ARCHIVES OF ELECTRICAL ENGINEERING VOL. 72(3), pp. 643–659 (2023)

DOI 10.24425/aee.2023.146042

Modelling of Li-Ion battery state-of-health with Gaussian processes

ADRIAN DUDEK, JERZY BARANOWSKI ✉

Department of Automatic Control and Robotics, AGH University of Science and Technology
Kraków, Poland
e-mail: {addudek/✉jb}@agh.edu.pl

(Received: 22.09.2022, revised: 13.04.2023)

Abstract: The problem of lithium-ion cells, which degrade in time on their own and while used, causes a significant decrease in total capacity and an increase in inner resistance. So, it is important to have a way to predict and simulate the remaining usability of batteries. The process and description of cell degradation are very complex and depend on various variables. Classical methods are based, on the one hand, on fitting a somewhat arbitrary parametric function to laboratory data and, on the other hand, on electrochemical modelling of the physics of degradation. Alternative solutions are machine learning ones or non-parametric ones like support-vector machines or the Gaussian process (GP), which we used in this case. Besides using the GP, our approach is based on current knowledge of how to use non-parametric approaches for modeling the electrochemical state of batteries. It also uses two different ways of dealing with GP problems, like maximum likelihood type II (ML-II) methods and the Monte Carlo Markov Chain (MCMC) sampling.

Key words: lithium-ion batteries, state-of-health, Gaussian process, diagnostics

1. Introduction

Lithium-ion batteries are currently standard for energy storage in portable consumer electronic devices such as mobile phones, notebooks, tablets and cameras [1]. Batteries are a complex chemistry and material system, and their electrical properties will change as they are used over time [2]. For example, their capacity will decrease, and their resistance will increase. Every improve of their capacity and lifespan requires major improvements in its design and chemistry [3]. In the case of such widespread use and drawback of constant wear, there is a need to predict its state-of-health (SoH).

The diagnostic of the SoH of a lithium battery is necessary to ensure the reliability and safety of the battery energy storage system and it also can minimize maintenance cost [4]. Battery aging is important to consider when predicting the battery's state of health, as it can show reduced capacity and increased resistance. The estimate of the remaining useful life is a key indicator for the management and performance of the battery and can also be seen as the time that elapses from now until the end of the battery's total useful life [5]. The SoH parameter is expressed as the ratio of the current cell capacity to its initial capacity (new cell), usually expressed in percentage. Most often, a cell is considered to be worn out when the SoH is below 80% [31].

Battery parameter prediction is mainly divided into methods based on physical-mathematical models (model-based) and data-driven models [32]. The model-based method considers the batteries aging as an empirical model process or as an electrical/electrochemical model. Unfortunately, it is a challenging task to develop a reasonable model that balances estimation accuracy and computational cost (due to the complex electrochemical process) [27]. That's why we often turn to a data-driven model.

Predicting lithium-ion battery degradation during design and operation is a significant challenge. Machine learning models have such a problem because the problem of black-box limitations still exists. Many mechanisms cause degradation, which may interact in various ways. A variety of factors affect it, including calendar time, high power usage, low-temperature usage, and combinations of these. However, battery end users can only measure a limited range of battery parameters. It is difficult to prioritize the mechanisms that drive ageing for a battery and use case [6].

There were several ways to predict battery health using data-driven methods such as: neural networks [7], support vector machines [8] and the Gaussian process [9–11, 29]. The approach proposed in our paper is based on a GP model in order to diagnose the SoH of the battery. We used an open-source dataset from the NASA Ames Research Centre regarding batteries, from which we selected features for GP fitting. Our GP models are based on two approaches of calculation: ML-II and MCMC. The first one comprises maximizing log-likelihood to optimize hyperparameters, and the second one is based on sampling the distributions. We also considered the multi-kernel approach of the GP to check the improvement of the results.

The novelty of our work consists in specific approaching and applicating Gaussian processes for use in the SoH estimation of Li-Ion batteries. The very concept of Gaussian processes has been well known for a long time, but it has been gaining popularity in recent years [23]. However, numerous ways of its implementation, variations and use are still under extensive research. [28, 30, 31]. As we mentioned, our proposal is based on the use of Markov Chains to sample data using the model we created earlier using STAN. This approach is not often used in cases of implementation of solutions using Gaussian processes [23]. For comparison, we also use python library implementations based on the aforementioned ML-II solution, which requires much less preparation.

Our motivation was to create an algorithm that would be able to diagnose the SoH of batteries using partial and time estimations of battery capacity changes based on the Gaussian process. The database used allowed for the study of a wide range of batteries applied in different conditions. In addition, various settings (including the mentioned multikernel approach) allowed us to check the quality of solutions for various initial assumptions. Moreover, we could examine the general quality of solutions less frequently used in GP implementations (like MCMC) and broaden the knowledge of the topic and encourage its further exploration.

The rest of the paper is organized as follows: first, we present a basic theory of the GP. Then, we discuss the dataset used in detail, as well as present the GP models. The next section introduces the results of our experiments and comparison of methods. The paper ends with conclusions.

2. Gaussian process

According to Rasmussen and Williams [12], the Gaussian process (GP) can be defined as *a collection of random variables, any Gaussian process finite number of which have a joint Gaussian distribution*. Such a definition captures the essence of the Gaussian process. If we have a real process $f(x)$, then we can define the mean function $m(x)$ and the covariance function $k(x,x')$ with the use of the following formulas:

$$m(x) = E\left[f(x)\right], \tag{1}$$

$$k(x,x') = E\left[f(x) - m(x) f(x') - m(x')\right]. \tag{2}$$

Then the definition of the GP can be formulated as:

$$f(x) \sim GP\left(m(x), k(x,x')\right). \tag{3}$$

There is only one way to fully determine the GP, which is to declare the mean and covariance function [13, 14]. The mean in most cases is set to “0”, because such a setting can be useful, simplify matters and is not difficult to fulfill. There are times when we would like to change the mean e.g., for a better model interpretability or the specification of our prior, but most of the time it can be left as 0 [12, 14]. The covariance function (also called the kernel function) represents a similarity between data points [15]. Usually, covariance is chosen from a set of already defined functions. In general, we should choose at least one that represents prior beliefs about the problem. However, any function that generates a positive definite covariance matrix is acceptable as a covariance function [16]. Moreover, it can be a challenging task to create new covariance functions that have practical value but are also correct.

