State of Health Forecasting of Lithium-Ion Batteries applicable in Real-World Operational Conditions

***Friedrich von Bülow*[a,*], *Joshua Mentz*[a], *Tobias Meisen*[b]**

[a] *Volkswagen AG, Berliner Ring 2, 38436 Wolfsburg, Germany*

[b] *Institute of Technologies and Management of the Digital Transformation, Bergische Universität Wuppertal, Rainer-Gruenter-Str. 21, 42119 Wuppertal, Germany*

[*] Corresponding author: Friedrich von Bülow, E-mail address: friedrich.von.buelow@volkswagen.de, Postal address: Volkswagen AG, Berliner Ring 2, 38436 Wolfsburg, Germany

Abstract

Currently, several methods for battery state of health (SOH) prediction exist which are applicable to battery electric vehicles (BEV). However, only few research has been conducted on SOH forecasting based on features that encode causes for battery ageing applicable in real world applications. This paper proposes a machine learning method for SOH forecasting applicable for BEV fleet managers and battery designers in real world applications. As model inputs, we use the battery's operation time within certain operation ranges defined by combinations of the battery signals current, state of charge (SOC) and temperature. Different variants of this temporal aggregation of the battery operation time and of the operation ranges of the battery signals are examined. Our findings state that combining different cycle window widths w_w to one training data set improves the generalization of the model. Also, we find that the fineness of the operational ranges of the signals does not limit the model's performance if w_w is larger than 100 cycles or different w_w are combined.

Keywords: Lithium-ion battery; battery electric vehicles; state of health; battery ageing; forecasting; machine learning

Credit author statement:

Friedrich von Bülow: Conceptualization, Methodology, Software, Investigation, Visualization, Writing - Original Draft, Writing - Review & Editing

Joshua Mentz: Software, Investigation, Visualization, Writing - Review & Editing

Tobias Meisen: Writing - Review & Editing

The results, opinions and conclusions expressed in this publication are not necessarily those of Volkswagen Aktiengesellschaft.

1 Introduction

With the transition to electric mobility the interest for the battery as core component of every battery electric vehicle (BEV) and its state of health (SOH) rises. The SOH of batteries is highly dependent on

the individual battery's usage, stress and environmental conditions. Much research has been conducted to estimate the current SOH, e.g. based on current battery behavior [1–5]. Other researchers have designed experiments to examine battery ageing, i.e. SOH degradation, under laboratory conditions to identify operational conditions which induce accelerated battery ageing. However, only little research has been done on SOH forecasting based on the battery usage applicable in real-world scenarios [6]. We observe that battery data from laboratory operation like [7–10] usually differs significantly from battery data from real-world operation of BEVs. Differences are the duration of hold mode with zero current, the variation of charging and discharging currents as well as the operation range of the state of charge (ΔSOC). Especially, due to regenerative braking in automotive applications charging and discharging mode alternate frequently. Many models using such laboratory battery data assume the same load during the whole battery life. However, batteries cycled in real-world operation, for example in automotive applications, may have a varying load over the life of the battery.

We perceive battery ageing as a state change from a current SOH to a future SOH due to ageing causes, so called stressor types, like those mentioned in section 2.1. We define SOH forecasting as a regression task to predict the future SOH from any input features that encode the battery operational load during a period of time until the future point in time. We differentiate SOH forecasting from the so-called SOH estimation in the literature. We perceive SOH estimation as the determination of the SOH from data at a certain point in time like a single charging cycle. Thus, compared to SOH forecasting, the SOH estimation does not model a change in state, but only a determination of state.

Nevertheless, a model for SOH forecasting becomes necessary in several application areas. For example, cloud-based fleet services depend on suitable SOH forecasting models. Assuming a certain usage scenario of a BEV fleet, the fleet manager can forecast the battery SOH of the fleet's vehicles and knows, when a vehicle replacement due to battery deterioration will be required. Furthermore, battery designers can conduct virtual battery ageing experiments by adapting the usage scenario data. As an example, the maximum discharge current can be limited at certain temperatures. After adapting the usage scenario data, the model will output its effect on the SOH. This enables a prescriptive analysis and recommendations for an operational strategy for BEV fleets including fleet charging management.

However, such new service concepts raise new challenges. When launching a new battery type with e.g. a new cell chemistry, a reliable model for battery ageing suitable for the new battery type is required quickly to support established services. Nevertheless, the available amount of training data of a new battery type is limited in the initial phase. Extensive data generation as solution is expensive and difficult as battery ageing is a lengthy process. Another solution is to transfer an established model for battery ageing of another battery type to the new battery type, as soon as a small amount of data of the new battery type has been gathered [11,12]. This method is called transfer learning and has been successfully applied in different domains, like computer vision [13]. The application of transfer learning for battery SOH forecasting models is a crucial part, as there are differences in battery systems like the nominal capacity, the cell anode and cathode materials as well as the applied load due to usage. However, the general electrochemical behavior of lithium-ion batteries is a major common characteristic which provides an excellent starting point for transfer learning.

Inspired by the applicability for BEV fleet managers and battery designers we define the following requirements of a model for SOH forecasting: First, data transmission cost from BEVs to e.g. a cloud shall be minimal so that the model can be scaled cost-efficiently. Second, when applying the model in production for SOH forecasting, a human would need to choose the input data as there is no data from the future battery load available in the present. Thus, the model's inputs need to be producible and interpretable by humans so that what-if-simulations can be run. Third, the model shall be applicable in real-world scenarios and, thus, be able to capture the higher variability of real-world battery operation compared to laboratory operation. Forth, the required amount of training data shall be minimal at least once an initial model has been trained so that it can be transferred easily to new batteries. Fifth, the model shall be applicable for 2nd life applications of batteries, i.e. below an SOH of 80%.

This paper addresses the aforementioned challenges and contributes a method for SOH forecasting fulfilling the defined requirements. The methods objective is to estimate the future SOH caused by battery ageing encoded in the stressor data. Further, we contribute an analysis of the temporal aggregation of the battery operation time and of the operation ranges of the battery signals to suitable stressor data before the transmission e.g. into a central cloud. This paper contributes a proof of

suitability of the proposed ML model for this problem, but an evaluation of transfer learning on the proposed model is out of scope for this paper.

The remainder of this paper is structured as follows: First, the foundations of battery ageing, the state of the art of battery SOH forecasting and remaining useful life (RUL) are described in section 2. Afterwards, in section 3, the method for SOH forecasting including the stressor extraction, the machine learning regression model and the model application is explained. The used data basis is presented in section 4. Subsequent, we present and discuss our results in section 5. Section 6 concludes our work.

2 State of the Art

2.1 Foundations of Battery Ageing

The SOH is commonly described by internal resistance (SOH_R) and remaining capacity (SOH_C) [14,15]. SOH_R is the relative change of internal ohmic resistance compared to a new battery. SOH_C denotes the remaining maximum capacity $C_{max}(t)$ relative to the initial maximum capacity of a new battery, also called nominal capacity C_{nom} [16].

$$SOH(t) = SOH_C(t) = \frac{C_{max}(t)}{C_{nom}} \quad (1)$$

In the following, we focus on the SOH_C, for simplicity referred to as SOH. Battery ageing can be structured into two causes: calendar ageing and cyclic ageing. Calendar ageing is associated with the storage of batteries, meaning no charging or discharging is applied. Hence, it is also called passive ageing. Cyclic aging corresponds to the impact of battery usage on the SOH, i.e. ageing due to charging and discharging [17].

High temperatures (T) and high state of charges (SOC) are causing fast battery calendar and cyclic ageing [18]. For example, a high SOC over 80% accelerates solid electrolyte interphase (SEI) growth [4]. Other stressors accelerating battery ageing are high charge and discharge C-rates[1] as well as a high ΔSOC [17,19]. Even though battery stressors are qualitatively known as displayed in Table 1, their single and joint impact on the battery's SOH has not been modeled quantitatively yet.

[1] C-rate in [1/h]=[A/Ah] is the current relative to the nominal capacity C_{nom}.

Table 1: Aging mechanisms and their accelerating stressors [20,21]

Battery component	Aging Mechanisms	Accelerated by
Anode	Lithium plating	↑ C-rate, ↓ T, ↑ SOC
	Electrolyte decomposition	↑ T, ↑ SOC
	SEI formation	↑ & ↓ SOC
	SEI decomposition	↑ C-rate, ↓ T
	SEI growth	↑ T, ↑ SOC
	Structural disordering	↑ C-rate, ↑ & ↓ SOC
	Corrosion and loss of electrical contact	↓ SOC
Separator	Blocked pores (Separator and Electrodes)	↑ T, ↑ SOC
Cathode	Dissolving of transition metals	↑ T,
	Binder decomposition	↑ T, ↑ SOC
	Structural disordering	↑ C-rate, ↑ T
	Corrosion and loss of electrical contact	↑ C-rate, ↑ T, ↑ SOC

2.2 Battery State of Health Forecasting

Few research has been conducted for the forecast of the SOH of lithium-ion batteries. Richardson et al. [6] presented a Gaussian process regression (GPR) model which forecasts the SOH only based on the charging cycle number of the battery as input data. As the authors themselves note, past data up to a given cycle does not have strong influence on distant future cycles because the model assumes no change in the usage of the battery. Furthermore, the influence of calendar ageing on battery ageing is neglected. However, in real-world operation of BEV fleets these are unrealistic assumptions (Req. 3).

In a later work, Richardson et al. [22] propose another GPR model this time for capacity fade (ΔQ) prediction. As inputs, they use the current capacity and the histogram-like time elapsed under certain operational conditions. The operational conditions are certain ranges of temperature and current. For example, the sum of all time spent in the range of 5°C to 40°C is a model input. However, the model does not distinguish between different charging currents as these are all considered in the current range below 2 A. Furthermore, no information on SOC and ΔSOC is considered as model input. In summary, like [6] the applicability of this model in real-world operation of BEV fleets is limited (Req. 3).

Lucu et al. [23] propose another Gaussian Process ageing model for capacity fade (ΔQ) prediction. They use ΔAh-throughput, reciprocal temperature, depth of discharge (DOD), average SOC, charging and discharging C-rate as input features. Their usage of DOD, charging and discharging C-rate as input features supports the analysis findings of the last two paragraphs. However, they do not combine battery signals relevant for ageing like SOC, current and temperature as input features. Additionally, they only model the capacity loss in the second ageing phase of linear degradation. However, modelling the SOH

below 80% is interesting for 2nd life applications like stationary energy storage for grid stabilization whose performance requirements need to be satisfied (Req. 5) [24,25].

