Forecasting Li-ion battery State of Charge using Long-Short-Term-Memory network

Irene Capodicasa[1,*], Tania Cerquitelli[2,*]

[1]Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin, 10129, Italy
[2]Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin, 10129, Italy

Abstract
Estimating the state of charge (SOC) for lithium-ion batteries (LIB) has become a highly desirable task, but also critical, especially as electrified vehicles become more common. However, due to the non-linear behaviour of these batteries, accurately estimating SOC remains a challenge. As a result, traditional theory-based methods are often being replaced by data-driven approaches, thanks to the greater availability of battery data and advances in artificial intelligence. Recurrent neural networks (RNNs), in particular, are promising methods to be exploited, because they can capture temporal dependencies and predict SOC without a battery model. Long short term memory (LSTM), a specific type of RNN, can accurately predict SOC values in real-time and forecast future SOC values within different time horizons.

Keywords
State of Charge, Long short term memory, battery, neural network, estimation, electric vehicle, time horizons

1. Introduction

Lithium-ion batteries have gained immense popularity in various industries, particularly in electric vehicles, and have become increasingly prevalent in recent years. LIBs are highly efficient, delivering a greater amount of energy for the same volume and mass compared to conventional batteries like lead-acid batteries. Accurately estimating the state of charge of a battery is crucial for making informed decisions at all stages of its life. SOC estimation also helps in enhancing vehicle performance, safety, and passenger comfort, reducing costs associated with battery over-sizing, and improving overall vehicle efficiency. However, direct measurement of SOC is not possible and must be estimated.

The key technologies for estimating the state of charge of lithium-ion batteries for electric vehicles can be divided into three main groups: (1) model-based methods such as simplified electrochemical models and equivalent circuit models [1][2][3][4], (2) machine learning methods , including neural networks [5][6] [7][8][9][10]; and (3) hybrid methods composed of two or more of the previously mentioned algorithms [2][1]. The most common model-based methods for SOC estimation are Coulomb counting and open-circuit voltage, which require many parameter measurements, usually affected by noise. More sophisticated models have been developed to deal with these uncertainties, including an equivalent circuit model (ECM) that requires extensive battery test models and parameters [1].

In recent times, data-driven techniques have gained significant popularity owing to advancements in artificial intelligence and machine learning, coupled with the wider availability of battery data.
The most commonly used algorithms are:

- Artificial Neural Networks (ANN), which have the ability to function under non-linear conditions, and utilize inputs such as battery terminal voltage, discharge or charge current, and temperature. Unfortunately, they require large amounts of training data to derive accurate data-driven models, thus high computational power and large memory storage are needed to perform the learning phase [5] efficiently;
- Support Vector Machine (SVM), which can deal with noisy data and incorporate knowledge from other indicators such as energy, power, etc., but on the other hand, training can be very time-consuming [6];
- some other data-driven algorithms, e.g., fuzzy logic and genetic algorithms (see [5] for further details).

Hybrid techniques are utilized to enhance the precision and effectiveness of battery models while circumventing the limitations of a single algorithm. The primary disadvantage of these techniques is their reliance on significant memory and computational power to execute complex mathematical calculations.

Battery modeling is a crucial step in developing a precise SOC estimation algorithm. However, the existing battery modeling approaches proposed in literature have

Published in the Workshop Proceedings of the EDBT/ICDT 2023 Joint Conference (March 28-March 31, 2023, Ioannina, Greece)
*Corresponding author.
s285205@studenti.polito.it (I. Capodicasa);
tania.cerquitelli@polito.it (T. Cerquitelli)

limitations in accurately assessing the battery's aging process and updating the models continuously. Further research is required to enhance the precision of battery modeling. The process of modeling the behavior of batteries and their adaptive control technology involves the utilization of expert system theories and artificial intelligence. With the huge amounts of data generated by energy storage systems, it is natural to utilize machine learning algorithms for state and parameter estimation, although the accuracy of these approaches is low, which may become a problem in the optimal monitoring of the LIB state.

