

Article

Artificial Feature Extraction for Estimating State-of-Temperature in Lithium-Ion-Cells Using Various Long Short-Term Memory Architectures

Mike Kopp ^{*}, Marco Ströbel , Alexander Fill, Julia Pross-Brakhage and Kai Peter Birke 

Electrical Energy Storage Systems, Institute of Photovoltaics, University of Stuttgart, 70596 Stuttgart, Germany; marco.stroebel@ipv.uni-stuttgart.de (M.S.); alexander.fill@ipv.uni-stuttgart.de (A.F.); julia.pross-brakhage.2@ipv.uni-stuttgart.de (J.P.-B.); peter.birke@ipv.uni-stuttgart.de (K.P.B.)

* Correspondence: mike.kopp.2@ipv.uni-stuttgart.de

Abstract: The temperature in each cell of a battery system should be monitored to correctly track aging behavior and ensure safety requirements. To eliminate the need for additional hardware components, a software based prediction model is needed to track the temperature behavior. This study looks at machine learning algorithms that learn physical behavior of non-linear systems based on sample data. Here, it is shown how to improve the prediction accuracy using a new method called “artificial feature extraction” compared to classical time series approaches. We show its effectiveness on tracking the temperature behavior of a Li-ion cell with limited training data at one defined ambient temperature. A custom measuring system was created capable of tracking the cell temperature, by installing a temperature sensor into the cell wrap instead of attaching it to the cell housing. Additionally, a custom early stopping algorithm was developed to eliminate the need for further hyperparameters. This study manifests that artificially training sub models that extract features with high accuracy aids models in predicting more complex physical behavior. On average, the prediction accuracy has been improved by $\Delta T_{\text{cell}} = 0.01$ °C for the training data and by $\Delta T_{\text{cell}} = 0.007$ °C for the validation data compared to the base model. In the field of electrical energy storage systems, this could reduce costs, increase safety and improve knowledge about the aging progress in an individual cell to sort out for second life applications.

Keywords: machine learning; state estimation; lithium-ion-battery; thermal management; artificial feature extraction; battery management system



Citation: Kopp, M.; Ströbel, M.; Fill, A.; Pross-Brakhage, J.; Birke, K.P. Artificial Feature Extraction for Estimating State-of-Temperature in Lithium-Ion-Cells Using Various Long Short-Term Memory Architectures. *Batteries* **2022**, *8*, 36. <https://doi.org/10.3390/batteries8040036>

Academic Editor: Carlos Ziebert

Received: 22 February 2022

Accepted: 11 April 2022

Published: 15 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Motivation

Lithium-ion-battery technology is a promising energy storage technology for transitioning to electro-mobility due to its advantages, such as low self-discharge rate, long cycle life, scalability, high energy/power density, etc. To ensure safety requirements and longevity, cell parameters such as cell voltage U , cell current I and ideally cell temperature T_{cell} need to be monitored at all times [1–3]. From these parameters, other vital cell states, such as state-of-charge SoC and state-of-health SoH, can be derived. The performance, longevity, safety, and applicability for second-life-applications are strongly dependent on the accuracy of these parameters. Obtaining the temperature T_{cell} is especially challenging for two reasons. First, most battery systems work with temperature sensors that are attached to the cells' housing. Thus, not the heat generation in the cell core is monitored, but the heat dissipation from the cell interior to the environment [3,4]. Furthermore, the cell temperature varies in a battery system depending on the location of the cell in a module [5,6]. Therefore, a sensor-based temperature tracking of each individual cell is recommended, however, not feasible in many applications. On the other hand, there are various software based prediction techniques such as mathematical–physical models and

machine learning. However, physical models are highly computationally intensive [2,7], rely on information that is not available in real life applications [8–10] or is too specific to apply for every scenario that a battery is exposed to in real life applications. Some popular examples are finite element analysis, simulating the thermal field distribution by the Newman model [11], or the heat generation model by the Bernadi model [12]. In contrast, there are data-driven models, with their main advantages being transferability and universality [13]. Most data-driven models aim to make predictions about SoC [14] or SoH [15]. Temperature predictions using time series forecasts are rather rare and very specific. For example, Kleiner et al. [16] uses real life temperature data to make predictions for the future. The emphasis is thermal cause and thermal behavior over a cells lifetime in contrast to more complex real life situations that can have varying results [17–19].

This paper aims to compare artificial neural network (ANN) architectures to track thermal behavior of a single li-ion battery cell (LIBC) with a limited amount of training data and at a constant ambient temperature. This will show the impact of artificial feature extraction to improve the tracking capability of a physical non-linear systems.

2. Theoretical Foundations

This section will provide theoretical foundations regarding temperature behavior in LIBCs and the machine learning aspects used in this paper. First, it is shown how heat is generated and dissipated. This will give an insight of the complexity of dynamic non-linear systems such as LIBCs. Secondly, a short introduction to the machine learning aspects are given. The focus lays on recurrent neural networks to generate a time series forecast. It is important to notice, this section does not represent the scientific contributions gained by this paper, but shows the groundwork laid down by others.

2.1. Temperature Behavior in Lithium-Ion-Battery-Cells

The goal of this subsection is to gain an understanding of the complexity and investigate the necessary information to predict the temperature behavior in a LIBC. Therefore, we will look at the physical-thermal behavior of a single cylindrical LIBC. If we assume the internal temperature as well as the heat generation in a cell to be evenly distributed, we can derive from the law of energy conservation:

$$c_{\text{cell}} m_{\text{cell}} \frac{dT_{\text{cell}}}{dt} = Q_{\text{gen}} + Q_{\text{dis}}, \quad (1)$$

with c_{cell} representing the specific heat capacity, m_{cell} the mass of the LIBC and $\frac{dT_{\text{cell}}}{dt}$ the temperature change over time t [20]. Q_{gen} is the heat generation rate and Q_{dis} the heat dissipation rate. Equation (1) assumes that every additional heat source is infinitely far from the LIBC. In other words, the ambient temperature T_{amb} is uniform.

2.1.1. Heat Generation

The heat generated inside a LIBC can be further subcategorized into reversible heat \dot{Q}_r , ohmic heat \dot{Q}_o , heat from material phase changes \dot{Q}_p and heat produced by the enthalpy absorbed or liberated from substances mixing \dot{Q}_m [3,20,21]. To simplify the model for heat generation, we will neglect voluminous expansion during charge and discharge cycles, assume that the active material is uniformly distributed and the diffusion coefficient of li-ion in solid particles is assumed as constant.

Reversible heat \dot{Q}_r is the energy needed to maintain the energy balance of the reaction caused by the gradient between the Gibbs free energy of product and reactant. Suppose the layer-by-layer cell structure is summarized as one block, the reversible heat \dot{Q}_r can be calculated by

$$\dot{Q}_r = IT_{\text{cell}} \frac{\partial U_{\text{OCV}}}{\partial T_{\text{cell}}}, \quad (2)$$

with U_{OCV} representing the open circuit voltage (OCV) potential [7,21,22]. U_{OCV} is a function of the State-of-Charge (SoC) without the effects of overvoltage taken into account. Ohmic heat \dot{Q}_o is generated by collisions of charged particles in the solid and electrolyte phase and can be simplified by

$$\dot{Q}_o = I(U - U_{OCV}), \quad (3)$$

with $(U - U_{OCV})$ filtering the overvoltage [21]. Thus, for \dot{Q}_o , only the potential caused by the current load is analyzed. The heat from material phase changes \dot{Q}_p can be expressed by

$$\dot{Q}_p = - \sum_i \Delta H_i^{avg} r_i, \quad (4)$$

and generally represents the heat either produced or consumed by the enthalpy change ΔH_i^{avg} of a chemical reaction i multiplied by the rate r_i at which the reaction i takes place [21,23,24]. Heat produced from mixing is calculated by

$$\dot{Q}_m = - \int \sum_i (\bar{H}_i - \bar{H}_i^{avg}) \frac{\partial c_i}{\partial t} dv, \quad (5)$$

and is caused by the enthalpy \bar{H}_i absorbed or liberated by two or more substances mixing [24]. To isolate \bar{H}_i , the average concentration of the enthalpy \bar{H}_i^{avg} of i has to be subtracted and then multiplied by the concentration change over time $\partial c_i / \partial t$. In a LIBC, this represents the formation and relaxation of concentration gradients within the LIBC and has to be integrated over its volume v . To obtain the total heat generated by a LIBC \dot{Q}_{gen} , all subcategories have to be summed up. However, it is worth knowing, that the impact of \dot{Q}_o and \dot{Q}_r far exceeds the impact of \dot{Q}_m and \dot{Q}_p .