The most basic and common kernel function is the Radial Basis Function (RBF). It is also known as the squared exponent (SE) and is defined by Eq. (4):

$$k\left(x_i, x_j\right) = \exp\left(-\frac{d\left(x_i, x_j\right)^2}{2l^2}\right). \tag{4}$$

Its main feature is that its value is usually only dependent on the distance from the specified point. The kernel used by the RBF algorithm is based on the length scale of l. Also $d(\cdot,\cdot)$ is the Euclidean distance. The RBF is infinitely differentiable. That means the GP with this kernel has a mean square derivative for all orders, despite that the RBF is very smooth. However, strong smoothness assumptions are unrealistic, and using the kernel from the Matérn family is recommended [12].

In the cases of battery health diagnostics and overall modeling, one of the most used kernels, besides for instance the RBF, are those from the Matérn family. The Matérn kernel family is a kind of generalization of the RBF function. The general formula for Matérn covariance functions is given by Eq. (5):

$$k\left(x_i, x_j\right) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d\left(x_i, x_j\right)\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(x_i, x_j)\right), \tag{5}$$

where: K_ν is the modified Bessel function, l is the length-scale parameter, $d(\cdot,\cdot)$ is the Euclidean distance and ν is the parameter, which allows one to change the smoothness of the function, giving an ability to flexibly control the function in relation to the one we want to model. These kernels are frequently used with parameter ν values of 3/2 applied in learning functions, which are at least once differentiable and 5/2 for functions at least twice differentiable [12].

In the case of modeling the electrochemical state of batteries, it is worth mentioning the Rational Quadratic (RQ) kernel, which can be seen as a scale mixture (an infinite sum) of RBF kernels with different characteristic length scales. Hyperparameters describing RQ are the length scale l and the scale mixture parameter α. The RQ kernel function is given by Formula (6):

$$k\left(x_i, x_j\right) = \left(1 + \frac{d\left(x_i, x_j\right)^2}{2\alpha l^2}\right)^{-\alpha}, \tag{6}$$

where in addition to hyperparameteres, $d(\cdot,\cdot)$, as previously, is the Euclidean distance.

Each of the selected covariance functions has a set of free parameters called hyperparameters. These values need to be determined in order to use the covariance function. Handling this task can be one of the main difficulties in using the GP. The ML-II method is widely used at present, and we calculate the parameters by edge likelihood optimization. Its advantage is that it is easy to use and the calculation process is not complicated, and its main disadvantage is its low precision. For example, it can suffer from the problem of multiple local maxima. There are also solutions to calculate hyperparameters such as statistical methods like the MCMC sampling of the declared prior distribution [12]. We used both approaches to solve the problem of diagnosis of battery cells to gain the SoH result, but more importantly, to research the benefits and drawback of each techniques, as well as future possibilities.

The GP is an approach that is quickly gaining popularity in recent years. The algorithm works best for problems with high dimensions, small samples, and nonlinearity [17]. The GP is a probabilistic technique for nonlinear regression that computes a posterior degradation estimate by constraining the prior distribution to fit the available training data. Recently, the GPR method has been favored by researchers because it is a probabilistic prediction model under the Bayesian framework [18]. Moreover, the most distinctive about the GP is that it provides not only the prediction but also a confidence interval as well as the distribution of all calculated values [19].

3. Data and methods

3.1. Battery dataset

The general problem of every lithium-ion battery diagnostics is the data related to it, more specifically, its overall lack of large, reliable, and comprehensive sets of data [20]. Some studies connected to this field of interests have made use of an open-source dataset from the NASA AMES research center, and much of this data shows an approximately constant degradation rate [21], so it could be easily used as a starting point in our research. Because of the aforementioned data issue, we also used one of the NASA resources, which is randomized battery usage dataset [22].

The dataset consists of several *18650 Li-Ion batteries* (28 in total), which were divided into 7 groups, based on the factors in which they were charged and discharged. Specifics about differences in load patterns can be seen in Table 1.

Table 1. Overview of load patters in NASA AMES battery dataset

Group	Description
1	– Repeatedly charged to 4.2 V using a **randomly** selected duration – Discharged to 3.2 V using a **randomized** sequence of discharging currents – Randomized sequence of discharging currents between 0.5 A and 4 A – Duration of charging between 0.5 h and 3 h – Room temperature
2	– Repeatedly charged to 4.2 V using a **preselected** duration – Discharged to 3.2 V using a **preselected** sequence of discharging currents – Sequences of discharging currents between 0.5 A and 4 A – Room temperature
3	– Operated using a sequence of charging/discharging currents – Sequences between –4.5 A and 4.5 A – Each loading period lasted 5 min – Room temperature
4	– Repeatedly charged to 4.2 V using a **randomly** selected duration – Discharged to 3.2 V using a **randomized** sequence of discharging currents – Randomized sequence of discharging currents between 0.5 A and 5 A – Customized probability distribution skewed towards selecting **higher** currents – Room temperature
5	– Repeatedly charged to 4.2 V using a **randomly** selected duration – Discharged to 3.2 V using a **randomized** sequence of discharging currents – Randomized sequence of discharging currents between 0.5 A and 5 A – Customized probability distribution skewed towards selecting **lower** currents – Room temperature
6	– Repeatedly charged to 4.2 V using a **randomly** selected duration – Discharged to 3.2 V using a **randomized** sequence of discharging currents – Randomized sequence of discharging currents between 0.5 A and 5 A – Customized probability distribution skewed towards selecting **lower** currents – Ambient temperature set to 40°C
7	– Repeatedly charged to 4.2 V using a **randomly** selected duration – Discharged to 3.2 V using a **randomized** sequence of discharging currents – Randomized sequence of discharging currents between 0.5 A and 5 A – Customized probability distribution skewed towards selecting **higher** currents – Ambient temperature set to 4°C

The dataset contained information about the start time of the experiment for each battery cell, as well as measurements of the current and voltage of the load pattern with time stamps. We used data from all 7 groups. The dataset was then divided into cycles, each of which ended with a check of the battery's remaining capacity. During the cycle, cells were charged or discharged according to the parameters based on the selected scenario, or the parameters were not used at all (details in Table 1, more information can be found in the dataset itself: Bole *et al*., 2014 [22]). From the whole dataset we extracted 3 features from each cycle, based on literature [20]:

- total time of battery life: $\boldsymbol{T}$,
- time of selected cycle: $\boldsymbol{t}$,
- charge throughput during cycle (which is the integral of current over time): $\boldsymbol{I}$.

Based on that information, our GP model is supposed to calculate an output $\boldsymbol{y}$ capacity difference $\boldsymbol{\Delta Q}$ during a single cycle.

