Applied Mathematical Modelling 122 (2023) 42–59

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A Physics-Constrained Bayesian neural network for battery remaining useful life prediction

David A. Najera-Flores[a,c], Zhen Hu[b], Mayank Chadha[a], Michael D. Todd[a,*]

[a] Department of Structural Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla CA 92093, USA
[b] Department of Industrial and Manufacturing Systems Engineering, University of Michigan-Dearborn, Dearborn MI 48128, USA
[c] ATA Engineering, Inc., 13290 Evening Creek Dr S, San Diego, CA 92128, USA

ARTICLE INFO

Article history:
Received 27 January 2023
Revised 12 April 2023
Accepted 30 May 2023
Available online 1 June 2023

Keywords:
Battery
Remaining useful life
Prognostics
Bayesian neural net
Physics constraint

ABSTRACT

In order to predict the remaining useful life (RUL) of lithium-ion batteries, a capacity degradation model may be developed using either simplified physical laws or machine learning-based methods. It is observed that even though degradation models based on simplified physical laws are easy to implement, they may result in large error in the application of failure prognostics. While data-driven prognostics models can provide more accurate degradation forecasting, they may require a large volume of training data and may invoke predictions inconsistent with physical laws. It is also very challenging for existing methods to predict the RUL at the early stages of battery life. In this paper, we propose a Bayesian physics-constrained neural network for battery RUL prediction by overcoming limitations of the current methods. In the proposed method, a neural differential operator is learned from the first 100 cycles of data. The neural differential operator is modeled with a Bayesian neural network architecture that separates the fixed history dependence from the time dependence to isolate epistemic uncertainty quantification. Using the battery dataset presented in the paper by Severson et al. as an example, we compare our proposed method with a simplified physics-based degradation forecasting model and two data-driven prognostics models. The results show that the proposed physics-constrained neural network can provide more accurate RUL estimation than the other methods with the same group of training data. Most importantly, the proposed method allows for RUL prediction at earlier stages of the battery life cycle.

1. Introduction

Lithium-ion battery health prognosis, or the ability to predict its remaining useful life, is crucial for ensuring the reliability and efficiency of the systems that rely on its power. Monitoring and forecasting the remaining battery capacity provides valuable insight into the overall performance of the battery and allows for the prediction of battery failure. These predictions facilitate timely maintenance and/or replacement, improves the overall energy efficiency of the system, and maximizes system operational availability (i.e., minimizing downtime). Battery prognosis is a crucial requirement for various systems, including electric vehicles (for optimizing range and performance), renewable energy systems (for improving en-

* Correspondence author.
E-mail address: mdtodd@eng.ucsd.edu (M.D. Todd).

https://doi.org/10.1016/j.apm.2023.05.038
0307-904X/

ergy storage), aerospace and defense applications (for predicting remaining battery life in satellites), Internet of Things (for monitoring device battery life to ensure seamless operation), among others.

Failure prognostics–the capability to predict remaining useful life (RUL) estimation–of lithium-ion batteries has been a very active research topic for a couple of decades with the rise of such battery usage across many applications [1–4]. Approaches to failure prognostics of batteries may be roughly classified into three categories, namely physics-based, data-driven, and hybrid approaches. This classification is similar to that for failure prognostics in other engineering domains [5]. Physics-based methods heavily rely on modeling battery degradation patterns using physics-based or physics-informed models, e.g., taking advantage of electro-chemical principles. The physics-based model could be a high-fidelity simulation model [6], an empirical/semi-empirical model [7], or a simplified model derived from physical laws [8]. For physics-based methods, failure prognostics of batteries usually consists of two main steps. The first step is to update the uncertain model parameters based on actual observations of a battery, using Bayesian methods [9–11], optimization-based methods [12], or other methods [13,14]. After the model updating, the physics-based model is then used to forecast the degradation of the battery and thus estimate its RUL. The commonly used models in the literature include the exponential function [15], the power-law function [11], the equivalent circuit models [16,17], and multi-physics simulation-based degradation models [18]. There are multiple advantages of physics-based methods. For instance, most of the physics-based methods do not require any run-to-failure data. Furthermore, physics-based methods have good extrapolation capabilities if the model accurately captures the underlying degradation principle of the batteries. However, this potential advantage becomes the main disadvantage, because the current state of the art knowledge of battery degradation is very limited due to the highly complicated degradation process [19]. The discrepancy between the model used for prediction and that of the underlying true physics could lead to large errors in the RUL prediction, resulting in sub-optimal if not costly consequences in the decision-making.

