

# Predicting Battery Discharge Behavior for Sparse Data with Augmentation<sup>\*</sup>

Liza Obermeier<sup>4</sup>[0009-0001-5372-4024], Muhammad Daniel Bin Mohd Khir<sup>2,4</sup>[0009-0009-8080-6007], Niklas Küchen<sup>3,4</sup>[0009-0005-7538-0276], Tetyana Turiy<sup>1,4</sup>[0009-0006-3003-6153], Johanna Dolinga<sup>4</sup>, Thomas Gründer<sup>5</sup>, Benjamin Landenberger<sup>5</sup>, and Andre Ebert<sup>4</sup>[0000-0002-0629-5722]

<sup>1</sup> TUM Technische Universität München, 80333 München, Germany

<sup>2</sup> Universität Augsburg, 86159 Augsburg, Germany

<sup>3</sup> HKA Karlsruhe, University of Applied Sciences, 76133 Karlsruhe, Germany

<sup>4</sup> inovex GmbH, 76131 Karlsruhe, Germany

{liza.obermeier,daniel.binmohdkhir,niklas.kuechen,tetyana.turiy,  
johanna.dolinga,andre.ebert}@inovex.de

<sup>5</sup> Testo SE & Co. KGaA, 79822 Titisee-Neustadt, Germany

{blandenberger,tgruender}@testo.de

**Abstract.** Optimizing the use and maintenance of battery dependent systems is of great economic and ecological importance. In practice, however, batteries are often replaced too early to avoid downtime at all costs. As a result, valuable capacity is thrown away, leading to unnecessarily high costs and ecological impacts. To address this challenge, this work presents a comprehensive end-to-end workflow for predicting battery discharge behavior using sparse multivariate time series data. The proposed methodology encompasses five key steps: 1) *Data Acquisition*, 2) *Augmentation*, 3) *Preprocessing*, 4) *Feature Engineering*, and 5) *Modeling*. One of our key contributions is to enrich the limited available training data by applying augmentation techniques. In extensive experiments, including 270 evaluations across *Survival Analysis* and *Iterative Regression*, we demonstrate the effectiveness of our method. In particular, we show that incorporating augmented data improves the *Integrated Brier Score* by up to 50.16%. Moreover, we propose a novel, use-case-centric metric called *Mean Divergence Time* which estimates how long time series predictions remain reliable within a defined tolerance. Our results show that an accurate modeling of battery discharge curves in our use case is possible for an average of 117.76 days. This provides sufficient time to optimize the planning of maintenance intervals. Thereby, our work contributes to the modeling of battery lifetimes with sparse data and sets the stage for future research on the use of augmentation techniques to increase efficiency and resource optimization.

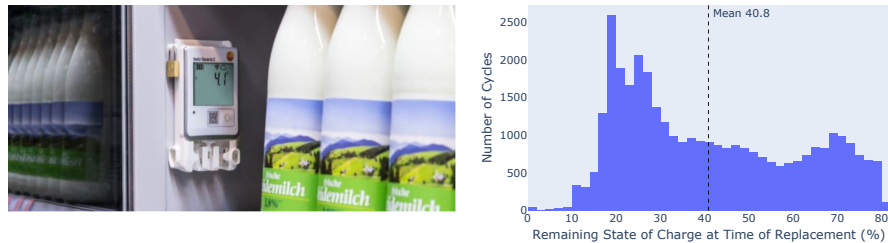
**Keywords:** Data Augmentation · Survival Analysis · Time Series Forecasting

---

<sup>\*</sup> Supported by *DeKIOps - Democratization of AI with understandable and easily accessible Machine Learning Operations* - a collaborative research project with *Fraunhofer IIS*, *senswork GmbH*, and *eresult GmbH* funded by the Bavarian State.

## 1 Introduction

Climate change and world wide resource shortage lead to challenges never seen before, economically as well as demographically. One way to address these challenges is the usage of Artificial Intelligence (AI), e.g., to enhance sustainability by significantly boosting efficiency in resource consumption. In particular, the accessibility of Machine Learning (ML) algorithms as well as their operation in real world scenarios remains challenging for domain experts without profound skills in AI development. Therefore, the *DeKIOps* research project aims to promote democratization of AI in the long-term by 1) reducing the complexity of technical interfaces for AI algorithms, e.g., by *AutoML*, and 2) pushing seamless interaction with AI at UI/UX level [4]. In this context, our research and the adjunctive use case described in this publication focus on resource-efficient prediction of battery lifetime and survival probabilities for distributed sensor platforms with sparse, multivariate, and augmented time series data. For modeling and evaluation, a historical dataset containing more than 35,677 battery discharge curves was provided by the research project’s industry partner *Testo SE & Co. KGaA* (Testo). The data was collected by *Saveris 2* sensor platforms (see Fig. 1a). An



(a) Testo *Saveris 2* platform monitoring the temperature of dairy products.

(b) Remaining SoC for 35,677 discharge cycles derived from 28,215 devices.

**Fig. 1:** Overview of a *Saveris 2* logger device provided by Testo and the State of Charge (SoC) distribution at replacement time.

analysis of remaining battery levels at replacement time strengthens the relevance of the proposed use case. Fig. 1b reveals that non-rechargeable batteries were discarded at an average State of Charge (SoC) of 40.8%. Projecting this premise onto real world settings indicates an enormous waste of resources. Besides that, a reliable long-term prediction of battery lifetime allows the optimized scheduling of maintenance windows. Additionally, downtimes can be minimized, and workforce as well as monetary expenses can be spared.

Still, a precise long-term prediction of battery lifetime is a non-trivial task. There are numerous influences suspected to affect discharge behavior, such as temperature, installed software, the connectivity, or logging intervals. Consequently, the relevance and individual effects of these factors need to be evaluated to finally extract expressive feature sets to create capable and robust predictors. Initially, de facto users often tend to have no or just little real world data at hand for modeling while high standards in data privacy and security aggravate

data sharing. Still today, mobile sensing devices have limited resources due to their design for long-term monitoring and their computation capabilities are not meant for the operation or training of complex models on device. To address those challenges, we highlight relevant related work in Sec. 2 and introduce a generic workflow for the preprocessing and augmentation of multivariate sensor data in Sec. 3. Special emphasis is laid on the handling of explicitly sparse datasets. Sec. 4 provides insights into 270 survival and regression experiments in order to evaluate the proposed workflow and the resulting predictors. Sec. 5 sums up this paper and provides a preview onto future work.