2.1.2. Heat Dissipation

With the aim of gaining a full understanding of the thermal behavior of a LIBC, not only heat generation but also, heat dissipation Q_{dis} needs to be taken into account. Q_{dis} describes the heat escaping from the LIBC and is caused by three main phenomena: Radiation heat \dot{Q}_{ra} , convection heat \dot{Q}_{cv} and conduction heat \dot{Q}_{cd} .

Radiation heat \dot{Q}_{ra} describes the energy transport from its surface to the surrounding atmosphere and can be calculated by Stefan–Boltzmann law with

$$\dot{Q}_{ra} = \varepsilon \sigma A (T_{out}^4 - T_{amb}^4), \quad (6)$$

where ε represents emissivity (dependent on the materials and colors used), σ is the Stefan–Boltzmann constant, A is the surface area of the LIBC, T_{out} is the surface temperature and T_{amb} the ambient temperature [25]. Heat convection \dot{Q}_{cv} can be further subcategorized into forced and natural heat convection. In this study, we will only look at the natural \dot{Q}_{cv} . A simplified approximation is given by Jossen [26] with

$$\dot{Q}_{cv} = A c_{K,n} (T_{out} - T_{amb}), \quad (7)$$

with $c_{K,n}$ representing the constant of natural convection [25]. Heat conduction \dot{Q}_{cd} is caused by mechanical contact from the LIBC surface area A to other materials as well as the electrical wiring of the terminals. With Fourier's law, we can derive

$$\dot{Q}_{cd} = kA \frac{dT_{out}}{dx}, \quad (8)$$

as a one-dimensional representation x with the thermal conductivity between the two materials.

To summarize, aside from material constants and dimensional details, we would need information about the OCV, enthalpy behavior and concentration changes of several reactions and temperature data. In practical applications, this information is not accessible.

This study aims to predict the temperature behavior with only the current profile I and its corresponding voltage response U without the need for solving differential equations.

2.2. Time Series Forecast

This section gives an overview of the machine learning aspects needed to follow along with this paper. ANNs are a subset of machine learning and at the heart of deep learning applications. A basic ANN is built with node layers, consisting of an input layer, a number of hidden layers, and an output layer [27]. Each layer has one or more neurons with individually associated weights and activation function. A training data set consists of labeled data points and is needed to train the weights [28]. During a supervised-learning-training process, the input layer provides the ANN with new information. This information is passed onto the first hidden layer, where it is forwarded to the following hidden layers. If all the information from one layer is passed onto each neuron from the next layer, it is called a fully connected layer [29]. The information passing through the neurons is altered by the weights. Once the information arrives at the output layer, it is compared with the label and an error value τ_{loss} is calculated via the mean squared error method. After a number of data points pass through the ANN (batch) a backpropagation algorithm corrects the weights of each neuron so that τ_{loss} is minimized [30]. This process is repeated for a number of iterations (epochs). For a time series forecast, it is not only important to know the information provided by the input layer, but it is crucial to know several time steps leading up to the information (window size h_{win}) and the previously predicted value. This is made possible by a recurrent neural network (RNN). A special kind of RNN is an architecture made out of long short-term memory (LSTM) cells [31]. This is a four layered ANN compromised into one neuron that can learn to filter information based on its importance [32]. To ensure that the ANN is learning the nature behind the predicting values and not fitting to the training data, a validation data set is needed as well. With the validation data set, an additional validation error $\tau_{\text{val,loss}}$ is calculated. If the behavior of τ_{loss} and $\tau_{\text{val,loss}}$ drift into different directions during the training process, overfitting occurs.

3. Experimental

3.1. Method of Measurement

The LIBC used in this study is a high power LIBC (INR18650-15L1) from Samsung. According to the manufacture, the LIBC can be cycled in a range of $2.5 \text{ V} \leq U \leq 4.5 \text{ V}$ and has a nominal discharge capacity of $C_N = 1.5 \text{ Ah}$. To avoid an offset from heat generation Q_{gen} to the actual measurement, a negative temperature coefficient (NTC)-sensor (NTC Thermistor 10k Perle SC30F103V by Amphenol) R_{sens} is placed in the center of the cell.

As seen in Figure 1 cylindrical cells come with a production-related cavity in the center of the cell wrap, where the sensor is located. To access the cavity, a through-hole was drilled in the center of the negative terminal. After positioning the sensor, the through-hole was sealed with a two-components-epoxide resin glue. This process took place in a glovebox filled with an argon atmosphere.

An electrochemical impedance spectroscopy (EIS) in the range of $0.01 \text{ Hz} \leq f \leq 10,000 \text{ Hz}$ had been carried out before and after the procedure. Figure 2 shows the results in a Nyquist plot with a negative $\text{Im}(Z)$ -axis. The graph marked as "Pre" represents the data measured before the LIBC was equipped with the sensor and the graph marked as "Post", the data after the sensor was inserted. Two regions are highlighted by a zoom window. The zoom window on the right focuses on the zero crossing of $\text{Im}(Z) = 0$. This is where the ohmic internal resistance R_i of the LIBC can be determined. The resistance after the procedure $R_{i,\text{post}} = 0.02240 \Omega$ has only a minor increase compared to before the procedure $R_{i,\text{pre}} = 0.02237 \Omega$. The zoom window on the right shows the other side of the semi-circle. The two high points (a) and (b) are $(0.03078 \Omega | -0.00080 \Omega)$ and $(0.03174 \Omega | -0.00082 \Omega)$, respectively. According to H. Ruan, B. Sun, and J. Jiang et al. [33], the accuracy in the low-frequency range can be drastically improved with the modified-electrochemical impedance

spectroscopy-based multi-time-scale fractional-order model. However, here the focus lies on the two marked regions, rather than on the diffusion dynamics. In addition, Figure 2 is meant to compare relative changes instead of absolute values.

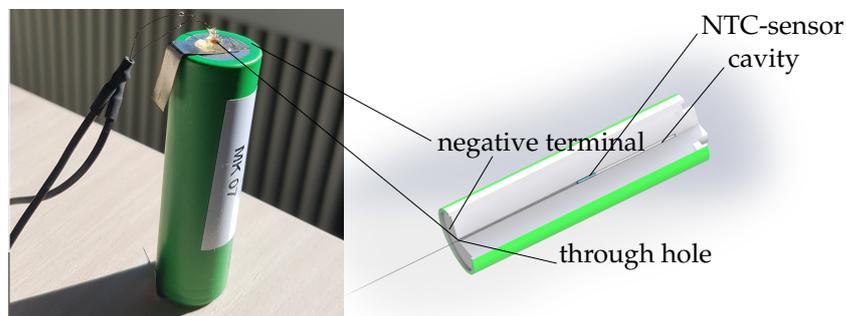


Figure 1. LIBC equipped with temperature sensor. On the left is a photograph of the prepared cell. On the right is a sectional view of sensor placement in LIBC. The figure shows the production-related cavity with an NTC-sensor placed in the middle. The cavity is accessed by a through-hole in the center of the negative terminal.