With the rapid development of machine learning (ML) and especially deep learning techniques in recent years, many data-driven approaches have been developed for prognostics to overcome the limitations of the lack of physical degradation knowledge [20–23]. In data-driven approaches, a model is constructed based on run-to-failure data to achieve a mapping between observables (voltage, current, etc.) and battery capacity or monitoring data and life directly [24]. Various ML models have been employed to construct such data-driven models for failure prognostics of batteries. For instance, numerous neural network-based data-driven models, including regular artificial neural network models and models based on deep neural networks, have been proposed for the capacity estimation of lithium-ion batteries [25–27]. The Gaussian process regression model-based method has also been proposed to estimate the RUL of batteries, even though it is not as widely used as neural network architectures [28,29]. Moreover, to overcome the limitations of individual ML models and improve the RUL estimation accuracy and robustness, ensemble learning methods have also been investigated in the literature and have shown promising performance [30,31]. Other researchers have shown promise with the use of recurrent neural network architectures [32,33], including combining these architectures with classical Bayesian approaches to enable uncertainty quantification [34]. The advantage of an ML-based data-driven model is that it is capable of generating the required mapping that is not conditioned upon an understanding of the underlying physics of degradation. The ML model learns the degradation pattern automatically purely based on run-to-failure data. The disadvantage is that the data used for training significantly affects the performance of the data-driven model. If the prediction region is outside of the training space, the data-driven models could have a very large prediction error. The model sometimes may also create predictions that violate the physical laws.

While both physics-based methods and emerging data-driven methods have their own advantages and have shown good performance for RUL estimation of batteries in some situations, their disadvantages have impeded the adoption of the methods in practical applications. This has motivated the development of hybrid approaches that combines physics-based knowledge with data-driven models [35,36]. The hybrid approaches have been explored in two main directions. In the first direction, run-to-failure data is used to correct the physics-based degradation models [37–39]. In the second direction, physics-based knowledge is incorporated into data-driven models to improve the extrapolation accuracy of data-driven models for failure prognostics of batteries [40–43]. A comprehensive review of various hybrid approaches for battery RUL estimation is available in [41]. It is believed that the hybrid approaches will outperform the performance of physics-based and data-driven methods by combining the advantages of these two types of methods.

Building upon the existing lines of inquiry into hybrid modeling, the proposed approach combines physical insights from physics-based forecasting models and the ability to assimilate existing experimental data to capture the complex performance evolution of lithium-ion batteries. The proposed approach seeks to overcome a number of challenges associated with existing approaches, namely the lack of uncertainty quantification in hybrid methods, the difficulty associated with training recurrent neural networks, and the expert knowledge required to enforce physics constraints. To this end, the proposed method leverages the observation that battery discharge capacity is a monotonically decreasing phenomenon to enforce physical constraints in a neural network ensemble. Furthermore, the network is trained to act as a neural differential operator that learns the discharge capacity rate instead of learning the discharge capacity directly. This formulation provides a direct way to enforce constraints on time derivatives as well as a way to perform discharge capacity forecasting through direct time integration of the neural operator (avoiding the need for recurrent architectures). To learn the operator, an architecture inspired by DeepONet [44] is used, consisting of a branch network that encodes the input function space (i.e., the domain of battery cells) and a trunk that encodes the domain of the output function (i.e., the number of cycles). To capture the epistemic uncertainty associated with lack of knowledge of the exact battery degradation phenomenon, the network is formulated as a Bayesian neural network capable of expressing uncertainty over its own predictions. The resulting

algorithm is a physics-constrained Bayesian neural network that provides high flexibility and improved accuracy for RUL prediction with respect to existing methods. Using the battery dataset presented in the paper by Severson et al. [45] as an example, the proposed method was compared with a simplified physics-based degradation forecasting model and two data-driven prognostics models. The results show that the proposed physics-constrained neural network can provide more accurate and precise RUL estimation than the other methods using the same training data.

This paper is organized as follows. Section 2 describes all the methods considered, including the proposed approach. Section 3 presents results and discussion for all the methods considered. Finally, Section 4 provides concluding remarks and future research directions.

2. Predictive approaches for battery RUL forecast

In this section, an implementation of the failure forecast model based on simplified physical laws and two data-driven failure prognostic methods are presented in Sections 2.1, 2.2.1, and 2.2.2 respectively. These three methods will be compared with this paper's proposed physics-constrained neural network method presented later in Section 2.3. In the following sections, capacity is used to refer to discharge capacity which is generally used to track capacity degradation. Capacity is always presented in Ampere-hour (Ah) units unless otherwise noted.