Existing approaches for SOH estimation [1–3,26–28] require data of the future charging cycles for prediction like the charging curve of current, voltage and temperature to estimate the SOH. These time series change when the battery ages, but are hard to interpret for humans. This means a user of the model, like a fleet manager, cannot derivate changes of the charging curve from an imagined usage scenario of her BEV fleet (Req. 2). For example, this applies to the Long short-term memory (LSTM) model for SOH estimation proposed by Song et al. [27] which uses features obtained from the voltage charging curve as inputs. Consequently, it will not be possible to apply the model for SOH forecasting. Overall, this makes the development of a new approach for real-world applications necessary that considers hold periods, different ΔSOCs, charging and discharging C-rates (Req. 3) and has model inputs which are producible and interpretable by humans (Req. 2).

2.3 Battery Remaining Useful Life Prediction

A similar task to the SOH forecasting is the prediction of the RUL. The RUL is defined as the number of charge-discharge cycles until a specific battery SOH is reached [29]. Usually for BEV batteries a SOH of 80% is required which is defined as the EOL of the first life (EOL_{80}) [24]. The RUL usually refers to full equivalent cycles, but real-world operational battery cycles have different ΔSOC (Req. 3). In the operation of BEVs, the RUL can therefore only serve as an indicator for the number of potentially non-full cycles until EOL.

Furthermore, a RUL prediction model only predict the number of cycles to a single SOH like EOL_{80}, so several RUL prediction models would be required depending on the SOH of interest. However, a single SOH model can forecast a range of SOHs. This is relevant for BEV fleet operation and for batteries in the 2nd life application which usually have a lower SOH (Req. 3 & 5). The inputs of the SOH forecasting models contain direct or indirect information on the number of cycles operated. Under this assumption, the inputs can be adapted until e.g. EOL_{80} is reached. Alternatively, the predicted SOH can be compared with the EOL threshold like 80%. From this point of view, SOH prediction can be seen as a generalization of RUL prediction.

Severson et al. [9] designed a feature-based regression model for RUL prediction using data from the first 100 cycles with a test root mean squared error (RMSE) of 9.1%. Their model uses the initial discharge capacity, charge time, cell can temperature and features derived from the discharge voltage curve as inputs. However, the features like those in the model of Song et al. [27] are not human-interpretable (Req. 2). Furthermore, the model assumes the same load during the whole battery life as it predicts the RUL only based on the first 100 cycles (Req. 3).

Patil et al. [30] use features of the discharging curve to predict the RUL. These features are expressing the fluctuation, the concavity/convexity, skewness and kurtosis of the signals voltage, temperature and current. However, as the authors themselves note, their model has only been investigated for constant load profiles, but load profiles in real-world application are non-constant (Req. 3).

3 Method

As mentioned in section 2.1, battery ageing is perceived as a state change from a current $SOH(t_1)$ to a future $SOH(t_2)$ due to ageing causes. The ageing causes are encoded in the battery operational data which consists of multidimensional time series signals of temperature, current and SOC. The United States Advanced Battery Consortium set the development objective for battery life of 15 years and 1000 cycles [31]. This means in real-world operation batteries age slowly so the relevant time series contain years of operational data at a sample rate in the range of seconds or milliseconds. As done by [22,23], data compression and feature extraction is necessary before the data is transmitted and used as model input (Req. 1 in section 1). For this task we propose the stressor extraction as first part of our SOH forecasting method in Figure 1. Additionally, the resulting stressor data is producible and interpretable by humans (Req. 2).

As second method part, the stressor data is input of a machine learning (ML) model, that outputs the state change ΔSOH from a current $SOH(t_1)$ to a future $SOH(t_2)$. The SOH values are assumed to be known for the training data. The two parts of the proposed SOH forecasting method are explained in detailed: The stressor extraction in section 3.1 and the ML model in section 3.2.

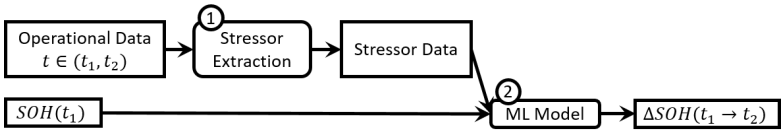
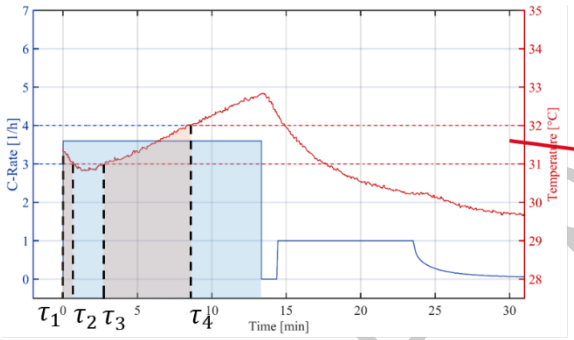


Figure 1: Model structure – stressor extraction (1) and ML model (2)

3.1 Stressor Extraction

In the stressor extraction, battery operational data is used to extract data of battery stressor types which are known to induce battery aging. A battery stressor type is defined by one or several relevant battery signals each limited by an interval (histogram-like binning). E.g., one stressor type is the battery operation within a C-rate of 3-4C and a temperature of 31-32°C as depicted in the plot in Figure 2. The corresponding stressor value is defined by Eq. (2) as time elapsed when the battery is operated within operational conditions of the stressor type. This corresponds to a two-dimensional extraction of [22].



t [h]	C-Rate [1/h] $t_{0<C\leq2}$	$t_{2<C\leq3}$	$t_{3<C\leq4}$	$t_{4<C\leq5}$	....
....	....	....	....	....	....
Temperature [°C] $t_{30<T\leq31}$	0.3189	0	0.0347	0	....
$t_{31<T\leq32}$	0.0473	$4.9 \cdot 10^{-5}$	0.1036	0	....
$t_{32<T\leq33}$	0.0111	0	0.0838	0	....
$t_{33<T\leq34}$	0	0	0	0	....
....	....	....	....	....	....

Figure 2: Exemplary stressor extraction from time series data with c-rate and temperature to form a stressor table which serves as model input feature. Signal interval widths is medium according to Table 3. "I & T" according to Table 4.[2]

$$t_{\substack{3<C\leq4, \\ 31<T\leq32}} = \int_{t_1}^{t_2} f(\tau)d\tau \text{ with } f(\tau) = \begin{cases} 1, & \begin{matrix}\{3 < C\text{-}rate(\tau) \leq 4\} \cap \\ \{31 < T(\tau) \leq 32\}\end{matrix} \\ 0, & \text{else} \end{cases} \quad (2)$$

When having several battery stressors, stressor tables like in Figure 2 can be formed, capturing all relevant battery stressor types. Figure 2 indicates that the battery has been operated 0.1036 h in the operational range during the depicted cycle. The stressor tables are finally vectorized to stressor data which have a suitable format for regression inputs.

The stressor extraction as a type of data preprocessing has hyperparameters. These are the signal interval width, cycle window width w_w, and cycle window shift w_s. First, the signal interval width for the stressor types used in Eq. (2) and displayed in Figure 2 is defining the sampling granularity of the

[2] Data from battery cell no. 1 of the 50th cycle from [9].

signals' bins. E.g., the temperature signal could be split into intervals of 1 °C or 2 °C. On the one hand, a finer signal interval width increases the number of input features as the stressor table in Figure 2 has more values, but will also provide more information to the ML regression model. On the other hand, a bigger signal interval width increases the robustness of the method because noisy signals more likely stay in the correct interval.

Second, one training sample is defined by the two points in time t_1 and t_2 which correspond to the current $SOH(t_1)$ and future $SOH(t_2)$. However, the stressor table in Figure 2 is only based on a single charging cycle, so t_1 and t_2 are close. Thus, we apply windowing to the battery cycles defined by a cycle window width w_w which defines the number of cycles between t_1 and t_2 as shown in Figure 3. This means the stressor tables of the single cycles are simply added up for the whole window. The cycle window width defines for how many future cycles the model learns to forecast the SOH.

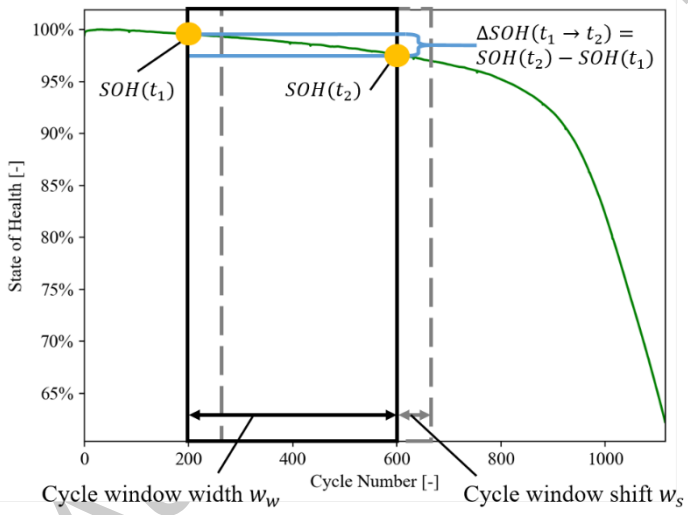


Figure 3: Hyperparameters of stressor extraction[3]

Third, to extract further training samples the cycle window is shifted by w_s as visualized by the grey box in Figure 3. When increasing w_w and w_s, the quantity of training samples decreases as fewer cycle windows can be generated from one battery. Exemplary, training samples consisting of the current $SOH(t_1)$ and vectorized stressor tables are shown in Table 2.

[3] SOH data from battery no. 25 of the 4th batch from [8].

Table 2: Exemplary part of input feature vector x_n and output value y_n. Signal interval widths is coarse according to Table 3. $w_w = 50$, $w_s = 50$. Data from [9].