In this paper, we present a data-driven methodology based on LSTM cells to estimate SOC accurately. The proposed approach maps battery measurement signals such as voltage, current, and temperature to the battery SOC. Specifically, the novel contribution of this paper is twofold:

- the application of the LSTM network to predict SOC at different time horizons. SOC works like a fuel gauge in a car, so an accurate prediction of SOC at a given time horizon can be critical for users to know how long they can drive their car before the battery dies or stops working. On the other hand, from car manufacturers perspective, SOC real-time estimation and prediction at different future horizons can help monitor battery status and plan future maintenance and repair work. In addition, it is crucial when the battery reaches the end of its first life, and manufacturers must decide whether it can be used for other purposes or disassemble it.
- With the proposed method, we show how powerful the LSTM is in predicting SOC: the key point is that data collected at different frequencies can lead to different results in predicting future values of SOC. This means that the granularity of the input data used to train an LSTM must be determined concerning the time horizon we want to use to estimate the battery SOC.

The remainder of this paper is organized as follows. Section 2 contains a literature review, highlighting the new contribution of this work. Section 3 introduces the data set and describes the data preprocessing and model-building steps. Section 4 presents the experimental results, and Section 5 provides some concluding remarks.

2. Related work

SOC estimation methods can be divided into the following three categories: the simplified electrochemical models (SEM)[1][2], the equivalent circuit models (ECM) [4][3][2] and the machine learning models [5][6] which includes neural network models [7][8][9][10].

By mathematically describing the internal processes based on electrochemical mechanisms, SEMs are capable of reflecting the battery characteristics: for example, Coulomb Counting is an ampere-hour (Ah) counting estimation method that integrates the discharging or charging current to determine the remaining charge in the battery; another method is open-circuit voltage (OCV), which employs the battery's stable electromotive force while in the open-circuit state, and uses the correlation between the OCV and SOC to approximate the SOC value [1]. The ECMs are created by combining resistors, capacitors, and voltage sources to generate a circuit network, which results in their high level of accuracy, robustness, and insensitivity to external disruptions [4]: the Kalman filter is one such example of an ECM that employs mathematical equations to recursively compute a linear optimal filtering solution for SOC estimation [3].

Machine learning methods for estimating SOC, which includes neural network methods, are summarized in [5]. The Support Vector Machine (SVM) is one of them: it has an excellent generalization capability compared to neural networks that may have local minimization problems. A regression SVM is applied by [6] to predict SOC of a LIB. For estimating SOC, neural network algorithms can generally be divided into recurrent and non-recurrent. Recurrent algorithms have a memory for the past, while non-recurrent algorithms depend on the data input at the current time step. At McMaster University, Carlos Vidal and his team have shown that training a feed-forward neural network is faster than training a recurrent network and that the error is smaller [7]. Creating a sufficiently large battery dataset to train a deep neural network (DNN) for SOC estimation can be challenging. To reduce the required test data, a Canadian university [9] proposes to use a specific DNN, the LSTM, which can estimate the SOC of different types of lithium-ion batteries. In [8], the authors propose a LSTM network to estimate SOC of a Panasonic 18650 battery cell, and show that it provides competitive estimation performance compared to other algorithms reported in the literature. In contrast to unidirectional RNNs, Yang's team at Beihang University proposed a model using a bidirectional LSTM and the study demonstrated the network's ability to comprehend the temporal information present in sequential sensor data obtained from LIBs. This includes variables such as voltage, current, and temperature measurements captured in both forward and backward directions. Additionally, the network effectively summarizes temporal dependencies from past and future contexts. [10].

3. Methodology

In this section, we provide a detailed description of the steps we took to obtain a SOC estimate: as shown in Figure 1, the first step is data collection of battery's parameters thanks to measurement sensors; the second step is to preprocess and prepare the data to obtain a training set and a test set to be fed into the network; the last step is LSTM training and performance evaluation.

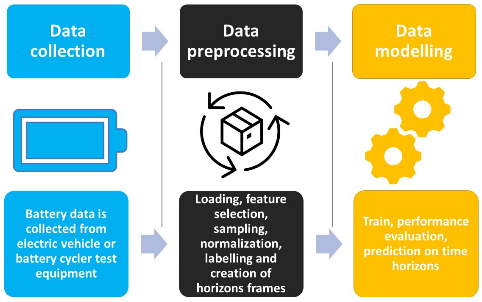

Figure 1: Figure showing the steps followed to obtain SOC estimation.