## 2 Related Work

The following section highlights relevant work that served as the basis for our proposed workflow and its evaluation.

*Survival analysis* is a well-known approach for end of life and hazard estimation in clinical scenarios. Multiple proposals predict battery life of medical equipment by utilizing *Kaplan-Meier* (KM) in combination with *log-rank test* and other procedures [9, 13]. Therewith, Kuhadja et al. manage to reduce the risk of unexpected battery failure from 28% to 7%. Similar approaches including semi-parametric models like *Cox Proportional-Hazards* (CPH) in combination with log-rank for time-to-event interpretation are described by [16, 6]. Here, Li-Ion battery degradation is predicted in form of the cell’s survival probability and a *Cumulative Hazard Function* (CHF) as battery risk. Additionally, Ibraheem et al. introduce a procedure to strengthen data region agnostic predictions, which might help to overcome cold-start problems in real life application [6].

Another approach for end-of-life prediction is offered by *Random Survival Forests* (RSF) as applied, e.g., by Frisk et al. for data-driven lead-acid battery prognostics in combination with augmentation techniques [5]. In general, examples for survival analysis using augmented data sources are sparse. Shourabizadeh et al. present *Classification-Augmented Survival Estimation* (CASE), a novel method for individualized long-term survival prediction, and employ it to forecast survival probabilities of patients for > 20 years after a liver transplantation. They prove superior performance for augmented data with improved AUCs from 0.69 to 0.88 and F1 scores from 0.32 to 0.73 as well as a better performance compared to vanilla approaches of KM, RSF, and CPH when benchmarked on common existing survival metrics [5, 6, 16]. Rajaram et al. propose data augmentation with cross-modal variational autoencoders (DACMVA) and evaluated their approach by predicting cancer. Hereby, they outperformed non-augmented baseline methods significantly by augmenting varying percentages of missing data. Even in very-sparse-data settings and in comparison to state-of-the-art methods, such as TDImpute or oversampling alone, significant performance improvements were attested [14].

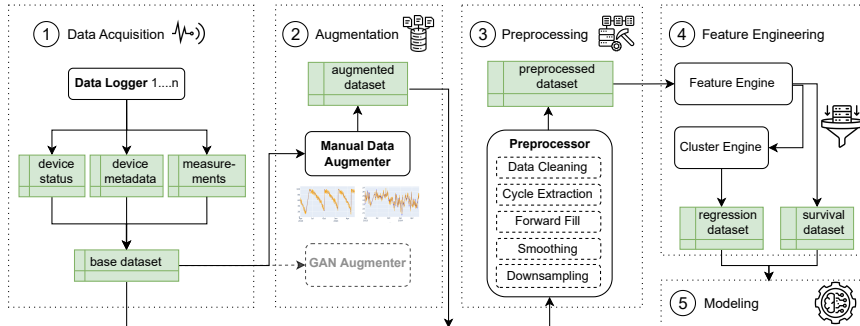
In contrast to the prediction of survival probability over time, there are also examples of predicting an expected time-to-event by utilizing iterative regression. Zhang et al. extract geometric features from the mid- to low-frequency impedance spectrum and use a probabilistic prediction based on linear regres-

sion to predict remaining useful life of a battery in early stages of operation [20]. A further common practice is the use of *Gaussian Mixture Regression* (GMR) models for the *State of Health* (SoH) prediction of batteries along with incremental capacity analysis [22, 10, 12]. In this context, Liu et al. show that SoH can be estimated accurately based on GMR even for different batteries, where MAE and RME rates of less than 1% are achievable [12]. Recent work deals with the appliance of XGBoost regression (XGB) instead of GMR in order to feature a precise lifetime estimation for retired energy storage batteries [1]. In this regard, Zhou et al. present a concept for fast capacity estimation of Li-Ion batteries based on XGB and electrochemical impedance spectroscopy. They validated on the NASA dataset with an accuracy of 99.78% for estimated lifetime [21].

All these studies dealing with survival analysis, end-of-lifetime prediction and techniques to augment domain-specific data are highly valuable inputs for our proposed workflow and its evaluation.

### 3 Concept

In this section, we propose a workflow for predicting battery lifetimes, starting with the acquisition of raw data. An overview is provided in Fig. 2. Its final goal is the creation of practically applicable models with low complexity that are capable of rendering highly accurate survival probability and time series predictions on basis of sparse data.



**Fig. 2:** High-level overview of the proposed workflow which extends a typical process chain for time series analysis covering 1) *Data Acquisition*, 2) *Augmentation*, 3) *Preprocessing*, 4) *Feature Engineering*, and 5) *Modeling* [3].

#### 3.1 Data Acquisition

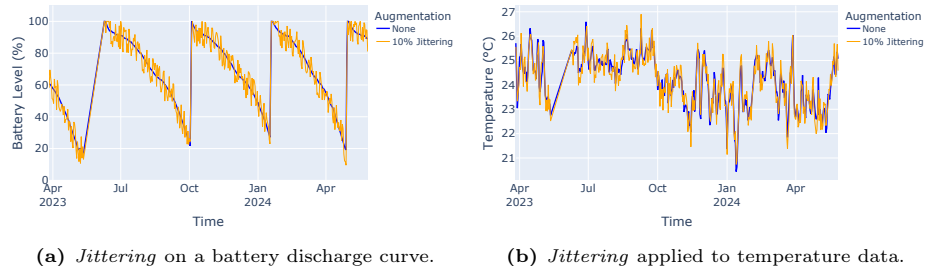
First step within the presented workflow is the acquisition of a comprehensive dataset as basis for further analysis and modeling. Therefore, we utilized historical information of 35,677 discharge curves from 28,215 different *Saveris 2* sensor devices. Thereby, the devices measure the ambient air temperature in a potential range of  $-30^{\circ}\text{C}$  to  $50^{\circ}\text{C}$  with an accuracy of  $\pm 0.5^{\circ}\text{C}$  and a corresponding resolution of  $0.1^{\circ}\text{C}$ . They are built for long lasting measurement periods up

to years. Measurement of other data types is viable by attaching different external probing units to the sensor platform. Initially, the obtained data splits into three individual datasets: 1) meta data, containing static information, such as the device model code, firmware version, and used battery type, 2) multivariate device status, e.g., battery State of Charge (SoC) and radio level and 3) the actual sensor measurements encoded as time series, i.e., the air temperature. All data points from 1), 2), and 3) are mappable by unique device ids that are used to merge them in accordance to their respecting timestamps. Thereby, a unified base dataset is formed, which is the origin for all preceding steps into both considered directions: survival analysis and time series forecasting.