It should be noted that all three methods are distinct and, as a result, are able to take different types of inputs. In particular, the neural network approach is the most flexible and can incorporate a larger number of inputs than the other methods without dramatically sacrificing efficiency (e.g., the GPR methods require the computation of the kernel matrix which becomes expensive with more inputs). This flexibility is leveraged to include additional potentially useful information as input to the neural network.

2.1. A Physics-Based Failure forecast model

The Failure Forecast Model (FFM) form proposed by Voight [46] successfully models many geophysical (see [47]) and material-level failure mechanisms (see [47–49]) that share a common self-accelerating degradation characteristic as the system approaches failure. We observe that the inverse of the battery capacity evolution accelerates with time and can, therefore, be modeled using the FFM. We assume a failure forecast model that has self-accelerating characteristics and perform Bayesian inference to predict the RUL and the uncertainty bounds in the prediction.

Let $C(t)$ denote the capacity of the battery at time (or equivalently, cycle) t. Let $R(t) = C(t)^{-1}$ denote the inverse of the discharge capacity. Since the discharge capacity is a monotonically decreasing phenomenon, the inverse of the discharge capacity is a monotonically increasing phenomenon. As per the general FFM model form, $R(t) = C(t)^{-1}$ is modeled by using a first order differential equation that takes the form

$$\frac{dR}{dt} = kR^{\alpha}, \text{ where } k > 0 \text{ and } \alpha > 1. \tag{1}$$

Here, the constants k and α are specific parameters for the physical process (discharge capacity degradation here). Using Eq. (1), the quantity $R(t)$ can be solved by assuming that $R(t = t_f) = R_f$, where, t_f is the time of failure, such that

$$R(t) = \left(R_f^{1-\alpha} + k(\alpha - 1)(t_f - t)\right)^{\frac{1}{1-\alpha}}. \tag{2}$$

The capacity failure threshold is defined as $C_f = R_f^{-1}$, which is assumed to be $C_f = 0.8$ in this paper. The discharge capacity $C(t)$ is obtained as

$$C(t) = \left(C_f^{\alpha-1} + k(\alpha - 1)(t_f - t)\right)^{\frac{1}{\alpha-1}}. \tag{3}$$

The value $\alpha = 2$ represents a special linear case of the FFM, such that

$$C(t)|_{\alpha=2} - C_f = k(t_f - t). \tag{4}$$

The value of $\alpha < 2$ represents the case where the discharge capacity follows a convex down trend (curving upward along the horizontal axis). On the other hand, $\alpha > 2$ represents the case where the discharge capacity follows a concave down trend (curving downward along the horizontal axis). As alpha increase, the value of $C(t)$ decays faster as the system approaches failure. Since discharge capacity is a monotonically degrading phenomenon with the rate of decay increasing over time, $\alpha \geq 2$ is of interest.

The goal here is to make a probabilistic estimate of the parameters k, α, t_f, and the noise parameter σ_l (standard deviation of the battery capacity data) using a portion of gathered capacity data $\mathscr{D}$. Let Θ denote the random vector consisting of the parameters to be estimated, i.e., $\{k, \alpha, t_f, \sigma_l\}$. The parameter vector can be inferred using Bayes theorem, such that

$$P(\Theta|\mathscr{D}) = \frac{P(\mathscr{D}|\Theta) \times P(\Theta)}{P(\mathscr{D})}. \tag{5}$$

In the equation above, $P(\Theta)$ denotes the prior belief of the parameter vector. Since $k > 0$, the prior for k is assumed to be a uniform distribution ranging from 0 to an arbitrarily large number (representing ∞). The prior for α is assumed to be

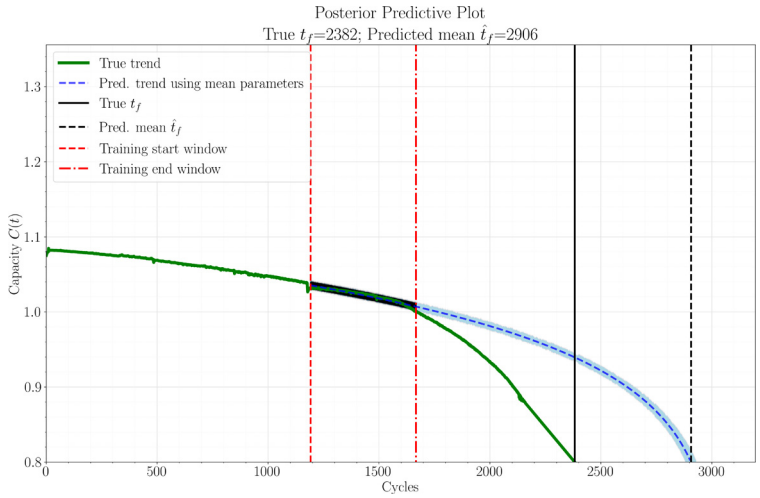

Fig. 1. Illustration of FFM used to generate posterior predictive plot.

