter drifts with aging, and temperature [3]. Combining some a-priori information, embedded in physical electrochemical models or electrical equivalent circuit models, with experimental data, model-based approach can result in reliable and accurate predictions [8]. However, they require extensive domain knowledge and relatively long development times. Common model-based approaches in recent publications include the usage of Sliding Mode Observer, Luenberger Observer, Kalman filters, Electrochemical Model, Equivalent Circuit Model, and Electrochemical Impedance Model [9].

Data-driven approaches, such as those based on machine learning (ML), are becoming more popular for estimating the SOC and battery state-of-health (SOH) due to the greater availability of battery data and improved computing power capabilities. SOC estimators based on neural networks (NN) have been studied extensively in the literature [2,9–11]. When using the NN model, a large amount of known input

[*] Corresponding author.
E-mail addresses: emanuele.buchicchio@ieee.org (E. Buchicchio), alessio.deangelis@unipg.it (A. De Angelis), francesco.santoni@unipg.it (F. Santoni), paolo.carbone@unipg.it (P. Carbone).

https://doi.org/10.1016/j.measurement.2023.113341
Received 14 April 2023; Received in revised form 11 July 2023; Accepted 16 July 2023
Available online 22 July 2023

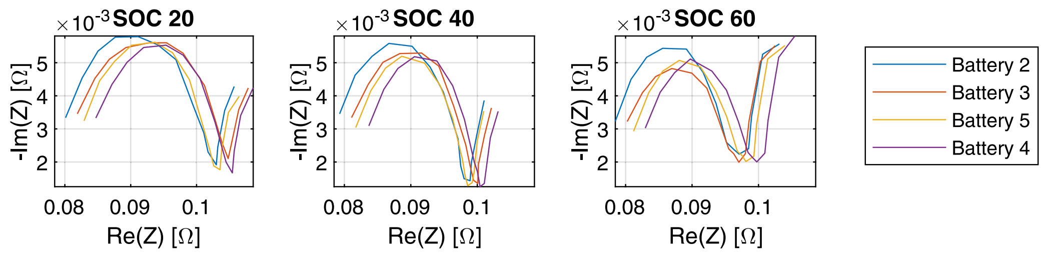


Fig. 1. EIS curve comparison. The shape of the curve is specific for each SOC internal state.

data and expected output data obtained from the battery charging and discharging experiments is required to train the network and extract the fitting relationship without an a-priori model of the battery.

Traditional ML techniques contain no more than one or two layers of non-linear and linear transformations. With the advent of faster computational power and an abundance of available real-world data, more complex architectures were investigated, which, in many cases, allowed researchers to make striking improvements in many applications. The ResNet deep learning architecture used in this work, called a deep residual network, won the 2015 ImageNet challenge with an error rate of 3.57% which even surpasses human-level accuracy valued at 5.1% [12] for the same task.

The convolutional neural networks (CNNs) are well-suited to image and signal processing tasks due to their ability to learn spatial hierarchies of features, which are useful for identifying patterns in signals. By using 1D convolutional layers, CNN can be applied to classify time series data such as audio, speech, and other signals [13]. 2D CNNs are commonly used for image classification tasks, but can also be used for other types of data that can be represented as images, as shown in [14,15]. In this regard, in [16], a 2D CNN is applied to a 2D representation of audio signals containing noise to recognize voiceprints from motor bearing failures. In some applications, such as [17], 2D CNN models achieve better performance than 1D CNN models.

In this work, we propose a battery SOC estimation method based on a visual representation of the EIS data. This visual representation is fed to a CNN that classifies the SOC. The proposed method is flexible because it can be easily reconfigured for other classification tasks and for using different visual representations of the data. The typical visual representation of EIS is based on the Nyquist plot. This plot shows the real versus the imaginary part of the impedance on the two axes. This work also considers the Bode plot, that is the magnitude and phase of frequency responses. We validated the proposed SOC estimation method on a dataset of repeated EIS measurements performed on cylindrical lithium-ion rechargeable batteries. We also evaluate the uncertainty associated with the measurement of the SOC classification results. The main contributions of this work, which extends the preliminary results in [18], can be summarized as follows:

1. a novel SOC estimation method is proposed, based on a CNN operating on a visual representation of EIS data. To the extent of the authors' knowledge, this is the first time that a 2D CNN is applied to a visual representation of EIS data for SOC estimation;
2. the proposed method is validated on a real-world EIS dataset resulting from measurement results obtained using cylindrical lithium-ion cells;
3. an EIS-specific data augmentation technique is developed and validated;
4. the uncertainty associated with the SOC classification provided by the proposed method is evaluated by means of Monte Carlo based techniques. This aspect is often overlooked in the literature on machine learning methods for SOC estimation.

2. Materials and methods

2.1. SoC estimation using electrochemical impedance spectroscopy and convolutional neural networks

EIS is a powerful tool for monitoring the SOC and SOH of rechargeable batteries [19]. It provides an estimate of the equivalent battery impedance $Z(f)$ by measuring, for example, the voltage variations at the battery contacts after a varying input current is applied. Visual representation of the results is often obtained by plotting the imaginary negative part of $Z(f)$ versus its real part.

Although it was easy to observe that different SOC values generate different shapes in the visual representations of the EIS, the relationships between SOC and $Z(f)$ are not obvious. This is shown in Fig. 1 which is based on measurements of the $Z(f)$ of four rechargeable batteries, characterized by three different SOCs.

Finding a closed-form expression of this relationship appeared to be an infeasible task. However, we could approximate this unknown function by extracting information from the experimental results. We can easily observe differences in the visual representation of the experimental data from different SOC conditions. EIS curves measured at the same SOC level showed similar shape patterns but exhibited translations in the complex plane. The magnitude of the translation varied for different batteries and even for the same battery across different measurements. Using an ML-based approach to relate SOC to EIS-derived data required the assumption that the EIS shape patterns depend on SOC and are invariant given the same battery type.

CNNs provide three primary advantages for image processing [20] over the traditional feed-forward neural network with fully connected layers. First, they have sparse connections instead of fully connected topologies, which leads to reduced parameters and allows processing high-dimensional data. Second, weight sharing across the entire image reduces memory requirements and causes translational equivariance property. Third, the pooling layer brings invariance to the local translation property. The invariance and equivariance properties make CNN an ideal candidate for performing classification tasks on EIS curves such as those graphed in Fig. 1. Given that deep CNNs are the defacto industry standard solution for image classification tasks, with many well-known architectures and pre-trained models available, we developed a CNN-based SOC estimator for visual representations of EIS data.

2.2. Electrochemical impedance spectroscopy data acquisition

We used a Keysight U2351 A data acquisition board to provide the excitation signal by means of a 16-bit digital-to-analog converter (DAC). Current and voltage signals were acquired utilizing two 16-bit analog-to-digital converter (ADC) channels. The excitation signal was a random-phase multisine, i.e., the sum of harmonically-related sinusoids [21] that allows to simultaneously excite the battery at a wide range of frequencies. In this case, the excited frequencies were 0.05, 0.1, 0.2, 0.4, 1, 2, 4, 10, 20, 40, 100, 200, 400, and 1000 Hz in all measurements. The excitation signal generated by the DAC was converted to a current signal by a voltage-to-current converter custom

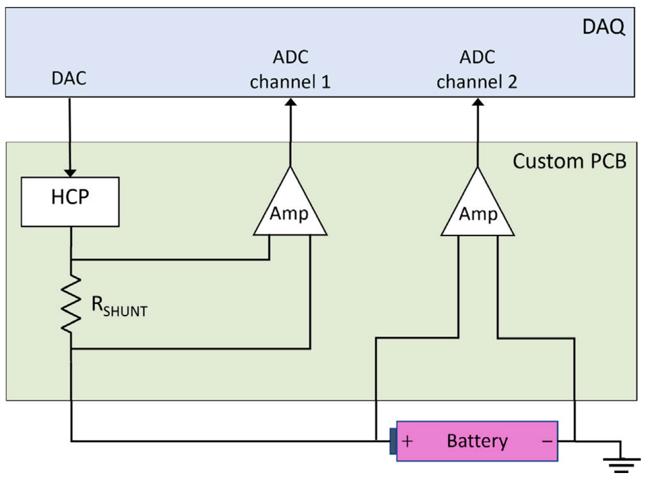


Fig. 2. Architecture of the EIS data acquisition system.
Source: Picture from [24].

circuit [22]. The current signal was measured across a known shunt resistor by means of an Instrumentation Amplifier (INA). The voltage signal across the battery was also measured by a second INA connected to the second ADC channel. The current and voltage signals acquired by the ADC channels were transferred to a PC to compute the discrete Fourier Transform (DFT) and to calculate the complex impedance value at each excited frequency. The current amplitude at each excited frequency was set to 50 mA, which resulted in a measurement uncertainty of approximately 0.1 mΩ, as characterized in [23]. A block diagram illustrating the architecture of the EIS data acquisition is shown in Fig. 2.

2.3. SOC estimation as a machine learning classification task

In machine learning, *classification* is the task of predicting the class to which an object belongs. Each object is uniquely represented by a set of values $x_1, x_2, \dots, x_l$, known as *features* in ML jargon. The l feature values, stacked together, in the common ML notation are labeled as the *feature vector* $x \in \mathbb{R}^l$. The goal is to design a classifier $f(x)$, so that given a value in a feature vector x we will be able to predict the class to which the object belongs. To formulate the task in mathematical terms, each class is represented by the class label variable y and $\hat{y}$ denotes the predicted class given the value of x and the set of possible predicted classes in the so-called *vocabulary*:

$$\hat{y} = \Phi(f(x)), \tag{1}$$

where $\Phi(\cdot)$ is a nonlinear function that maps any possible $f(x)$ values to one on the class labels. The set of possible SOC class *vocabulary* V considered in this work was 100%, 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, and 10%.