### 3.2 Augmentation

Subsequently, we describe the 2) *Augmentation* engine as depicted in Fig. 2. As proposed by Wen et al., we evaluated two approaches for data augmentation, namely *Manual Data Augmentation* and *Advanced Approaches*, e.g., the utilization of *Generative Adversarial Networks* (GANs) [18].

**Manual Data Augmentation** In order to maintain inherited patterns and distribution of the original data, the input for the augmentation step is raw data as provided by 1) *Data Acquisition*. Up-following, we introduce different man-

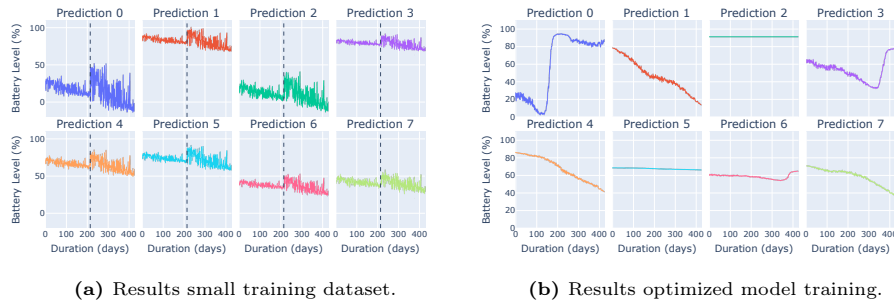


**Fig. 3:** Applied *Jittering* on SoC and temperature data of the *Saveris 2* platform.

ual augmentation approaches and discuss their applicability for our scenario. This envisages the availability of limited resources, which points to easy-to-use techniques of low complexity, i.e., those of the time-domain, possibly disregarding aspects of frequency and time-frequency. *Cropping* and *Window Slicing*, are efficient sub-sampling methods used to extract slices from time series [2]. For a time series  $X = (x_1, \dots, x_n)$  of length  $n$ , a sliced time series is defined as  $X'(i, j) = (x_i, \dots, x_j)$  with  $1 \leq i \leq j \leq n$ . As learned from the provided historical information, the de facto available data (in our use case) is sparse and includes a significant amount of right- and left-censored samples, whereby these approaches are not suitable. *Flipping* mirrors data on a specific axis, e.g.,  $X' = (-x_1, \dots, -x_n)$ . When applied to battery charge curves downward trends in the data are lost. In contrast, *Scaling* modifies the magnitude of the individual signal components while preserving its original shape:  $X'(\alpha) = (\alpha x_1, \dots, \alpha x_n)$  with  $\alpha > 0$  [7]. This can yield physically unrealistic curves that no longer follow

electrochemical laws and therefore do not reflect real battery behavior. *Jittering* adds random noise to the time series data while retaining the temporal dimension:  $X'(\epsilon) = (x_1 + \epsilon_1, \dots, x_n + \epsilon_n)$ , whereby  $\epsilon_t$  is the noise added at time  $t$ . Since it does not discard data points and also retains the main characteristics of the original data, it is perceived as adequate augmentation strategy (see Fig. 3).

**Generative Adversarial Networks** We also experimented with GANs for augmentation as documented in related research and various open source implementations, e.g., *DoppelGANger* (DGAN) [11, 19]. Utilizing this, we trained a

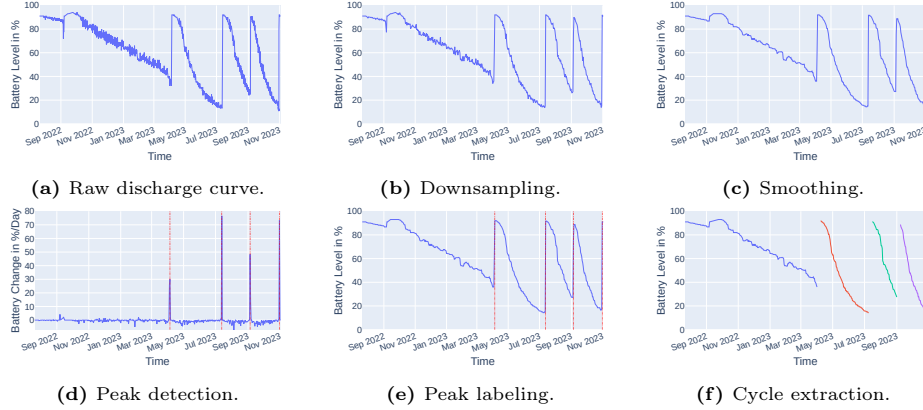


**Fig. 4:** Augmentation examples of time series with *DoppelGANger* using small (see Fig. 4a) versus large (see Fig. 4b) amounts of training data plus optimized hyperparameters.

baseline model with a sparse data subset of 200 devices and evaluated its generative performance. As shown in Fig. 4a, the resulting sequences are noisy and they do not correlate with the actual shape of real world discharge curves. Moreover, the entire range of battery states occurring in real scenarios is not covered. To overcome these limitations, we enlarged the training dataset by factor 10 and applied hyperparameter tuning. Those measures drastically reduced noise and produced much more usable shapes, as depicted in Fig. 4b. Despite promising results, *Deep Learning* and GAN-based approaches require large amounts of data not available in the proposed scenario. One could argue that pre-trained network layers may offer a solution to this limitation, but due to commonly applied restrictions in data privacy pre-trained structures are not applicable in general. To sum up, GANs are not suitable for our use case that focuses on low-complexity algorithmic approaches, sparse data usage, and resource efficiency.