3.2. Gaussian process model

In order to create the model, we took into account the principles of the Gaussian process from section 2 and tried to use them considering the state-of-health diagnostic based on the dataset from section 3.1. Our first approach consists of the ML-II method, which is one of the most widely used methods of GP implementation [23]. We optimize the parameters by using a sort of likelihood function. Its advantage is that it is easy to set up and has a cheaper computation process, in exchange for overall accuracy and a less informative model. For some applications, it can experience a problem with having multiple local maxima [23].

To create a model using the ML-II method, we implemented the Gaussian processes from the python scikit-learn library. This implementation allowed us to create a GP model, which can use predefined kernel functions such as: the RBF, Matern family and RQ. Considering the fact that these kernels (especially Matern one) seem succesfull in research of battery diagnostics [9], we had the full possibility of testing this approach. Additionally, we could use some other kernels as points of reference. In this approach we were only able to pass information about kernels and optionally set the starting point of hyperparameters, which will be optimized automatically during the process with the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LM-BFGS) optimizer. As a result of fitting, we receive only predicted values corresponding with the standard deviation.

In the second approach, we developed other methods which do not rely on marginal likelihood maximizing. We created an STAN model of the GP and used it with the Markov Chain Monte Carlo (MCMC) method of sampling distributions. The main drawback of such an approach is the requirement of additional and specific knowledge, as well as longer computation time, but in return, we gain access to full information about the distribution of our results [24]. This approach requires building a model from scratch, with all assumptions about prior distribution, kernel matrix operations, generating quantities, etc. It is also the best way to code expert knowledge about the problem into our model. The returned distributions should allow us to come up with more complex conclusions and future use, which compensate for the difficulties of use.

A detailed description of the implementation and use of both algorithms can be found in Fig. 1, in the form of a diagram. Details and theoretical background can be found in C.E. Rasmussen and C.K.I. Williams, Gaussian processes for Machine Learning [12].

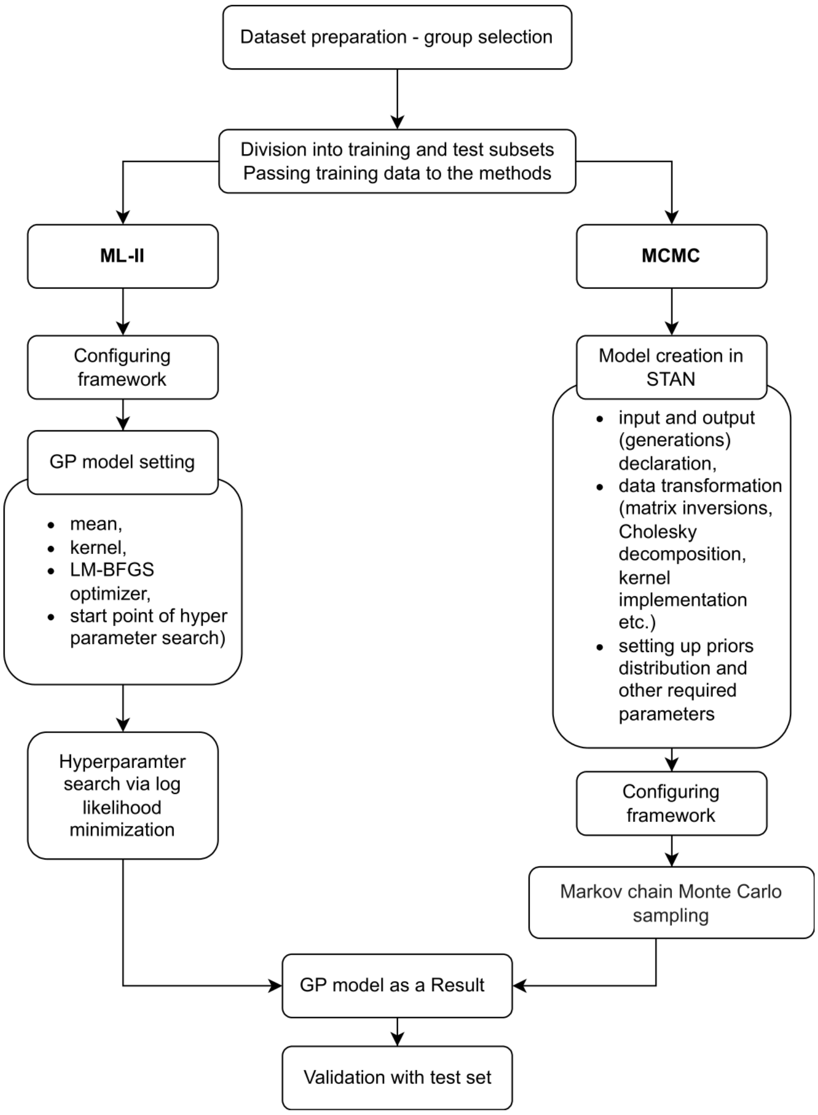

Dataset preparation - group selection
Division into training and test subsets
Passing training data to the methods
ML-II
MCMC
Configuring framework
GP model setting
mean,
kernel,
LM-BFGS optimizer,
start point of hyper parameter search)
Hyperparamter search via log likelihood minimization
Model creation in STAN
input and output (generations) declaration,
data transformation (matrix inversions, Cholesky decomposition, kernel implementation etc.)
setting up priors distribution and other required parameters
Configuring framework
Markov chain Monte Carlo sampling
GP model as a Result
Validation with test set

Fig. 1. Diagram showing the course of the algorithm divided into parts ML-II and MCMC

Setting the framework means setting all the necessary tools to work together in a specific structure, e.g. as a python script. ML-II, after providing the data, only requires setting the initial model values (average, hyperparameters) and optional settings for the optimizer. MCMC, on the other hand, requires creating the entire model manually with all the necessary data transitions (e.g. operations on the covariance matrix) and setting the initial distributions for the priorities. In addition, the necessary elements for samples generation. The sampling process is time-consuming but needs to be performed only once per model.

4. Results

4.1. ML-II approach

In order to evaluate the created model with this method we decided to use 4 different kernels: the RBF, Matern (3/2 and 5/2), RQ and Exp-Sine-Squared (ESS). The scikit-learn framework also allows one to easily combine different kernels into a multi-kernel approach, so we also considered that in our tests. The entire workflow consists of some principals. First, as the dataset relies on 7 different scenarios of load patterns, we created separate GP models for each scenario, which were then fitted using training data from each group. Validation of the models is done based on test data. Then we consider checking examples of batteries in terms of SoH diagnostic based on the learned model outcomes for a specific group. SoH prediction is based on knowledge of the battery capacity at the start and information about its life cycle necessary to select features and perform calculations. The model predicts changes in capacity for each cycle measurements (between moments in time when we collect data according to which we perform calculations with the GP model) and then the difference is subtracted from the previously known (estimated) capacity, which allows us to diagnose a current SoH.