The raw features vector was made of complex impedance values measured at frequencies $[0.05, 0.1, 0.2, 0.4, 1, 2, 4, 10, 20, 40, 100, 200, 400, 1000]$ Hz for a specific SOC. A two-dimensional visual representation of the impedance curve in the complex plane was used to build a 2D feature vector for each measured EIS (Fig. 3). Any possible visual representation of the impedance curve could be used to build the feature vector x. In this work, we used the Bode diagram and Nyquist diagram representations.

2.4. Training and test dataset

We compiled and published several open-access datasets obtained through EIS, by measuring brand-new Samsung ICR18650-26J lithium-Ion batteries at different SOCs. In this work we focus on a dataset of EIS measurement that includes four batteries [24]. For each battery, six measurement repetitions were provided at ten different SOCs. The data acquisition procedure was described in 2.2. A detailed description of the measurement system is available in [23], and an excerpt of the employed dataset is shown in Fig. 4.

Two sets of EIS measurements had been excluded from training and validation as a test dataset for the final system performance assessment. These two test cases were representative of two different scenarios: (1) a new set of EIS data on a battery that was originally included in model training (*new measurement* test case); (2) a set of EIS data originating from a battery never seen by the model during training and validation (*new battery* test case).

We randomly chose one of the four batteries for the *new battery* test case. All measurements performed on the battery with BATTERY_ID 6 were assigned to the test dataset and never fed to the model training pipeline. For the *new measurement* test case, we randomly chose a set of EIS from one measurement cycle. The 10 EIS spectra obtained from measure 05_8 (measure number 8 on battery 5) were assigned to the test dataset and never fed to the model training pipeline.

2.5. Training and validation workflow

We have developed a fully automated pipeline for training deep learning-based classifiers using the open-source libraries PyTorch and FastAI [26]. The workflow consisted of three steps (Fig. 5):

1. Step 1: data acquisition;
2. Step 2: data selection;
3. Step 3: model training.

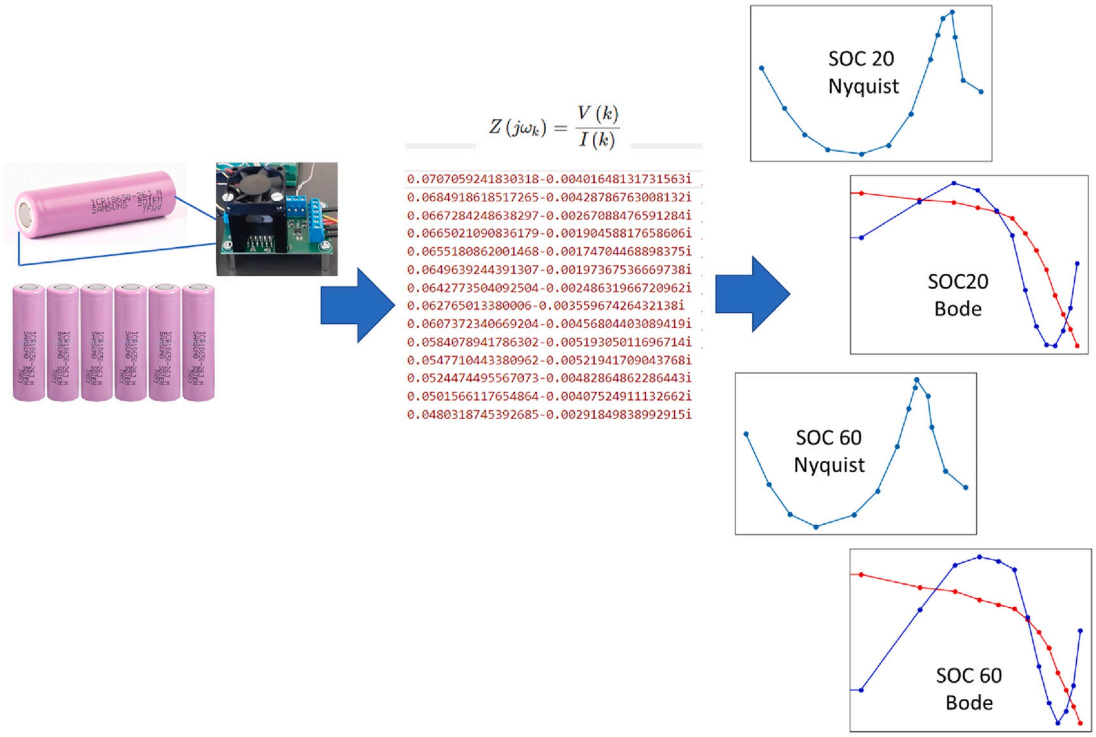
The *Data acquisition* step produced a CSV file containing EIS spectra at each SOC. In *Data selection*, the available data were split into training ad test datasets. The test dataset was only used for testing and not for training. The training dataset was further split to hold out 20% data for validation. We implemented different training strategies for the *model training step*. In this work, we performed two different experiments adopting the *validation hold* strategy for the first and the *leave k-out cross-validation* strategy for the second. In the *validation hold* experiment, we trained a single model using 80% of the available data, while 20% of the data was set aside and used exclusively for validation. The model's performance was then verified using the test dataset, which contains completely new measurements. In the second experiment, we implemented the *leave k-out* strategy as *leave one battery out* to investigate the model performance dependency from a particular choice of training dataset and the effects of specific battery inclusion in the training process. We ran three training sessions, excluding data related to one of the batteries each time. This experiment produced three trained models that can be used alone or combined in an *ensemble model* for SOC inference.

We published all software developed for this work in the open-source project repository, including several Jupyter notebooks with scripts to reproduce the experiment described in this work. Refer to Appendix A for more details.

2.6. Efficient deep CNN training for SoC estimation

Many pre-trained deep learning models have proven adequate for image/video classification tasks. We chose the ResNet18 CNN because the residual network architecture achieves good results in image classification tasks and is relatively fast to train [12]. Transfer learning from general-purpose image classification pre-trained models allows fast fine-tuning training of the deep CNN model.

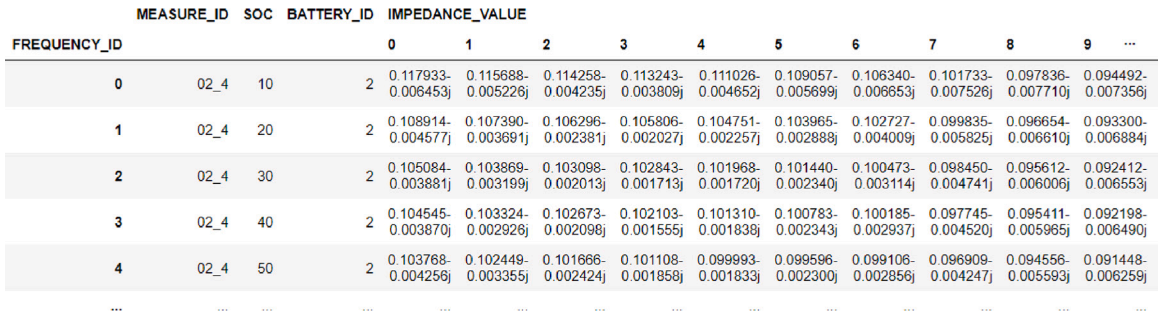
The key idea behind ResNets is the introduction of residual blocks (Fig. 6) that contain an identity shortcut connection, which skips one or more layers. The ResNet paper [12], empirically demonstrated that the addition of skip connections improve the accuracy and make the optimization of these residual networks much easier.



$$Z(j\omega_k) = \frac{V(k)}{I(k)}$$

```
0.0707059241830318-0.00401648131731563i
0.0684918618517265-0.00428786763008132i
0.0667284248638297-0.00267088476591284i
0.0665021090836179-0.00190458817658606i
0.0655180862001468-0.00174704468898375i
0.0649639244391307-0.00197367536669738i
0.0642773504092504-0.00248631966720962i
0.062765013380006-0.00355967426432138i
0.0607372340669204-0.00456804403089419i
0.0584078941786302-0.00519305011696714i
0.0547710443380962-0.00521941709043768i
0.0524474495567073-0.00482864862286443i
0.0501566117654864-0.00407524911132662i
0.0480318745392685-0.00291849838992915i
```

Fig. 3. EIS of the batteries were acquired at different SOC using the system presented in [23]. The complex impedance values were then encoded in 2D visual representation as Bode and Nyquist diagram to feed the convolutional neural network.



	MEASURE_ID	SOC	BATTERY_ID	IMPEDANCE_VALUE										
FREQUENCY_ID				0	1	2	3	4	5	6	7	8	9	...
0	02_4	10	2	0.117933-0.006453j	0.115688-0.005226j	0.114258-0.004235j	0.113243-0.003809j	0.111026-0.004652j	0.109057-0.005699j	0.106340-0.006653j	0.101733-0.007526j	0.097836-0.007710j	0.094492-0.007356j	
1	02_4	20	2	0.108914-0.004577j	0.107390-0.003691j	0.106296-0.002381j	0.105806-0.002027j	0.104751-0.002257j	0.103965-0.002888j	0.102727-0.004009j	0.099835-0.005825j	0.096654-0.006610j	0.093300-0.006884j	
2	02_4	30	2	0.105084-0.003881j	0.103869-0.003199j	0.103098-0.002013j	0.102843-0.001713j	0.101968-0.001720j	0.101440-0.002340j	0.100473-0.003114j	0.098450-0.004741j	0.095612-0.006006j	0.092412-0.006553j	
3	02_4	40	2	0.104545-0.003870j	0.103324-0.002926j	0.102673-0.002098j	0.102103-0.001555j	0.101310-0.001838j	0.100783-0.002343j	0.100185-0.002937j	0.097745-0.004520j	0.095411-0.005965j	0.092198-0.006490j	
4	02_4	50	2	0.103768-0.004256j	0.102449-0.003355j	0.101666-0.002424j	0.101108-0.001858j	0.099993-0.001833j	0.099596-0.002300j	0.099106-0.002856j	0.096909-0.004247j	0.094556-0.005593j	0.091448-0.006259j	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Fig. 4. Excerpt of the EIS measurement dataset [24], where $MEASURE\_ID$ is the unique identification code of each EIS measurement, SOC is the state of charge of the battery, $BATTERY\_ID$ is the unique identification code of the battery, $FREQUENCY\_ID$ is the key to the frequency value array (labeled in increasing order), and $IMPEDANCE\_VALUE$ is the measured complex impedance value in Ohm expressed in Cartesian form.