### 3.3 Preprocessing

The base dataset and the augmented dataset emerging from Sec. 3.1 and 3.2 are now fed into the *Preprocessor*. Fig. 5a shows a raw discharge curve consisting of multiple *cycles*, corresponding to the time between battery installation and replacement. Extracting individual cycles is challenging as the timestamps of battery replacement are unlabeled. Some curves fluctuate heavily and prevent the identification of obvious trends, others show anomalous discharge behaviour in terms of static SoC over time. Other challenges arise due to inconsistent measurement intervals.



**Fig. 5:** Overview across the transformations applied in the preprocessing pipeline.

**Downsampling, Smoothing, and Forward Fill** First, all measurement intervals are downsampled to a daily frequency (see Fig.5b). Here, a daily sampling frequency holds an appropriate balance between precision and generalization, for long-term maintenance intervals are commonly planned on a daily basis. Subsequently, median smoothing with a rolling window of 5 days is used to further stabilize the battery curve (see Fig.5c). The median ensures the preservation of expressive SoC values for subsequent peak detection. Occurring measurement interruptions, e.g., due to offline devices, lead to gaps within the dataset ranging from days to months. This was addressed by applying data imputation with forward filling [15].

**Cycle Extraction** Subsequently, a peak detection algorithm is applied to the smoothed discharge curves. First, the derivative of the discharge curve is determined using *Forward Difference Approximation*  $f'(t) = \frac{f(t+1) - f(t)}{\Delta t}$  (see Fig. 5d). By calculating the difference between neighboring data points, i.e.,  $f'(t) = f(t+1) - f(t)$ , the absolute change in SoC can be used to identify peaks (see Fig. 5e). Due to residual noise, a simple peak detection based on sign changes within derivatives is not applicable. Since battery changes constitute sudden and significant spikes in the SoC, a comparably high threshold can be selected to yield robust results (see Fig. 5f).

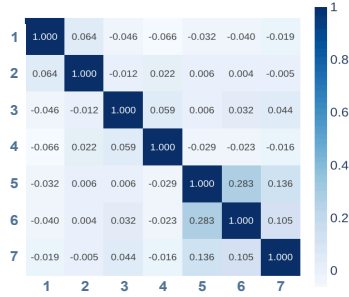
**Data Cleaning** Finally, insufficient cycles, i.e., those without expressive SoC range (28.9%) or with less than 25 observations (6.6%), are identified and removed as well as data that is left- and right-censored. The resulting preprocessed dataset contains 35,677 cycles. All relevant features were downsampled to a daily basis. Daily mean was used to ensure continuous availability for numerical features, for categorical ones a majority voting was conducted.

### 3.4 Feature Engineering

As part of feature engineering, we derive expressive features for each modeling approach. Prior to this, we briefly describe the respective inputs and targets.

Survival analysis methods operate on tabular time-to-event data associating events of interest with timestamps [16]. Targets are binary indicators signaling whether or not an event occurred. As batteries are changed at varying SoC levels, their replacement is unsuitable as target event. Trained models would predict battery replacement times with an unknown residual SoC, yielding inconsistent predictions. Instead, we predefine a SoC threshold as target event to model the time when a particular battery level is reached. Therefore, we truncate all cycles at the SoC threshold and consider those not surpassing it as right-censored. The threshold is selected in a way that ensures an optimized battery usage without service interruption while offering end users enough flexibility to effectively plan maintenance intervals. On average, batteries are replaced when they have a residual charge of 40.8% (see Fig. 1b). We chose a target SoC of 20% as it already substantially improves the current situation.

Input for the regression models are multivariate time series. The target is the battery level shifted by the prediction horizon, i.e., their goal is to replicate battery cycles. In our scenario, the prediction horizon is set to 1 day.



**Tab. 1:** Description of continuous variables.

Variable	Description
1	SoC (percent)
2	SoC delta to previous day (percent)
3	Air temperature (C°)
4	Radio level (percent)
5	Sensor interval (min)
6	Daytime communication interval (min)
7	Nighttime communication interval (min)

**Fig. 6:** Pearson's correlation coefficients of continuous variables [17].

To obtain a non-redundant feature set, we investigate the correlation of the preprocessed variables (see Fig. 6). Pearson's correlation coefficients  $\rho$  with  $-1 \leq \rho \leq 1$  express the linear correlation of two variables. Negative values show an opposing, positive values an equal trend. If  $\rho(v_1, v_2) = 0$ , then  $v_1$  and  $v_2$  are uncorrelated. The coefficients reveal a generally low degree of collinearity in the preprocessed dataset. None of the variables has a notable linear relationship to the SoC. As a result, the variables 2 to 7 in Tab. 1 are considered uninformative.

For survival analysis, the cycle duration in days, the minimum and maximum SoC, and the one-hot-encoded categorical variables, i.e., firmware version, device model code, and battery type (lithium or alkaline-manganese), are used as final features. The categorical features as well as the median SoC over a 5-day and a 50-day rolling window serve as input for regression modeling. The rolling window aggregations are used to provide information about short- and long-term trends.

### 3.5 Modeling

To evaluate the preceding workflow, we trained *Kaplan-Meier* (KM), *Cox Proportional-Hazards* (CPH), and *Random Survival Forest* (RSF) models for survival analysis

(see Tab. 2). For time series prediction with iterative regression we used *Linear Regression* (LR), *XGBoost* (XGB), *Decision Tree* (DT) (see Tab. 4). The selection and configuration originates from related work in Sec. 2.

## 4 Evaluation

Up-following, we present results for two independent evaluation series. One includes 126 survival analysis experiments (see Sec. 4.1) and one with 144 experiments focusing on time series prediction by iterative regression (see Sec. 4.2).

### 4.1 Survival analysis experiments

**Tab. 2:** Overview of the different indices and algorithms from which the number of conducted survival experiments is derived.