The results of the model performance based on the test data set can be seen in Fig. 2. It presents several plots, with different kernels, where the relationship between the prediction of the capacity change and the measured value is shown. Red dots represent the mean value of prediction when blue bars show a 95% confidence interval. The method we chose to evaluate the models was to calculate the Root-Mean-Square Error (RMSE) value of prediction for each group and the whole test set, respectively. The concrete results of the RMSE for the total set are provided in Table 2.

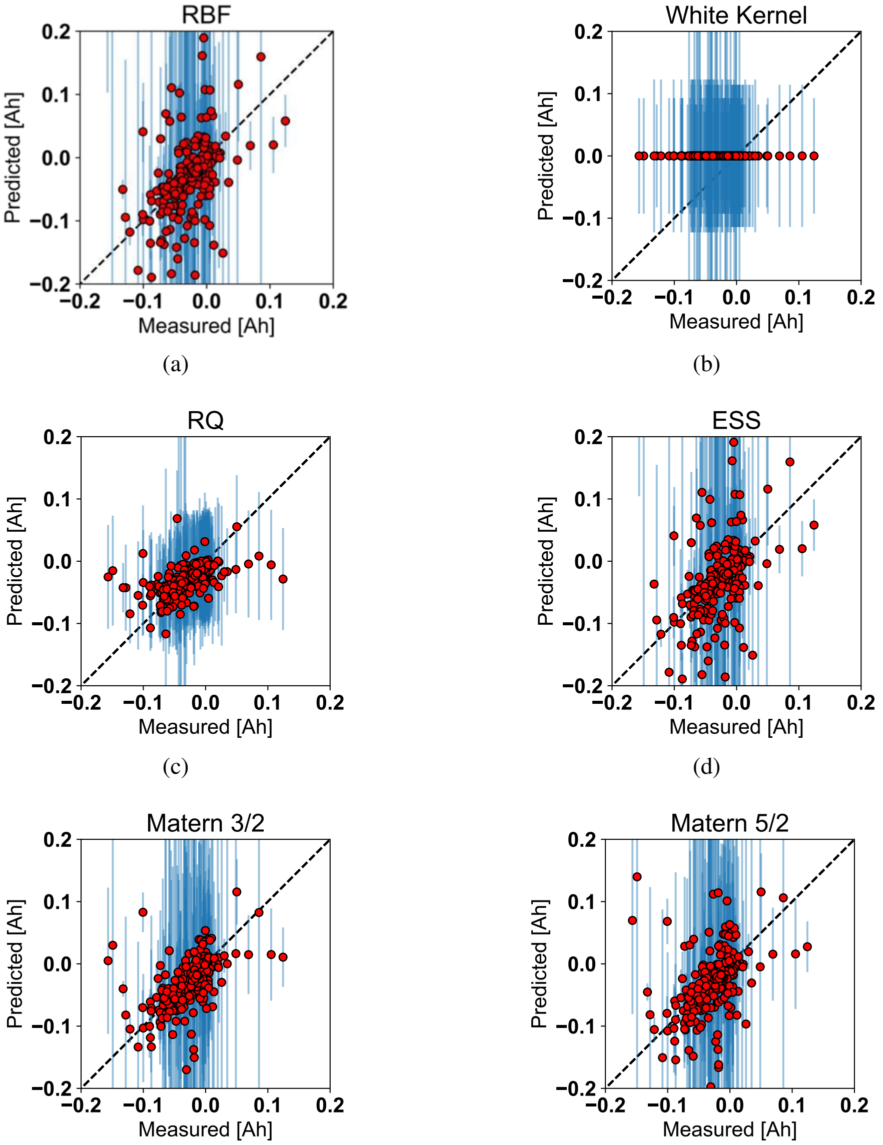
Additionally, for the best-performing kernels, we decided to test the multi-kernel approach. The scikit-learn framework allows one to combine kernels by simply adding them to each other, with maintaining the ability to autotune hyperparameters. The results for the mentioned multi-kernels are shown in Fig. 3, as well as the RMSE values in Table 2.

The results show that the multi-kernel approach increases the accuracy of the overall model and the multi-kernel model themselves are at a similar level of accuracy. For the best-performing

Table 2. Total RMSE results of GP models with ML-II approach

RMSE	**Kernel**
0.096	ESS
0.057	RQ
0.185	White
0.096	RBF
0.058	Ma 3/2
0.065	Ma 5/2
0.055	Ma 3/2 + RQ
0.058	Ma 3/2 + Ma 5/2
0.061	Ma 3/2 + Ma 5/2 + RBF
0.054	Ma 3/2 + Ma 5/2 + RQ

(in terms of RMSE) model, we created the SoH prediction, which is based on the knowledge of a basic battery capacity and then predicts its changes based on usage in the considered intervals. Example plots for the batteries from different groups can be seen in Fig. 4.

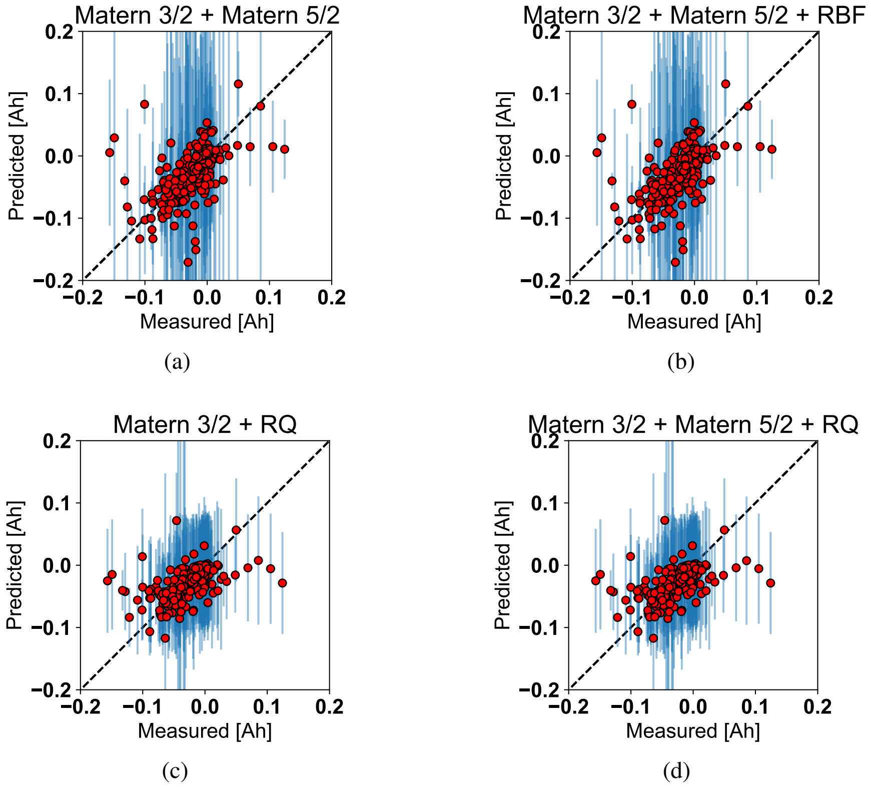


(a) (b)