The optimal learning rate for training has been estimated with the Cyclical Learning Rates method [27] to avoid time-consuming multiple runs to perform hyperparameter sweeps. The algorithm implementation in the FastAI library suggested a learning rate in the range 10^{-2}–10^{-3}. We fine-tune the model with a sequence of freeze, fit-one-cycle, unfreeze, and fit-one-cycle operations using the *discriminative learning rate fine tuning* method developed in [28] and implemented in FastAI library [26].

After some tests, we set the training script to run for 50 epochs, in many cases, fewer epochs would suffice to reach accuracy and validation loss plateau as shown in Fig. 7.

The time required for model training and validation heavily depends on the hardware configuration. Using the AWS Studio Lab free compute service as recommended in Appendix A, we measured the following average wall clock time intervals:

1. Image dataset generation takes about 60 s for each SOC class with 2000 images per class.
2. Resnet18 CNN model fine tuning (50 epochs) 240–300 s
3. Inference on a batch of 2000 test images: 1 s

The typical compute instance configuration, from AWS Studio Lab free service, was Single Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50 GHz with 2 cores. 16 GB of RAM. Equipped with a single NVIDIA T4 GPU 16 GB RAM. The hardware configuration is not always the same due to the best-effort resource allocation policy of the free service.

2.7. Data augmentation

Measuring EIS spectra at different SOC levels is time-consuming and requires performing battery charge and discharge cycles, including relaxation times. Every measurement series with the data acquisition system described in Section 2.2 lasts at least one day. Therefore, the number of measurements available for model training was limited, and data augmentation techniques had to be adopted. Standard data augmentation methods for image datasets apply a variety of transformations such as rotation, perspective distortion, translation, zooming, cropping, and alterations in brightness, contrast, and color hue. In the case of the EIS curve images, instead, these standard methods lead to unrealistic curve distortions and information loss. Thus, it was necessary to generate synthetic EIS measurement results. The generation

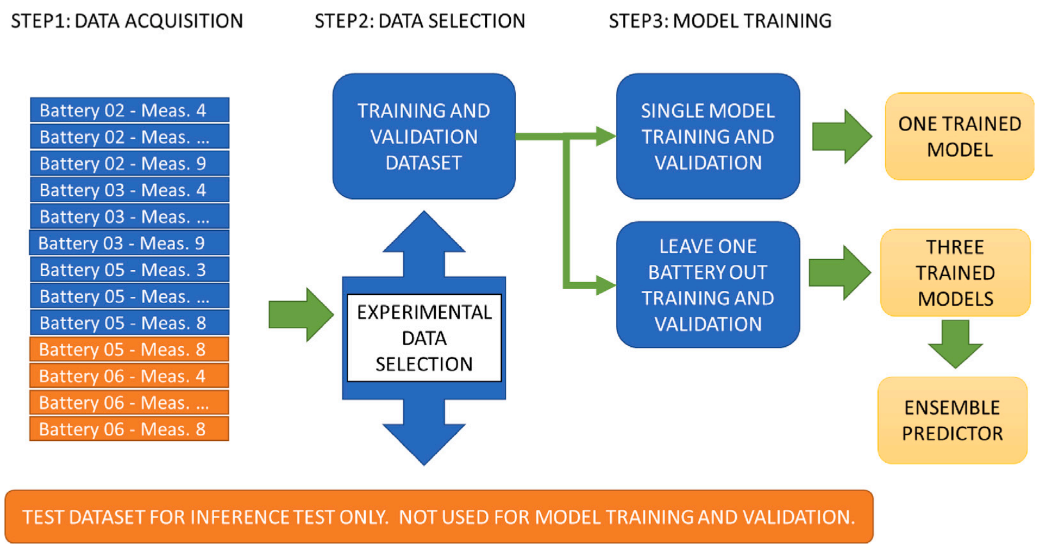


Fig. 5. SOC estimator training and validation workflow overview. Two different sets of EIS measurements have been kept out from training and validation as a test dataset for the final system performance assessment. Some measurements from battery 05 for the *new measurement* test case and all the measurements from battery 06 for the *new battery test case*. The battery and measurement identification code is referred to the dataset [25].

of realistic simulated data was made possible by characterizing the uncertainty of the impedance measurement system.

For model training, we developed an *ad-hoc* data augmentation technique based on simulating additional measurements by adding white Gaussian noise within the uncertainty band estimated in [23], as shown in Fig. 8. The training pipeline generated K different visual representations for each available EIS spectrum, with K equal to the augmentation factor parameter set in the experimental setup. The results shown in this article were obtained with $K = 10$.

2.8. Model interpretation with class activation map analysis

In order to verify that the SOC classification was based on the visual representation of the EIS spectra and not on some other random areas of image fed to the neural network, we used Class Activation Map (CAM) [32] and Gradient-weighted Class Activation Map (Grad-CAM) techniques [33]. The (GRAD) CAM are popular techniques for CNN image classification interpretation that provide a visual representation of the regions of an image that are most important for a given classification task [34].

CAMs work generating a heatmap by computing a weighted sum of the feature maps of the last convolutional layer, where the weights are the weights of the fully-connected layer.

Grad-CAM works by computing the gradient of the output of the model with respect to the feature maps of the last convolutional layer, then performing a global average pooling to obtain the class scores. The heatmap is generated by computing the weighted sum of the feature maps of the last convolutional layer, where the weights are the gradients of the output of the model with respect to the feature maps. This method is applicable to any CNN architecture, and does not require the layer to be global average pooled. Instead, it can be applied to any convolutional feature map, which increases its flexibility with respect to CAM.

CAM highlights areas of an image that are most important for the particular prediction according to the weights of the fully-connected layer, while Grad-CAM highlights areas of an image that have the highest gradients with respect to the output of the model [33]. In practice, this means that while CAM highlights areas of the image that are most important according to the model's learned features, Grad-CAM highlights areas of the image that influence the model's output most.

An example of Grad-CAM and CAM computed from the trained model and the visual representation of an EIS spectrum from the dataset described in Section 2.4 is shown in Fig. 9.

3. Results

3.1. Model validation

The SOC estimator model trained in the *validation hold* single model experiment achieves up to 100% accuracy score on the validation dataset. The accuracy depends on the amount of measurement noise used for data augmentation. Using a realistic amount of additive white Gaussian noise according to the uncertainty band of the impedance measurement system estimated in [23], the model trained using data from the second dataset achieves an overall 94% (88% using Bode diagram representation) accuracy. The three models trained in the *leave one battery out* experiment achieved accuracy scores between 89% and 93% (90% and 92% using Bode diagram). On the validation dataset the model achieves 100% classification accuracy for most SOC classes.

To verify that the SOC estimation produced is based on a reasonable set of image features, we implemented the CAM and GRAD-CAM model interpretation methods described in Section 2.8. The resulting heat maps, computed on validation data, as shown in Fig. 9, indicate that the main contributions to the SOC classification came from the valley and peak areas of the curves, whose shape changes with SOC. This result confirms that the model predictions are based on relevant image features.

3.2. Test in a SOC measurement system

Observing the visual representation of the EIS data resulting from the data pre-processing (Fig. 3), we could see that are shape variations across different SOC for the same battery but also for the same SOC across different batteries (Fig. 10). Therefore, to evaluate the SOC estimator performance on realistic conditions, we arranged two different test cases, representative of several use cases of the system:

1. New battery test: SOC estimation of a battery similar (same model) to the batteries used for model training but never connected to the system before.
2. New measure test: SOC estimation of one of the batteries used for training a few days after the model training.