uniform distribution ranging from 2 to 10. The prior to σ_l is chosen to be half normal, a weakly informative distribution as suggested by Gelman et al. [50,51]. Since t_f is positive and greater than the current time, the prior for the time of failure is assumed to be a uniform distribution from the current time t to an arbitrarily large number (representing ∞). The term $P(\mathscr{D})$ in Eq. (5) is the evidence and is simply a normalizing quantity. Finally, the term $P(\mathscr{D}|\Theta)$ denotes the likelihood of observing data $\mathscr{D}$ given the parameter vector Θ. If a specific forward measurement model is adopted, a tailored likelihood function could be derived from that. In this paper, the FFM (Eq. (3)) itself is used to model the likelihood function. To capture the heteroscedastic converging effect of the noise structure, the inference is performed in the logarithmic space $\tilde{\mathscr{D}} = \log(\mathscr{D})$. The likelihood of observing $\tilde{\mathscr{D}}$ given the parameters vector $\Theta = \{k, \alpha, t_f, \sigma_l\}$, denoted by $P(\tilde{\mathscr{D}}|\Theta)$, is modeled using a normal distribution with the standard deviation σ_l and mean μ_l, i.e., $P(\tilde{\mathscr{D}}|\Theta) = N(\tilde{\mathscr{D}}|\mu_l, \sigma_l)$, where the mean of the likelihood function is given by

$$\mu_l = \log\left(\left(C_f^{\alpha-1} + k(\alpha-1)(t_f - t)\right)^{\frac{1}{\alpha-1}}\right). \tag{6}$$

To obtain the joint posterior distribution of parameters constituting the parameter vector Θ using Eq. (5), a sampling-based Markov Chain Monte Carlo (MCMC) approach is implemented using a Metropolis sampler within the PyMC3 Python package. Once the parameter vector is inferred using the recorded data $\tilde{\mathscr{D}}$, the future discharge capacity data $\hat{\tilde{\mathscr{D}}}$ (called the posterior predictive distribution) may be forecast using the samples of the inferred parameters run on the FFM model that was used to construct the likelihood function. Mathematically, $\hat{\tilde{\mathscr{D}}}$ is conditioned on the recorded data $\tilde{\mathscr{D}}$, such that

$$P(\hat{\tilde{\mathscr{D}}}|\tilde{\mathscr{D}}) = \int_{\Omega_\Theta} P(\hat{\tilde{\mathscr{D}}}|\Theta) P(\Theta|\tilde{\mathscr{D}}) \, \mathrm{d}\Theta. \tag{7}$$

Fig. 1 illustrates an example of a posterior predictive plot inferred using a window of data that forecasts the time of failure of the battery's capacity as well as yields uncertainty bounds of the prediction. We observe that the prediction overshoots the true trend for the considered window. This is because the assumed failure model described in Eq. (3) does not fully capture the characteristics of battery capacity curves. It is designed for forecasting damage or fracture type of failure, where degradation notably accelerates towards the end. However, the capacity degradation of a battery behaves differently. Unlike the fracture type of failure, the slope of the capacity curve is almost constant towards the end of the battery's life. Consequently, the FFM model does not fully capture this degradation characteristic, resulting in an overshooting of the prediction until the data window comes closer to the failure threshold.

To predict the distribution of the time of failure (and by extension RUL), parameter estimations are performed using a moving window of recorded data because the data contained in the moving window will contain the latest stage information about the capacity degradation that informs the degradation curve. A data window size of 20% is used. Let $\hat{t}_f$ denote a realization of time, t denote the current instance of time, then the corresponding RUL denoted by $\hat{L}(\hat{t}_f)$ is given as

$$\hat{L}(\hat{t}_f, t) = \hat{t}_f - t. \tag{8}$$

We note that this is not a recursive approach since a particular data window does not affect the inferred parameters obtained by any other window and each window acts as an independent source of available data. For the purpose of demonstration, we assume non-overlapping windows spanning the entire time/cycle domain as shown later in Fig. 5 and 6.

2.2. Data-Driven prognostic models

Two different types of data-driven prognostic models, including a Gaussian process regression (GPR)-based model and a similarity-based model, will be provided in this subsection. Both models use the run-to-failure data of batteries to build a degradation model for failure prognostics.