3.1. Data collection

Data plays a crucial role in driving innovative advancements in battery development, modeling, and management. For instance, the development of a battery management system (BMS) to regulate battery operations necessitates the usage of data both for its creation, as well as for the training and calibration of the models employed to estimate battery states, such as State of Health (SOH), SOC, and Remaining Useful Life (RUL). The authors in [11] presented an overview of publicly available battery datasets: electric vehicle (EV) battery requirements differ from those for laptops, cell phones, stationary energy storage, and other devices. Thus, application-specific data are needed for:

- cycle aging data: typically, input data include in-cycle measurements of current, voltage, and temperature and per-cycle measurements of capacity and internal resistance or impedance;
- drive cycle data: input data are collected by cycling batteries according to the drive schedules;
- chemistry cell modeling: is mainly based on the short-term responses of current and voltage and focuses on the impedance variance at different battery SOC levels and temperatures;
- calendar aging: data include information related to battery cycler such as voltage, current, capacity, and energy from periodic characterization tests.

Various countries and organizations create driving cycles, which are employed to evaluate vehicles' performance in terms of factors like fuel consumption, pollutant emissions, and traffic impact. Cycling batteries can gather a lot of data based on these driving schedules, which can then be utilized for SOC estimation algorithms under realistic conditions. The universally recognized driving cycle tables can be classified into European, American, and Asian driving cycles [11].

Most used in literature are the following American (US) driving cycles:

- highway fuel economy test (HWFET) is a chassis dynamometer driving schedule representing highway driving conditions under 60 mph, for the determination of fuel economy of light-duty vehicles over highway driving cycle;
- federal test procedure (FTP-75) has been created by US EPA (Environmental Protection Agency) to represent a commuting cycle with a part of urban driving including frequent stops, and a part of highway driving;
- US06 is a supplemental federal test procedure cycle that represents aggressive, high speed and/or high acceleration driving behaviour;
- LA92 unified dynamometer driving schedule was developed as an emission inventory improvement tool, and is for Class 3 heavy-duty vehicles (power-to-mass ratio is greater than 34);
- urban dynamometer driving schedule (UDDS) is also known as "the city test" and represents city driving conditions. It is used for light-duty vehicle testing.

3.2. Data preprocessing

3.2.1. Data gathering

In this paper, the dataset used for training, testing and validation of the network was released by the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, and is freely available online [12].

All tests were performed in a thermal chamber with cell test equipment showed in Figure 2: the control computer contains the databases with the driving schedules and then, after the termal chamber is set to the desired ambient temperature and the cell is fully charged, the system starts to record the current drive cycle. The 3Ah LG 18650HG2 battery cell was subjected to four drive cycles, UDDS, HWFET, LA92, US06, and eight drive cycles (mix 1-8) consisting of a random mix of UDDS, HWFET, LA92, and US06. Following every test, the battery was charged at a rate of 1C with 50mA until it reached a voltage of 4.2V, after which it was turned off. It was ensured that the battery temperature remained at or above 22°C

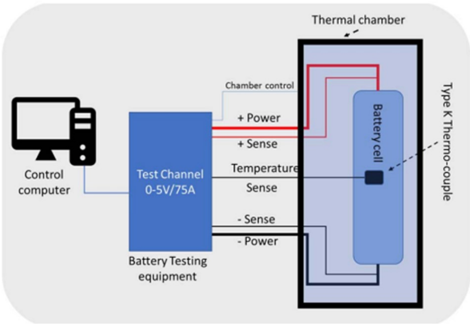

Figure 2: Battery test equipment for data monitoring.

throughout this process. The cycler collects the following measurements:

- Time (time in seconds)
- TimeStamp (timestamp in MM/DD/YYYY HH:MM:SS AM format)
- Voltage (measured cell terminal voltage, sense leads welded directly to battery terminal)
- Current (measure current in amps)
- Capacity (measured amp-hours (Ah), with Ah counter, typically reset after each charge, test, or drive cycle)
- Energy (measured watt-hours (Wh), with Wh counter, reset after each charge, test, or drive cycle)
- Battery_Temp_degC (battery case temperature, at the middle of battery, in degrees Celsius)

The training data set consists of the eight mixed driving cycles and their corresponding charges, while the test consists of the UDDS, LA92, and US06 driving cycles and their corresponding charges. All the measurements refers to tests performed only in a 25°C thermal chamber since analyzing how different temperatures can affect the SOC estimation is out of our work's scope.