The models trained during both *leave one battery out* and *validation hold* experiments were evaluated on the two test cases. The overall multi-class classification accuracy was computed using functions im-

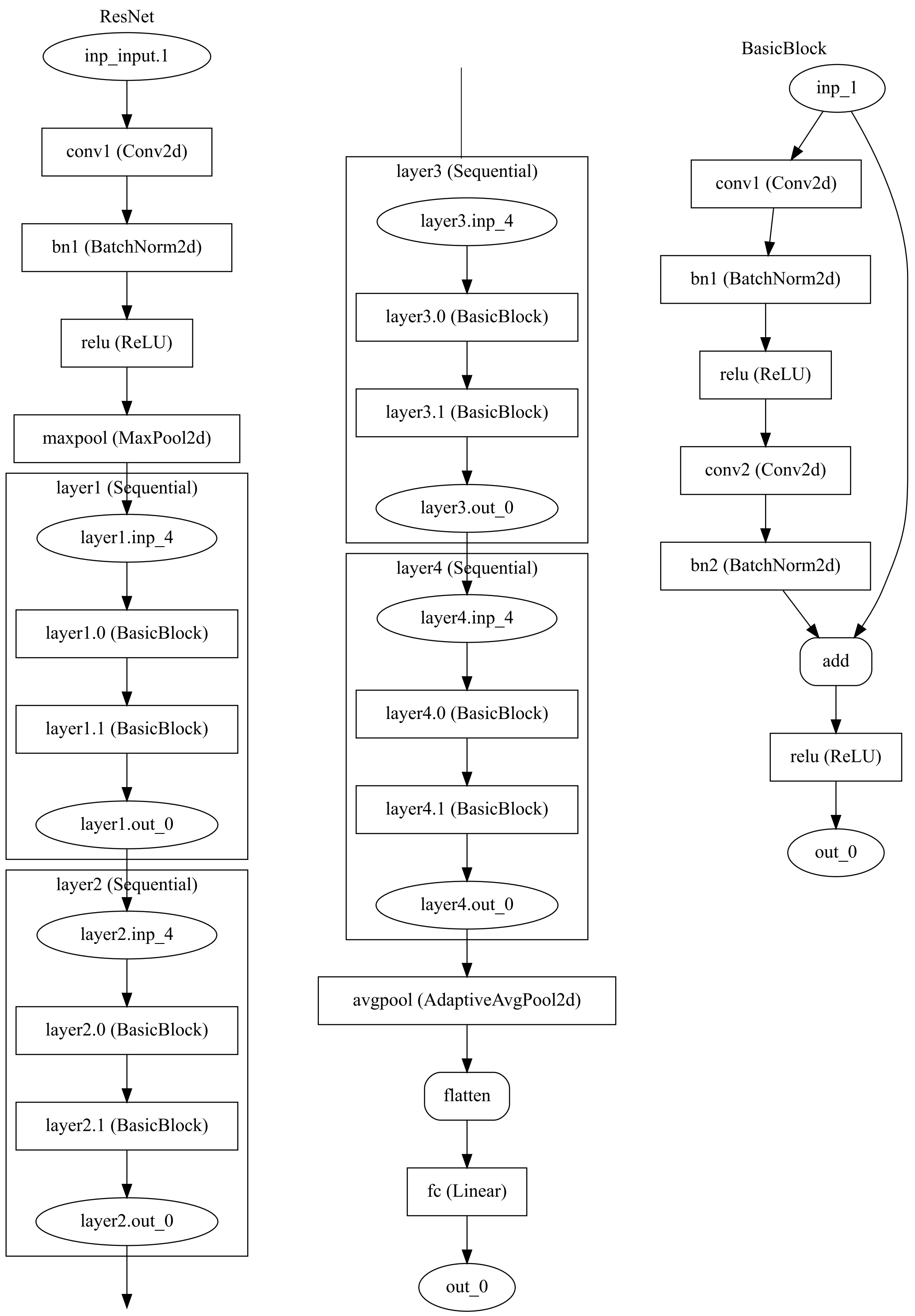


Fig. 6. A ResNet high level network architecture overview (on the left) and basic ResNet block detail (on the right). Images generated from the model using Graphviz (https://graphviz.org/).

plemented by *SciKit Learn* [35] open-source library. The model trained in *validation hold* experiment achieves 0.93 accuracy on *new measure* test and 0.88 accuracy on *new battery*. The performance of the classification model on a new battery is significantly lower with respect to batteries used for the training.

The accuracy dropped on a new battery due to the variability of EIS spectra across different batteries. The impact of the variability across the batteries on the SOC classification accuracy was confirmed by results of the *leave one battery out* cross-validation: the three models trained using the Bode diagram visual representation scored an accuracy of 0.81 (leave out battery 02), 0.80 (leave out battery 03) and 0.52 (leave out battery 05). The resulting ensemble model achieves 0.81 accuracy on test data. We got similar results also with the three models trained using the Nyquist EIS representation. The models' accuracy was 0.56 (leave out battery 02), 0.70 (leave out battery 03), and 0.67 (leave out battery 05). The ensemble accuracy is 0.77.

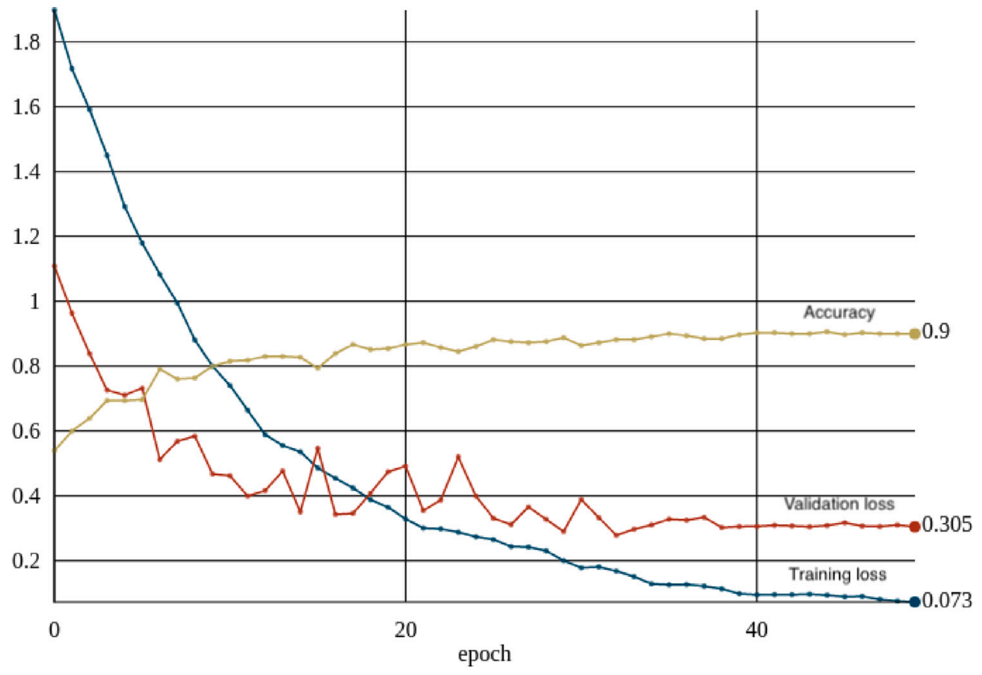


Fig. 7. Example of accuracy versus epochs plot over 50 epochs. This particular plot was generated from data of a leave one battery 02 out experiment with Bode feature extraction methods. Similar curve shapes have been observed during other training experiments.

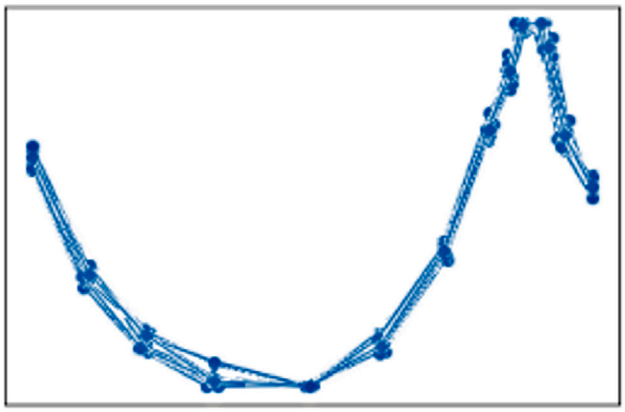


Fig. 8. Example overlay of the visual representation of synthetic data generated from an actual EIS measurement result. The synthetic curves were generated by adding Gaussian noise with standard deviation 0.1 mΩ according to the uncertainty band of the measurement system, estimated in [23].

Repeating the experiment with a higher data augmentation factor led to a marginal performance improvement of the order of 0.01 in the accuracy score with a significant impact on the computational cost of the training phase. Results of the different experiments are summarized in Table 1.

It is worth noting that in the *new measurement* test case, almost all predictions were within one class error range from the actual SOC class, as illustrated by the confusion matrix in Fig. 11. This results in a reliable coarse estimate of the SOC, which can be acceptable in several real-world scenarios.

3.3. System calibration

We deployed the training model in an inference pipeline to simulate usage in a practical SOC estimation system. In field applications, the model training should be performed during the system calibration procedure to fine-tune the model on the actual battery that must be monitored.

Table 1
Summary of accuracy achieved on the two test datasets by models trained with data augmentation factor 10.

Training	Test case	Representation	Accuracy
Leave One Out	New Measure	Bode	0.91
Leave One Out	New Measure	Nyquist	0.83
Leave One Out	New Battery	Bode	0.81
Leave One Out	New Battery	Nyquist	0.77
Validation Hold	New Measure	Nyquist	0.84
Validation Hold	New Measure	Bode	0.93
Validation Hold	New Battery	Nyquist	0.84
Validation Hold	New Battery	Bode	0.88

Experimental data revealed high variability across different batteries and also significant differences between measurements on the same battery at different times. For a field application, it is therefore necessary to tune the estimation model each time the battery is replaced and periodically to compensate for changes in the internal state of the battery that occur over time. This procedure, depicted in Fig. 12, can be performed as part of a periodic calibration process using hardware onboard the device powered by the battery or a dedicated external instrument that may be included in the docking station of the device (e.g., in the charging station of an electric vehicle).

To study this realistic case, we simulated this field application using the *new measurement* test case, described in Sections 2.4 and 2.5. After fine-tuning training on a set of EIS measurements from a specific battery, the classification model achieved accuracy in the range of 0.95–0.96 on the validation dataset and in the range of 0.92–0.93 on test data from a new set of EIS measurements from the same battery. These results confirm that a periodic system calibration with model fine-tuning on the last EIS measurements can compensate for model parameter drift due to battery aging and variation in operation environmental conditions.