2.2.1. A GPR-Based method

Assuming that N run-to-failure data-sets of battery capacity with different lengths exist, denoted as $\mathbf{C}_i,\ i = 1, \cdots, N$, where $\mathbf{C}_i = \{C_{i,j}, j = 1, \cdots, T_i\}$ is the i-th capacity dataset with a length of T_i. In order to build a GPR model to capture the degradation pattern of the battery, an finite impulse response (FIR) filter with a Hanning window is first applied to the original capacity curves to remove high-frequency noise and outliers in the data and thus smooth the capacity degradation curves.

Since GPR model training will be computationally very slow for a large number data set and further suffers from the curse of dimensionality, a down-sampling step is performed to reduce the size of the data after the smoothing of the degradation curves. Denoting the run-to-failure data after down-sampling as $\tilde{\mathbf{C}}_i = \{C_{i,k}, k = 1, \cdots, M_i\},\ \forall i = 1, \cdots, N$, where M_i is the length of the i-th capacity dataset after down-sampling, the finite difference of the capacity between two adjacent time steps is computed as

$$\Delta C_{i,k} = C_{i,k} - C_{i,k+1},\ i = 1, \cdots, N; k = 1, \cdots, M_i - 1, \tag{9}$$

where $\Delta C_{i,k}$ is the capacity degradation of the i-th battery from time step t_k to time step t_{k+1} after down-sampling. Note that the finite difference is computed by subtracting $C_{i,k+1}$ from $C_{i,k}$ since the capacity is decreasing over time.

To learn the underlying dynamics of the capacity degradation, a nonlinear autoregressive exogenous model (NARX) using the GPR model is constructed. NARX model is a nonlinear time-series modeling method that describe the response of a time series at the current time instant as a nonlinear function of the past values of the time series and current and past values of the driving factors (i.e., exogenous inputs). Various methods can be employed to model such a nonlinear functional relationship. When a GPR model is used to build this nonlinear function, the resulting model is called a GP-NRAX model [52]. The GP-NARX model in this paper is given by

$$\Delta \hat{C}_k = \mathcal{G}(\Delta C_{k-1}, \Delta C_{k-2}, \cdots, \Delta C_{k-p}; \theta), \tag{10}$$

in which $\mathcal{G}(\cdot)$ is the GPR model with hyper-parameters θ, $\Delta \hat{C}_k \sim \mathcal{N}(\mu_{C,k}, \sigma^2_{C,k})$ is the predicted capacity degradation at time step t_k which follows a Gaussian distribution with a mean of $\mu_{C,k}$ and a standard deviation of $\sigma_{C,k}$, and p is the number of lags in the NARX model. The hyper-parameters θ are estimated using the maximum likelihood estimate method based on the training data of $\Delta C_{i,k}$ obtained in Eq. (9).

For some situations, additional information from the run-to-failure data may be available. For instance, for the data used in the numerical example, the first 100 cycles of the discharge capacity curves are available. Denoting those additional parameters as ω and incorporating them into the GPR model, the GP-NARX model given in Eq. (10) can be rewritten as

$$\Delta \hat{C}_k = \mathcal{G}(\Delta C_{k-1}, \Delta C_{k-2}, \cdots, \Delta C_{k-p}, \omega; \theta). \tag{11}$$

In the above equations, the optimal down-sampling rate and the number of lags p need to be determined through a group of validation data that is not used in the test. After the construction of the GP-NARX model, for a new battery with measured capacity data $C_{new,q},\ q = 1, \cdots, T$, where T is the current number of measured time steps, we first predict the battery capacity degradation of the new battery at time step t_{T+1} as

$$\Delta \hat{C}_{new,T+1} = \mathcal{G}(\Delta C_{new,T}, \Delta C_{new,T-2}, \cdots, \Delta C_{new,T-p+1}, \omega; \theta). \tag{12}$$

Since $\Delta \hat{C}_{new,T+1} \sim \mathcal{N}(\mu_{new,T+1}, \sigma^2_{new,T+1})$, we first generate N_{MCS} samples of $\Delta \hat{C}$ according to the Gaussian distribution using Monte Carlo simulation (MCS). Let the MCS samples be $\Delta C_{new,r},\ r = 1, \cdots, N_{MCS}$, the capacity at time step t_{T+1} is given as

$$\hat{C}_{T+1,r} = C_{new,T} - \Delta C_{new,r},\ r = 1, \cdots, N_{MCS}, \tag{13}$$

where $\hat{C}_{T+1,r},\ r = 1, \cdots, N_{MCS}$ are the predicted MCS samples of battery capacity at time step t_{T+1} due to the uncertainty in the GPR prediction.

The above two equations (i.e., Eqs. (12) and (13)) are implemented recursively over time to forecast battery capacity over a long time period, and thereby allows one to predict the RUL of a new battery.