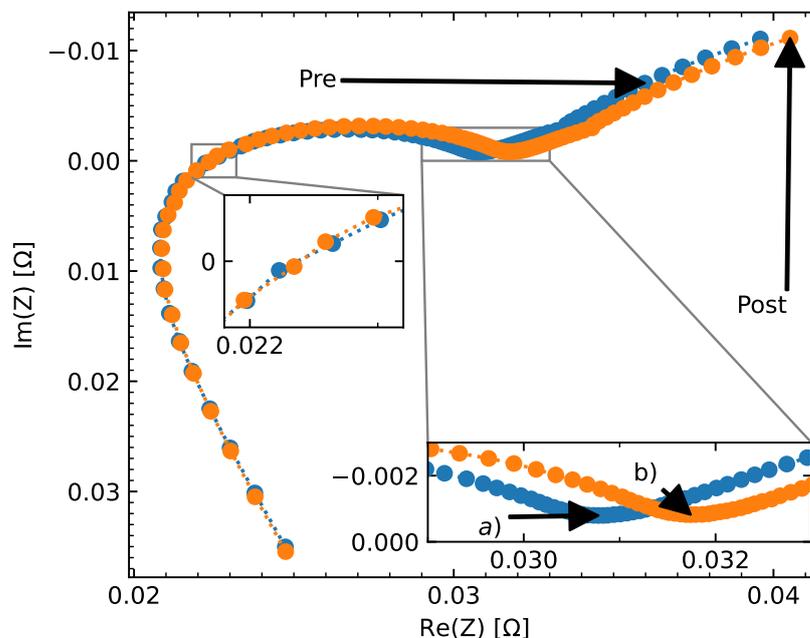


Figure 2. Nyquist plot of the EIS measurement carried out before (Pre) and after (Post) the LIBC was equipped with a temperature sensor. Point (a) and (b) compare the two high points of the first semi-circle.

To read and interpret the sensor values, a custom measuring system was designed. The resistance value from the sensor was measured in a voltage divider against a SL1TTE1002F by KOA Speer Electronics inc, and digitalized by an ADS1115 by Adafruit. The data is then processed with an Arduino Mega 2560 and logged onto a Micro-SD-Card. The measuring system is able to log roughly 4.5 data points per second. In post production, the digitalized data was calculated to temperature values according to the sensors manual. In order to get a sense of the noise produced, every data point is logged and averaged out. To account for contact resistance and other influences by the circuit, the sensors were validated at various ambient temperatures $T_{amb,c} = (283.15 \text{ K}, 298.15 \text{ K}, 313.26 \text{ K}, 333.15 \text{ K})$. The equation

$$R_{sens,x,new} = 1.027R_{sens,x} - R_{offset} \quad (9)$$

is capable to compensate the offset with $R_{\text{offset}} = 80.561 \Omega$. The cell was placed in the climate chamber IPP260plus by Memmert and kept at $T_{\text{amb}} = 298.15 \text{ K}$ for the entirety of the experiment.

3.2. Current Profile

Figure 3 shows the data used to train the models. First T_{cell} as measured as in Section 3.1 is presented. Derived from T_{cell} is the linear approximation of $dT_{\text{cell}}/dt(t_i) = (T_{\text{cell}}(t_i) - T_{\text{cell}}(t_{i-1})) / (t_i - t_{i-1})$ followed by the current profile I and its corresponding voltage response U . The profile can be subdivided into four sections. Each section is presented with a sample of an enlarged view of the temperature T_{cell} as well as the current I in Figure 4. First, the LIBC is strained with 21 full constant current ($C = 1 \text{ C}$) charge/discharge cycles with no rest in-between. An enlarged view is given in Figure 4a). This is followed by three full random constant current charge/discharge cycles in the range of $-4 \text{ A} \leq I \leq 4 \text{ A}$ and can be viewed in Figure 4b). Positive values represent charging- and negative values discharge-mode. To ensure that the temperature T_{cell} starts every cycle with $T_{\text{cell}} = T_{\text{amb}}$, there is an $I = 0 \text{ A}$ phase for $t = 1 \text{ h}$ in-between each charge/discharge cycle. After that nine different pulse cycles are implemented. The first four cycles are $t = 10 \text{ s}$ pulses, with $t = 10 \text{ s}$ rest in-between discharge cycles with random current values analogous to section (b) and the remaining cycles are charge-/discharge cycles for $t = 30 \text{ s}$ with no rest in between. One example can be viewed in Figure 4c). It is important to use full cycles in order to ensure the models ability to learn the temperature T_{cell} to SoC relationship. The last section is meant to increase the level of complexity. The LIBC starts with SoC = 50 %. From there, a random value in the range of $-4 \text{ A} \leq I \leq 4 \text{ A}$ is applied to the LIBC for a random time segment in the range of $5 \text{ s} \leq t \leq 40 \text{ s}$. A zoomed view is extracted in Figure 4d).

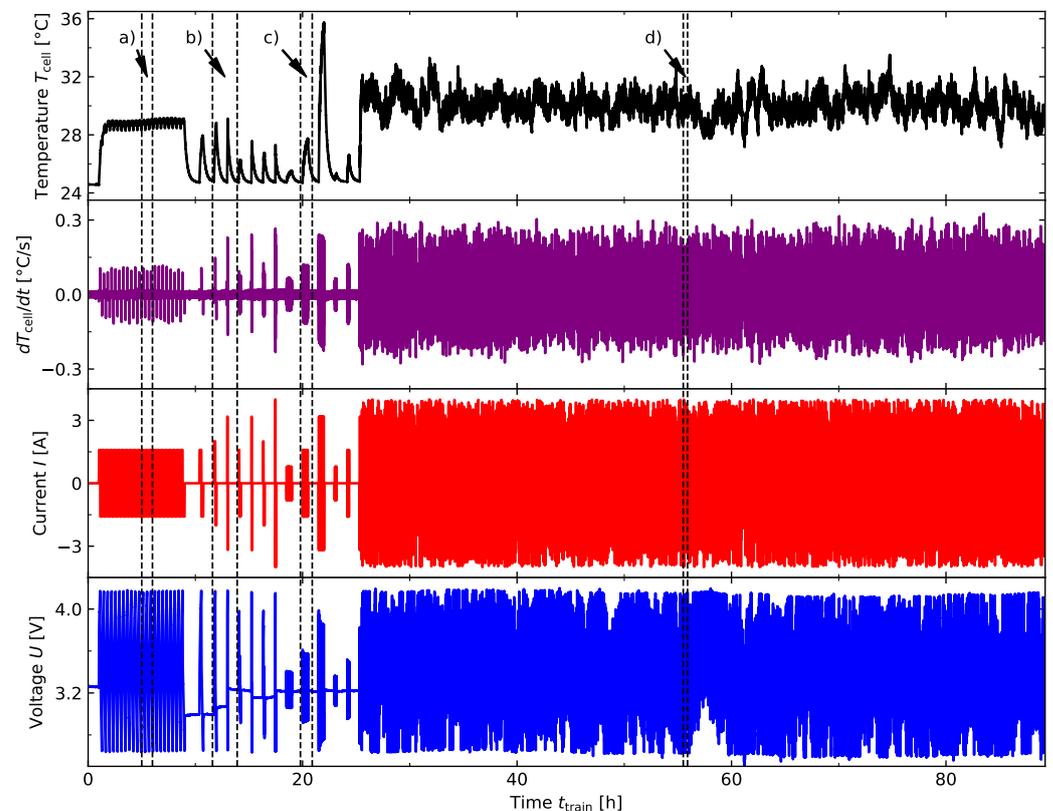


Figure 3. Selected data to train the algorithm. Presented are four subfigures showing temperature response $T_{\text{cell}}(t_{\text{train}})$, temperature change over time $dT_{\text{cell}}/dt(t_{\text{train}})$, current profile $I(t_{\text{train}})$ and voltage response $U(t_{\text{train}})$. Four areas are marked (a–d).

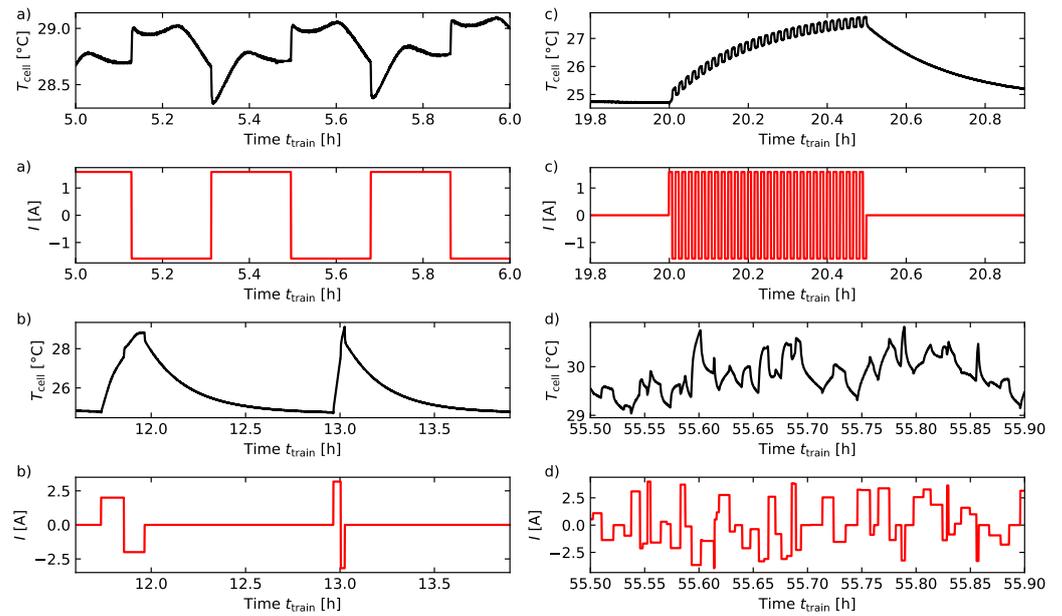


Figure 4. Enlarged view of the areas marked in Figure 3. Presented are temperature response $T_{cell}(t_{train})$ and current profile $I(t_{train})$ of each of the marked areas (a–d).