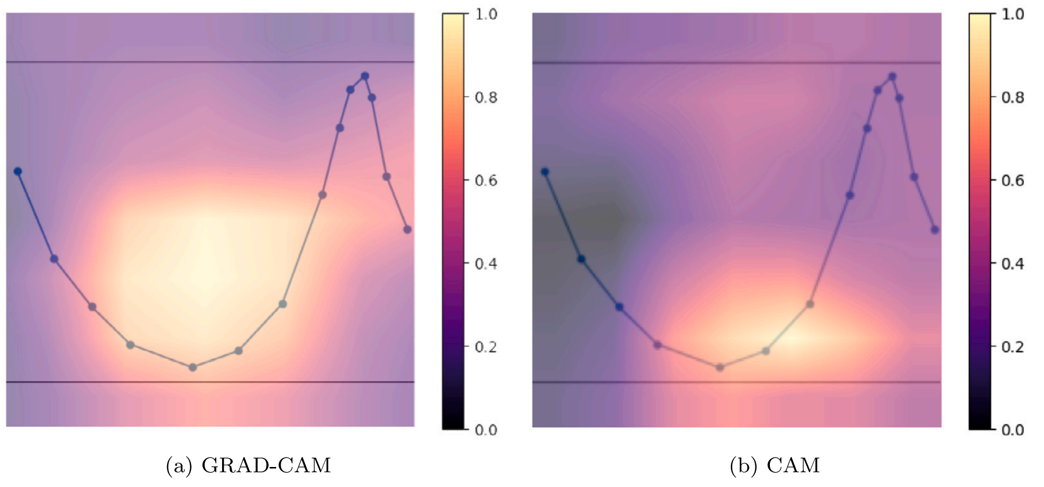


Fig. 9. Overlay of Class Activation Map (CAM) [left] and Gradient Class Activation Map (GRAD-CAM) [right] over the same EIS curve measured at SOC 20% on battery 05. The brighter colors indicate a relatively more significant contribution of feature (image pixels) in the area to the final classification. CAM and GRAD-CAM were computed following the method described in chapter 18 of [29] as implemented in *ML 4 Measurement library* [30] and in *Amalgam FastAI Extension* [31].

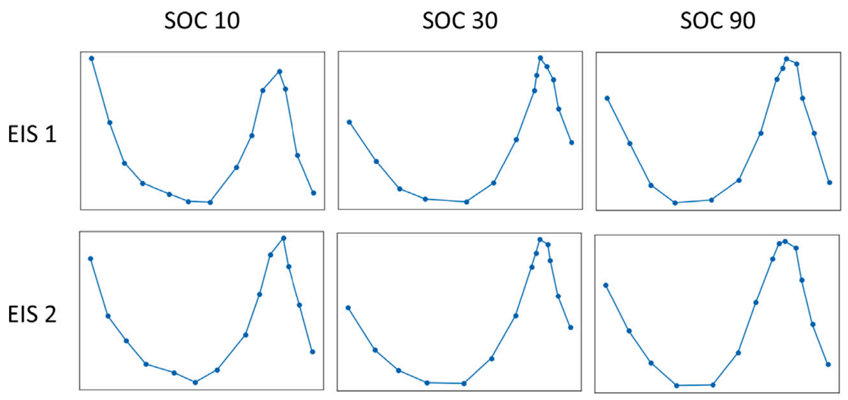


Fig. 10. EIS curves of two batteries at SOC 10%, 30%, and 90% with differences across SOC for the same battery and across batteries for the same SOC.

4. SOC estimation uncertainty

The *guide to the expression of measurement uncertainty* (GUM, JCGM 100) is the most widely adopted standard for uncertainty evaluation in metrology. The Monte-Carlo simulation (MCS) method, described in the standard *propagation of distributions by a Monte Carlo method* (GUMS1, JCGM 101) is an alternative for measurement uncertainty estimation where the GUM method is not applicable. The general procedure for the MCS method to estimate the uncertainty of a measurement is as follows:

1. Determine the number of simulations run n
2. Generate random variables for each uncertainty contributor (input) X_i $i = 0, 1, \dots, n$
3. Calculate the measurement output according to the mathematical model of the measurement $Y_i = f(x_i)$
4. If the statistical distribution of the measurement results Y_i from the simulation is known (in general, a Gaussian distribution is assumed), then the standard deviation of Y_i is calculated and used as the estimated uncertainty u. If the statistical distribution of the measurement results Y_i is not known and cannot be assumed, then the user should sort all the outputs Y_i in ascending order and take the value Y_i at the positions 2.5% (as the lower limit) and 97.5% (as the upper limit). These two limits are then used as the uncertainty interval (95% confidence interval).

The measurement result from our SOC classification system is a vector with values from neurons in the output layer of the network, one for each of the 10 classes of the *vocabulary*. The output values are normalized through a softmax activation function applied to the final layer of the network. The softmax function is used to convert a vector of arbitrary real values into a vector Y with values within 0 and 1, with the sum of all 10 components equal to 1. The predicted class label $\hat{y}$ is typically obtained by selecting the class associated with the neuron with the highest output value.

Despite the proven performance of standard deterministic deep learning methods in solving many real-world problems, they produce single-point predictions $\hat{y} = \Phi(f(x))$ and do not provide information on the reliability of their predictions. To address this issue, the Bayesian deep learning and Bayesian NNs approach should be adopted [36,37]. This approach involves placing a priori distribution over the model weights and then using Bayesian inference to obtain a posterior distribution over the weights given the observed data. This posterior distribution $p(\hat{y}|x)$ can then be used to quantify the uncertainty in the model predictions. Computing an exact posterior inference is difficult and often a mathematically intractable problem [38]. Therefore, practical applications should rely on a Bayesian inference approximation. Bayesian approximation and ensemble learning techniques are two widely used types of uncertainty evaluation methods [37,39]. In some applications, MCS is also used together with test data augmentation, such as in [40].

Gal and Ghahramani in [41] empirically showed that passing a point estimate through the softmax activation function results in overconfident predictions, while passing a distribution of estimates through the softmax yields more reasonable and lower confidence scores. The

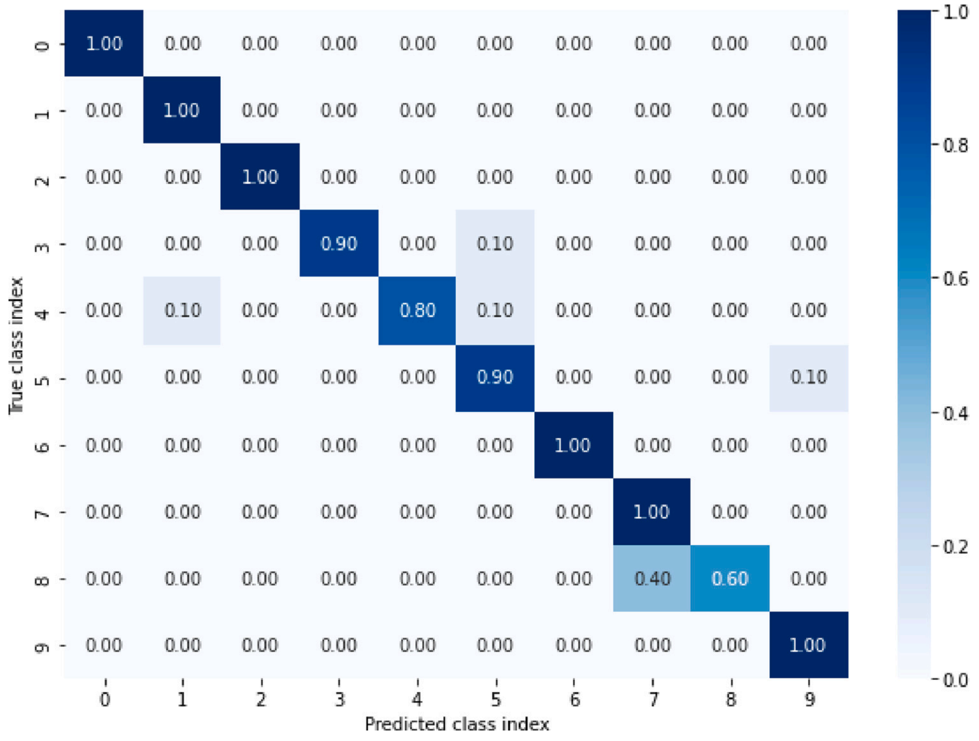


Fig. 11. Example of confusion matrix on *new measurement* test dataset. Most of the prediction errors are in one-class error range from the real class.

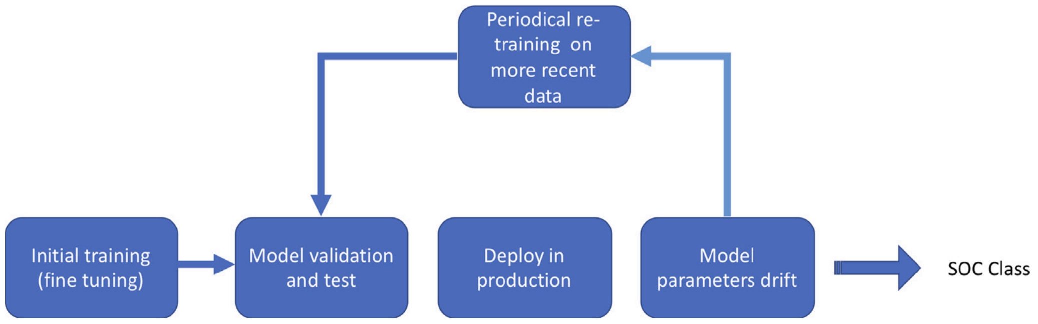

Periodical re-training on more recent data
Initial training (fine tuning)
Model validation and test
Deploy in production
Model parameters drift
SOC Class

Fig. 12. Periodical system calibration process.

authors then proposed the *Monte Carlo dropout* (MCD) method as a Bayesian approximation to improve the quality of uncertainty evaluation and mitigate the overconfidence issue. In this method, dropout is applied during both training and inference, which can be seen as an approximation to Bayesian deep learning. The dropout procedure generates multiple models, each of which can be used to make a prediction. The spread of the predictions provides an estimate of the uncertainty.