			Input vector x_n					Output y_n
	Sample n	t [h] Cycles	Feature 1 $SOH(t_1)$	Feature 2 $t_{0<I\leq3,\ 0<T\leq28}$	Feature 3 $t_{0<I\leq3,\ 28<T\leq31}$	Feature 4 $t_{0<I\leq3,\ 31<T\leq34}$	… …	ΔSOH
Cell 1	1	1-50	97.33 %	0h	15.27h	2.73h	…	0.55%
	2	26-75	97.87 %	0h	14.04h	2.70h	…	-0.02%
	3	51-100	97.88 %	0h	14.1h	2.63h	…	-0.07%
	…	…	…	…	…	…	…	…
	47	1216-1175	93.80 %	0h	10.93h	3.51h	…	-0.35%
Cell 2	48	1-50	97.75 %	0h	14.16h	2.56h	…	0.58%
	49	26-75	98.37 %	0h	14.49h	2.24h	…	-0.07%
…	…	…	…	…	…	…	…	…

When w_s is greater than w_w, battery operational data of some cycles would be skipped. In the case of a $w_s = 50$ and $w_w = 25$, we would generate windows from cycle 1-26, 51-76, but the cycles 27-50 would not be used. Thus, in this case, we set the $w_s = w_w$ to avoid skipping any data. The EOL of the battery is in most cases reached after a number of cycles that is not $w_s \cdot n + w_w$ with $n \in \mathbb{N}$. To avoid shorter windows than w_w at the EOL, the last window is until the last cycle (EOL) and begins by w_w before the EOL.

Figure 1 indicates that the model output is the future $\Delta SOH(t_1 \rightarrow t_2)$. An alternative model output, would be the $SOH(t_2) = SOH(t_1) + \Delta SOH(t_1, t_2)$. The work of [22,23] is using the capacity fade ΔQ as output, which is similar to the ΔSOH. [22] applies a specialization of the presented stressor extraction with w_w and w_s of one cycle. When applying their model for several cycles in a loop, we expect propagation of the model error.

3.2 Machine Learning Regression Model

The ML model uses the current $SOH(t_1)$ and the stressor data as features according to Table 2. Once a model of a certain battery type has been trained, we want to minimize the amount of required training data to obtain a model for a new battery type (Req. 4 in section 1). Thus, we require a ML model that is suitable for transfer learning. Therefore, we choose multilayer perceptrons (MLP) as they store different information on different layers, which allows adaption to similar problems via transfer learning. Other models like support-vector regression (SVR) and random forest regression (RFR) are out of scope

of this paper as they do not allow transfer learning. Only for comparison to [22,23] we also implement GPR.

MLP is a type of feedforward artificial neural network (ANN) used for regression tasks. According to the universal approximation theorem, an MLP with a single hidden layer can approximate any smooth function from one finite dimensional space to another to any desired degree of accuracy as long as sufficient hidden neurons are provided. However, no statement is made about the optimality of this approximation regarding the learning time, the ease of implementation or generalization [32–34].

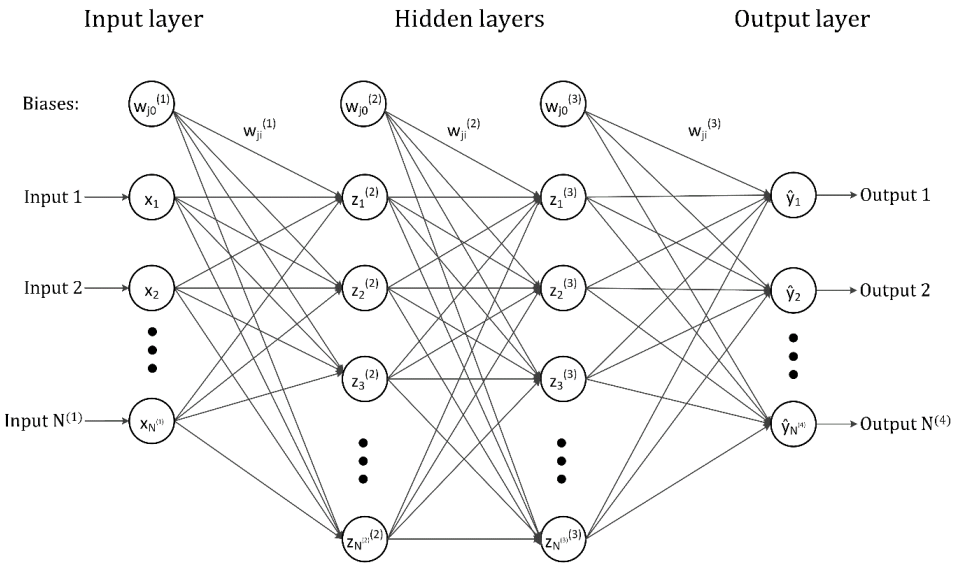


Figure 4: Example of a MLP with $N^{(1)}$ inputs, two hidden layers with $N^{(2)}$ and $N^{(3)}$ neurons and $N^{(4)}$ outputs.

As depicted in Figure 4, a MLP can be described as a directed acyclic graph connecting L layers, numbered from $l = 1$ to L. The input layer $l = 1$ is imaginary and corresponds to the inputs x of the MLP. The last layer $l = L$ corresponds to the outputs y and the predicted outputs $\hat{y}$. Between these two layers there may be several hidden layers $l = 2, \dots, L-1$, but at least one hidden layer is necessary. Each layer consists of $N^{(l)}$ neurons. Except of the input layer, i.e., for all $l \geq 2$, the value $z_j^{(l)}$ of the neuron j in the layer l depends on the values of the neurons $z_j^{(l-1)}$ in the previous layer $l-1$ with the weights $w_{ji}^{(l-1)}$ and biases $b_{j0}^{(l-1)}$ given:

$$z_j^{(l)} = h\left(a_j^{(l)}\right) = h\left(\sum_{i=1}^{N} w_{ji}^{(l-1)} \cdot z_{ji}^{(l-1)} + b_{j0}^{(l-1)}\right) \tag{3}$$

The weight $w_{ji}^{(l-1)}$ describes the connection strength of the neuron i of the previous layer $l-1$ to the neuron j of the next layer l. The bias $b_{j0}^{(l-1)}$ is independent of the input $z_{ji}^{(l-1)}$ and useful for including any fixed offset in the data like unforeseen or nonobservable factors. The weighted sum of all inputs and the bias $a_j^{(l)}$ is used as argument of the nonlinear activation function $h(\cdot)$ to calculate the value $z_j^{(l)}$ of the neuron j in the next layer l. Common activation functions are the hyperbolic tangent (tanh) and the rectifier. Equation (3) corresponds to a single neuron in the network and is used for the process of forward propagation that calculates the predicted output $\hat{y}_n$ of a MLP for a given input vector x_n.

MLP training requires a data set $S = (x_n, y_n)$ with inputs $x_n \in X \subset \mathbb{R}^m$ and outputs $y_n \in Y \subset \mathbb{R}^o$ with the samples $n = 1,2,\dots,N$. The dimensions of the input and output vector are specified by m and o respectively. For this method, m depends on the number of entries in the stressor tables and $o = 1$ for the ΔSOH. Training an MLP is the process of minimizing a cost function $J(\boldsymbol{W}, \boldsymbol{b})$, also referred to as loss or error function, by finding optimal parameters $\boldsymbol{W}$ and $\boldsymbol{b}$, respectively weights and biases. The cost function measures the deviation of the target outputs y_n and the outputs predicted by the network $\hat{y}_n(x_n; \boldsymbol{W}, \boldsymbol{b})$ which is parameterized by $\boldsymbol{W}$ and $\boldsymbol{b}$. A common cost function for regression problems is the mean squared error (MSE) as shown in Eq. (4).

$$\min_{\boldsymbol{W},\boldsymbol{b}} J(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{2N} \sum_{i=1}^{N} \| y_n - \hat{y}_n(x_n; \boldsymbol{W}, \boldsymbol{b}) \|^2 \tag{4}$$

Further information on MLPs can be found in [32–34].

3.3 Model Application

When applying the model, several variants exist: First, the model can be used for forecasting based on historical data. This would imply a forecast under the assumption of the same operational load as in the past. Second, the input features of historical data can be adapted under the assumption of changes in the operational load. This could be the case for the change of the operational strategy of a BEV fleet. This way the model can also be applied by battery designers like described in section 1. Third, explainable AI (XAI) methods can be applied to the SOH forecasting model to quantify the influence of relevant battery signals and value intervals on battery ageing.

The model's requirements listed in section 1 are fulfilled: First, compared to models using operational time series data of e.g. current, temperature, and SOC directly as inputs, this model enables the aggregation of the time series data already on the device incorporating the battery (e.g. a BEV). The operational raw data does not need to be transmitted to e.g. a cloud, but only the aggregated tables are transmitted. This improves security and saves transmission and storage costs. Second, as presented in section 1, by applying transfer learning the size of the training data may be reduced significantly, once an initial model for a battery type has been trained. Third, this model enables battery developers and fleet managers to connect different ways of battery operation with battery ageing. The way of battery operation is specified by the inputs of the method (stressor tables of T, I, SOC). Humans may influence the way of battery operation, e.g. the battery temperature by the cooling system configuration, the maximum discharge and charge current configured in the BMS and the preferred SOC operational range. Other models require e.g. battery operational time series data that is no easily changeable, producible or interpretable by humans. Forth, the presented method is applicable in real-world operational conditions e.g. of a BEV fleets because the selected feature space could capture the higher variability of real-world operation compared to laboratory operation. Fifth, the method could work with SOH below 80% by learning the increasing SOH degradation depending on the battery operation encoded in the stressor tables. This enables the model's applicability for 2nd life applications of batteries.

4 Data basis

The training data suitable for the method described in section 3 should be obtained from several batteries which have aged under a wide range of operational scenarios. This enables the model to forecast the SOH of batteries given an ageing scenario encoded in the stressors.

This method is demonstrated on a public fast charging dataset [9]. It consists of 46 commercial lithium-ion batteries cycled under fast charging conditions up to 80% SOH. These lithium-ion phosphate (LFP)/graphite cells, manufactured by A123 systems (APR18650M1A), were cycled in a forced convection temperature chamber set to 30°C under varied fast charging conditions but identical discharging conditions. All cells are charged with a two-step fast-charging protocol. This protocol has the format "C1(Q1)-C2", in which C1 and C2 are the C-rates of the first and second constant-current

steps (CC1 and CC2 respectively) and Q1 is the SOC at which the current switches. C1 and C2 range from 3 to 8C, while Q1 ranges from 15 to 80 % SOC.[4] The second current step ends at 80% SOC, after which the cells charge with another constant-current step at 1C (CC3) followed by a constant-voltage phase (CV). The upper and lower cutoff potentials are 3.6 V and 2.0 V respectively, which are consistent with the manufacturer's specifications. These cutoff potentials are fixed for all current steps. After some cycling, the cells may hit the upper cutoff potential during fast charging, leading to significant constant-voltage charging. All cells discharge at 4C. The cells have a nominal capacity of 1.1 Ah and a nominal voltage of 3.3 V.