(c) (d)

(e) (f)

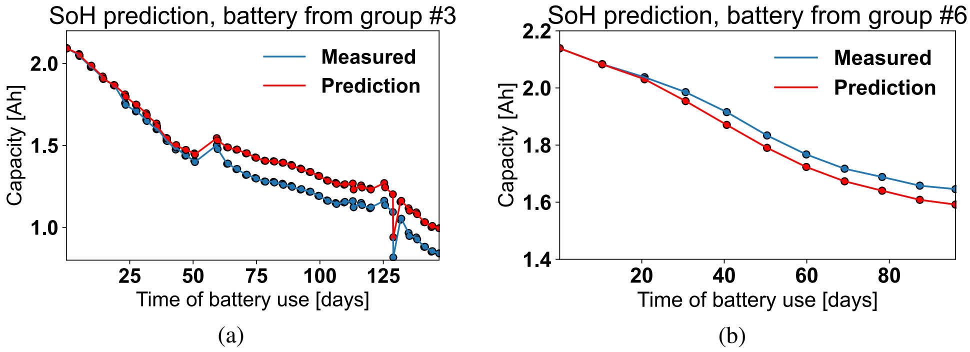
Fig. 2. Results of GP models with different kernels Radial-Basis Function (a); white kernel (b); Rational Quadratic (c); Exp-Sine-Squared (d); Matern 3/2 (e) and Matern 5/2 (f); plot represents prediction of model in relation to measured values; red dots are mean of prediction and the closer they are to the black dotted line, the smaller deviation from real values is; blue bars represent 95% confidence interval for each sample



(a) (b)

(c) (d)

Fig. 3. Results of GP models with different multikernes: sum of Matern family (a); sum of both Materns and Radial Basis Function (b); sum of Matern 3/2 and Rational Quadratic (c) and sum of both Materns and Rational Quadratic (d); plot represents prediction of model in relation to measured values; red dots are mean of prediction and the closer they are to the black dotted line, the smaller deviation from real values is; blue bars represent 95% confidence interval for each sample



(a) (b)

Fig. 4. Results of SoH prediction with created GP model with ML-II approach: red dots show the predicted values, while blue ones are measured values; left plot (a) shows example battery from group 3 and right plot (b) shows example battery from group 6 (which has ambient temeperature set to 40°C)

4.2. MCMC sampling approach

Contrary to the ML-II approach, the MCMC sampling is characterized by the need to create a separate Stan model. The model is based on the hierarchical model approach, where each of the three levels consists of posteriors over parameters, hyperparameters and model structure. This requires setting up our priors for each of them. The overall accuracy depends on the correctness of their selection. A detailed description of the hierarchical model can be found in [12]. The model we created took data in the form of feature vectors and then created a covariance matrix corresponding to the selected kernel from them. Additionally, to speed up the calculations, we used the Cholesky decomposition mechanism. Initial settings for hyperparameters were set based on generally used distributions based on examples and applications of GP modeling in the Stan as well as our predictions. The output of our system was a generated distribution of predicted values for the test set. We created two sets of priors (one based on example usage and second set with our slight changes). The results for the RBF and Matern kernel with the test set can be seen in Fig. 5, as well as the RMSE results for expected values of posterior distribution in Table 3. Additionaly, we present the SoH prediction for example batteries in Fig. 6.

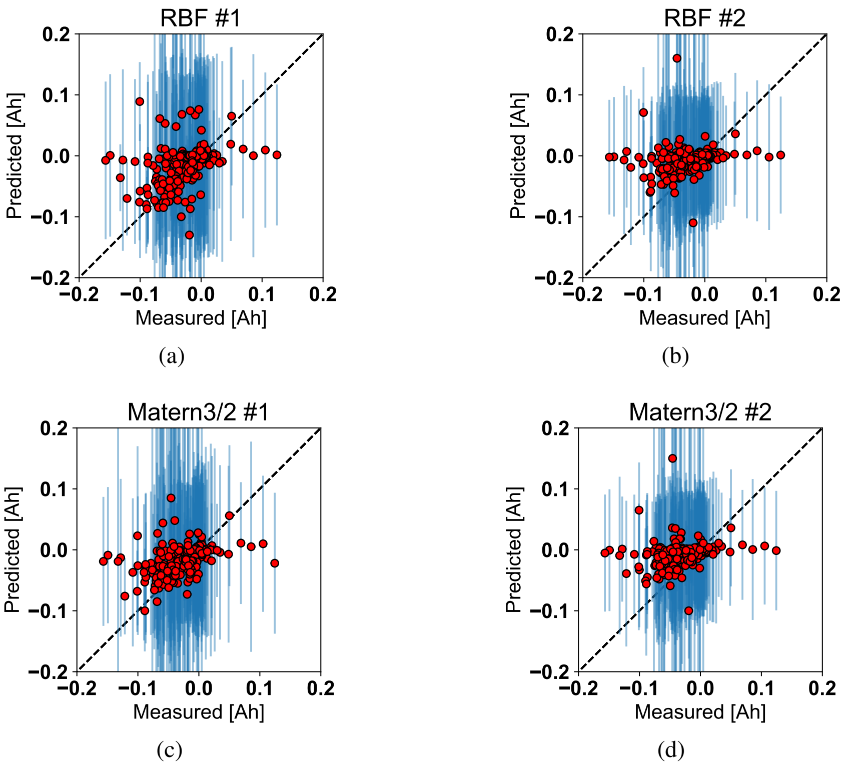
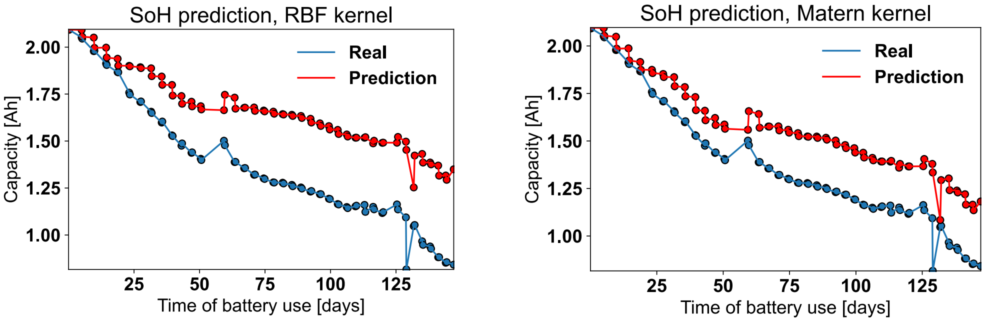