2.2.2. Similarity-Based approach

The similarity-based method proposed by Wang et al. in Ref [53]. for data-driven prognostics is also employed in this paper as a benchmark method for comparison. The basic idea of the similarity-based method is to match the degradation pattern of a new battery to that of the run-to-failure data in the library and thus estimate the RUL of the new battery based on the actual life of the existing units. The approach will be briefly summarized herein, and interested readers are directed to Ref [53]. for details. As shown in Fig. 2, a subset of the capacity data of the new battery is first extracted using a window size ΔT and denote the extracted data as $\mathbf{C}_s = \{C_{new,T-\Delta T}, \cdots, C_{new,T}\}$. For a given run-to-failure capacity curve $\mathbf{C}_i$ in the

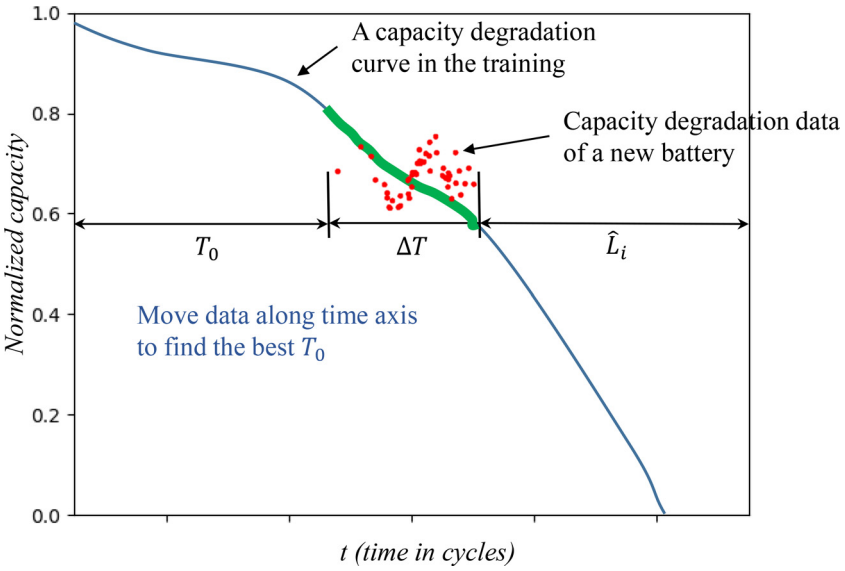

Fig. 2. Illustration of similarity-based RUL estimation method. Note that the data and curve shown in this figure are for explanation purpose only. They are not real capacity data and curve..

training data,the time instant T_0^* that $\mathbf{C}_i$ has the highest similarity with $\mathbf{C}_s$ is identified by solving the following optimization model:

$$\begin{aligned} &T_0^* = \arg\min_{T_0}\{f(\mathbf{C}_s,\ \mathbf{C}_i(T_0))\},\\ &s.t.\\ &T_0 \in [0,\ T_{e,i} - \Delta T], \end{aligned} \tag{14}$$

where $\mathbf{C}_i(T_0) = \{C_{i,T_0}, \cdots, C_{i,T_0+\Delta T-1}\}$ are the capacity data of the i-th capacity degradation curve starting from time T_0 with a window size ΔT, $f(\mathbf{C}_s, \mathbf{C}_i(T_0))$ is a loss function that quantifies the similarity of the two data-sets $\mathbf{C}_s$ and $\mathbf{C}_i(T_0)$, and $T_{e,i}$ is the end of life of the i-th battery obtained based on the capacity degradation curve. The loss function $f(\mathbf{C}_s, \mathbf{C}_i(T_0))$ could be defined in many different ways, such as likelihood function, Euclidean distance, or correlation measures. In this paper, the Euclidean distance is employed as the loss function.

The above optimization model basically moves the capacity degradation data of the new battery along the time axis to find the best match between the capacity degradation pattern of the new battery and that of the battery used for training (see the illustration in Fig. 2).

After the identification of T_0^*, the similarity, S_i, between the degradation pattern of the new battery and that of the i-th battery for training, is given as

$$S_i = f(\mathbf{C}_s, \mathbf{C}(T_0^*)), i = 1, \cdots, N. \tag{15}$$

The corresponding RUL estimate based on the i-th degradation curve is given by

$$\hat{L}_i = T_{e,i} - T_0^* - \Delta T, i = 1, \cdots, N. \tag{16}$$

Finally, the $S_i, i = 1, \cdots, N$, are used as weights and $\hat{L}_i, i = 1, \cdots, N$ as corresponding RUL estimates, to implement a re-sampling of $\hat{L}_i$ based on the weights S_i to obtain the probabilistic distribution of the RUL estimate of the new battery. In this paper, the re-sampling of $\hat{L}_i$ is performed using the particle filtering method.