Available signals in the dataset are current, voltage, and temperature with a sampling rate of approximately 0.018 s. We did not apply any down sampling to the time series data. A sampling rate is sufficiently high in our opinion, if it captures short spikes, especially of the current. This is assumed to be the case in the controllers and the battery management system (BMS) anyways. Thus, we do not consider additional down sampling before aggregation of the time series data to stressor tables. The SOH values are defined by the discharged electric charge of the corresponding cycle.[5] The SOC signal is calculated offline relatively to the discharged electric charge of the previous cycle. Some SOH values corresponding to single cycles are inconsistently lower by at least 10 % when comparing to the neighboring SOH values. These values are interpolated linearly using the neighboring SOH values. We did not add noise to the signals because we firstly assume that the precision of the signals used by the BMS is also sufficient for our method. Secondly, the method is robust for noise of time series of current, temperature, and SOC because of the binning in the stressor tables. We do not use cells no. 1 and 19 as they show abnormal behavior. For example, cycle 11 of cell no. 1 is 11.8 times longer as the previous cycle because the switching from charging to discharging happens late.[6] This would skew the min-max-normalization. We use cells no. 3, 7, and 8 as validation cells which means that they are not part of the training, validation or test set.

[4] C1 and C2 ϵ {3, 3.6, 4, 4.4, 4.8, 5.4, 6, 7, 8}C. Q1 ϵ {15, 25, 30, 35, 40, 50, 60, 70, 80} % SOC.
[5] The discharged electric charge is specified by the end value of the signal "Qd".
[6] A similar situation is observed for cycle 851 of cell no.1 and the first 51 cycles of cell no. 19.

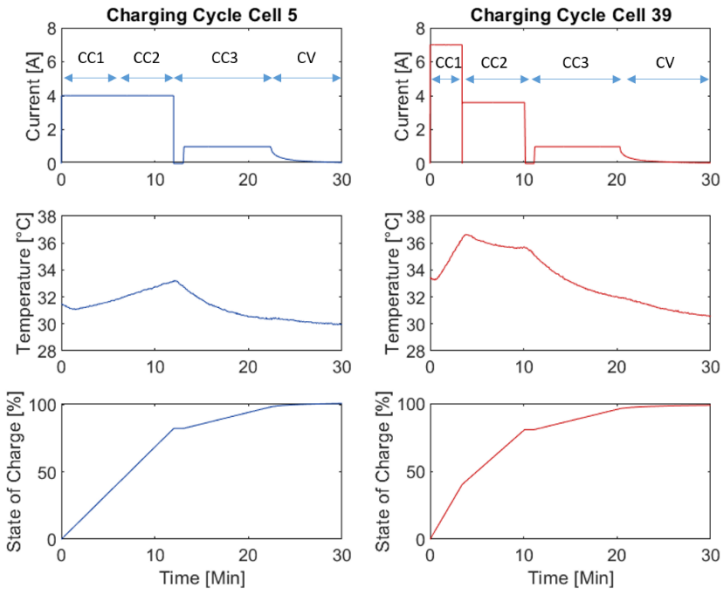


Figure 5: Different fast charging protocol within the data set [8]

5 Results and Discussion

5.1 Design of Experiments

For developing this method for SOH forecasting it is essential to find suitable hyperparameters for the stressor extraction described in section 3.1. The examined hyperparameters are the signal interval width for current, temperature, and SOC as well as the cycle window width w_w and shift w_s. We aim at finding the right balance of input features and training samples. Further, we examine different combinations of the signals of current, temperature, and SOC to stressor tables for charging, discharging, and hold mode. In addition, we examine the capability of the SOH forecasting model to generalize.

Experiment 1a: Single window width w_w

The cycle window width w_w determines how many cycles into the future the model can forecast the SOH following Figure 3. To examine, how well the model forecasts small and big SOH degradations, we use different w_w. The maximal w_w is limited by the battery cell no. 2 which ages the fastest. Thus, we create data sets with w_w of 25, 50, 100, 200, 400, and 530 cycles (W1 to W6).

Experiment 1b: Combined window width w_w

We expect that a model trained with a data set of a certain w_w will only be suitable to predict on samples of that or a similar w_w. This would limit the practical applicability of the model for example for fleet managers because they may be interested into the ageing after different amounts of cycles using one

model. Thus, we combine the aforementioned w_w to further data sets as follows: {25, 50}, {50, 100}, {25, 50, 100}, {200, 400, 530}, {50, 200, 530}, and {50, 200, 400, 530} (W7 to W12). However, we cannot combine different signal interval widths as different signal interval widths have different input feature shapes requiring different input layers.

Experiment 1c: Window shift $w_s = 50$

A small w_s will improve the ability of the MLP to interpolate as more samples are available, but the size of the training data will increase as well. When applying the method e.g. in BEVs the stressor tables will not be transmitted and saved in a central cloud after every battery cycle, but only after the battery has aged measurably. Thus, we create the data sets for experiments 1a and b with $w_s = 25$ and for experiment 1c with $w_s = 50$ (see Figure 3).

Further, the data sets W1 to W12 each contain either fine, medium, or coarse signal interval widths as displayed in Table 3 (F, M, and C respectively). Thus, we create 36 data sets for each w_s. These data sets consist of stressor tables using two combined signals (2D) in the operational modes of charging, discharging, and hold as shown in Table 4 (2D stressor table, variant A).

Table 3: Signal interval width for current, temperature and SOC

	Signal interval width		
	Current	**Temperature**	**SOC**
Fine (F)	0.5C	0.5 °C	5 %
Medium (M)	1C	1 °C	10 % at 0 and 100%, else 20 %
Coarse (C)	3C	3 °C	20 %

Table 4: Combined signals for 2D stressor tables, variant A and B

	Charging	**Discharging**	**Hold**
variant A	T & SOC I & SOC I & T	T & SOC I & SOC I & T	T & SOC
variant B	T & SOC I & SOC	T & SOC I & SOC	T & SOC

First experiments deepened in section 5.2 show that the coarse signal interval width (C) with w_s of 25 and w_w of 200, 400, and {25, 50, 100} cycles (W4, W5, and W9 each with C-2D,variant A) result in a coefficient of determination (R^2) greater than 0.99. Thus, we create further data sets to examine

variations of the stressor tables for w_w W4, W5, and W9 as shown in Table 5. These all have a coarse signal interval width.

Table 5: Data sets created for the experiments

Exper. No.	Abbreviation	Signal interval width	Cycle window width w_w	Cycle window shift w_s	Stressor tables
1	**2D,Variant A**	F, M, C	W1-12	25 and 50	A
2	**2D,Variant B**	C See	W4, W5, and W9	25	B See
3	**3D**	C Table 3	W4, W5, and W9	25	A Table 4
4	**2D,Variant A**	C	1 cycle (W13)	1	A

Experiment 2: 2D stressor tables , variant B

Second, we create a data set with 2D stressor tables, but we change the signal combinations as shown in Table 4 (Wx-C-2D,variant B). In comparison to variant A in Experiment 1, this variant B does not contain any explicit stressor combination of current and temperature. We assume that this eases the applicability of the model as the dependency of current and temperature does not need to be considered explicitly when designing a sample for a forecast by a human (Req. 2).

Experiment 3: 3D stressor tables

Third, as stated in section 2.1, current, temperature, and SOC are relevant signals inducing battery ageing. To examine them as joint stressors, we create a data set with 3D stressor tables combining current, temperature, and SOC over all battery operational modes (Wx-C-3D).

Experiment 4: Window width $w_w = 1$ and window shift $w_s = 1$

Forth, Richardson et al. [22] use a data set with $w_w = w_s = 1$. For comparison with them, we further create a data set with $w_w = w_s = 1$ using the 2D stressor table, variant A (W13-C-2D, variant A).

For each data set a corresponding hyperparameter optimization was executed using Hyperopt version 0.2.4 with Tree-structured Parzen Estimator (TPE) which resulted in a final model for each data set [35,36]. Hyperopt parameters were set to maximum of 1,000 evaluations and 100 random startup evaluations. Each data set was split for training, validation, and test by the ratio of 80:10:10. We apply min-max-normalization to each data set. The MSE on the validation data set was set as optimization metric for the hyperparameter optimization. The variable hyperparameters of the ANN and their range are chosen form experience and shown in Table 6. The constant hyperparameters of the ANN are the

optimizer Adam, 80 epochs, MSE as loss function, maximum of ten hidden layers and a linear activation function of the output layer. Version 2.0.4 of TensorFlow was used as back-end including version 2.2.4 of Keras.

Table 6: Overview of the used hyperparameters of the artificial neural network

Hyperparameter		**Values**
Activation function of hidden layers		Sigmoid, ReLU,[7] tanh
Batch size		32, 64, 128
Learning rate α		0.01, 0.001, 0.0001
Number of layers		[1, 10], in steps of 1
Neurons per layer		[5, 200], in steps of 10
Choose one, others are 0:	**Regularization parameter** $\boldsymbol{\lambda_1}$ **or** $\boldsymbol{\lambda_2}$	0.01, 0.001, 0.0001
	Dropout rate	0.1, 0.2, 0.5

5.2 Evaluation of Model Performance

We first evaluate the results of the 36 models trained on data sets with $w_s = 25$ (Experiment 1a and b). Afterwards we compare the results of $w_s = 25$ and $w_s = 50$ with each other (Experiment 1c). Then, we analyze the adapted 2D and 3D stressor tables (Experiments 2 & 3 respectively). We further assess $w_w = w_s = 1$ (Experiment 4). Finally, we evaluate MLPs trained with 2D stressor tables each with $w_s = 1$ and $w_s = 25$. We also evaluate GPR fitted on the last data set.

The hyperparameters of four MLPs are displayed in Appendix 8.2 (Table 11). The hyperparameter optimization never converges choosing the maximum of the value range for the number of layers and neurons per layer in Table 6. This underlines a suitable choice of the value range of the hyperparameters.

As shown in Table 10, data sets with a large w_w have a higher mean ΔSOH, i.e. a higher mean output value. The variance of the output values also increases resulting in a greater variance of ΔSOH the models needs to fit. Thus, when comparing models with different w_w, i.e., a different output value distribution, the coefficient of determination (R^2) is suitable because it normalizes the MSE by the variance σ_y^2 as shown in Eq. (5) [37]. However, when comparing models with the same w_w the RMSE is still favorable as it is non-relative, thus indicating the error in the real output value unit.