Figure 5 shows the data used to validate the models. Every scenario described for the training data is implemented here as well. However, it was ensured that the current values I differ from the training data set. Analogous to Figure 4, there is an enlarged view of all scenarios in Figure 6 showing T_{cell} as well as the current I .

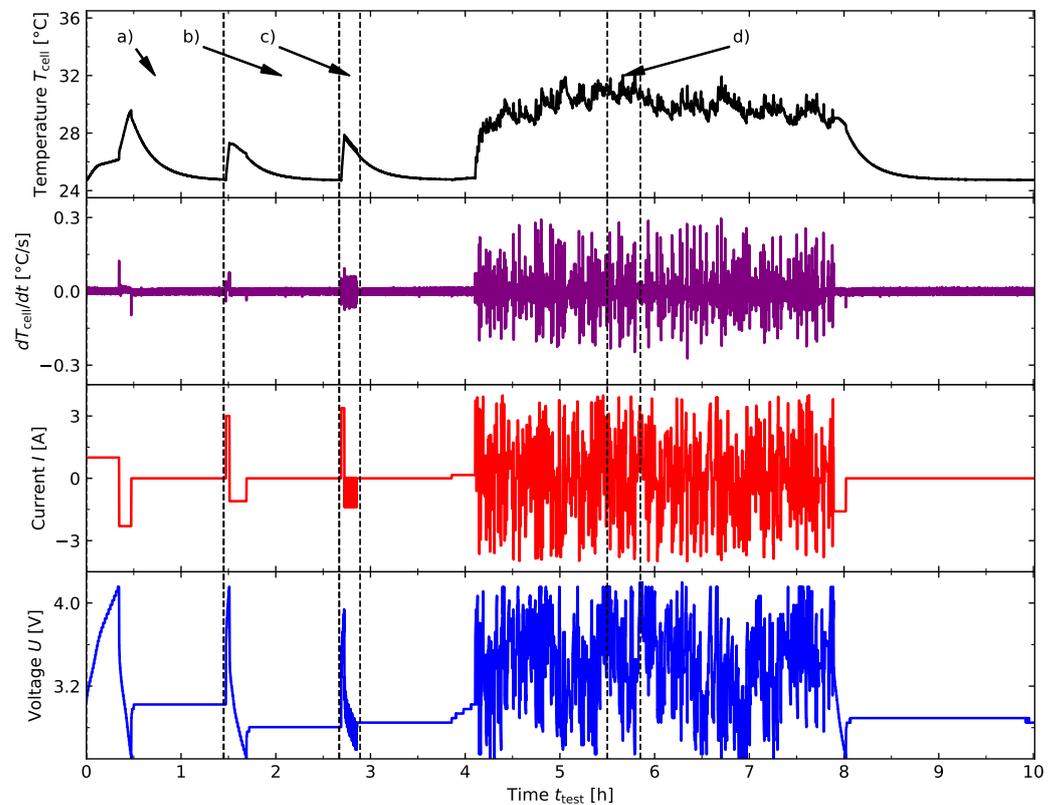


Figure 5. Selected data to test the algorithm. Presented are four subfigures showing temperature response $T_{cell}(t_{test})$, temperature change over time $dT_{cell}/dt(t_{test})$, current profile $I(t_{test})$ and voltage response $U(t_{test})$. Four areas are marked (a–d).

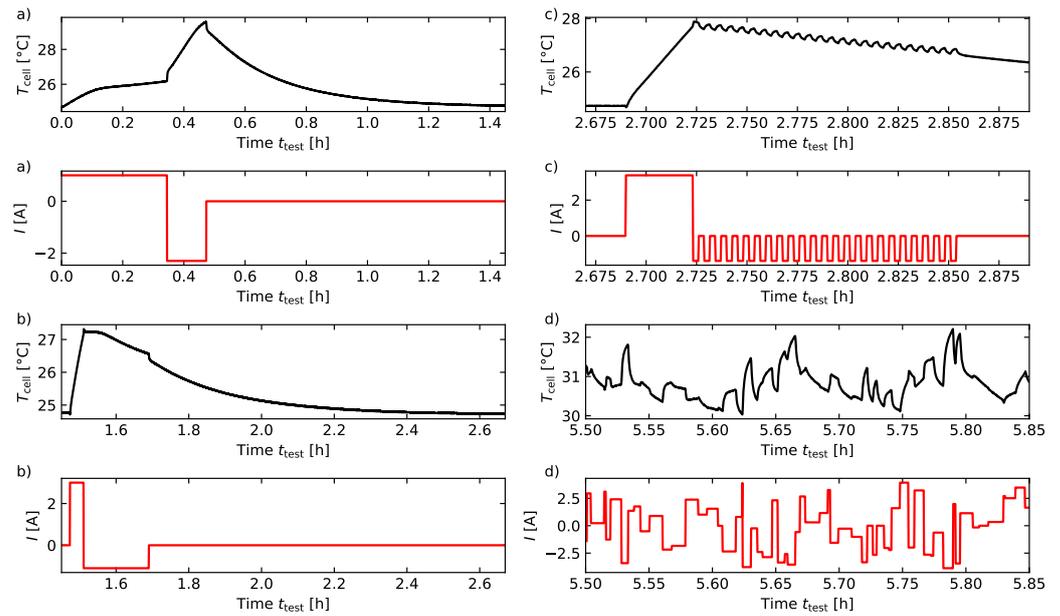


Figure 6. Enlarged view of the areas marked in Figure 5. Presented are temperature response $T_{\text{cell}}(t_{\text{test}})$ and current profile $I(t_{\text{test}})$ of each of the marked areas (a–d).

3.3. Data Preparation

The LIBC was cycled by the PEC ACT0550. In order to merge the temperature data with the cell data, the time stamps (t_{pec} for PEC and t_{temp} for the measuring system) of the two systems have to be synchronized. Analyzing the data, leads to the equation

$$t_{\text{pec}} = 1.0009t_{\text{temp}} - t_{\text{offset}} \quad (10)$$

with the intercept $t_{\text{offset}} = 2.5011$ s representing the offset in which the two systems measuring state were started. The cell is placed in a climate chamber as described in Section 3.1.

Finally, the data was normalized. Since I -data consists of positive as well as negative values, the scaling range is $-1 \leq I \leq 1$. The scaling range for U and T are $0 \leq U \leq 1$ and $0 \leq T \leq 1$, respectively. This impacts the resolution by 50 %; however, during the research conducted, it was revealed, that the ANN benefits more from learning the importance of the value zero.

4. ANN-Architectures

Every sub-model is constructed the same way as a function of:

- Input block (red) consists of the input nodes (U, I) as described in Section 2.1.
- Hidden block (green) is a function of the number of hidden layers h_{lay} and the number of neurons h_{neu} . Each neuron is a LSTM neuron and each layer is fully connected as described in Section 2.2.
- dT block (purple) is a fully trained LSTM model built as in Figure 7—Model-dT. It is structured like Model A; however, it was trained to predict the linear approximation dT_{cell}/dt as calculated in Section 3.2.
- Output block (blue) is a single LSTM neuron to bundle all the information coming from the hidden block.