Fig. 5. Results of GP model with MCMC sampling: RBF kernel with first set of hyperparameters prior (a); RBF kernel with second set of hyperparameters prior (b); Matern 3/2 kernel with first set of hyperparameters prior (c) and Matern 3/2 kernel with second set hyperparameters prior (d); plot represents prediction of model in relation to measured values; red dots are mean of prediction and the closer they are to the black dotted line, the smaller deviation from real values is; blue bars represent 95% confidence interval for each sample

Table 3. RMSE results for expected values of GP models with MCMC sampling

RMSE	Kernel	Prior set
0.153	RBF	1
0.079	RBF	2
0.183	Ma 3/2	1
0.184	Ma 3/2	2



(a) (b)

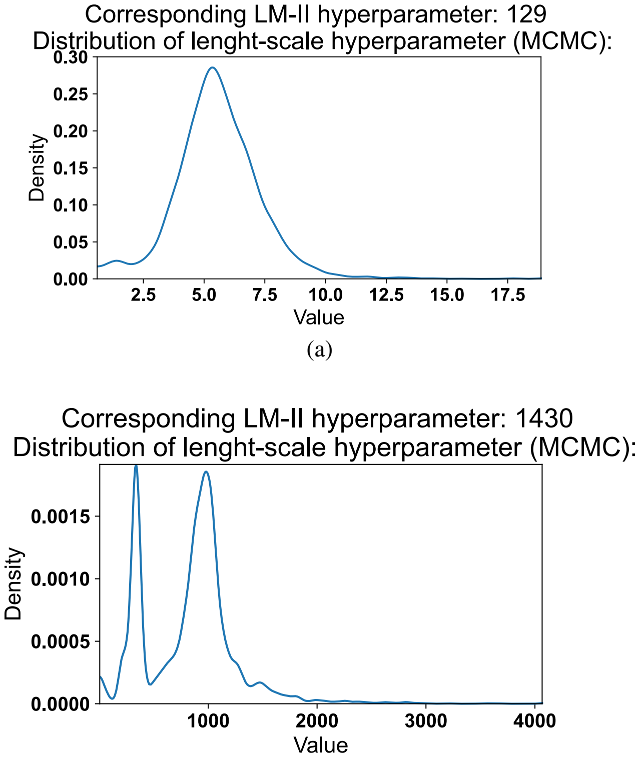
Fig. 6. Results of SoH prediction with created hierarchical GP model with MCMC sampling; red dots show the predicted values, while blue one are measured values; left plot (a) shows example battery predicted with RBF kernel and right plot (b) with Matern 3/2 kernel

First of all, it seems clear that the general prior assumption we used, does not fit SoH diagnostics very well. Secondly, the RMSE calculated over the expected values from posterior distribution is not the best way to measure model accuracy, because even though RBF kernels seem to get a better score, they still have worse results then the predicted example battery, as well as have higher uncertainty than Matern kernels, which can be seen in the plot in Fig. 4. The hierarchical model resulted in much more information about the problem, with slightly worse accuracy of prediction at the cost of a more complicated setup and higher computing time. As the setup of priors was not done perfectly, there is still room to improve it in future research.

4.3. Comparison of presented approaches

First of all, it seems normal to compare the accuracy of both methods in terms of RMSE value, but it seems to be correct mainly in the ML-II solution, where firstly you can see solid differences between kernels, and secondly, this tool is mainly based on returning predictions, where the MCMC sampling returns to us more of the entire posterior of the model because this is its main advantage. The RMSE for the second case was calculated based on the expected values and not the whole distributions. In addition, the parameters in the case of ML-II were autotuned, where for the Stan model they were limited by the selection of distributions from the base GP-use cases, and not adapted to a given problem.

Because of it, hyperparameters resulting from autotuning in the case of the RBF and Matern differ from the mean value of the posterior distribution. Figure 7 shows the comparison of distributions of length-scale hyperparameters for the same example cases. The mean value to variant (a) is 5.5, while for variant (b) is 830. At the same time the ML-II method tuned the same parameter to 129 in variant (a) and 1 430 in variant (b). We also have to be aware that the Stan model uses an additional parameter called *magnitude* [25], which even further precludes a direct comparison of the two methods. Here, we will only focus on a comparison of length-scale hyperparameters, as these are common for both approaches.

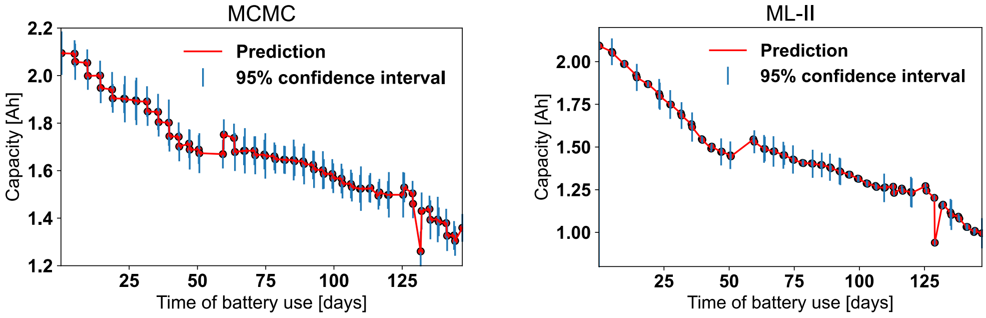


(a)

(b)

Fig. 7. Plots represent posterior distribution of length-scale hyperparameter aquired from model with MCMC sampling method; top plot (a) shows the distribution for case of Matern 3/2 kernel for third group from battery dataset (mean value: 5.5), correspoing value from ML-II method for this case is 129; bottom plot (b) shows the distribution for case of RBF kernel for fifth group from battery dataset (mean value: 830), correspoing value from ML-II method for this case is 180

Additionally, the MCMC sampling provides us with a more consistent confidence interval, while the scikit-learn framework sometimes has even some trouble catching the uncertainty. This can be noted in the plots in Figs. 2, 3 and 5, where we can see that the confidence interval is not consistent across all predictions, while in the case of the MCMC sampling, it remains constant for the whole result (each sample). We can also better visualise that in the plot of the SoH diagnostic of the example battery when drawing uncertainty of prediction, as shown in Fig. 8.