[7] Rectified linear unit (ReLU) refers to a neuron that employs the rectifier activation function.

$$R^2 = \frac{SSE}{SST} = 1 - \frac{SSR}{SST} = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} = 1 - \frac{N \cdot MSE}{N \cdot \sigma_y^2} = 1 - \frac{MSE}{\sigma_y^2} = 1 - \frac{RMSE^2}{\sigma_y^2} \quad (5)$$

Where SST is the sum of squared totals, SSE the sum of squared explanations, SSR the sum of squared residuals, N the number of samples in the data set, y_i the true output value, $\bar{y}$ the mean, σ_y^2 the variance of these values, and $\hat{y}_i$ the predicted output value of the model. R^2 represents the percentage of variance explained by the model. The optimal score is 1.0.

Experiment 1a: Single window width w_w

For each data set W1 to W12, we observe a positive influence of finer signal interval width on the models' RMSEs as depicted in Figure 7. A finer signal interval increases the number of features as shown in Table 10. This improves the distinguishability of the samples and thus reduces the RMSE. This effect seems to diminish from W1 to W6 in Figure 6 when looking at R^2. However, as noted earlier, R^2 is an unsuitable metric for the analysis of the signal interval width due to its normalization. The described effect can still be observed when zooming into Figure 6.

Despite the use of regularization techniques, we observe overfitting for all models without combined w_w in Figure 7 (W1 to 6). For these models the RMSE on the test data set is on average 65% worse than on the training data set. It has to be noted that the RMSE from W1 to W6 also increases because the mean of the output value increases (see Table 10). Thus, in the following we look at R^2 as a relative RMSE (see Eq. (5)).

When increasing w_w for the coarse signal interval width, measured by R^2 in Figure 6 the models fit better. For W4 to W6 in Figure 6 for all three data sets R^2 is above 0.99, meaning these models explain at least 99% of the variance in the output data. For these models, R^2 on the training data set is on average 0.49% better than on the test data set. For W1 to W3, this value is 2.05%. Thus, the generalizability is better for W4 to W6 than for W1 to W3. One reason for this might be the data set size which decreases from W1 to W6 (see Table 10). The more important reason is seen in the output value distribution also shown in Table 10. From W1 to W6 the mean of the output values increases which means the model is trained to learn greater SOH degradations. The variance of the output values also increases which means

that the model learns a greater variety of SOH degradations. Both ease the distinguishability of the samples for the model.

From this observation we conclude that a data set with samples that cover a longer w_w is more suitable for model training. A longer w_w also makes sense for the practical applicability for fleet operators: They are more interested in the battery ageing after a longer period like more than 200 cycles (W5). They are not so much interested in the battery ageing after only 25 cycles (W1). Assuming an intensive use of the BEVs with two charging events per day, these two w_w would correspond to a forecast of three or only 0.5 months.

Experiment 1b: Combined window width w_w

We also examined the combination of different w_w to one data set (W7 to W12) which increases the data set size as shown in Table 10. When combining different w_w for the coarse signal interval width, we observe different effects: Combining W1 and W2 to W7 does not improve R^2 on the training data, but improves the generalization on validation and test data. Remarkably, separately for W1, W2, and W3 R^2 is 0.9714 on average for the test data, but when combining W1, W2, and W3 to W9 R^2 is 0.993 on the test data. Also, W9 achieves a higher R^2 compared to W7 and W8 on the test data. Thus, we conclude that combining data sets with short w_w which are similar to each other does not provide enough incentive for generalization. We observe improved generalization with combined w_w also for W10 to W12, though the effect decreases for combining longer w_w. The models W7 to W12 generalize better because the models have to abstract features from samples with different w_w. Thus, combining different w_w into one data set is favorable.

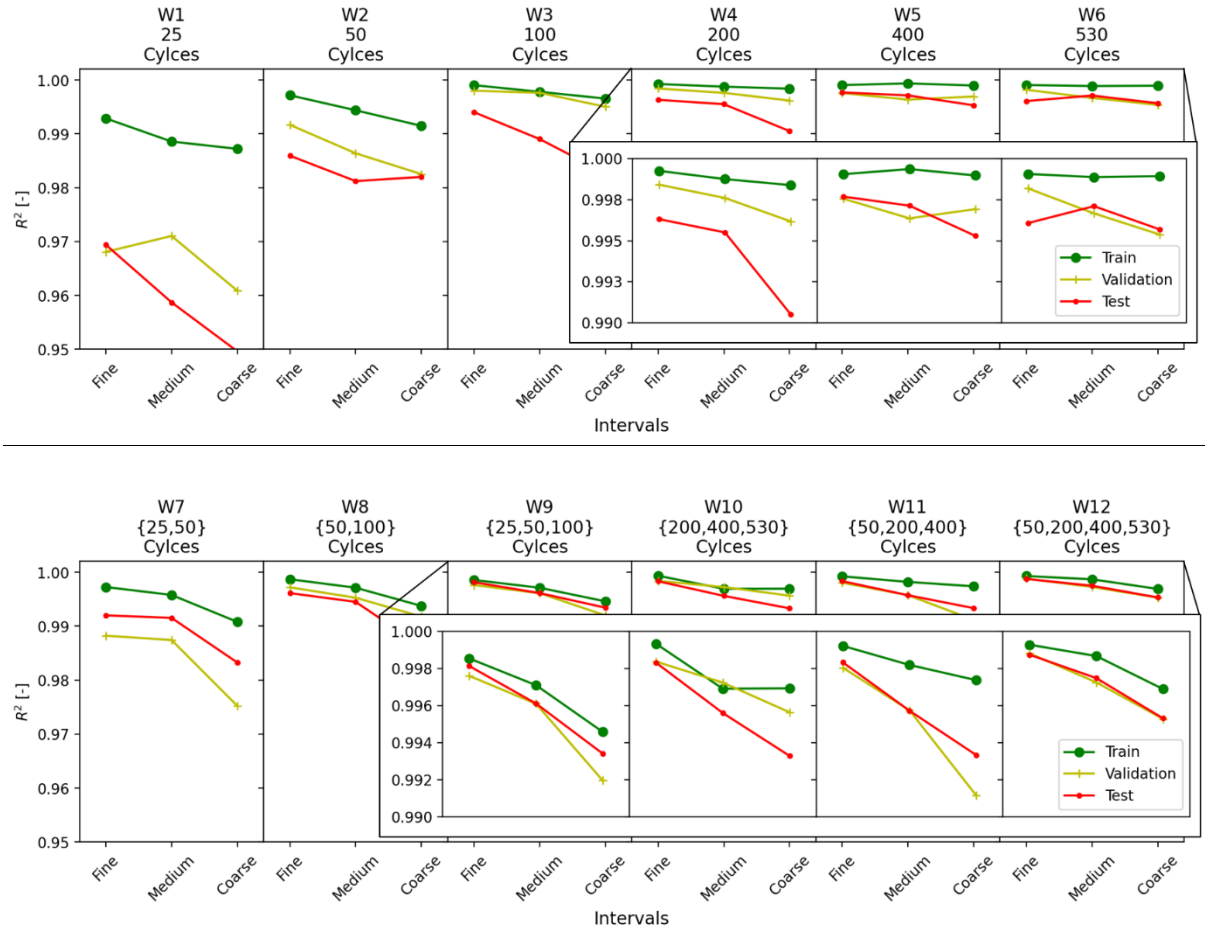


Figure 6: Model performance W1 to 12 with $w_s = 25 - R^2$
(Additional boxes show zoom of W4 to W6 and W9 to W12 respectively)

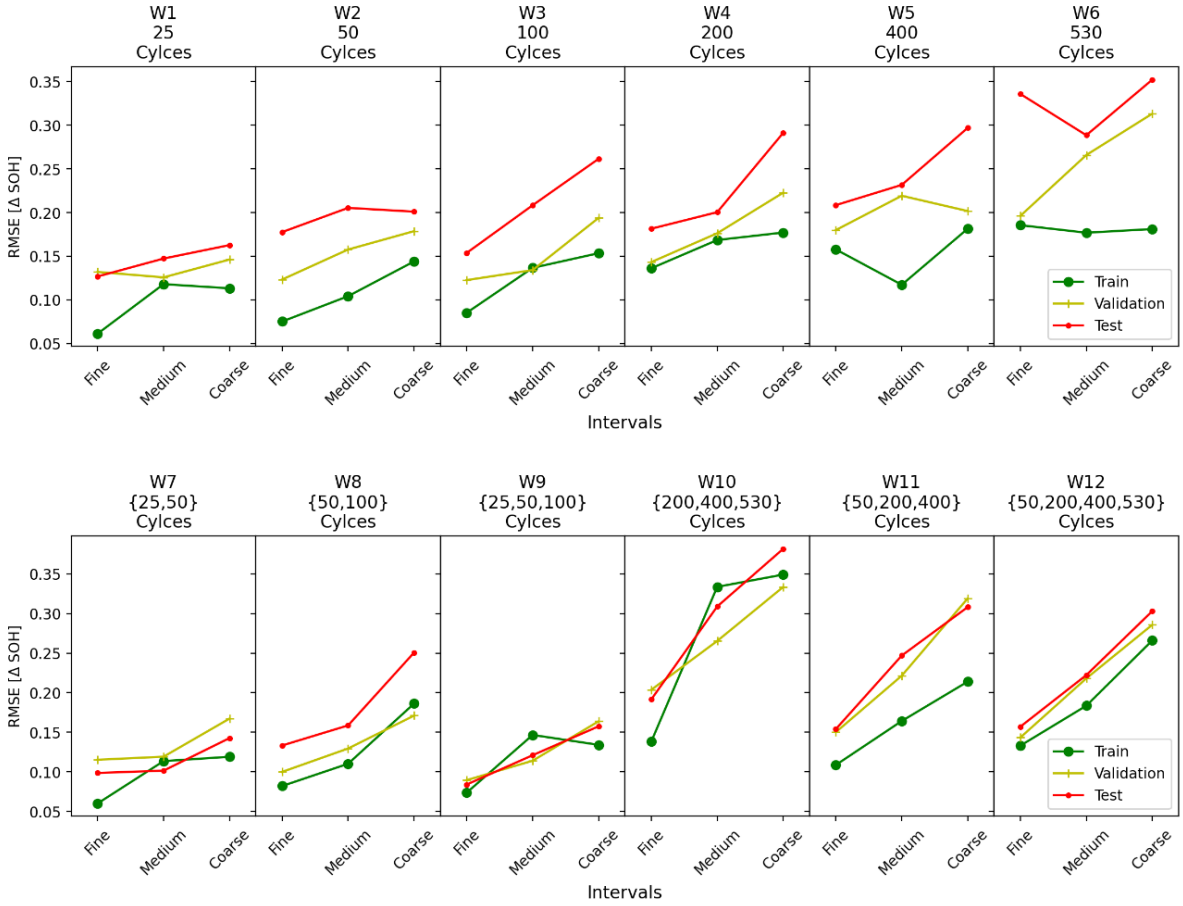


Figure 7: Model performance W1 to W12 with $w_s = 25$ - RMSE

Experiment 1c: Window shift $w_s = 50$

After examining models trained on data sets with $w_s = 25$, the same configurations only with $w_s = 50$ are examined. The results for R^2 and RMSE show slightly inferior results for all models (see Appendix 8.2, Table 13). When increasing w_s from 25 of 50 the number of samples is roughly halved. With fewer samples, the models were neither able to fit nor to generalize as well as they did with $w_s = 25$. Furthermore, the single samples have more overlapping cycles with $w_s = 25$ which leads to better training results without the necessity for more battery time series data. Also, the MLP has more sample points to interpolate in between.