3.2.2. Feature selection: correlation

The features relating to the battery are voltage, current, capacity, energy, and temperature. The response variable, the state of charge of the battery, is obtained from the capacity (Ah) of the battery calculated by the battery cycler (detailed procedure is explained in Section 3.2.5). To measure how much the relationship between dataset features and target variable (SOC) is close to a linear function, the Pearson correlation coefficient is used: the more the absolute value of the correlation coefficient is higher, the more the correlation is stronger [13].

From the correlation matrix in Figure 3, the strongest correlation is between capacity and energy and both are

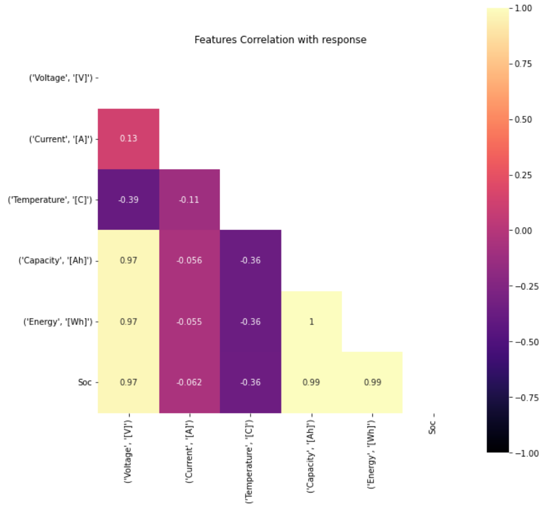

Figure 3: This figure shows the correlation between features and response variable (SOC).

strongly correlated with SOC. Unfortunately, as mentioned earlier, capacity was used to evaluate SOC, so both capacity and energy must be excluded as predictors. So the variables used as inputs are voltage (V), current (A) and temperature (°C).

3.2.3. Sampling

The collected data from battery cyclers may have different frequencies: in our dataset [12], data related to driving cycles and mix have a time step of 0.1 seconds, while data related to the charging phase have a slower dynamic and were considered less important, so they were stored at a lower data rate of 1 minute.

As mentioned in Section 3.2.1, both the training and test data contains drive cycles and charges, so both must be up-sampled to obtain the same frequency.

3.2.4. Normalization

Since machine learning algorithms generally do not work well with numerical attributes with different scales, the dataset must be rescaled. To ensure that input features are within a bounded range of 0 to 1, a technique called min-max scaling, or normalization, is employed. While there are several other methods available, min-max scaling is preferred specifically for neural networks as unbounded input features may pose difficulties [14].

3.2.5. Data Labelling

Prediction algorithms require apriori knowledge about the values to be predicted (i.e., SOC in our research). Unfortunately, SOC cannot be measured directly but can be easily estimated. Following the SOC definition, it is defined as the battery percentage remaining charge and is obtained by dividing its remaining capacity (Ah), by its nominal capacity. Each label (SOC) is associated with the corresponding features for the real-time prediction of SOC. On the other hand, when predicting SOC for different future time horizons, each label is shifted with respect to its initial position by the number of rows needed to reach that time horizon: for example, if we try to predict SOC within 10 minutes and the data is sampled at $\frac{1}{60}$ Hz, each label will be shifted by 10 rows.

3.3. Data modelling

Long Short Term Memory is a type of recurrent neural network. RNNs have an internal unit that can form a cycle to show the state history of the previous input. A general LSTM unit consists of:

- input gate $i_k = \mu\left(W_{\Psi i}\Psi_k + W_{hi}h_{k-1} + b_i\right)$ which controls which value of the input should be used to modify the memory. The sigmoid function μ decides which values to let through 0 or 1;
- memory cell $c_k = f_k c_{k-1} + \tanh\left(W_{\Psi c}\Psi_c + W_{hc}h_{k-1} + b_c\right)$ where $\tanh$ function gives weights to the values which are passed, deciding their level of importance ranging from -1 to 1;
- forget gate $f_k = \mu\left(W_{\Psi f}\Psi_k + W_{hf}h_{k-1} + b_f\right)$ that regulates the details to be discarded from the block using the sigmoid function μ;
- output gate $o_k = \mu\left(W_{\Psi o}\Psi_k + W_{ho}h_{k-1} + b_o\right)$ that is the result of the input and the memory cell gate. Again here, the sigmoid function can be zero-valued, so it can inhibit the flow of information to the next computational node;
- state of the cell $h_k = o_k \tanh c_k$ where $\tanh$ function gives weights to the values which are passed, deciding their level of importance ranging from -1 to 1.