The minimal time span necessary to predict a temperature value is depended on the window size h_{win} . This means if $h_{\text{win}} = x$ the ANN will be able to make a prediction of $T(x)$ based on the input values of $I(x - h_{\text{win}})$ to $I(x)$ and $U(x - h_{\text{win}})$ to $U(x)$. In this study, three different model architectures are being investigated. The schematic of each model is represented in Figure 7. Model-A will function as the base model consisting of input-,

hidden- and output blocks. Model-B has the same structure as Model-A except in-between the input block and hidden block is a fully trained dT block (Model-dT). Therefore, Model-B takes advantage of a form of artificial feature extraction, since the feature $dT_{cell}/dt(t)$ is not a product of how a model was trained, but has been trained in a controlled environment to produce $dT_{cell}/dt(t)$. Model-C includes the dT block as well. However, the hidden block is being supplied with the information from the input block as well as the dT block.

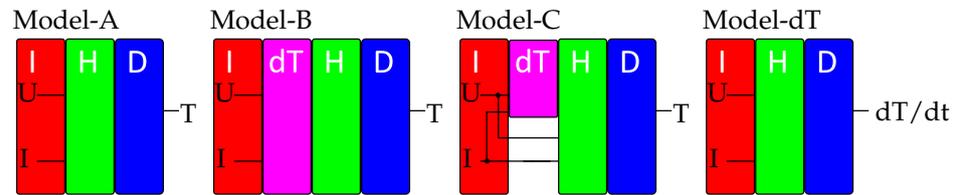


Figure 7. Layer schematic of different models. I-block (red) is the input layer, H-block (green) the hidden layer, D-block (blue) the output layer and dT-block (purple) summarizes the pre trained Model-dT.

Figure 8 illustrates an example of a model structure like Model-A or Model-dT. White neurons represent input neurons and yellow neurons represent LSTM neurons. Each neuron is fully connected to the upcoming layer. The background colors are analogous to Figure 7. This example uses three neurons for two layers in the hidden layer. The input and output structure is the same across all models. Model-B and Model-C have a fully trained dT-Model in-between input and hidden layer as described.

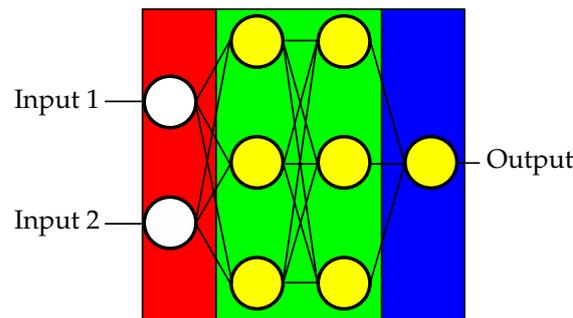


Figure 8. Example of a single model like Model-A or Model-dT with two inputs (Input 1 and Input 2) and one output (Output). White neurons represent input neurons and yellow neurons represent LSTM neurons.

Five hyperparameters were investigated:

- Window size h_{win} : This is the number of data points the ANN will see at each time step. If the algorithm is supposed to estimate a temperature value for $T(100)$ and $h_{win} = 10$, then the ANN would be fed $T(89) - T(99)$. In order to make all architectures comparable, h_{win} will only be dynamic at the base model (Model-A). For Model-B, -C and -dT, h_{win} will be static. In case a value $T(x)$ needs to be estimated with a window size $h_{win} = y$ with $y > x$, the data padding approach will be used [34]. This means, that the values will be automatically filled in to create an array with the appropriate dimensions.
- Number of neurons h_{neu} : This value represents the total number of neurons in the hidden block.
- Number of hidden layers h_{lay} : This is representative of the number of hidden layers in the hidden block. The number of neurons h_{npl} of each individual layer l can be calculated by

$$h_{npl}(l) = \begin{cases} \left\lfloor \frac{h_{neu}}{h_{lay}} \right\rfloor, & \left\lfloor \frac{h_{neu}}{h_{lay}} \right\rfloor - \frac{h_{neu}}{h_{lay}} < \frac{l}{h_{lay}} \\ \left\lfloor \frac{h_{neu}}{h_{lay}} \right\rfloor + 1, & \left\lfloor \frac{h_{neu}}{h_{lay}} \right\rfloor - \frac{h_{neu}}{h_{lay}} \geq \frac{l}{h_{lay}} \end{cases}$$

with $h_{\text{neu}} \geq h_{\text{lay}}$ and $h_{\text{lay}} \geq l$

- Learning rate: During the backpropagation process, the weights are being adjusted according to Section 2.2. Learning rate h_{lear} is the order of magnitude by which the weights are adjusted. A strong h_{lear} is unlikely to find the optimal solution while a small h_{lear} will make it challenging to reach a conclusion in a reasonable time frame. This is especially important when using an early stopping approach. In addition, this study uses an ADAM (derived from adaptive moment estimation) optimization algorithm to dynamically change the h_{lear} during the training process [35].
- Drop rate h_{drop} : This parameter is meant to counteract overfitting by stochastically taking weights out of the equation. If we assume $h_{\text{drop}} = 0.1$, this would mean every neuron has a 10 % chance of being bypassed.

To bypass the need for the number of training epochs n_{ep} being yet another hyperparameter, a custom early-stopping approach has been integrated. If the training process is running for $n_{\text{ep}} > 10$, the training will stop if either $\tau_{\text{loss}}(n_{\text{ep}}) < \tau_{\text{loss}}(1)$ or if the slope a_{reg} of

$$a_{\text{reg}} = \overline{\tau_{\text{loss}}(n_{\text{ep}})} - \overline{n_{\text{ep}}} \frac{\sum_{i=n_{\text{ep}}-10}^{n_{\text{ep}}} (n_{\text{ep},i} - \overline{n_{\text{ep}}})(\tau_{\text{loss}}(n_{\text{ep},i}) - \overline{\tau_{\text{loss}}(n_{\text{ep}})})}{\sum_{i=n_{\text{ep}}-10}^{n_{\text{ep}}} (n_{\text{ep},i} - \overline{n_{\text{ep}}})^2}, \quad (11)$$

with

$$\overline{n_{\text{ep}}} = \frac{1}{10} \sum_{i=n_{\text{ep}}-10}^{n_{\text{ep}}} n_{\text{ep},i}, \quad (12)$$

and

$$\overline{\tau_{\text{loss}}(n_{\text{ep}})} = \frac{1}{10} \sum_{i=n_{\text{ep}}-10}^{n_{\text{ep}}} \tau_{\text{loss}}(n_{\text{ep},i}), \quad (13)$$

is positive.

To evaluate the models with individual hyperparameters, a constrained global optimization package built upon Bayesian inference and the Gaussian process were used. It attempts to find the maximum value of an unknown high cost function in as many few iterations as possible by adjusting the aforementioned hyperparameters [36]. It starts with a random grid search with $n_{\text{grid}} = 10$ epochs. From there, the algorithm picks the best result as a starting point for the Bayesian optimization that runs for a maximum of $n_{\text{bays}} = 100$ epochs. Because of the stochastic nature of ANN every iteration went through the training process twice and the arithmetic mean of the two error rates was calculated.

5. Results and Discussion

In this section, the results of all the models will be presented, compared and discussed. Table 1 shows the best results achieved in this experiment and is structured in the “Bayesian” and “Individual” areas. All models are trained to reduce τ_{train} , and the hyperparameter h_{win} is set static after the hyperparameter investigation of Model-A is completed as described in Section 3.

The area “Bayesian” is the arithmetic mean of the two results for the training loss τ_{train} with the corresponding hyperparameters. Model-C was able to achieve the best results. To reduce the impact of the arithmetic mean, every individual model trained in “Bayesian” was investigated for “Individual”. Again, Model-C outperformed Model-A and Model-B.

The results shown in Table 1 are illustrated in Figure 9 for the training data set and Figure 10 for the validation data set.