Experiment 2: 2D stressor tables, variant B

The data sets with coarse 2D stressor tables variant B have 30 features less than with 2D stressor tables variant A. Switching from 2D stressor tables variant A to B did not worsen the RMSE or R^2 of the models. Thus, we see the combination of battery signals into 2D stressor tables with variant B as sufficient for learning SOH degradation.

Table 7: Results of models trained on 2D stressor tables, variant B, and $w_s = 25$ (Wx-C-2D,variant B)

Interval	Window width w_w	RMSE			R^2		
		Train	Validation	Test	Train	Validation	Test
Coarse	200 cycles (W4-C-2D, variant B)	0.1538	0.208	0.3034	0.9987	0.9966	0.9897
	400 cycles (W5-C-2D, variant B)	0.1978	0.1794	0.3186	0.9989	0.9976	0.9946
	{25, 50, 100} cycles (W9-C-2D, variant B)	0.1321	0.1723	0.1568	0.9947	0.9911	0.9935

Experiment 3: 3D stressor tables

The data sets with coarse 3D stressor tables combining current, temperature, and SOC over all battery operational modes (W4-/W5-/W9-C-3D) have 20 more features than with 2D stressor tables, variant A. The results of models trained on coarse 3D stressor tables and $w_s = 25$ are shown in Table 10. Switching from 2D to 3D stressor tables did not improve the RMSE or R^2 of the models significantly. Thus, we see the combination of battery signals into 2D stressor tables as sufficient for learning SOH degradation.

Table 8: Results of models trained on 3D stressor tables and $w_s = 25$ (Wx-C-3D)

Interval	Window width w_w	RMSE			R^2		
		Train	Validation	Test	Train	Validation	Test
Coarse	200 cycles (W4-C-3D)	0.2159	0.234	0.2845	0.9981	0.9955	0.9901
	400 cycles (W5-C-3D)	0.2193	0.2507	0.3828	0.9981	0.9952	0.9922
	{25, 50, 100} cycles (W9-C-3D)	0.1226	0.1728	0.1581	0.9954	0.9911	0.9933

Experiment 4: Window width $w_w = 1$ and window shift $w_s = 1$

The features used by Richardson et al. [22] correspond to $w_w = w_s = 1$ in the presented method. Their best validation RMSE for predicting the capacity change of a battery cell is 0.0201. We compare their best model with our coarse data set with $w_w = w_s = 1$ and 2D stressor tables variant A. On this data set, we reach a test RMSE of 0.0357. In comparison, our method performs worse by 43%, but [22] did not provide the RMSE on the training data set. Thus, potential overfitting of their model cannot be checked. For our model, RMSE on the test data set is only 9% worse than on the training data set. Further, [22] do not publish R^2 of their models, but on the aforementioned data set our test RMSE of 0.0357 corresponds to R^2 of only 0.388. The worst test R^2 of all other examined variations shown in

Figure 6 is 0.949. Concluding R^2 of $w_w = w_s = 1$ is 59% worse than the worst of the other w_w (W1 to W12).

Table 9: Results of models trained on 2D stressor tables, variant A, and $w_w = w_s = 1$ (W13-C-2D, variant A)

Interval	Window width w_w	RMSE			R^2		
		Train	Validation	Test	Train	Validation	Test
Coarse	1 cycle (W13-C-2D, variant A)	0.0325	0.0312	0.0357	0.5499	0.4436	0.3881

This becomes also obvious in Figure 8 which depicts the model fit of three different data sets each split into training, validation, and test data by comparing the predicted versus measured ΔSOH values. A good fit would show predictions that are close to the true values. Thus, the blue dots should be closely around the orange identity line. The mentioned data set with $w_w = w_s = 1$ is shown in part b) of Figure 8. Not even on the training data set a good fit is reached. In addition, the overfitting is evident as the deviation from the identity line is even larger for validation and test data. Contrarily, the data set W9 with $w_s = 25$ and 2D stressor table, variant A, depicted in in part a) shows a very good fit and generalization which are important for practical application. Richardson et al. [22] use GPR with a Matérn-Kernel with $\nu = 2.5$. When using the same data as in part a), but using GPR as [22] instead of MLP as in part a), we achieve the results shown in part c) of Figure 8. Compared to c) the overfitting is even higher. Compared to a) the RMSE on the training data is better, but overfitting occurs. The comparison with a) indicates that GPR overfits more easily on the same data. Thus, we favor MLP.

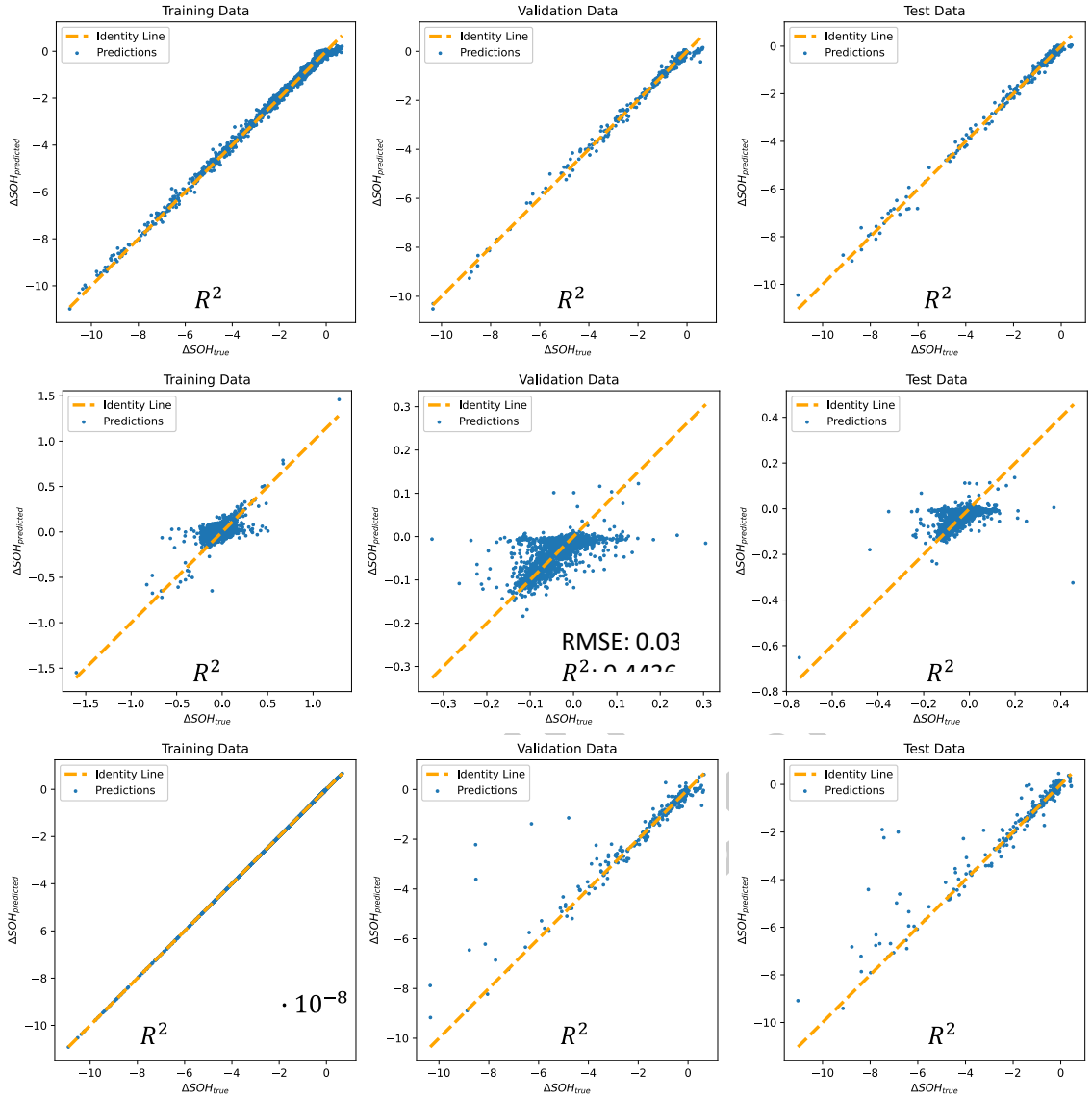


Figure 8: Predicted versus measured ΔSOH values for **a)** MLP, $w_w = \{25, 50, 100\}$ (W9-C-2D, variant A), $w_s = 25$; **b)** MLP, $w_w = w_s = 1$ (W13-C-2D, variant A); **c)** GPR, $w_w = \{25, 50, 100\}$ (W9-C-2D, variant A), $w_s = 25$ (Same data as in a) but with GPR); Common data set characteristics: Coarse, 2D stressor table, variant A.

Summarizing the results, combining different w_w to one training data set improves generalization. Also, models trained with features extracted for fine signal intervals provide more precise forecasts than coarse signal intervals. $w_s = 25$ is favorable to $w_s = 50$. Using 3D stressor tables does not improve the model's forecast compared to 2D stressor tables, while 2D stressor tables variant B compared to variant A of the 2D stressor tables does not worsen the model's forecast.

The results are limited by the used data set. The battery data was obtained from batteries cycles with 100% ΔSOC. Cycles with smaller ΔSOC were neither part of the data basis, nor could they be used for validation. Furthermore, discharging currents were constant over the entire data set. So, findings for

non-constant discharging currents as in automotive applications are limited. Still the models and their input features can capture non-constant discharging currents. Furthermore, it has to be noted that the used data is on battery cell level, but the battery system's SOH is also of interest.