Parameter	Options	Multiplicator
Number of devices $n$	[10, 20, 40, 80, 160, 320]	6
Added parts $p$ of augmented data	[1, 2, 4]	3
Strength of jittering $s_j$	[0.1, 0.2]	2
Model types <i>Survival Analysis</i> $m_s$	[KM, CPH, RSF]	3

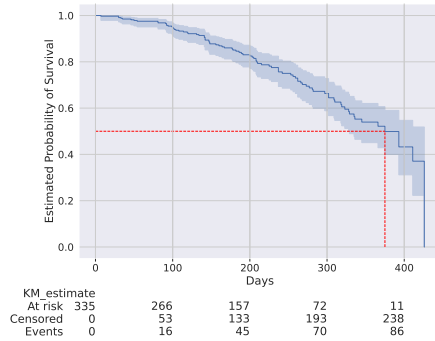
Tab. 2 provides insights into different indices and modeling algorithms derived from related work (see Sec. 2). They form the basis for the final number of experiments conducted. As depicted,  $n(6) \times m_s(3) = 18$  non-augmented and  $n(6) \times p(3) \times s_j(2) \times m_s(3) = 108$  augmented datasets are created, leading to a total of 126 experiments performed for models of type  $m_s$ . In each experiment, a 4-fold cross-validation was carried out to train and validate 4 individual models. To allow for better comparability, we subjected each trained model to a final performance benchmark on the same holdout dataset (see results in Tab. 3). As a performance metric, we used *Integrated Brier Score* (IBS). It ranges between 1 and 0 with 0 being the best possible score and reflects how accurate a survival function is at a given time  $t$ . The results show that - especially when the number of available devices to supply training data is low ( $n = 10$ ) - the addition of augmented data leads to a significant performance gain<sup>6</sup> (on average +42.17% over all models), with RSF profiting the most from augmentation with a performance gain of 50.16%. The benefit diminishes, when more real data is available ( $n = 20$ , on average +6.93%) and is barely noticeable for  $n > 20$ .

Overall, the best IBS was achieved by utilizing RSF. Astoundingly, augmentation enables us to use data of only  $n = 10$  devices in order to achieve results comparable to those of  $n = 80$  devices without augmentation. This highlights the strong boost for predictions offered by augmentation techniques. Fig. 7 visualizes the predictions of all models trained on the same sample dataset with 335 discharge cycles extracted from  $n = 320$  devices. Fig. 7a shows the estimated probability of survival with KM, whereby the predicted maximum range is  $d \approx 426$  days with a moderately widening confidence interval. The horizontal downward steps represent the number of cycles where the target event occurred

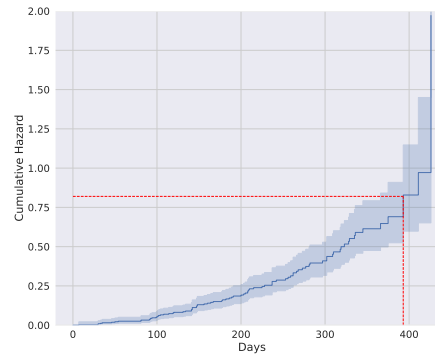
<sup>6</sup> Performance gain (in %) per model type listed in Tab. 3 is defined as  $1 - \frac{\min(\text{IBS})}{\text{IBS}_{p=0}}$ .

**Tab. 3:** Comparison of mean IBS values for all experiments per model type, added parts  $p$  of augmented data and number of devices  $n$  available for training.

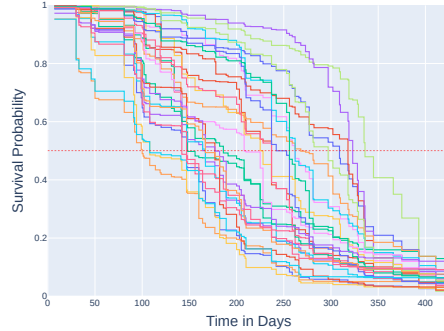
Model type	Parts $p$ of aug. data	$\emptyset$ IBS					
		n=10	n=20	n=40	n=80	n=160	n=320
KM	0	0.317	0.205	0.244	0.235	0.239	<b>0.245</b>
KM	1	0.265	<b>0.191</b>	0.244	0.230	<b>0.221</b>	0.248
KM	2	<b>0.167</b>	0.197	0.243	<b>0.228</b>	0.223	0.247
KM	4	<b>0.167</b>	0.197	<b>0.236</b>	0.231	0.224	0.247
CPH	0	0.317	0.209	0.221	0.206	0.179	<b>0.174</b>
CPH	1	0.293	0.184	0.221	<b>0.192</b>	0.171	0.178
CPH	2	<b>0.225</b>	<b>0.185</b>	0.220	0.195	0.171	0.176
CPH	4	<b>0.225</b>	<b>0.185</b>	<b>0.213</b>	0.195	<b>0.170</b>	0.177
RSF	0	0.317	0.166	0.164	<b>0.148</b>	0.127	0.093
RSF	1	0.263	<b>0.154</b>	<b>0.162</b>	0.150	<b>0.119</b>	0.095
RSF	2	0.160	0.157	<b>0.162</b>	0.153	<b>0.119</b>	<b>0.092</b>
RSF	4	<b>0.158</b>	0.157	0.166	0.156	0.121	<b>0.092</b>



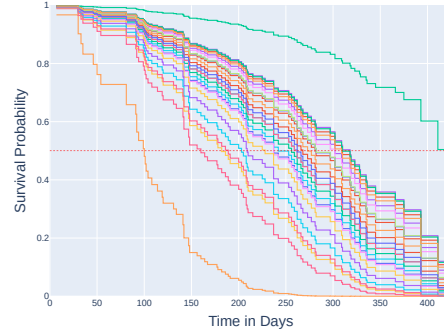
(a) Probability of survival with *Kaplan-Meier* estimator.



(b) *Cumulative Hazard Function* modeled via *Nelson-Aalen Fitter* [8].



(c) Example predictions for survival probabilities with *Random Survival Forest*.



(d) Example predictions for survival probabilities with *Cox Proportional Hazards*.