(a) (b)

Fig. 8. Uncertainty of SoH prediction for example battery with hierarchical GP model with MCMC sampling (a) and ML-II approach (b); red dots show the predicted values, while blue lines are 95% confidence interval

ML-II in general is the fastest way to calculated GP's hyperparameters. It has a variety of implementations across different frameworks, but the principles are the same and based on maximum log-likelihood optimization. In the case of the framework which we used, it allowed us to set starting points for searching the parameters for kernel functions (as well as kernel functions) with the output of predicted (mean) value and its standard deviation. Additionaly, we can only add parameters to the diagonal of the covariance matrix. This makes the approach comprehensive and uncomplicated, and the use of the ML-II method for parameter autotuning is fast. The cost incurred in such an approach is the inaccuracy of the solution due to its tendency to overfit [23], having very little output information about the problem, and the frequent problems with catching the uncertainty [26].

MCMC requires much more in terms of preparation of the model (it has to be done by hand and with some expert knowledge) and the computation cost (sampling just a few chains can require a lot of time in certain problems). On the other hand, as the output we got not only the prediction but the whole distribution of it, as well as the distribution of parameters, from which we can calculate values that we need to improve research [24]. In our case we did not use the full potential of MCMC with a Stan model. There is still room to increase the accuracy and flexibility of our model (e.g. better prior selection), but we wanted to signal the existence of other approaches worth considering for their own properties.

5. Conclusion

Diagnosing the SoH with the use of the GP is something that is constantly being developed. This paper mainly focused on creating the multi-kernel approach with ML-II hyperparameter tuning as well as checking possible alternatives based on creating a hierarchical model with the MCMC sampling. The overall results of the ML-II approach are quite promising in terms of using the GP to battery SoH diagnostics. We compared a few known kernels when dealing with the mentioned problem, where the Matern family with RQ showed the best results in terms of the RMSE. Then we tried to combine covariance functions with the best accuracy from the previous step in order to check if the multi-kernel approach can improve accuracy. The combination of Ma 3/2, Ma 5/2 and RQ resulted in the best RMSE score of 0.054 on the test set.

The used framework shows an uncomplicated way of GP modelling, with the possibility of coding previous knowledge about the problem with a choice of the kernel function and its start point of hyperparameter autotuning. Optimizing hyperparametres do the rest of the work within a reasonable time and the overall accuracy of prediction is satisfactory. Also, some limitations of this method are noticeable, e.g. sometimes it has some trouble with catching the uncertainty, results depend on the quality of optimizing and tends to overfit. A current state of implementation consists of models fitted into scenarios of battery usage, but in future development we probably could generalize the model even for situations not included in the NASA dataset.

Additionally, we tried an alternative approach creating a hierarchical model with the MCMC sampling method. Based on the theory from Rasmussen and Williams [12] and exemplary modeled problems, we implemented an STAN model for the GP, specifying priors and the kernel function. We used the two different kernels (RBF and Ma 3/2), as well as two different prior sets (basic one from STAN recommendation and the second one with our twists). The result showed that changes in prior assumptions can lead to significant differences in the posterior distributions. The best RMSE value was not a breakthrough in our research. The reason is that the best accuracy was not our first concern, but rather presenting a new possibility of GP modelling and its properties. The results of this approach cannot be directly compared to the presented ML-II method, because of not using its full potential. Furthermore, we should focus on the various pros and cons of both methods. The model with the MCMC sampling gives us the posterior distribution of each of the levels of a hierarchical model. This delivers more information about problems, which can lead to an additional test and meaningful conclusions. The main drawback is the need to accurately declare the prior distributions (based on the expert knowledge at best) and computational complexity.

In conclusion, both of the presented approaches show promise for the future development of the GP modeling of battery SoH diagnostics. The ML-II approach showed that autotuned parameters can lead to accurate models with valid kernel selection, while the method based on the MCMC sampling returns more information about problems but requires better knowledge of the problems and can outperform other approaches if set correctly. The paper showed a very early stage of development, especially in terms of the approach with a hierarchical model. It needs to be refined by adding expert knowledge in process creation. After such transformations, further comparative and testing stages can be specified, e.g. through measures other than the RMSE, like the mean absolute percentage error (MAPE). The current state shows the possibilities of an MCMC model and its meaningfulness for future research in this field, which will probably lead to a more general model. The improved MCMC model may not only be generalized but also become a competitor for other dominant solutions, which are used in the fields of capacity and SoH diagnostics, so our project will surely be developed.

Acknowledgements

This research was funded by AGH's Research University Excellence Initiative under the project "Interpretable methods of process diagnosis using statistics and machine learning" and by the Polish National Science Centre project "Process Fault Prediction and Detection" contract no. UMO-2021/41/B/ST7/03851.

References

[1] Tagade P., Hariharan K.S., Ramachandran S., Khandelwal A., Naha A., Kolake S.M., Han S.H., *Deep Gaussian process regression for lithium-ion battery health prognosis and degradation mode diagnosis*, Journal of Power Sources, vol. 445, 227281 (2020), DOI: 10.1016/j.jpowsour.2019.227281.

[2] Li X., Wang Z., Yan J., *Prognostic health condition for lithium battery using the partial incremental capacity and Gaussian process regression*, Journal of Power Sources, vol. 421, pp. 56–67 (2019), DOI: 10.1016/j.jpowsour.2019.03.008.

[3] Dillon S.J., Sun K., *Microstructural design considerations for Li-ion battery systems*, Current Opinion in Solid State and Materials Science, vol. 16, no. 4, pp. 153–162 (2012), DOI: 10.1016/j.cossms.2012.03.002.

[4] Li X., Yuan C., Li X., Wang Z., *State of health estimation for Li-Ion battery using incremental capacity analysis and Gaussian process regression*, Energy, vol. 190, 116467 (2020), DOI: 10.1016/j.energy.2019.116467.

[5] Garay F., Huaman W., Vargas-Machuca J., *State of health diagnostic and remain useful life prognostic for lithium-ion battery by combining multi-kernel in Gaussian process regression*, 2021 IEEE XXVIII International Conference on Electronics, Electrical Engineering and Computing, pp. 1–4 (2021), DOI: 10.1109/INTERCON52678.2021.9532733.

[6] Greenbank S., Howey D., *Automated Feature Extraction and Selection for Data-Driven Models of Rapid Battery Capacity Fade and End of Life*, in IEEE Transactions on Industrial Informatics, vol. 18, no. 5, pp. 2965–2973 (2022), DOI: 10.1109/TII.2021.3106593.