6 Conclusion

Machine learning models for battery SOH degradation [6,22,23] exist. Each of them has certain weaknesses limiting model application in automotive context of BEVs (Req. 1-5). For example, they face variations in charging and discharging current which is not considered by the models (Req. 3). We propose a ML method for SOH forecasting applicable for BEV fleet managers and battery designers in real world applications based on MLPs. As model inputs, we used the battery operation time within certain operational ranges defined by combinations of the battery signals current, SOC, and temperature during a certain number of cycles. The chosen input features are not limited to the application of laboratory data, but applicable to real-world operation of BEV fleets. For example, highly fluctuating discharging currents, variable ΔSOC for each cycle and long hold periods for calendar ageing can be captured by the model (Req. 3). Our findings state that combining different cycle window widths w_w to one training data set improves the generalization of the model. Furthermore, the fineness of the operational ranges of the signals does not limit the model if the w_w is larger than 100 cycles or different w_w are combined. Also, we observed that GPR overfits more easily than MLP on the same stressor data. BEV fleet managers can improve the operation and replacement of their fleet members by applying the proposed SOH forecasting model. As a next step, model validation with data obtained from real-world operational data shall be executed. However, the differences of SOH forecasting on battery cell and system level has to be examined more closely. Also, XAI shall be applied to give insights into the ANN. When adding noise to the time series signals, the influence of data accuracy on the performance of the method could be evaluated. After proofing the suitability of the proposed ML model for SOH forecasting in this paper, we will evaluate transfer learning on the proposed model to enable quick applicability on new batteries as argued in section 1.

7 References

[1] Y. Fan, F. Xiao, C. Li, G. Yang, X. Tang, A novel deep learning framework for state of health estimation of lithium-ion battery, J. of Energy Storage. 32 (2020) 101741. 10.1016/j.est.2020.101741.

[2] J. Yu, B. Mo, D. Tang, J. Yang, J. Wan, J. Liu, Indirect state-of-health estimation for lithium-ion batteries under randomized use, Energies. 10 (2017) 2012. 10.3390/en10122012.

[3] J. Wu, Y. Wang, X. Zhang, Z. Chen, A novel state of health estimation method of li-ion battery using group method of data handling, J. of Power Sources. 327 (2016) 457–464. 10.1016/j.jpowsour.2016.07.065.

[4] A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard, D. Riu, A review on lithium-ion battery ageing mechanisms and estimations for automotive applications, J. of Power Sources. 241 (2013) 680–689. 10.1016/j.jpowsour.2013.05.040.

[5] E. Locorotondo, V. Cultrera, L. Pugi, L. Berzi, M. Pierini, G. Lutzemberger, Development of a battery real-time state of health diagnosis based on fast impedance measurements, J. of Energy Storage. 38 (2021) 102566. 10.1016/j.est.2021.102566.

[6] R.R. Richardson, M.A. Osborne, D.A. Howey, Gaussian process regression for forecasting battery state of health, J. of Power Sources. 357 (2017) 209–219. 10.1016/j.jpowsour.2017.05.004.

[7] B. Bole, C.S. Kulkarni, M. Daigle, Randomized Battery Usage Data Set. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#batteryrnddischarge (accessed 4.02.2021).

[8] P.M. Attia, A. Grover, N. Jin, K.A. Severson, T.M. Markov, Y.-H. Liao, M.H. Chen, B. Cheong, N. Perkins, Z. Yang, P.K. Herring, M. Aykol, S.J. Harris, R.D. Braatz, S. Ermon, W.C. Chueh, Closed-loop optimization of fast-charging protocols for batteries with machine learning, Nature. 578 (2020) 397–402. 10.1038/s41586-020-1994-5.

[9] K.A. Severson, P.M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M.H. Chen, M. Aykol, P.K. Herring, D. Fraggedakis, M.Z. Bazant, S.J. Harris, W.C. Chueh, R.D. Braatz, Data-driven prediction of battery cycle life before capacity degradation, Nat. Energy. 4 (2019) 383–391. 10.1038/s41560-019-0356-8.

[10] B. Saha, K. Goebel, Battery Data Set. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#battery (accessed 4.01.2021).

[11] F. von Bülow, F. Heinrich, T. Meisen, Fleet Management Approach for Manufacturers displayed at the Use Case of Battery Electric Vehicles, in: IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021, unpublished.

[12] S.J. Pan, Q. Yang, A Survey on Transfer Learning, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359. 10.1109/TKDE.2009.191.

[13] L. Shao, F. Zhu, X. Li, Transfer learning for visual categorization: a survey, IEEE Transactions on Neural Networks and Learning Systems. 26 (2015) 1019–1034. 10.1109/TNNLS.2014.2330900.

[14] W. Waag, C. Fleischer, D.U. Sauer, Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles, J. of Power Sources. 258 (2014) 321–339. 10.1016/j.jpowsour.2014.02.064.

[15] L. Chen, Z. Lü, W. Lin, J. Li, H. Pan, A new state-of-health estimation method for lithium-ion batteries through the intrinsic relationship between ohmic internal resistance and capacity, Measurement. 116 (2018) 586–595. 10.1016/j.measurement.2017.11.016.

[16] M.S.H. Lipu, M.A. Hannan, A. Hussain, M.M. Hoque, P.J. Ker, M.H.M. Saad, A. Ayob, A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles, Journal of Cleaner Production. 205 (2018) 115–133. 10.1016/j.jclepro.2018.09.065.

[17] T. Gewald, A. Candussio, L. Wildfeuer, D. Lehmkuhl, A. Hahn, M. Lienkamp, Accelerated aging characterization of lithium-ion cells, Batteries. 6 (2020) 6. 10.3390/batteries6010006.

[18] B.P. Matadi, S. Geniès, A. Delaille, T. Waldmann, M. Kasper, M. Wohlfahrt-Mehrens, F. Aguesse, E. Bekaert, I. Jiménez-Gordon, L. Daniel, X. Fleury, M. Bardet, J.-F. Martin, Y. Bultel, Effects of biphenyl polymerization on lithium deposition in commercial graphite/NMC lithium-ion pouch-cells during calendar aging at high temperature, J. Electrochem. Soc. 164 (2017) A1089-A1097. 10.1149/2.0631706jes.

[19] A. Marongiu, M. Roscher, D.U. Sauer, Influence of the vehicle-to-grid strategy on the aging behavior of lithium battery electric vehicles, Appl. Energy. 137 (2015) 899–912. 10.1016/j.apenergy.2014.06.063.

[20] C. Birkl, Diagnosis and Prognosis of Degradation in Lithium-Ion Batteries, Oxford, 2017.

[21] T.T. Nguyen, Big Data Alterungsanalyse von Fahrzeugantriebsbatterien zur Klassifizierung für stationäre Anwendungen, München, 2019.

[22] R.R. Richardson, M.A. Osborne, D.A. Howey, Battery health prediction under generalized conditions using a Gaussian process transition model, J. of Energy Storage. 23 (2019) 320–328. 10.1016/j.est.2019.03.022.

[23] M. Lucu, E. Martinez-Laserna, I. Gandiaga, K. Liu, H. Camblong, W.D. Widanage, J. Marco, Data-driven nonparametric Li-ion battery ageing model aiming at learning from real operation data - Part B, J. of Energy Storage. 30 (2020) 101410. 10.1016/j.est.2020.101410.

[24] D.U. Sauer, H. Wenzl, BATTERIES | Lifetime Prediction, in: J. Garche (Ed.), Encyclopedia of electrochemical power sources, Elsevier, Amsterdam, The Netherlands, 2009, pp. 522–538.

[25] A. Warnecke, Degradation Mechanisms in NMC Based Lithium-Ion Batteries, Aachen, 2017.

[26] Z. Guo, X. Qiu, G. Hou, B.Y. Liaw, C. Zhang, State of health estimation for lithium ion batteries based on charging curves, J. of Power Sources. 249 (2014) 457–462. 10.1016/j.jpowsour.2013.10.114.

[27] L. Song, K. Zhang, T. Liang, X. Han, Y. Zhang, Intelligent state of health estimation for lithium-ion battery pack based on big data analysis, J. of Energy Storage. 32 (2020) 101836. 10.1016/j.est.2020.101836.

[28] D. Yang, X. Zhang, R. Pan, Y. Wang, Z. Chen, A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve, J. of Power Sources. 384 (2018) 387–395. 10.1016/j.jpowsour.2018.03.015.

[29] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, K. Dietmayer, Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods, J. of Power Sources. 239 (2013) 680–688. 10.1016/j.jpowsour.2012.11.146.

[30] M.A. Patil, P. Tagade, K.S. Hariharan, S.M. Kolake, T. Song, T. Yeo, S. Doo, A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation, Appl. Energy. 159 (2015) 285–297. 10.1016/j.apenergy.2015.08.119.

[31] United States Advanced Battery Consortium (USABC), Goals for Advanced Batteries for EVs. http://www.uscar.org/commands/files_download.php?files_id=364, 2019 (accessed 4.05.2021).

[32] C.M. Bishop, Pattern recognition and machine learning, 8th ed., Springer, New York, NY, USA, 2009.

[33] M.T. Hagan, H.B. Demuth, M.H. Beale, O. de Jesús, Neural network design, 2nd ed., 2016.

[34] S.S. Haykin, Neural networks and learning machines, 3rd ed., Pearson, New York, NY, USA, 2009.

[35] J.S. Bergstra, D. Yamins, D. Cox, Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architecture, PMLR. 28 (2013) 115–123.

[36] J.S. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for Hyper-Parameter Optimization, in: 24th International Conference on Neural Information Processing Systems (NIPS), 2011, pp. 2546–2554.

[37] D. Chicco, M.J. Warrens, G. Jurman, The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation, PeerJ. Computer science. 7 (2021) e623. 10.7717/peerj-cs.623.