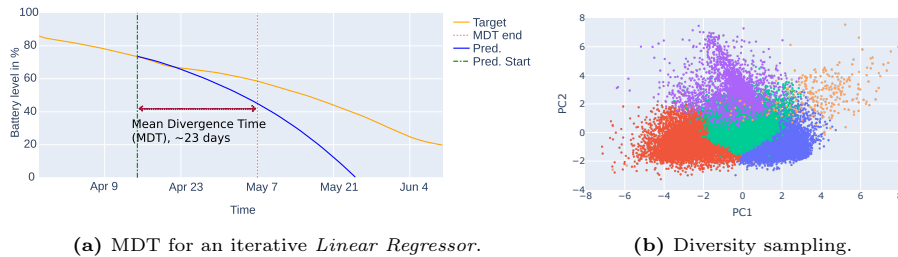
**Fig. 7:** Probability of survival (Fig. 7a), *Cumulative Hazard Function* (Fig. 7b), and exemplary results of predicted survival probabilities for different models (Fig. 7c and 7d), all based on the same sample dataset.

at a particular time step (i.e., the SoC threshold was reached). It is noticeable that the expected event, i.e., a SoC of 20%, occurs suddenly for most devices at risk when survival probability is lower than  $s(t) = 0.5$  after  $d \approx 375$  days of survival. A complementary observation can be made for the CHF as depicted in Fig. 7b. A massive increase of cumulative hazard can be observed at  $H(t) \approx 0.82$  after  $d \approx 393$  days of survival. Additionally, Fig. 7c and 7d show example predictions of the trained RSF and CPH models. The red dotted line marks the survival probability  $s(t) = 0.5$  and simultaneously indicates the inflection point, where the frequency of observed events declines with a decreasing probability of survival. It is visible, that CPH tends to predict slower battery discharge rates in comparison to RSF models. As Tab. 3 shows, the RSF predictions are more trustworthy compared to CPH. As the median survival times for CPH are heavily skewed to  $> 250$  days, they capture the event distribution less accurately as shown in Fig. 7d.

#### 4.2 Time series prediction with iterative regression

Now we present the results for time series prediction by utilizing 1) unsupervised learning for diversity sampling in combination with 2) different iterative regression approaches. Thereby, a total of  $n(6) \times m_r(4) \times c(6) = 144$  experiments are conducted with  $n = 6$  different amounts of devices used for training,  $m_r = 4$  regression model types, and  $c = 6$  clusters. Besides the 3 model types listed in Sec. 3.5, we use a constant baseline (BL) defined as  $BL(x_t) = x_{t-1}$  for benchmarking. Within each of the 144 experiments, multiple models were trained. One without and the others with augmented data according to the parameters listed in Tab. 2.

For performance evaluation, we introduce a novel use-case-centric metric called *Mean Divergence Time* (MDT), which is tailored to measure how long a prediction remains reliable within a defined tolerance interval. It corresponds to the observed average time span over all trained models within an experiment where the predictions do not deviate more than 10% from the true SoC, allowing a tolerance window of  $t_{mdt}$  days (see Fig. 8a).

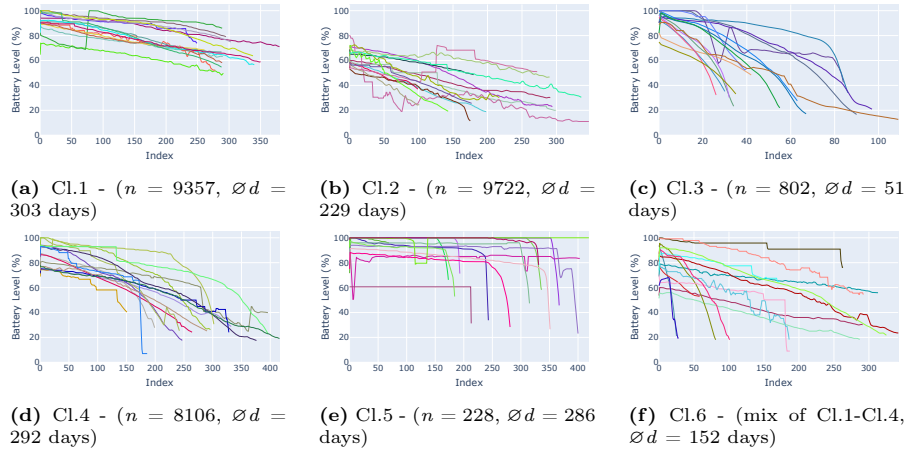


**Fig. 8:** Examples of MDT determination and diversity sampling with k-Means.

Thereby,  $t_{mdt}$  is highly application-dependent. Here, for the prediction of battery lifetime for mobile sensor platforms,  $t_{mdt} = 5$  proved to be adequate.

Consequently, all extracted cycles were clustered with k-Means for a subsequent diversity sampling, whereby  $k = 5$  was derived using the elbow method (see 8b). First, regression models were trained for each cluster separately to actually strengthen overfitting by preferring similar cycles. The basic assumption is, that this enhances the prediction performance for cases, where models need to be created with limited data originating from devices operating in the same or similar environments.

Fig. 9 illustrates the actual clusters, where  $n$  corresponds to the number of included devices and  $d$  to the average duration in days of the assigned cycles: Cl.1 with mainly flat and long cycles (see Fig. 9a), Cl.2 with a maximum initial SoC of 80% (see Fig. 9b), Cl.3 with shorter cycles (see Fig. 9c), Cl.4 with steep cycles of a longer runtime (see Fig. 9d), Cl.5 with outliers exhibiting a constant SoC (see Fig. 9e and also the orange cluster in Fig. 8b), and finally Cl.6 which represents a diversity sampled mixture of all clusters excluding the outliers of Cl.5 (see Fig. 9f). Tab. 4 shows the determined MDT values of the best models



**Fig. 9:** Random samples per cluster with  $n$  devices and an average duration of  $d$  days.

for each cluster. Models trained without augmented data are marked with asterisks. In 110 of 144 experiments (76.39%), augmentation led to better results. The best scores were achieved for Cl.1 with a mean MDT of 117.76 days, while Cl.3 scored worst with an average MDT of 18.54 days. Cl.6 incorporates a diversity sampled data basis and achieved a competitive MDT of 53.82 days. Cl.5 exclusively contains outliers with a constant SoC, thereby they are easiest to model with our baseline BL. Hence, BL experiments can be ignored for overall performance contemplation, resulting in 89 out of  $n(6) \times m_r(3) \times c(6) = 108$  remaining experiments (82.41%) achieving better performance with augmentation. These results are highly valuable for maintenance planning and again, a significant performance boost due to augmentation for the prediction capabilities in iterative regression is evident.