An alternative approach to uncertainty quantification is the *deep ensemble* [42,43] method, in which a single model is trained multiple times with different hyperparameters values and random seeds, and the predictions are combined to produce a final overall prediction.

The MCD method is the best fit for transfer learning scenarios when the cost of the multiple models training required by the deep ensemble method makes the procedure practically infeasible. The dropout layer [44] is usually already present in the deep NN architecture, but is enabled only during the training phase as normalization to avoid overfitting and then turned off during inference. MCD requires only performing a set of inference operations while keeping the dropout layer and collecting the outputs.

We applied the MCD method along with the MCS over a set of samples obtained from the same EIS spectrum with additive Gaussian noise. In some cases, MCD produced a distribution of predictions output values with a smaller variance compared to the classic MCS, while the two distributions are comparable in other cases. In the example of a SOC 10% EIS from the test dataset shown in Fig. 13, the MCD produced a set of output values with a standard deviation of 0.3002 while the standard deviation of values produced by MCS was 0.2981. We observed that distributions of output values computed on EIS spectra from the validation dataset could be much sharper, with a standard deviation in the order of $10^{-7} \div 10^{-5}$.

We could also analyze some incorrect predictions by applying the trained models to the *new battery* test dataset. In the example shown in Figs. 14 and 15, an EIS curve measured at SOC 30% was classified as SOC 40%. The standard deviation is 0.012 for the MCD output values and 0.154 for MCS. We noted that the MCD method also produced sharper output-value distributions in the observed cases of incorrect SOC classification.

4.1. Softmax normalization

The numeric outputs of the last linear layer of a multiclass classification neural network in the ML jargon are known as *logit* values. The softmax function Eq. (2) transforms logits into a discrete probability

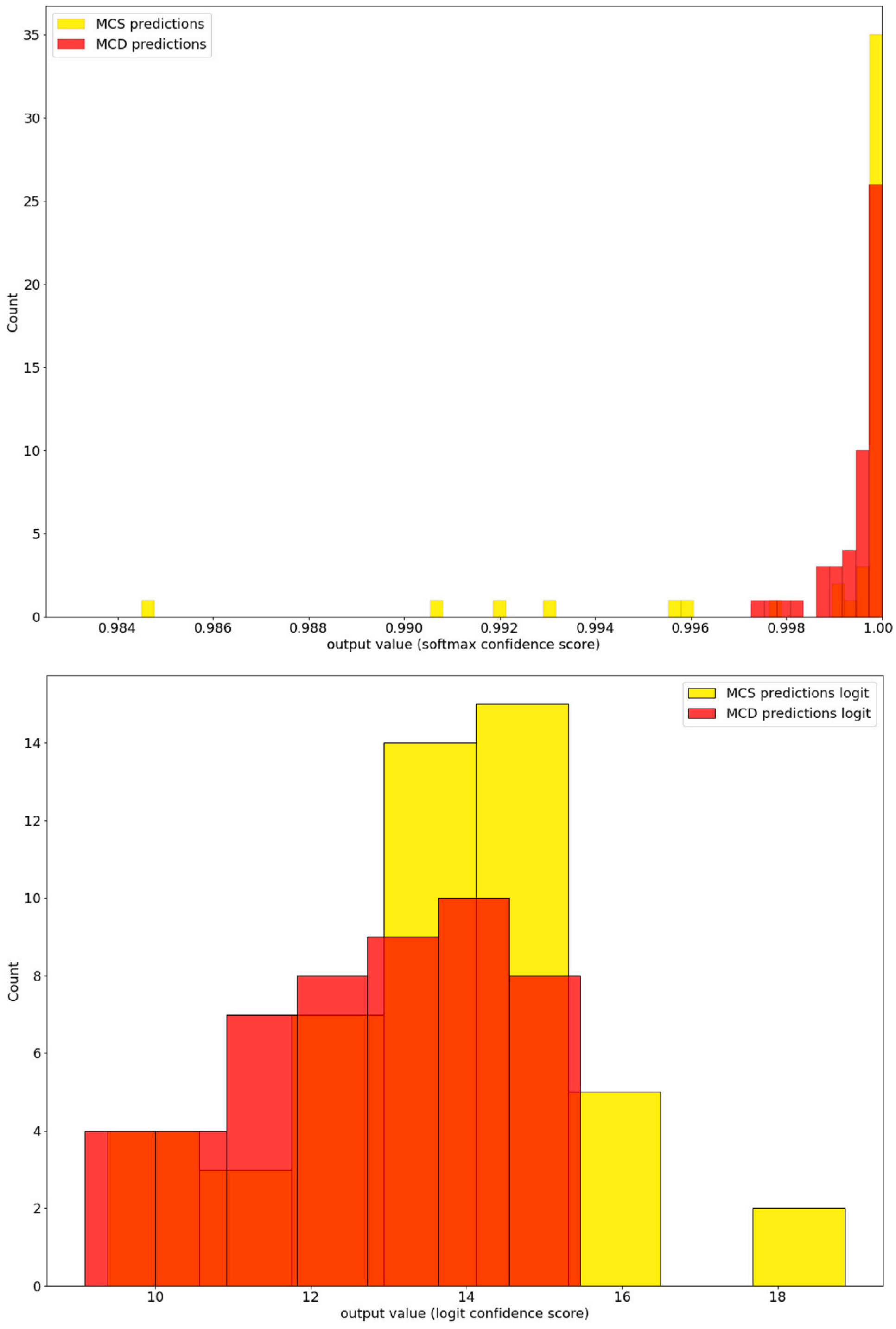


Fig. 13. Distributions of the output values of the SOC 10% class output neuron when the NN is fed with an actual SOC 10% EIS spectrum from *new battery* test dataset. The distribution was computed using MCD and MCS on 50 samples. In the top figure softmax normalization is active, while in the bottom figure the raw logits values are shown.

distribution over all possible classes. The function takes as input a vector of arbitrary length and outputs a vector of the same length, where each element is between 0 and 1 and the sum of all elements is equal to 1.

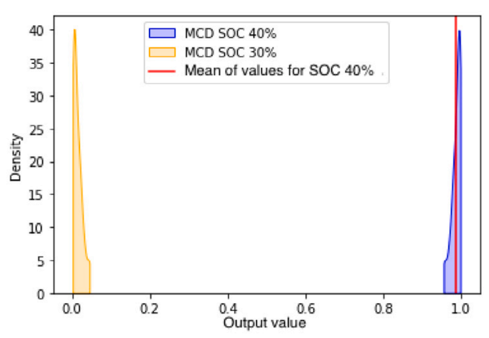
Given an input vector $\mathbf{z} = (z_1, z_2, \dots, z_k)$, the softmax function is defined as follows:

$$softmax(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad i = 1, 2, \dots, k \tag{2}$$

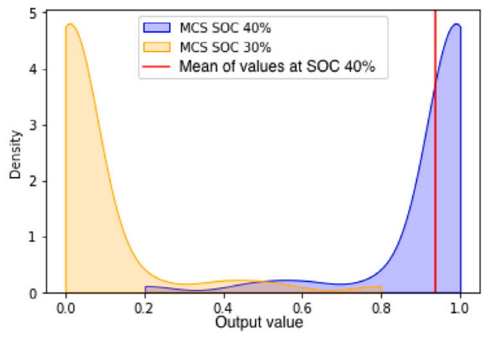
where $softmax(\mathbf{z})_i$ denotes the ith element of the output vector of the softmax function.

The numerator e^{z_i} computes the exponential of the ith element of the input vector, ensuring that the resulting values are not negative. The denominator $\sum_{j=1}^{k} e^{z_j}$ computes the sum of the exponentials of all elements in the input vector, normalizing the values and ensuring that their sum is equal to 1. The division of the numerator by the denominator produces the probability of the ith class.

Softmax is frequently added to the last layer of an image classification network such as VGG16 used in ImageNet competitions. Although the softmax layer is not included in the ResNet CNN, the *learner.get_preds* methods of the FastAI library, used in this work, apply by default a softmax normalization before returning the classification



(a) MCD



(b) MCS

Fig. 14. Output value distributions of the output neurons associated with the SOC 30% and 40% classes in a test case with incorrect SOC classification. The actual SOC was 30% while the system's overall prediction was 40%. MCD and MCS uncertainty estimation produced two different output values distributions. Sample size 50 was used for both computations. These curves were obtained using the kernel density smoothing method.

model output.[1] The softmax normalization could have various effects on raw logit values. In general, the softmax normalization produces a sharper value distribution. When all output logit values are very close, the softmax normalization amplifies the differences and boosts the confidence score value on the class with the highest logit value. Even when all logit outputs are very low (low activation of all output neurons). The effect of softmax on uncertainty estimation with MCD and MCS methods is illustrated in Figs. 14 and 15.

5. Discussion

The results presented in Section 3 demonstrate the feasibility of battery SOC estimation using EIS measurement data and 2D CNN. Using the Bode diagram visual representation, the trained system achieved 0.93 overall classification accuracy on the test dataset. This specific representation gives better results than the Nyquist diagram, possibly because the information about the frequency is embedded in the Bode representation, while it is not present in the Nyquist diagram. The classification approach implies an intrinsic uncertainty associated with the SOC estimation that depends on the extension and the number of classes. In our case, we trained the model using 10 classes and a 10% SOC step.

Our validation and test results also indicated that battery SOC estimation is affected by the *domain shift* issue: the inference accuracy of the trained model was limited by the variability across different batteries and different charge–discharge cycles. In our *new battery* test case, the accuracy drops to 0.88 until a new calibration on the particular battery in use was performed. A possible solution is the periodical calibration process described in Section 3.3.