8 Appendix

8.1 Dataset characteristics

Table 10: Dataset characteristics, 2D stressor tables, variant A (Wx-x-2D, variant A)

Interval	**Window width w_w [cycles]**	**# features m**	**Window shift $w_s = 25$**		**Window shift $w_s = 50$**	
			# samples N	**Output value distribution ΔSOH (μ, σ)**	**# samples N**	**Output value distribution ΔSOH (μ, σ)**
Fine		2802		(-0.51, 0.72)		(-0.51, 0.72)
Medium	25 (W1)	404	1387	(-0.51, 0.72)	1387	(-0.51, 0.72)
Coarse		157		(-0.51, 0.72)		(-0.51, 0.72)
	50 (W2)		1346	(-1.00, 1.35)	705	(-1.08, 1.47)
	100 (W3)		1264	(-1.87, 2.41)	664	(-2.03, 2.63)
Coarse	200 (W4)	157	1100	(-3.23, 3.66)	582	(-3.51, 4.04)
	400 (W5)		772	(-5.51, 4.67)	418	(-5.91, 5.11)
	530 (W6)		557	(-7.16, 5.03)	309	(-7.60, 5.41)
	{25, 50} (W7)		2733	(-0.75, 1.11)	2733	(-0.75, 1.11)
	{50, 100} (W8)		2610	(-1.42, 1.98)	1369	(-1.54, 2.17)
	{25, 50, 100} (W9)		3997	(-1.11, 1.71)	3997	(-1.11, 1.71)
Coarse	{200, 400, 530} (W10)	157	2429	(-486, 4.62)	1309	(-5.24, 5.03)
	{50, 200, 400} (W11)		3218	(-2.84, 3.71)	1705	(-3.09, 4.07)
	{50, 200, 400, 530} (W12)		3775	(-3.48, 4.22)	2014	(-3.78, 4.60)

8.2 Hyperparameters of the Multilayer Perceptrons

Table 11: Hyperparameters, model complexity, and metrics of MLPs trained on W4, W5, W9 with $w_s = 25$ as well as $w_w = w_s = 1$. Common data set characteristics: Coarse, 2D stressor tables variant A (Wx-C-2D, variant A)

		Coarse, 2D stressor table, variant A			
		200 cycles (W4)	**400 cycles (W5)**	**{25, 50, 100} cycles (W9)**	**$w_w = w_s = 1$ (W13)**
Hyperparameters					
Activation Function		ReLU	ReLU	ReLU	ReLU
Batch Size		64	128	32	32
Learning Rate α		0.001	0.001	0.0001	0.001
Regularization $\{\lambda_1, \lambda_2\}$		{0, 0.001}	{0, 0.001}	{0, 0.001}	{0, 0}
Dropout rate		0	0	0	0.1
MLP layout		[330, 280, 155, 330, 305, 330, 105]	[330, 130, 255, 205, 305, 155]	[280, 180, 55, 55, 280, 130]	[330, 330]
Model Complexity					
No. of Hidden Layers		7	6	6	2
No. of Model Parameter		476,651	291,471	160,196	148,501
Metrics					
RMSE	**Train**	0.177	0.1813	0.134	0.03247
	Validation	0.2225	0.2016	0.1638	0.03118
	Test	0.2909	0.2967	0.1574	0.03572
R^2	**Train**	0.9984	0.999	0.9946	0.54993
	Validation	0.9962	0.9969	0.992	0.44363
	Test	0.9905	0.9953	0.9934	0.38807

8.3 Results 2D stressor tables, variant A

Table 12: Results of models trained on different data set configurations with 2D stressor tables, variant A, and $w_w = 25$ (Wx-2D, variant A, corresponding to Figure 6 and Figure 7)

Interval	Window width w_w [cycles]	RMSE			R^2		
		Train	Validation	Test	Train	Validation	Test
Fine	25 (W1)	0.0613	0.132	0.1267	0.9929	0.9681	0.9694
	50 (W2)	0.0752	0.1233	0.1775	0.9972	0.9917	0.9859
	100 (W3)	0.0852	0.1226	0.1537	0.999	0.998	0.994
	200 (W4)	0.1361	0.1432	0.1814	0.9993	0.9984	0.9963
	400 (W5)	0.1577	0.1798	0.2084	0.9991	0.9976	0.9977
	530 (W6)	0.1854	0.1961	0.3355	0.9991	0.9982	0.9961
	{25, 50} (W7)	0.0601	0.1152	0.0985	0.9972	0.9882	0.992
	{50, 100} (W8)	0.0821	0.0999	0.1333	0.9987	0.9972	0.9961
	{25, 50, 100} (W9)	0.0737	0.0895	0.084	0.9985	0.9976	0.9981
	{200, 400, 530} (W10)	0.1382	0.2036	0.1915	0.9993	0.9984	0.9983
	{50, 200, 400} (W11)	0.1086	0.1501	0.154	0.9992	0.998	0.9983
	{50, 200, 400, 530} (W12)	0.1331	0.1434	0.1571	0.9993	0.9988	0.9987
Medium	25 (W1)	0.1179	0.1257	0.1472	0.9886	0.971	0.9587
	50 (W2)	0.1042	0.1576	0.2052	0.9944	0.9864	0.9812
	100 (W3)	0.1366	0.1341	0.2082	0.9978	0.9976	0.989
	200 (W4)	0.1684	0.1762	0.2003	0.9988	0.9976	0.9955
	400 (W5)	0.1172	0.2191	0.2315	0.9994	0.9964	0.9971
	530 (W6)	0.1768	0.2657	0.2883	0.9989	0.9967	0.9971
	{25, 50} (W7)	0.1135	0.1191	0.1014	0.9958	0.9874	0.9915
	{50, 100} (W8)	0.1102	0.1294	0.1586	0.9971	0.9953	0.9945
	{25, 50, 100} (W9)	0.1466	0.114	0.121	0.9971	0.9961	0.9961
	{200, 400, 530} (W10)	0.3337	0.2653	0.3088	0.9969	0.9972	0.9956
	{50, 200, 400} (W11)	0.1642	0.2214	0.2468	0.9982	0.9957	0.9957
	{50, 200, 400, 530} (W12)	0.1835	0.218	0.2222	0.9987	0.9973	0.9975
Coarse	25 (W1)	0.1132	0.1461	0.1626	0.9872	0.9609	0.9496
	50 (W2)	0.1439	0.1786	0.2009	0.9915	0.9825	0.982
	100 (W3)	0.1534	0.1937	0.2613	0.9965	0.995	0.9827
	200 (W4)	0.177	0.2225	0.2909	0.9984	0.9962	0.9905
	400 (W5)	0.1813	0.2016	0.2967	0.999	0.9969	0.9953
	530 (W6)	0.181	0.313	0.3519	0.9989	0.9954	0.9957
	{25, 50} (W7)	0.119	0.1672	0.1426	0.9908	0.9752	0.9833
	{50, 100} (W8)	0.1861	0.171	0.2505	0.9937	0.9917	0.9863
	{25, 50, 100} (W9)	0.134	0.1638	0.1574	0.9946	0.992	0.9934
	{200, 400, 530} (W10)	0.3491	0.3334	0.3816	0.9969	0.9956	0.9933
	{50, 200, 400} (W11)	0.2139	0.319	0.3081	0.9974	0.9912	0.9933
	{50, 200, 400, 530} (W12)	0.2659	0.2857	0.3029	0.9969	0.9953	0.9953

Table 13: Results of models trained on different data set configurations with 2D stressor tables, variant A, and $w_w = 50$ (Wx-2D, variant A)

Interval	Window width w_w [cycles]	RMSE			R^2		
		Train	Validation	Test	Train	Validation	Test
Fine	25 (W1)	0.0642	0.1334	0.1227	0.9936	0.9674	0.9713
	50 (W2)	0.0774	0.1722	0.2156	0.9952	0.9872	0.9808
	100 (W3)	0.1105	0.2928	0.2483	0.9987	0.9887	0.9906
	200 (W4)	0.1611	0.2477	0.377	0.998	0.9924	0.9907
	400 (W5)	0.1432	0.2415	0.3847	0.9993	0.9968	0.9952
	530 (W6)	0.2936	0.407	0.6596	0.9967	0.9929	0.9837
	{25, 50} (W7)	0.0649	0.1159	0.1063	0.9974	0.9881	0.9907
	{50, 100} (W8)	0.082	0.1194	0.1395	0.9989	0.9965	0.9952
	{25, 50, 100} (W9)	0.0755	0.0906	0.0883	0.9983	0.9975	0.9979
	{200, 400, 530} (W10)	0.142	0.2356	0.2719	0.9993	0.9979	0.9971
	{50, 200, 400} (W11)	0.112	0.1961	0.2136	0.9993	0.9975	0.9973
	{50, 200, 400, 530} (W12)	0.1422	0.2024	0.2185	0.9994	0.9981	0.9976
Medium	25 (W1)	0.0715	0.1352	0.1377	0.9912	0.9665	0.9639
	50 (W2)	0.1313	0.1767	0.2139	0.9907	0.9865	0.9811
	100 (W3)	0.1717	0.3932	0.3681	0.9951	0.9796	0.9793
	200 (W4)	0.3094	0.274	0.4767	0.9949	0.9907	0.9851
	400 (W5)	0.2953	0.3428	0.4357	0.9983	0.9935	0.9938
	530 (W6)	0.3727	0.5268	0.4758	0.9966	0.9881	0.9915
	{25, 50} (W7)	0.1041	0.1327	0.119	0.9919	0.9844	0.9883
	{50, 100} (W8)	0.0949	0.1512	0.1999	0.9981	0.9943	0.9901
	{25, 50, 100} (W9)	0.1391	0.1144	0.1003	0.9973	0.9961	0.9973
	{200, 400, 530} (W10)	0.2366	0.3913	0.4573	0.9976	0.9943	0.9918
	{50, 200, 400} (W11)	0.2048	0.271	0.3325	0.9976	0.9952	0.9934
	{50, 200, 400, 530} (W12)	0.1632	0.2951	0.3612	0.9987	0.9959	0.9935
Coarse	25 (W1)	0.1201	0.1423	0.1693	0.986	0.9629	0.9454
	50 (W2)	0.2532	0.242	0.3285	0.9815	0.9748	0.9555
	100 (W3)	0.4565	0.4325	0.3209	0.9879	0.9753	0.9843
	200 (W4)	0.3143	0.2928	0.5666	0.9949	0.9893	0.9789
	400 (W5)	0.1847	0.2553	0.4244	0.9991	0.9964	0.9941
	530 (W6)	0.213	0.5367	0.4227	0.9988	0.9876	0.9933
	{25, 50} (W7)	0.1342	0.1725	0.145	0.9888	0.9736	0.9827
	{50, 100} (W8)	0.1796	0.1933	0.2142	0.9942	0.9907	0.9886
	{25, 50, 100} (W9)	0.1361	0.1659	0.1644	0.9944	0.9918	0.9928
	{200, 400, 530} (W10)	0.2316	0.3864	0.5554	0.9979	0.9944	0.9879
	{50, 200, 400} (W11)	0.2557	0.437	0.5167	0.9968	0.9875	0.984
	{50, 200, 400, 530} (W12)	0.2744	0.4847	0.4165	0.9963	0.989	0.9913