**Tab. 4:** Best models per Cluster (Cl.) sorted by  $n$  devices used for model training - absence of augmented data in training dataset is marked with asterisks.

Cl.	$n$	Model	MDT $\uparrow$	Cl.	$n$	Model	MDT $\uparrow$	Cl.	$n$	Model	MDT $\uparrow$																																																																																																
1	10	XGB	108.36	2	10	XGB	84.51	3	10	DT	15.41																																																																																																
	20	DT	111.18		40	LR	118.06		80	DT-*	116.35	160	LR	122.29	320	LR	130.34	4	10	DT	62.53	5	10	BL	107.51	6	10	DT	75.06	20	DT	81.28	40	DT	80.78	80	DT	81.38	160	LR	85.12	320	LR-*	87.21		10	XGB	84.51		20	LR	70.16		20	DT-*	22.56	40	LR	85.53	80	LR-*	80.89	160	LR	88.82	320	LR	87.98		10	DT	15.41		40	LR	85.53		40	DT	19.94	20	DT-*	22.56	40	DT	19.94	80	DT	18.02	160	XGB-*	17.79	320	XGB	17.52												
	40	LR	118.06		80	DT-*	116.35		160	LR	122.29	320	LR	130.34	4	10	DT		62.53	5	10		BL	107.51	6		10	DT	75.06	20	DT	81.28	40	DT	80.78	80	DT	81.38	160	LR	85.12	320	LR-*	87.21			10	XGB		84.51		20		LR	70.16		20	DT-*	22.56	40	LR	85.53	80	LR-*	80.89	160	LR	88.82		320	LR	87.98			10	DT		15.41		40	LR	85.53		40	DT	19.94	20	DT-*	22.56	40	DT	19.94	80	DT	18.02	160	XGB-*	17.79	320	XGB	17.52						
	80	DT-*	116.35		160	LR	122.29		320	LR	130.34	4	10	DT		62.53	5		10		BL		107.51	6			10	DT	75.06	20	DT	81.28	40	DT	80.78	80	DT	81.38	160	LR	85.12	320	LR-*	87.21				10		XGB		84.51			20		LR	70.16		20	DT-*	22.56	40	LR	85.53	80	LR-*	80.89		160	LR	88.82			320	LR		87.98			10	DT		15.41		40	LR	85.53		40	DT	19.94	20	DT-*	22.56	40	DT	19.94	80	DT	18.02	160	XGB-*	17.79	320	XGB	17.52
	160	LR	122.29																																																																																																								
320	LR	130.34	4	10	DT	62.53	5	10	BL	107.51	6		10	DT		75.06			20		DT		81.28				40	DT	80.78	80	DT	81.38	160	LR	85.12	320	LR-*	87.21		10	XGB	84.51		20				LR		70.16					20		DT-*	22.56		40	LR	85.53	80	LR-*	80.89	160	LR	88.82		320	LR	87.98				10		DT			15.41			40		LR	85.53			40	DT	19.94	20	DT-*	22.56	40	DT	19.94	80	DT	18.02	160	XGB-*	17.79	320	XGB	17.52
4	10	DT		62.53	5	10		BL	107.51	6			10	DT		75.06																																																																																											
	20	DT		81.28		40		DT	80.78				80	DT	81.38	160		LR	85.12	320	LR-*	87.21			10	XGB	84.51		20	LR	70.16		20	DT-*	22.56	40	LR	85.53		80	LR-*	80.89		160	LR	88.82		320	LR	87.98			10		DT	15.41		40		LR	85.53		40	DT	19.94	20	DT-*	22.56	40	DT	19.94	80	DT	18.02		160	XGB-*	17.79	320		XGB		17.52																								
	40	DT		80.78		80		DT	81.38			160	LR	85.12	320	LR-*	87.21		10	XGB	84.51			20	LR	70.16			20	DT-*	22.56		40	LR	85.53	80	LR-*	80.89		160	LR	88.82		320	LR	87.98		10	DT	15.41				40	LR	85.53			40	DT	19.94		20	DT-*	22.56	40	DT	19.94	80	DT	18.02	160	XGB-*	17.79		320	XGB	17.52																													
	80	DT		81.38		160		LR	85.12			320	LR-*	87.21		10	XGB		84.51		20			LR	70.16				20	DT-*	22.56		40	LR	85.53	80	LR-*	80.89		160	LR	88.82		320	LR	87.98			10	DT				15.41		40			LR	85.53			40	DT	19.94	20	DT-*	22.56	40	DT	19.94	80	DT	18.02		160	XGB-*	17.79	320	XGB	17.52																										
	160	LR	85.12																																																																																																								
320	LR-*	87.21		10	XGB	84.51		20	LR	70.16		20	DT-*	22.56		40	LR		85.53		80			LR-*	80.89				160	LR	88.82		320	LR	87.98		10	DT	15.41		40	LR	85.53		40	DT			19.94	20		DT-*		22.56		40			DT	19.94			80	DT	18.02	160	XGB-*	17.79	320	XGB	17.52																																				
	10	XGB		84.51		20		LR	70.16			20	DT-*	22.56																																																																																													
	40	LR		85.53		80		LR-*	80.89			160	LR	88.82		320	LR	87.98			10	DT	15.41		40		LR	85.53		40	DT	19.94	20	DT-*	22.56		40	DT	19.94		80	DT	18.02		160	XGB-*	17.79		320	XGB	17.52																																																								
	80	LR-*		80.89		160		LR	88.82			320	LR	87.98		10	DT	15.41			40	LR	85.53			40	DT	19.94		20	DT-*	22.56	40	DT	19.94		80	DT	18.02		160	XGB-*	17.79		320	XGB	17.52																																																												
	160	LR		88.82		320		LR	87.98				10	DT		15.41		40			LR	85.53				40	DT	19.94		20	DT-*	22.56	40	DT	19.94		80	DT	18.02		160	XGB-*	17.79		320	XGB	17.52																																																												
	320	LR	87.98																																																																																																								
	10	DT	15.41		40	LR	85.53		40	DT	19.94																																																																																																
	20	DT-*	22.56		40	DT	19.94		80	DT	18.02		160	XGB-*		17.79		320	XGB		17.52																																																																																						
	40	DT	19.94		80	DT	18.02		160	XGB-*	17.79		320	XGB	17.52																																																																																												
	80	DT	18.02		160	XGB-*	17.79		320	XGB	17.52																																																																																																
	160	XGB-*	17.79																																																																																																								
320	XGB	17.52																																																																																																									