Each gate has its set of network weights denoted by W and a bias b is added at each matrix multiplication to increase the flexibility of the network to the data. Ψ denotes the vector of inputs to the network, which are the voltage, current, and temperature of the battery measured at time step k.

3.3.1. Training

The training dataset consists of $\Psi_k = [V(k), I(k), T(k)]$ as input and SOC as the response variable to be predicted. Since neural networks require large amounts of data to be trained, they are collected in smaller fixed-size mini-batches to shorten the training time of the network, and each of these batches is then fed into the LSTM network [14]. A forward pass begins when the training data (all batches) are fed into the network, and ends when the SOC estimates are generated at each time step k. Each forward pass is followed by a backward pass where the network weights and biases are updated: this cycle is referred to as epoch, and is denoted by ϵ.

Training a very large neural network can be very slow. Beyond the mini-batches, one way to speed up training is to use a faster optimizer than gradient descent [14]. The most commonly used one [8][9][7][10] is adaptive moment estimation (Adam): it keeps track of an exponentially decaying average of past gradients and also an exponentially decaying average of past quadratic gradients of the loss function, when weights and biases are updated. Once the network has computed all the hidden states h_k of the last epoch, the estimated SOC is obtained by a fully (dense) connected layer:

$$SOC_k = V_y h_k + b_y$$

where V_y is the weights matrix and b_y is the bias corresponding to the last fully connected layer. A loss function is needed to measure the prediction's accuracy at each time step. Mean absolute error (MAE) is chosen because of its simple structure and easy calculations [14]:

$$MAE = \frac{1}{N}\sum_{k=0}^{N}|SOC_k - SOC_k^*|$$

where N is the length of the sequence and SOC_k and SOC_K^* are the estimated and true values of the battery's state of charge.

3.3.2. Evaluation

The validation of the algorithm is done using the test set consisting of the three previously mentioned driving cycles and their respective charges. Only one forward run is required here since all parameters have already been learned during training. The measures used to validate the results obtained are again MAE and the root mean squared error [14]:

$$RMSE = \sqrt{\sum_{k=0}^{N}\frac{1}{N}\left(SOC_k - SOC_k^*\right)^2}.$$

3.3.3. Prediction

SOC prediction is performed again with the test set. We do not show prediction results here because the built-in model is not properly optimized, and this task is outside the scope of our work.

4. Preliminary experimental results

As mentioned above, the vector of collected inputs is defined as $\Psi_k = [V(k), I(k), T(k)]$, where $V(k), I(k), T(k)$ are the voltage, current, and temperature measurements of the battery at time step k, respectively. The train set consists of the eight mix and the corresponding charges, and the test set consists of the UDDS, LA92, and US06 drive cycles and the corresponding charges. After performing all the preprocessing steps explained in Section 3.2, both the training and test set are split into batches to better feed the network.

The model is implemented in Python 3.10.6 version with Tensorflow 2.9.1 package: it is built sequentially, starting with the input layer, then an LSTM layer, and finally, a dense layer that computes the output. The parameters of the network, summarized in Figure 4, were kept constant in all experiments in order not to influence the results:

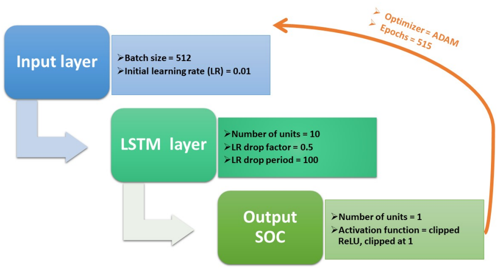

Figure 4: Network structure with respective parameters.