2.3. Physics-Constrained bayesian neural network

Neural networks have been shown to be universal function approximators [54,55] as well as universal nonlinear operator approximators [44,56]. Their extensive expressive power comes from performing a series of function composition operations. On its own, the individual functions that make up a neural network are often differentiable linear or simple nonlinear functions. The key aspect is that these operations are differentiable and thus, amenable to be used in conjunction with gradient-based optimization algorithms. However, because of the emergent complexity that neural networks exhibit, they can often overfit and produce non-physical behavior. To this end, physics constraints can be introduced into the network via penalty terms in the loss function [57]. The constraints introduced in this work will be described in the next section. Neural networks are particularly amenable for imposing Neumann-type constraints because gradients are easily computed with auto-differentiation. For these reasons, neural networks were chosen as the proposed approach.

Moreover, the present problem exhibits significant variance in the battery RUL. For this reason, it is desirable to opt for a Bayesian neural network that can model the epistemic uncertainty associated with the predictions. In this work, it is assumed that epistemic uncertainty is associated with the parametric part of the model and not the time-dependent portion. The rationale behind this choice is that there is model-form uncertainty associated with the parametric model due to not having access to direct state variables associated with the physical phenomena and instead relying on indirect measurements. The effects of this assumption could be explored in future work. The time-dependent trunk only takes time as input while the parametric branch of the network takes as input the following vector

$$\lambda = \{C_0, C_{20}, C_{40}, C_{60}, C_{80}, C_{100}, \mathbf{var}(Q_{100-10}), \int_0^{100} C dt\}. \tag{17}$$

This vector of input parameters seeks to capture the trend of the RUL curve in the first 100 cycles. Through an iterative process, it was determined that 5 sample points from the initial RUL curve were sufficient. In addition to these samples, it was determined that the area under the curve (AUC), computed by integrating the first 100 steps of the RUL curve, was also a useful predictor. In addition to these features, the authors of [45] found that the voltage discharge curves from early cycles correlated well with the final RUL value. For this reason, the variance of the difference between the voltage discharge curve at 100 and at 10 cycles is used as a predictor as well.

2.3.1. Bayesian neural network overview

Unlike a traditional neural network which learns to predict point estimates, a Bayesian Neural Network learns the posterior distribution, $P(w|\mathcal{D})$, over the weights given an assumed prior distribution, $P(w)$ following Bayes theorem as

$$P(w|\mathcal{D}) = \frac{P(\mathcal{D}|w)P(w)}{\int P(\mathcal{D}|w')P(w')\mathrm{d}w'}, \tag{18}$$

where w are the model weights to be estimated, $\mathcal{D}$ represents the training data, and $P(\mathcal{D}|w)$ is the likelihood of observing $\mathcal{D}$ for given weights w.

As described in [58], obtaining the exact predictive posterior distribution is intractable for neural networks of any practical size due to the challenge of computing the term in the denominator of Bayes theorem. A common approach for approximating the predictive posterior is to rely on variational inference, as described in [59,60]. In this formulation, we begin by minimizing the Kullback-Leibler (KL) divergence between a variational posterior distribution $q(w\|\mu)$ and the desired posterior distribution $P(w|\mathcal{D})$. The KL divergence term is defined as

$$\mathbf{KL}(q(w|\mu)||P(w|\mathcal{D})) = \int q(w|\mu) \log\left(\frac{q(w|\mu)}{P(w|\mathcal{D})}\right)\mathrm{d}w, \tag{19}$$

where μ are the parameters of the variational distribution $q(w|\mu)$.

This formulation leads to the familiar problem of minimizing the variational free energy or maximizing the expected lower bound (ELBO) which consists of two terms: a KL distance between the variational distribution and the weights prior distribution (effectively acting as a complexity cost) and the negative log-likelihood term that effectively captures the reconstruction loss term. The total loss term is defined as

$$\mathcal{H}(\mathcal{D}, \mu) = \mathbf{KL}[q(w|\mu)\|P(w)] - \mathbf{E}_{q(w\|\mu)}[\log P(\mathcal{D}|w)], \tag{20}$$

This formulation is used to train the network by employing the Bayes by Backprop algorithm, as described in [58]. In practice, the loss function is approximated by employing Monte Carlo sampling such that the loss over a batch of size n is computed as

$$\mathcal{H}(\mathcal{D}, \mu) \approx \sum_i^n \log q\left(w^{(i)}|\mu\right) - \log P\left(w^{(i)}\right) - \log P\left(\mathcal{D}|\left(w^{(i)}\right)\right), \tag{21}$$

where $w^{(i)}$ represents the i-th Monte Carlo sample drawn from the variational posterior $q(w|\mu)$. In this way, the network can be trained more or less as a traditional neural network using the tools available in Tensorflow and Tensorflow Probability [61]. Specifically, dense variational layers are used in Tensorflow to keep track of the variational posterior distribution.