## 5 Summary and Outlook

Within this publication, we present a generic workflow to train survival and iterative regression models capable of time series prediction and modeling battery discharge behavior based on sparse, multivariate time series. Their effectiveness was verified in overall 270 experiments resulting in highly competitive performance indicators. We proved that augmentation boosts IBS significantly and achieved a maximum performance gain of 50.16% in scenarios with sparse data availability. For time series forecasting, we introduced the *Mean Divergence Time* as a to our knowledge novel, use-case-centric metric for benchmarking the applicability of time series predictions in the real world. We achieved an MDT value of 117.76 days in best cases. Again, 82.41% of the best-performing regression models utilized augmentation.

There are still open questions of technical nature and use-case-specific. First, we mainly focus on manual augmentation and feature engineering. Sophisticated (hyper-)parameter tuning is still pending. When disregarding constraints of resource scarcity, more complex augmentation techniques and prediction models become feasible, e.g., *Autoencoders*, or *Deep Learning*. Therefore, this work provides a strong fundament for subsequent use cases of higher complexity, e.g., the optimization of operational intervals with outlier detection. These and other aspects will be covered in our ongoing research within the *DeKIops* project.

## References

- Chen, X., Hua, J., Yin, C., Chen, Z., Chen, Y., Li, Z.: A precise life estimation method for retired energy storage batteries based on energy storage batteries attenuation characteristics and xgboost algorithm. *IEEE Access* **11**, 135968–135978 (2023)
- Cui, Z., Chen, W., Chen, Y.: Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995* (2016)
- Ebert, A., Beck, M.T., Mattausch, A., Belzner, L., Linnhoff-Popien, C.: Qualitative assessment of recurrent human motion. In: 2017 25th European Signal Processing Conference (EUSIPCO). pp. 306–310. IEEE (2017)

4. Ebert, A., Kempter, J., Siebold, M., Pesch, R., Turiy, T., Tchuinkam, T., Sune, T.C.: Autotim - an open-source service for automated provisioning and operation of time series based machine learning models. In: IFIP International Conference on Artificial Intelligence Applications and Innovations. pp. 266–278. Springer (2023)
5. Frisk, E., Krysander, M., Larsson, E.: Data-driven lead-acid battery prognostics using random survival forests. In: Annual Conference of the PHM Society. vol. 6 (2014)
6. Ibraheem, R., Cannings, T.I., Sell, T., dos Reis, G.: Robust survival model for the prediction of li-ion battery lifetime reliability and risk functions. *Energy and AI* **19**, 100465 (2025)
7. Iglesias, G., Talavera, E., González-Prieto, Á., Mozo, A., Gómez-Canaval, S.: Data augmentation techniques in time series domain: A survey and taxonomy. *Neural Computing and Applications* **35**(14), 10123–10145 (2023)
8. Jiang, R.: A bias-corrected nelson-aalen estimator. In: IOP Conference Series: Materials Science and Engineering. vol. 1043, p. 022013. IOP Publishing (2021)
9. Kuhajda, D.: Using survival analysis to evaluate medical equipment battery life. *Biomedical Instrumentation & Technology* **50**(3), 184–189 (2016)
10. Li, K., Xie, N.: Battery health prognostics based on improved incremental capacity using a hybrid grey modeling and gaussian process regression. *Energy* p. 131888 (2024)
11. Lin, Z., Jain, A., Wang, C., Fanti, G., Sekar, V.: Generating high-fidelity, synthetic time series datasets with doppelganger. arXiv preprint arXiv:1909.13403 (2019)
12. Liu, K., Li, Y., Hu, X., Lucu, M., Widanage, W.D.: Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries. *IEEE Transactions on Industrial Informatics* **16**(6), 3767–3777 (2019)
13. Ondo, W.G., Meilak, C., Vuong, K.D.: Predictors of battery life for the activa® solettra 7426 neurostimulator. *Parkinsonism & Related Disorders* **13**(4), 240–242 (2007)
14. Rajaram, S., Mitchell, C.S.: Data augmentation with cross-modal variational autoencoders (dacmva) for cancer survival prediction. *Information* **15**(1), 7 (2023)
15. Ribeiro, S.M., Castro, C.: Missing data in time series: A review of imputation methods and case study. *Learning and Nonlinear Models* **20**(1), 31–46 (2022)
16. Schober, P., Vetter, T.R.: Survival analysis and interpretation of time-to-event data: The tortoise and the hare. *Anesthesia & Analgesia* **127**(3), 792–798 (2018)
17. Sedgwick, P.: Pearson’s correlation coefficient. *Bmj* **345** (2012)
18. Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H.: Time series data augmentation for deep learning: A survey. arXiv preprint arXiv:2002.12478 (2020)
19. Yoon, J., Jarrett, D., Van der Schaar, M.: Time-series generative adversarial networks. *Advances in neural information processing systems* **32** (2019)
20. Zhang, S., Yuan, W., Wang, Y., Cheng, S., Wang, J.: Early-stage lifetime prediction for lithium-ion batteries: Linear regression-ensemble learning hybrid model based on impedance spectroscopy geometry. *Journal of Power Sources* **617**, 235153 (2024)
21. Zhou, X., Wang, X., Yuan, Y., Dai, H., Wei, X.: Fast capacity estimation for lithium-ion batteries based on xgboost and electrochemical impedance spectroscopy at various state of charge and temperature. *Automotive Innovation* pp. 1–19 (2024)
22. Zhou, Z., Duan, B., Kang, Y., Shang, Y., Zhang, Q., Zhang, C.: Practical state of health estimation for lifepo 4 batteries based on gaussian mixture regression and incremental capacity analysis. *IEEE Transactions on Industrial Electronics* **70**(3), 2576–2585 (2022)