The following results are obtained by considering two different sampling frequencies: 1Hz (one observation per second) and $\frac{1}{60}$Hz (one observation per minute). Since we consider two different data granularities, aggregate mean and standard deviation of voltage and current were calculated to avoid information loss when down-sampling the data set: thus, the vector of inputs fed into the network is

$$\Psi_k = [V(k), I(k), T(k), MeanV(k), MeanI(k), StdV(k), StdI(k)].$$

4.1. Sampling to 1Hz

Recalling the different frequencies of stored data of drive cycles (and mix) and charges, to obtain train and test sampled to 1Hz we down-sample driving cycles from 10 to 1Hz and up-sample charges from $\frac{1}{60}$Hz to 1Hz. The results obtained are presented in Table (a): the training errors with a time horizon of 0, 10 minutes and 20 minutes are quite low; on the other hand, when the time horizon is higher and we predict SOC within the next 30 minutes, the training errors doubled with respect to results obtained for real-time estimation (h = 0), and validation errors are quite high. This is why we need to up-sample: considering a dataset with observations every second is not optimal to predict with reasonable accuracy within a horizon of 30 minutes.

The results in Table (a) are also confirmed by evaluating the Pearson correlation coefficient between SOC values corresponding to different time intervals (Lags):

- Lag is of 1 second: 0.9939
- Lag is of 10 minutes: 0.9263
- Lag is of 20 minutes: 0.7968
- Lag is of 30 minutes: 0.5404

When two values of SOC have a difference of half an hour, the correlation coefficient drops significantly, confirming that this algorithm with fine-grained data cannot correctly predict SOC for 30 minutes or larger time horizons.

4.2. Sampling to $\frac{1}{60}$Hz

To obtain train and test sampled to 1Hz, driving cycles are down-sampled from 10 to $\frac{1}{60}$Hz. The collected results are shown in Table (b). When predicting real-time SOC (h = 0), both train and validation errors are lower than the ones in Table (a). When the time horizon is of 10 minutes, the training error is bigger, but the validation error is lower. By looking at results with a time horizon of 20 minutes, there are no significant differences between the prediction with the fine-grained and coarse-grained data set. The same happens when the time horizon considered is of 30 minutes.

By looking at the Pearson correlation coefficient between SOC values corresponding to different time intervals (Lags):

- Lag is of 1 minute: 0.9965
- Lag is of 10 minutes: 0.9692
- Lag is of 20 minutes: 0.8991
- Lag is of 30 minutes: 0.7938

The coefficients are more significant than the ones obtained when the frequency of input data is 1Hz, but for time horizons of 20 and 30 minutes, the coefficients are too small, confirming that to predict SOC for those time horizons, a more coarse-grained data set is needed.

The improvements obtained with real-time SOC estimation (h = 0) are due to the increasing information about the past given by the aggregation measures computed. When the time horizon is of 20 and 30 minutes the algorithm overfits when trying to predict with a fine-grained dataset: the training error is relatively low, but

	h	MAE	RMSE
Training errors	0min	0.027	0.050
	10min	0.026	0.049
	20min	0.039	0.070
	30min	0.065	0.117
Test errors	0min	0.139	0.304
	10min	0.131	0.296
	20min	0.145	0.292
	30min	0.151	0.267

(a) Table to represent errors obtained with different time horizons (h) with observations frequency of 1 Hz.

	h	MAE	RMSE
Training errors	0min	0.012	0.035
	10min	0.034	0.116
	20min	0.040	0.068
	30min	0.042	0.067
Test errors	0min	0.128	0.305
	10min	0.064	0.286
	20min	0.149	0.280
	30min	0.168	0.261

(b) Table to represent errors obtained with different time horizons (h) with observations frequency of $\frac{1}{60}$ Hz.

the validation error is very high; moreover, when we try to up-sample, dataset size reduces a lot, and this aspect can influence the network performance negatively so that when we validate the network, we cannot see improvements, such as the ones with a time horizon of 10 minutes.

5. Discussion

In the final analysis, this work shows how the input data's different granularity can affect the LSTM network's performance in predicting SOC within different time horizons: the more distant the considered horizon, the more we need to aggregate the data and collect them in a coarse-grained dataset. The results are promising, especially for a time horizon of 10 minutes. A very important aspect that should be considered for improvements of this work and future developments is the reduction of the size of the considered dataset: this can affect the performance of the estimation, since neural network-based algorithms always require a significant amount of data to derive proper models. The more distant the horizon, the larger the period of the collected data needs to be.