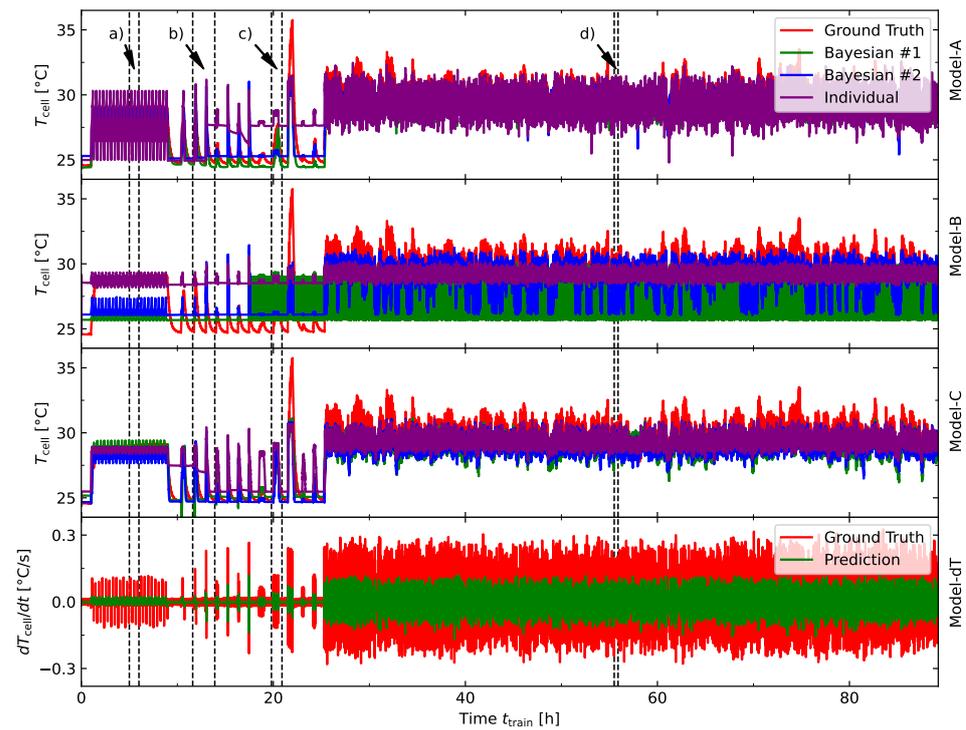


Figure 9. Predicted values of all models in four subfigures are given for the training data set. Model-A to Model-C show ground truth, Bayesian #1, Bayesian #2 and the most promising individual results. Model-dT illustrates ground truth and the predicted values of the dT-Model. Four areas are marked from (a–d).

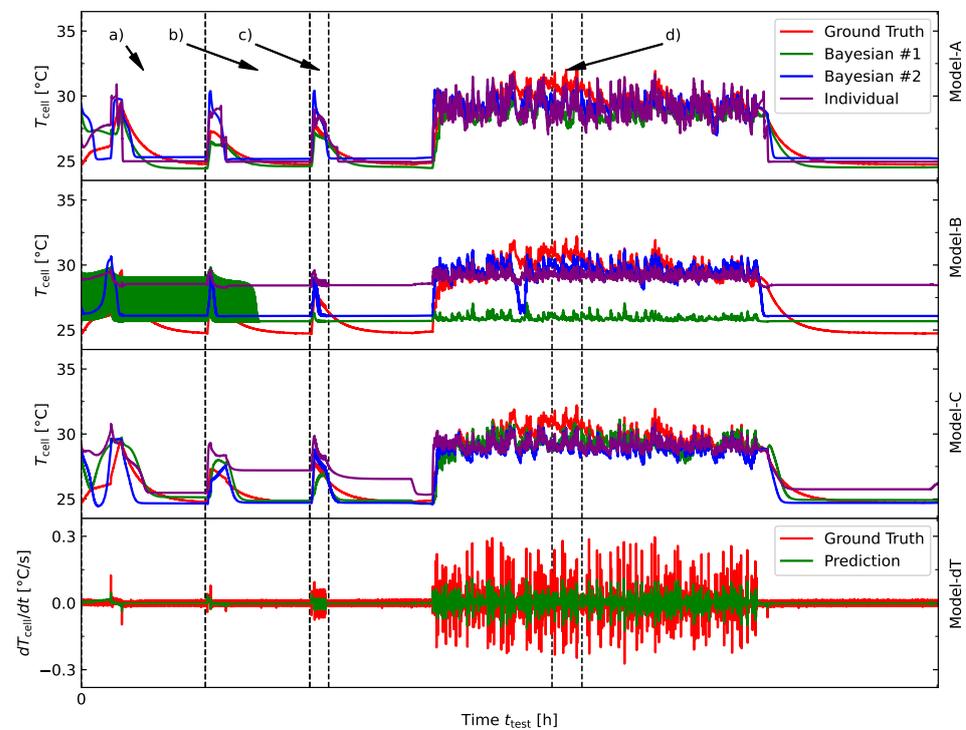


Figure 10. Predicted values of all models in four subfigures are given for the testing data set. Model-A to Model-C show ground truth, Bayesian #1, Bayesian #2 and the most promising individual results. Model-dT illustrates ground truth and the predicted values of the dT-Model. Four areas are marked from (a–d).

Table 1. Hyperparameter results. The table shows the hyperparameter configuration for the hyperparameters h_{win} , h_{lay} , h_{neu} , h_{drop} and h_{learn} for each model. The “Bayesian” area shows the arithmetic mean of two results with the same hyperparameter configuration and “Individual” area shows the results for each individual model.

| | Model | τ_{train} | h_{win} | h_{lay} | h_{neu} | h_{drop} | h_{learn} |
|------------|-------|-----------------------|-----------|-----------|-----------|------------|-----------------------|
| Bayesian | A | 9.02×10^{-3} | 8 | 2 | 5 | 0.0670 | 4.23×10^{-6} |
| | B | 5.48×10^{-2} | 8 | 1 | 5 | 0.0818 | 1.72×10^{-6} |
| | C | 7.44×10^{-3} | 8 | 1 | 5 | 0.0876 | 8.95×10^{-6} |
| Individual | A | 2.05×10^{-3} | 10 | 1 | 28 | 0.0419 | 6.88×10^{-6} |
| | B | 5.73×10^{-3} | 8 | 1 | 24 | 0.0687 | 8.36×10^{-6} |
| | C | 1.04×10^{-3} | 8 | 1 | 82 | 0.0373 | 2.12×10^{-5} |

Both figures show the predicted temperature values for the two models trained in the area “Bayesian” (“Bayesian #1” and “Bayesian #2”) and the overall best result in the area “Individual” for all models (Model-A, Model-B, Model-C and Model-dT) in contrast to the measured values (Ground Truth). In addition, each figure is subdivided into four areas (a)–(d).

These areas provide four enlarged views for a more detailed analysis on the training data in Figure 11 and validation data in Figure 12. The time span and position of all areas are identical to the areas in Figures 3 and 5, respectively.

The success rate of the Model-dT is higher than predicting the absolute value. Calculating the linear approximation dT/dt as shown in Section 3.2 will amplify the noise produced by T_{cell} . Figures 9 and 10 show, that the ANN successfully performs a noise reduction and therefore, compresses the upper and lower limits. However, it filters out the information when heat is generated Q_{gen} ($dT_{cell}/dt(t_i) > 0$) and when heat is dissipated Q_{dis} ($dT_{cell}/dt(t_i) < 0$). Thus, it aids Model-C to produce absolute values, which is to be taken as the key message of this research. Model-dT by itself is not sufficient to extrapolate the absolute values since every error is being summed up, resulting in a temperature drift which is not suitable for real life applications.