The data augmentation technique we introduced in Section 2.7 is specific to EIS data and is helpful for model training and validation. It also enables test-time data augmentation for uncertainty quantification using the MCS method on EIS-based models. The proposed technique can be applied to any EIS measurement dataset and is not limited to CNN and visual representation methods used in this work.

The results presented in Section 4 show that the two considered methods for uncertainty quantification, i.e., MCS and MCD, give comparable results. This comparison is a relevant contribution of the present work, since, to the extent of the authors' knowledge, these two methods have not yet been compared experimentally on the same dataset. Therefore, based on these results, the Monte Carlo GUM approach for uncertainty evaluation can be applied to machine learning models used in measurement processes. Figs. 14 and 15 clearly show that softmax normalization influences the distribution of output values. Given that the use of the softmax function, often in combination with the *cross entropy* loss function, is very common in multiclass classification applications, the effect of softmax normalization on uncertainty quantification should be further investigated.

6. Conclusions

This paper proposed a machine learning approach based on EIS data and 2D CNN. The method was validated on an open-access dataset acquired from Samsung ICR18650-26J lithium-Ion batteries. The results show that the method achieves a 0.93 overall classification accuracy on the *new measurement* test dataset and 0.88 overall classification accuracy on the *new battery* test dataset. A data augmentation technique was introduced, which is specific to EIS data, to be used for model training. This technique also enables test-time data augmentation for uncertainty quantification using the MCS method. An uncertainty quantification procedure for SOC estimation using MCS and MCD was provided. Our experimental results showed that the MCS and MCD methods give comparable results on the considered test case. The uncertainty quantification enables the definition of uncertainty and confidence thresholds for *uncertainty informed* [39] deep learning models suitable for high confidence predictions in critical application fields, such as biomedicine, robotics, and autonomous vehicles.

CRediT authorship contribution statement

Emanuele Buchicchio: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing, Validation. **Alessio De Angelis:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing, Validation. **Francesco Santoni:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Validation. **Paolo Carbone:** Conceptualization, Methodology, Writing – review & editing, Supervision.

Declaration of competing interest

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

Data availability

The raw current and voltage data acquisition are published on the Mendeley Data repository and available to download from https://data.mendeley.com/datasets/zdsgxwksn5.

The EIS dataset required to reproduce the above findings is published on the Mendeley Data repository and available to download from https://data.mendeley.com/datasets/mbv3bx847g.

A detailed description of the EIS dataset is also available in [24].

[1] See https://docs.fast.ai/learner.html#learner.get_preds.

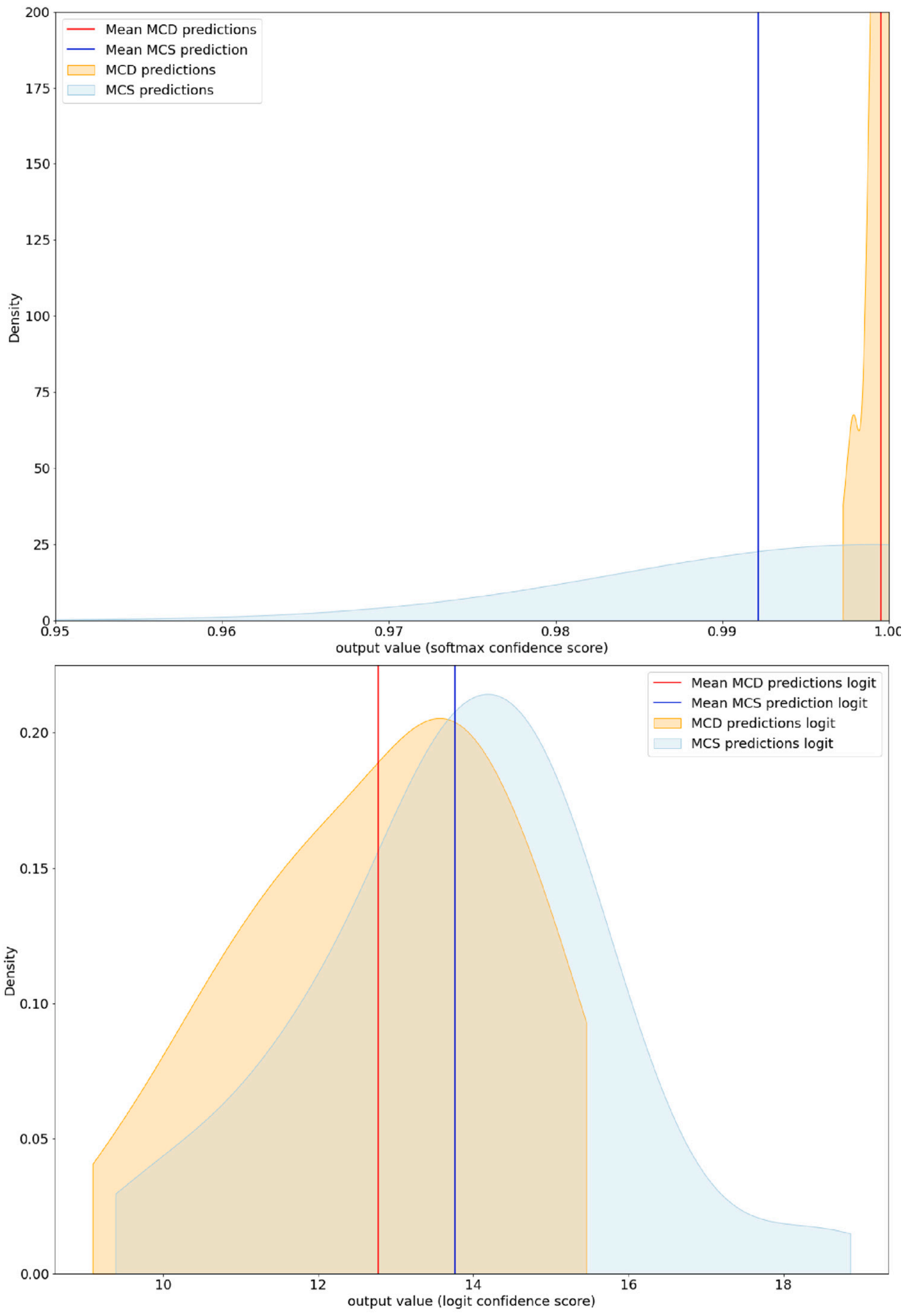


Fig. 15. Comparison of the distribution of the output value of the output neuron associated with incorrect SOC 40% class. The distribution was computed with MCD and MCS method with sample size 50. These curves were obtained using the kernel density smoothing method. In the top figure softmax normalization is active, while in the bottom figure the raw logits values are shown.

Appendix A. Computational reproducibility

We embrace Reproducibility Enhancement Principles (REP) from [45] publishing all software and data used for this work. All software developed or used for this work is publicly available and open source. We published all algorithms we develop on *ML Measurement library* repository [30]. We also published several Jupyter notebooks to reproduce the result discussed in this paper in a dedicated public repository hosted on GitHub.[2] These notebooks are directly executable on free cloud machine learning computing platforms such as *Google Colab* or *Sage Maker Studio Lab*. On Google Colab, the notebooks will make use of the free Google storage service associated with the account

[2] https://github.com/electrical-and-electronic-measurement/state-of-charge-with-EIS-and-CNN

of the reader to permanently store all trained models and generated images across Colab sessions. Sage Maker Studio Lab includes a permanent storage service itself. While paid service subscriptions include more computational resources, the hardware and software configuration available with the free service allows running all experiments from scratch within a few hours using the GPU-accelerated runtimes. The notebook itself will take care of the installation of missing third-party software components in the *setup* phase.

References

[1] Z. Chen, H. Zhao, X. Shu, Y. Zhang, J. Shen, Y. Liu, Synthetic state of charge estimation for lithium-ion batteries based on long short-term memory network modeling and adaptive H-Infinity filter, Energy 228 (2021) 120630, http://dx.doi.org/10.1016/j.energy.2021.120630, URL https://www.sciencedirect.com/science/article/pii/S0360544221008793.

[2] E. Chemali, P.J. Kollmeyer, M. Preindl, A. Emadi, State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach, J. Power Sources 400 (2018) 242–255, http://dx.doi.org/10.1016/j.jpowsour.2018.06.104, URL https://linkinghub.elsevier.com/retrieve/pii/S0378775318307080.

[3] C. Hu, L. Ma, S. Guo, G. Guo, Z. Han, Deep learning enabled state-of-charge estimation of LiFePO4 batteries: A systematic validation on state-of-the-art charging protocols, Energy 246 (2022) 123404, http://dx.doi.org/10.1016/J.ENERGY.2022.123404, URL https://linkinghub.elsevier.com/retrieve/pii/S0360544222003073.

[4] M.E. Orazem, B. Tribollet, Electrochemical impedance spectroscopy, The ECS Series of Texts and Monographs, Wiley, 2008, https://onlinelibrary.wiley.com/doi/book/10.1002/9780470381588.

[5] Y. Fernández Pulido, C. Blanco, D. Anseán, V.M. García, F. Ferrero, M. Valledor, Determination of suitable parameters for battery analysis by Electrochemical Impedance Spectroscopy, Measurement 106 (2017) 1–11, http://dx.doi.org/10.1016/j.measurement.2017.04.022, URL https://www.sciencedirect.com/science/article/pii/S0263224117302476.

[6] C. Chang, S. Wang, C. Tao, J. Jiang, Y. Jiang, L. Wang, An improvement of equivalent circuit model for state of health estimation of lithium-ion batteries based on mid-frequency and low-frequency electrochemical impedance spectroscopy, Measurement 202 (2022) 111795, http://dx.doi.org/10.1016/j.measurement.2022.111795, URL https://www.sciencedirect.com/science/article/pii/S0263224122009976.