Thanks to a promising estimation, some business developments can be defined: the integration of anomaly detection techniques in the battery management system (BMS) based on future predictions of SOC, a predictive maintenance plan for the battery, an appropriate target for the second life of the battery to make the most of its remaining useful life; the implementation of a user interface that makes it easier for the owner of an electric vehicle to monitor the battery parameters.

Even though LSTM can self-learn its own parameters, the results obtained in this paper are promising and can be improved thanks to the optimization of LSTM parameters, which will be addressed in future work. Furthermore, it is essential to have a kind of regulation that guarantees the uniformity of electric vehicle battery data, and accessibility for analysts to train the algorithms to improve SOC estimation easily: most innovative ideas have been developed in the laboratory environment, so further improvements are needed in the area of real-time vehicle monitoring.

Acknowledgments

This work was partially supported by

- the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”).
- the grant “National Centre for HPC, Big Data and Quantum Computing,” CN000013 (approved under the M42C Call for Proposals - Investment 1.4 - Notice“National Centers” - D.D. No. 3138 of 16.12.2021, admitted for funding by MUR Decree No. 1031 of 17.06.2022).

References

[1] R. Zhang, B. Xia, B. Li, L. Cao, Y. Lai, W. Zheng, H. Wang, W. Wang, State of the art of lithium-ion battery soc estimation for electrical vehicles, Energies 11 (2018). URL: https://www.mdpi.com/1996-1073/11/7/1820. doi:10.3390/en11071820.

[2] Y. Wang, C. Liu, R. Pan, Z. Chen, Modeling and state-of-charge prediction of lithium-ion battery and ultracapacitor hybrids with a co-estimator, Energy 121 (2017) 739–750. doi:10.1016/j.energy.2017.01.044.

[3] M. A. Hannan, M. S. Lipu, A. Hussain, A. Mohamed, A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations, Renewable and Sustainable Energy Reviews 78 (2017) 834–854. URL: https://www.sciencedirect.com/science/article/abs/pii/S1364032117306275. doi:10.1016/J.RSER.2017.05.001.

[4] L. Wu, K. Liu, H. Pang, J. Jin, Online soc estimation based on simplified electrochemical model for lithium-ion batteries considering current bias†, Energies 14 (2021). doi:10.3390/EN14175265.

[5] C. Vidal, P. Malysz, P. Kollmeyer, A. Emadi, Machine learning applied to electrified vehicle battery state of charge and state of health estimation: State-of-the-art, 2020. doi:10.1109/ACCESS.2020.2980961.

[6] J. C. A. Anton, P. J. G. Nieto, C. B. Viejo, J. A. V. Vilan, Support vector machines used to estimate the battery state of charge, IEEE Transactions on Power Electronics 28 (2013) 5919–5926. doi:10.1109/TPEL.2013.2243918.

[7] C. Vidal, P. Malysz, M. Naguib, A. Emadi, P. J. Kollmeyer, Estimating battery state of charge using recurrent and non-recurrent neural networks, Journal of Energy Storage 47 (2022). doi:10.1016/j.est.2021.103660.

[8] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, A. Emadi, Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries, IEEE Transactions on Industrial Electronics 65 (2018) 6730–6739. doi:10.1109/TIE.2017.2787586.

[9] C. Vidal, P. Kollmeyer, E. Chemali, A. Emadi, Li-ion battery state of charge estimation using long short-term memory recurrent neural network with transfer learning (2019).

[10] C. Bian, H. He, S. Yang, Stacked bidirectional long short-term memory networks for state-of-charge estimation of lithium-ion batteries, Energy 191 (2020). doi:10.1016/j.energy.2019.116538.

[11] G. dos Reis, C. Strange, M. Yadav, S. Li, Lithium-ion battery data and where to find it, 2021. doi:10.1016/j.egyai.2021.100081.

[12] P. Kollmeyer, C. Vidal, M. Naguib, M. Skells, Lg 18650hg2 li-ion battery data and example deep neural network xev soc estimator script 2 (2020). doi:10.17632/CP3473X7XV.2.

[13] S. Turney, Pearson correlation coefficient (r) | guide examples, 2022. URL: https://www.scribbr.com/statistics/pearson-correlation-coefficient/.

[14] B. Farnham, S. Tokyo, B. Boston, F. Sebastopol, T. Beijing, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd ed., 2019. (book).