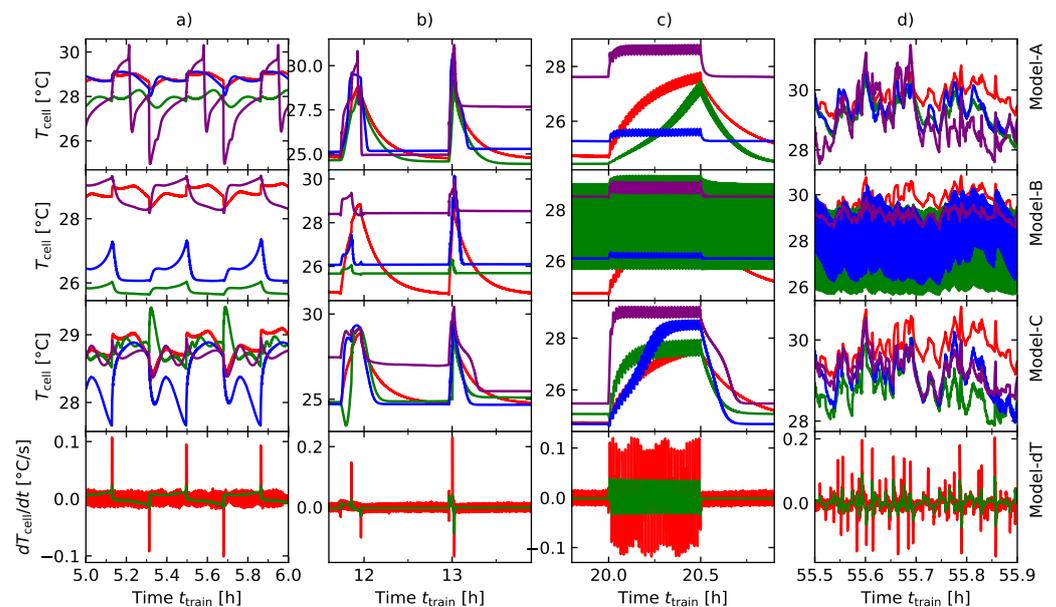


Figure 11. Predicted values of all models in four subfigures are given for the training data set. Model-A to Model-C show ground truth, Bayesian #1, Bayesian #2 and the most promising individual results. Model-dT illustrates ground truth and the predicted values of the dT-Model. Four areas are marked from (a–d).

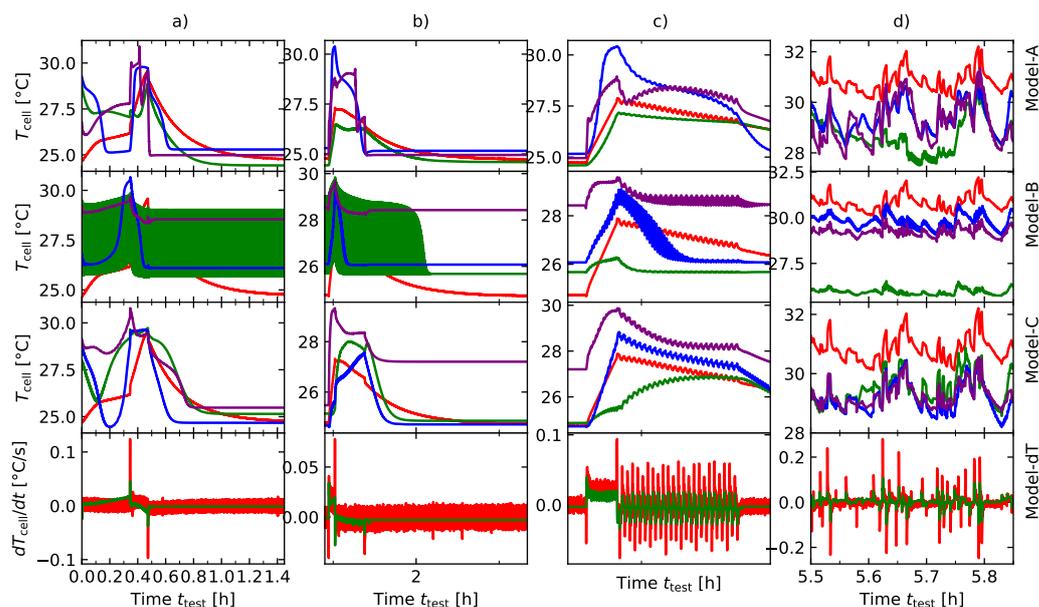


Figure 12. Predicted values of all models in four subfigures are given for the testing data set. Model-A to Model-C show ground truth, Bayesian #1, Bayesian #2 and the most promising individual results. Model-dT illustrates ground truth and the predicted values of the dT-Model. Four areas are marked from (a–d).

The predictions in Figure 9 show similar performance results to the predictions in Figure 10. Therefore, no signs of overfitting occurred.

The models do not aim to simply classify the sections in Section 3.2 and predict a mean value. Analyzing the enlarged sections (Figures 11a–d and 12a–d) shows the models' ability to learn the physical behavior of the LIBC. The model knows when T_{cell} increases/decreases depending on the previous time step, and the active current load and voltage response that had been given in h_{win} . Therefore, the model understands when the effects of Q_{gen} or Q_{dis} dominate.

Due to the limited amount of training data, not every situation gets a sufficient amount of attention. This is most notably when the LIBC heats up from T_{amb} or cools down to T_{amb} .

The “Individual” predictions perform visually the poorest of all graphs. This shows the importance of having more than one ANN trained with the same hyperparameter settings to reduce the effects of the stochastic nature.

The overall investigated temperature range is $24.53 \text{ °C} \pm 0.01 \text{ °C} \leq T_{\text{cell}} \leq 35.76 \text{ °C} \pm 0.01 \text{ °C}$. In this spectrum, Model-A achieved an accuracy of $\Delta T_{\text{cell}} = 0.023 \text{ °C}$ during training and $\Delta T_{\text{cell}} = 0.015 \text{ °C}$ during validation, Model-B $\Delta T_{\text{cell}} = 0.064 \text{ °C}$ during training and $\Delta T_{\text{cell}} = 0.043 \text{ °C}$ during validation and Model-C $\Delta T_{\text{cell}} = 0.012 \text{ °C}$ during training and $\Delta T_{\text{cell}} = 0.008 \text{ °C}$ during validation.

Model-B shows the poorest performance. This is most likely due to the dimensionality reduction. Where the input layer of Model-A has a dimension of two (U, I) and Model-C a dimension of three ($U, I, dT_{\text{cell}}/dt(t_i)$), the hidden block in Model-B is only provided one dimension ($dT_{\text{cell}}/dt(t_i)$). Model-A and Model-C both show that the physical temperature behavior of a cell was learned. This is best illustrated in section (d) of Figures 11 and 12. The LIBC can correctly predict the trend when to cool down and when to heat up based on the current inputs (U, I) and the previous cell states. Model-C outperformed Model-A and Model-B.

6. Conclusions and Future Work

The impact of artificial feature extraction using the first derivative in combination with ANN has been investigated. In this regard, the following steps have been taken to show its effectiveness:

- A LIBC has been prepared with an internal NTC-temperature sensor with the aim to prevent a time delay from heat generation to heat dissipation.
- A custom measuring system was designed to track the temperature and to synchronize the temperature data with the data of the battery system.
- Using this approach, a training- and validation dataset was created to investigate three LSTM-architectures. A hyperparameter analysis for each model has been carried out to find the optimal model structure for each sub model. Model-A architecture is the base model, Model-B architecture uses an additional dT-layer, that has been separately trained to forecast the linear approximation of $dT_{\text{cell}}/dt(t_i)$ and the third Model-C benefits from both approaches (Model-A and Model-B). Model-C was able to outperform Model-A and Model-B, which shows that artificial feature extraction is a useful method to improve model accuracy in the non-linear state of temperature prediction in LIBCs. This method made it possible to increase the accuracy by $\Delta T_{\text{cell}} = 0.01$ °C for the training data and by $\Delta T_{\text{cell}} = 0.007$ °C for the validation data compared to the base model with only the information of the current profile I and its corresponding voltage response U .

To show the full potential of data driven temperature prediction models for lithium-ion-battery systems, I would like to make the following suggestions for the data set used to train the model:

- Broader temperature range;
- Variable ambient temperature;
- Implementing real life drive cycles.

In addition, the transferability needs to be looked at. That would include transferability between cell technologies and extrapolation from cell to module level. To compensate for the high computational effort to train an ANNs and the need for training data, a cloud server can be implemented, trying to find patterns across battery technologies and different topologies.