[7] I. Babaeiyazdi, A. Rezaei-Zare, S. Shokrzadeh, State of charge prediction of EV Li-ion batteries using EIS: A machine learning approach, Energy 223 (2021) 120116, http://dx.doi.org/10.1016/j.energy.2021.120116, URL https://www.sciencedirect.com/science/article/pii/S0360544221003650.

[8] C. Woosung, S. Heon-Cheol, K.J. Man, C. Jae-Young, Y. Won-Sub, Modeling and applications of electrochemical impedance spectroscopy (EIS) for lithium-ion batteries, J. Electrochem. Sci. Technol. 11 (1) (2020) 1–13, http://dx.doi.org/10.33961/jecst.2019.00528, arXiv:http://www.jecst.org/journal/view.php?number=315. URL http://www.jecst.org/journal/view.php?number=315.

[9] M.A. Hannan, D.N. How, M.B. Mansor, M.S.H. Lipu, P. Ker, K. Muttaqi, State-of-charge estimation of li-ion battery using gated recurrent unit with one-cycle learning rate policy, IEEE Trans. Ind. Appl. 57 (2021) 2964–2971, http://dx.doi.org/10.1109/TIA.2021.3065194.

[10] S. Hong, H. Hwang, D. Kim, S. Cui, I. Joe, Real driving cycle-based state of charge prediction for ev batteries using deep learning methods, Appl. Sci. (Switzerland) 11 (2021) http://dx.doi.org/10.3390/app112311285.

[11] Y. Zhang, R. Xiong, H. He, M.G. Pecht, Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries, IEEE Trans. Veh. Technol. 67 (2018) 5695–5705, http://dx.doi.org/10.1109/TVT.2018.2805189.

[12] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.

[13] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, 1D convolutional neural networks and applications: A survey, Mech. Syst. Signal Process. 151 (2021) 107398, http://dx.doi.org/10.1016/j.ymssp.2020.107398, URL https://www.sciencedirect.com/science/article/pii/S0888327020307846.

[14] L. Wen, X. Li, L. Gao, Y. Zhang, A new convolutional neural network-based data-driven fault diagnosis method, IEEE Trans. Ind. Electron. 65 (7) (2018) 5990–5998, http://dx.doi.org/10.1109/TIE.2017.2774777.

[15] S.K. Khare, V. Bajaj, Time–frequency representation and convolutional neural network-based emotion recognition, IEEE Trans. Neural Netw. Learn. Syst. 32 (7) (2021) 2901–2909, http://dx.doi.org/10.1109/TNNLS.2020.3008938.

[16] S. Shan, J. Liu, S. Wu, Y. Shao, H. Li, A motor bearing fault voiceprint recognition method based on Mel-CNN model, Measurement 207 (2023) 112408, http://dx.doi.org/10.1016/j.measurement.2022.112408, URL https://www.sciencedirect.com/science/article/pii/S0263224122016050.

[17] A. Ullah, S.u. Rehman, S. Tu, R.M. Mehmood, Fawad, M. Ehatisham-ul haq, A hybrid deep CNN model for abnormal arrhythmia detection based on cardiac ECG signal, Sensors 21 (3) (2021) http://dx.doi.org/10.3390/s21030951, URL https://www.mdpi.com/1424-8220/21/3/951.

[18] E. Buchicchio, A. De Angelis, F. Santoni, P. Carbone, Lithium-Ion Batteries state of charge estimation based on electrochemical impedance spectroscopy and convolutional neural network, in: Proceedings of the 25th IMEKO TC4 International Symposium, IMEKO, 2022, pp. 90–95, URL https://imeko-tc4-2022.org/procs/IMEKO_2022_Proceedings.pdf.

[19] A. Guha, A. Patra, Online estimation of the electrochemical impedance spectrum and remaining useful life of lithium-ion batteries, IEEE Trans. Instrum. Meas. 67 (2018) http://dx.doi.org/10.1109/TIM.2018.2809138.

[20] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, URLhttp://www.deeplearningbook.org.

[21] R. Pintelon, J. Schoukens, System Identification: A Frequency Domain Approach, John Wiley & Sons, 2012.

[22] A. De Angelis, P. Carbone, A. Moschitta, M. Crescentini, R. Ramilli, P.A. Traverso, A fast and simple broadband EIS measurement system for li-ion batteries, in: 24th IMEKO TC4 International Symposium and 22nd International Workshop on ADC and DAC Modelling and Testing, 2020, URLhttps://www.imeko.org/publications/tc4-2020/IMEKO-TC4-2020-30.pdf.

[23] A. De Angelis, E. Buchicchio, F. Santoni, A. Moschitta, P. Carbone, Uncertainty characterization of a practical system for broadband measurement of battery EIS, IEEE Trans. Instrum. Meas. 71 (2022) 1–9, http://dx.doi.org/10.1109/TIM.2022.3156994.

[24] E. Buchicchio, A. De Angelis, F. Santoni, P. Carbone, F. Bianconi, F. Smeraldi, Dataset on broadband electrochemical impedance spectroscopy of lithium-ion batteries for different values of the state-of-charge, Data in Brief 45 (2022) 108589, http://dx.doi.org/10.1016/j.dib.2022.108589, URL https://www.sciencedirect.com/science/article/pii/S235234092200796X.

[25] E. Buchicchio, A. De Angelis, F. Santoni, P. Carbone, Dataset on broadband electrochemical impedance spectroscopy of lithium-ion batteries for different values of the state of charge, Mendeley Data (2022) http://dx.doi.org/10.17632/MBV3BX847G.3.

[26] J. Howard, S. Gugger, FastAI: A layered API for deep learning, Information (Switzerland) 11 (2020) http://dx.doi.org/10.3390/info11020108.

[27] L.N. Smith, Cyclical learning rates for training neural networks, in: Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, 2017, http://dx.doi.org/10.1109/WACV.2017.58.

[28] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339, http://dx.doi.org/10.18653/v1/P18-1031, URL https://aclanthology.org/P18-1031.

[29] J. Howard, S. Gugger, Deep Learning for Coders with Fastai and Pytorch: AI Applications Without a PhD, O'Reilly Media, Incorporated, 2020, URL https://books.google.no/books?id=xd6LxgEACAAJ.

[30] E. Buchicchio, Machine learning for measurement: v0.2.0, 2022, http://dx.doi.org/10.5281/zenodo.6672602, *Zenodo*.

[31] R. Somani, Amalgam Fast AI Extention, 2022, URL https://github.com/emanbuc/amalgam-fastAI-extention.

[32] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 2921–2929, http://dx.doi.org/10.1109/CVPR.2016.319.

[33] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, Int. J. Comput. Vis. 128 (2) (2019) 336–359, http://dx.doi.org/10.1007/s11263-019-01228-7, URL https://link.springer.com/article/10.1007/s11263-019-01228-7.

[34] A. Schöttl, A light-weight method to foster the (grad)cam interpretability and explainability of classification networks, in: 2020 10th International Conference on Advanced Computer Information Technologies, ACIT, 2020, pp. 348–351, http://dx.doi.org/10.1109/ACIT49673.2020.9208950.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[36] S. Theodoridis, Chapter 12 - Bayesian learning: Inference and the EM algorithm, in: S. Theodoridis (Ed.), Machine Learning, second ed., Academic Press, 2020, pp. 595–646, http://dx.doi.org/10.1016/B978-0-12-818803-3.00023-4, URL https://www.sciencedirect.com/science/article/pii/B9780128188033000234.

[37] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U.R. Acharya, V. Makarenkov, S. Nahavandi, A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Inf. Fusion 76 (2021) 243–297, http://dx.doi.org/10.1016/j.inffus.2021.05.008, URL https://www.sciencedirect.com/science/article/pii/S1566253521001081.

[38] S. Theodoridis, Chapter 13 - Bayesian learning: Approximate inference and nonparametric models, in: S. Theodoridis (Ed.), Machine Learning, second ed., Academic Press, 2020, pp. 647–730, http://dx.doi.org/10.1016/B978-0-12-818803-3.00025-8, URL https://www.sciencedirect.com/science/article/pii/B9780128188033000258.

[39] J.M. Dolezal, A. Srisuwananukorn, D. Karpeyev, S. Ramesh, S. Kochanny, B. Cody, A.S. Mansfield, S. Rakshit, R. Bansal, M.C. Bois, et al., Uncertainty-informed deep learning models enable high-confidence predictions for digital histopathology, Nature Commun. 13 (1) (2022) http://dx.doi.org/10.1038/s41467-022-34025-x.

[40] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, T. Vercauteren, Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks, Neurocomputing 338 (2019) 34–45, http://dx.doi.org/10.1016/j.neucom.2019.01.103, URL https://www.sciencedirect.com/science/article/pii/S0925231219301961.

[41] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, 2015, http://dx.doi.org/10.48550/ARXIV.1506.02142, URL https://arxiv.org/abs/1506.02142.

[42] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS '17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6405–6416.

[43] F. Wenzel, J. Snoek, D. Tran, R. Jenatton, Hyperparameter ensembles for robustness and uncertainty quantification, Adv. Neural Inf. Process. Syst. 33 (2020) 6514–6527.

[44] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, 2012, http://dx.doi.org/10.48550/ARXIV.1207.0580, URL https://arxiv.org/abs/1207.0580.

[45] V. Stodden, M. McNutt, D.H. Bailey, E. Deelman, Y. Gil, B. Hanson, M.A. Heroux, J.P. Ioannidis, M. Taufer, Enhancing reproducibility for computational methods, Science 354 (6317) (2016) 1240–1241, http://dx.doi.org/10.1126/science.aah6168, arXiv:https://www.science.org/doi/pdf/10.1126/science.aah6168. URL https://www.science.org/doi/abs/10.1126/science.aah6168.