[7] Liu J., Saxena A., Goebel K., Saha B., Wang W., *An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries*, in Conference of the Prognostics and Health Management Society (2010), DOI: 10.36001/phmconf.2010.v2i1.1896.

[8] Fermín P., McTurk E., Allerhand M., Medina-Lopez E., Anjos M.F., Sylvester J., dos Reis G., *Identification and machine learning prediction of knee-point and knee-onset in capacity degradation curves of lithium-ion cells*, Energy and AI, vol. 1, in press (2020), DOI: 10.1016/j.egyai.2020.100006.

[9] Richardson R.R., Osborne M.A., Howey D.A., *Gaussian process regression for forecasting battery state of health*, Journal of Power Sources, vol. 357, pp. 209–219 (2017), DOI: 10.1016/j.jpowsour.2017.05.004.

[10] Tagade P., Hariharan K.S., Ramachandran S., Khandelwal A., Naha A., Mayya S., Kolake S., Han H., *Deep Gausian process regression for lithium-ion battery health prognosis and degradation mode diagnosis*, Journal of Power Sources, vol. 445, pp. 227–281 (2020), DOI: 10.1016/j.jpowsour.2019.227281.

[11] Zheng X., Deng X., *State-of-health prediction for lithium-ion bat- teries with multiple Gaussian process regression model*, IEEE Access, vol. 7, pp. 150383–150394 (2019), DOI: 10.1109/ACCESS.2019.2947294.

[12] Rasmussen C., Williams C.K.I., *Gaussian Processes in machine learning*, MIT Press (2006), DOI: 10.7551/mitpress/3206.003.0001.

[13] Davis R.A., *Encyclopedia of Environmetrics, Gaussian Process*, In Encyclopedia of Environmetrics; American Cancer Society (2006), DOI: 10.1002/9780470057339.vag002.

[14] Garnett R., *Bayesian Optimization*, Cambridge University Press (2022).

[15] Blum M., Riedmiller M., *Optimization of gaussian process hyperparameters using Rprop*, pp. 339–344 (2013).

[16] Raes W., Dhaene T., Stevens N., *On the Usage of Gaussian Processes for Visible Light Positioning with Real Radiation Patterns*, In Proceedings of the 2021 17th International Symposium on Wireless Communication Systems (ISWCS), pp. 1–6 (2021), DOI: 10.1109/ISWCS49558.2021.9562197.

[17] Chen T., Morris J., Martin E., *Gaussian process regression for multivariate spectroscopic calibration*, Chemometrics and Intelligent Laboratory Systems, vol. 87, no. 1, pp. 59–71 (2007), DOI: 10.1016/j.chemolab.2006.09.004.

[18] He Y.J., Shen J.N., Shen J.F., Ma Z.F., *State of health estimation of lithium-ion batteries: A multiscale Gaussian process regression modeling approach*, AIChE Journal, vol. 61, no. 5, pp. 1589–1600 (2015), DOI: 10.1002/aic.14760.

[19] Brevault L., Balesdent M., Hebbal A., *Overview of Gaussian process based multi-fidelity techniques with variable relationship between fidelities* (2020), DOI: 10.48550/arXiv.2006.16728.

[20] Richardson R.R., Osborne M.A., Howey D.A., *Battery health prediction under generalized conditions using a Gaussian process transition model*, Journal of Energy Storage, vol. 23, pp. 320–32 (2019), DOI: 10.1016/j.est.2019.03.022.

[21] Greenbank S., Howey D., *Automated feature selection for data-driven models of rapid battery capacity fade and end of life*, Transactions on Industrial Informatics, Institute of Electrical and Electronics Engineers (IEEE), vol. 18, no. 5, pp. 2965–2973 (2022), DOI: 10.1109%2Ftii.2021.3106593.

[22] Bole B., Kulkarni C., Daigle M., *Randomized battery usage data set, NASA AMES Prognostics Data Repository*, NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA (2014).

[23] Dudek A., Baranowski J., *Gaussian Processes for Signal Processing and Representation in Control Engineering*, Appl. Sci., vol. 12, no. 10, 4946 (2022), DOI: 10.3390/app12104946.

[24] Titsias M., Lawrence N., Rattray M., *Markov chain Monte Carlo algorithms for Gaussian processes. Inference and Estimation in Probabilistic Time-Series Models*, Bayesian Time Series Models, pp. 295–316 (2008), DOI: 10.1017/CBO9780511984679.015.

[25] Vanhatalo J., *Sparse Log Gaussian Process in Spatial Epidemiology*, Gaussian Processes in Practice in Proceedings of Machine Learning Research, vol. 1, pp. 73–89 (2022).

[26] Dudek A., Baranowski J., Mularczyk R., *Transient Anomaly Detection Using Gaussian Process Depth Analysis*, In Proceedings of the 2021 25th International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 221–226 (2021), DOI: 10.1109/MMAR49549.2021.9528470.

[27] Cai L., Lin J., Liao X., *An estimation model for state of health of lithium-ion batteries using energy-based features*, J. Energy Storage, vol. 46 (2021), DOI: 10.1016/j.est.2021.103846.

[28] Burzyński D., *Useful energy prediction model of a Lithium-ion cell operating on various duty cycles*, Eksploatacja i Niezawodność, vol. 24, no. 2, pp. 317–329 (2022), DOI: 10.17531/ein.2022.2.13.

[29] Liu K., Tang X., Teodorescu R., Gao F., Meng J., *Future Ageing Trajectory Prediction for Lithium-ion Battery Considering the Knee Point Effect*, IEEE Trans. Energy Convers., vol. 8969, no. c, pp. 1–10 (2021), DOI: 10.1109/TEC.2021.3130600.

[30] Liu K., Shang Y., Ouyang Q., Widanage W.D., *A Data-Driven Approach with Uncertainty Quantification for Predicting Future Capacities and Remaining Useful Life of Lithium-ion Battery*, IEEE Trans. Ind. Electron., vol. 68, no. 4, pp. 3170–3180 (2021), DOI: 10.1109/TIE.2020.2973876.

[31] Burzyński D., Kasprzyk L., *A novel method for the modeling of the state of health of lithium-ion cells using machine learning for practical applications*, Knowledge-Based Syst., vol. 219, 106900 (2021), DOI: 10.1016/j.knosys.2021.106900.

[32] Su C., Chen H., Wen Z., *Prediction of remaining useful life for lithium-ion battery with multiple health indicators*, Eksploatacja i Niezawodność, vol. 23, no. 1, pp. 176–183 (2021), DOI: 10.17531/ein.2021.1.18.