Author Contributions: M.K.: conceptualization, methodology, software, validation, formal analysis, investigation, writing—original draft, visualization; M.S.: conceptualization, writing—review and editing, supervision, validation; A.F.: writing—review and editing, validation; J.P.-B.: writing—review and editing; K.P.B.: conceptualization, writing—review and editing, supervision, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Bundesministerium für Wirtschaft und Energie [19121014D].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: I would like to thank the Bundesministerium für Wirtschaft und Energie for financial resources and the project management organisation TÜV Rheinland Consulting GmbH.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lyu, Y.; Siddique, A.; Hasibul Majid, S.; Biglarbegian, M.; Gadsden, S.; Mahmud, S. Electric vehicle battery thermal management system with thermoelectric cooling. *Energy Rep.* **2019**, *5*, 822–827. [[CrossRef](#)]
2. Li, W.; Rentemeister, M.; Badedo, J.; Jöst, D.; Schulte, D.; Sauer, D.U. Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation. *J. Energy Storage* **2020**, *30*, 101557. [[CrossRef](#)]
3. Ma, S.; Jiang, M.; Tao, P.; Song, C.; Wu, J.; Wang, J.; Deng, T.; Shang, W. Temperature effect and thermal impact in lithium-ion batteries: A review. *Prog. Nat. Sci. Mater. Int.* **2018**, *28*, 653–666. [[CrossRef](#)]
4. Yang, Y.; Huang, X.; Cao, Z.; Chen, G. Thermally conductive separator with hierarchical nano/microstructures for improving thermal management of batteries. *Nano Energy* **2016**, *22*, 301–309. [[CrossRef](#)]
5. Cavalheiro, G.; Iriyama, T.; Nelson, G.; Huang, S.; Zhang, G. Effects of Non-Uniform Temperature Distribution on Degradation of Lithium-ion Batteries. *J. Electrochem. Energy Convers. Storage* **2019**, *17*, 021101. [[CrossRef](#)]

6. Fill, A.; Avdyli, A.; Birke, K.P. Online Data-based Cell State Estimation of a Lithium-Ion Battery. In Proceedings of the 2020 2nd IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES), Cagliari, Italy, 1–3 September 2020; Volume 1, pp. 351–356. [[CrossRef](#)]
7. Guo, G.; Long, B.; Cheng, B.; Zhou, S.; Xu, P.; Cao, B. Three-dimensional thermal finite element modeling of lithium-ion battery in thermal abuse application. *J. Power Sources* **2010**, *195*, 2393–2398. [[CrossRef](#)]
8. Spinner, N.S.; Love, C.T.; Rose-Pehrsson, S.L.; Tuttle, S.G. Expanding the Operational Limits of the Single-Point Impedance Diagnostic for Internal Temperature Monitoring of Lithium-ion Batteries. *Electrochim. Acta* **2015**, *174*, 488–493. [[CrossRef](#)]
9. Rajmakers, L.; Danilov, D.; Lammeren, J.; Lammers, M.; Notten, P. Sensorless Battery Temperature Measurements based on Electrochemical Impedance Spectroscopy. *J. Power Sources* **2013**, *247*, 539–544. [[CrossRef](#)]
10. Wang, S.; Fernandez, C.; Yu, C.; Fan, Y.; Cao, W.; Stroe, D.I. A novel charged state prediction method of the lithium ion battery packs based on the composite equivalent modeling and improved splice Kalman filtering algorithm. *J. Power Sources* **2020**, *471*, 228450. [[CrossRef](#)]
11. Thomas, K.E.; Newman, J. Thermal Modeling of Porous Insertion Electrodes. *J. Electrochem. Soc.* **2003**, *150*, A176. [[CrossRef](#)]
12. Zhang, J.; Huang, J.; Li, Z.; Wu, B.; Nie, Z.; Sun, Y.; An, F.; Wu, N. Comparison and validation of methods for estimating heat generation rate of large-format lithium-ion batteries. *J. Therm. Anal. Calorim.* **2014**, *117*, 447–461. [[CrossRef](#)]
13. Au Severson, K.A.; Attia, P.M.; Jin, N.; Perkins, N.; Jiang, B.; Yang, Z.; Chen, M.H.; Aykol, M.; Herring, P.K.; Fragedakis, D.; et al. Data-driven prediction of battery cycle life before capacity degradation. *Nat. Energy* **2019**, *4*, 383–391. [[CrossRef](#)]
14. Yang, F.; Li, W.; Li, C.; Miao, Q. State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network. *Energy* **2019**, *175*, 66–75. [[CrossRef](#)]
15. Li, P.; Zhang, Z.; Xiong, Q.; Ding, B.; Hou, J.; Luo, D.; Rong, Y.; Li, S. State-of-health estimation and remaining useful life prediction for the lithium-ion battery based on a variant long short term memory neural network. *J. Power Sources* **2020**, *459*, 228069. [[CrossRef](#)]
16. Kleiner, J.; Stuckenberg, M.; Komsiyyska, L.; Endisch, C. Real-time core temperature prediction of prismatic automotive lithium-ion battery cells based on artificial neural networks. *J. Energy Storage* **2021**, *39*, 102588. [[CrossRef](#)]
17. Feng, F.; Teng, S.; Liu, K.; Xie, J.; Xie, Y.; Liu, B.; Li, K. Co-estimation of lithium-ion battery state of charge and state of temperature based on a hybrid electrochemical-thermal-neural-network model. *J. Power Sources* **2020**, *455*, 227935. [[CrossRef](#)]
18. Zhu, S.; He, C.; Zhao, N.; Sha, J. Data-driven analysis on thermal effects and temperature changes of lithium-ion battery. *J. Power Sources* **2021**, *482*, 228983. [[CrossRef](#)]
19. Wu, B.; Han, S.; Shin, K.G.; Lu, W. Application of artificial neural networks in design of lithium-ion batteries. *J. Power Sources* **2018**, *395*, 128–136. [[CrossRef](#)]
20. Zhu, C.; Li, X.; Song, L.; Xiang, L. Development of a theoretically based thermal model for lithium ion battery pack. *J. Power Sources* **2013**, *223*, 155–164. [[CrossRef](#)]
21. Rao, L.; Newman, J. Heat-Generation Rate and General Energy Balance for Insertion Battery Systems. *J. Electrochem. Soc.* **1997**, *144*, 2697–2704. [[CrossRef](#)]
22. Abdul-Quadir, Y.; Laurila, T.; Karppinen, J.; Jalkanen, K.; Vuorilehto, K.; Skogström, L.; Paulasto-Kröckel, M. Heat generation in high power prismatic Li-ion battery cell with LiMnNiCoO₂ cathode material. *Int. J. Energy Res.* **2014**, *38*, 1424–1437. [[CrossRef](#)]
23. Bandhauer, T.; Garimella, S.; Fuller, T. A Critical Review of Thermal Issues in Lithium-Ion Batteries. *J. Electrochem. Soc.* **2011**, *158*, R1. [[CrossRef](#)]
24. Thomas, K.E.; Newman, J. Heats of mixing and of entropy in porous insertion electrodes. *J. Power Sources* **2003**, *119–121*, 844–849. [[CrossRef](#)]
25. Chen, S.; Wan, C.; Wang, Y. Thermal analysis of lithium-ion batteries. *J. Power Sources* **2005**, *140*, 111–124. [[CrossRef](#)]
26. Keil, P.; Jossen, A. Thermal FDM Battery Model. In Proceedings of the 2020 Advanced Battery Power–Kraftwerk Batterie, Münster, Germany, 23–25 March 2012.
27. Dongare, A.; Kharde, R.R.; Kachare, A.D. Introduction to Artificial Neural Network. *Int. J. Eng. Innov. Technol. (IJEIT)* **2012**, *2*, 189–194.
28. Lipton, Z. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv* **2015**, arXiv:1506.00019.
29. Bird, J.; Ekart, A.; Buckingham, C.; Faria, D. Evolutionary Optimisation of Fully Connected Artificial Neural Network Topology. In Proceedings of the SAI Computing Conference 2019, Las Vegas, NV, USA, 25–26 April 2019.
30. Rumelhart, D.E.; McClelland, J.L. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations* MIT Press: Cambridge, MA, USA, 1987; pp. 992–997.
31. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [[CrossRef](#)]
32. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
33. Ruan, H.; Sun, B.; Jiang, J.; Zhang, W.; He, X.; Su, X.; Bian, J.; Gao, W. A modified-electrochemical impedance spectroscopy-based multi-time-scale fractional-order model for lithium-ion batteries. *Electrochim. Acta* **2021**, *394*, 139066. [[CrossRef](#)]
34. Dwarampudi, M.; Reddy, N.V.S. Effects of padding on LSTMs and CNNs. *arXiv* **2019**, arXiv:1903.07288.

-
35. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
 36. Nogueira, F. Bayesian Optimization: Open Source Constrained Global Optimization Tool for Python. Available online: <https://github.com/fmfn/BayesianOptimization> (accessed on 9 June 2021).