2.3.2. Neural network architecture and constraints

A Bayesian Neural Network architecture was used to model the battery discharge capacity rate. A Bayesian neural network, unlike a traditional network, treats its weights as random variables as described previously. As a result, the network functions as a probabilistic model where the weight distributions have been determined through variational inference (i.e., during training). The trained network functions as a neural differential operator that can be deployed by performing time integration over the desired inference time range. The concept of a neural differential operator refers to a neural network that learns the mapping between state variables and the derivative of a field variable (e.g., the discharge capacity) and it has been used in previous work [62] by the authors. The network architecture was inspired by DeepONet [44] and is shown in Fig. 3. DeepONet leverages the concept of neural networks as universal operator approximators (based on the universal nonlinear operator approximation theorem introduced by Chen and Chen [63]). DeepONet consists of

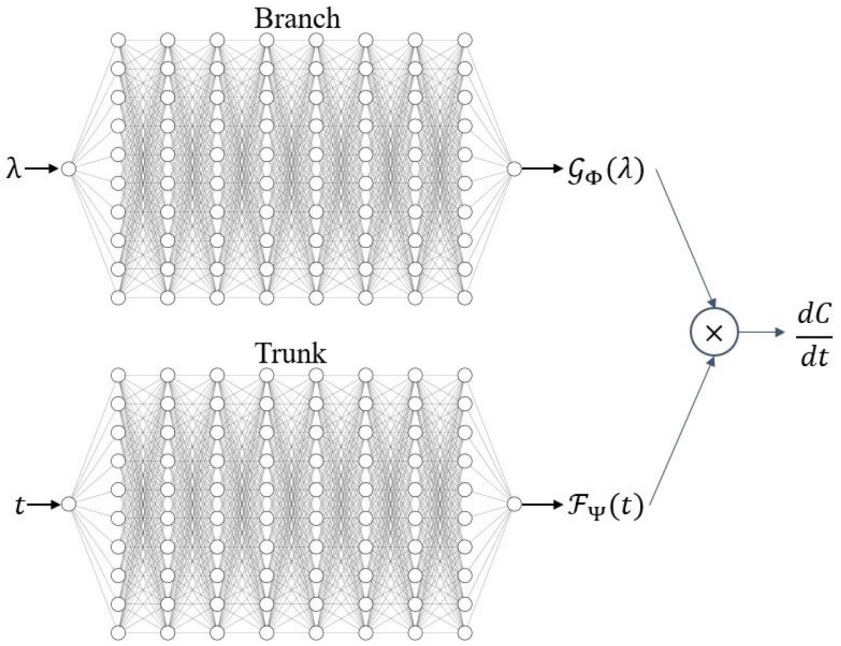

Fig. 3. Schematic of general neural network architecture.

- a branch network (i.e., $\mathcal{G}$) that encodes the discrete input function space and,
- a trunk network (i.e., $\mathcal{F}$) that encodes the domain of the output functions.

This architecture is desirable here because the objective is to construct the neural network as an operator that models the time derivative of the capacity discharge over a function space representing the different battery cells (parameterized by their initial capacity performance) and evaluated at an arbitrary point along their lifetime.

Consistent with Fig. 3, the trunk $\mathcal{F}$ processes the time dependence and the branch $\mathcal{G}$ handles the history or parametric dependence as

$$\frac{dC}{dt} \sim p\left(\frac{dC}{dt}|\lambda, t\right) = \mathcal{G}_\Phi(\lambda)\mathcal{F}_\Psi(t), \tag{22}$$

where Φ and Ψ are the weights of the branch and the trunk, respectively, and λ is the vector of input parameters related to the first 100 cycles of the discharge capacity curve. The branch learns the encoding over the discrete universe of battery cells (based on limited observations) while the trunk encodes the output domain, i.e., the number of cycles. In this case, both networks are exactly the same but they could be different if, for example, the branch needed to act as a functional over image data or high-dimensional functions.

The two networks consisted of 8 layers with 10 units, each with a nonlinear swish activation function. The swish activation function is defined as $f(x) = x/(1 + \exp(-x))$ and was chosen because it is part of the Tauber-Wiener functions used in DeepONet [44] and was shown to behave well based on experiments performed by the authors. The last layer of the branch network has a variational dense layer with a trainable posterior approximate Gaussian mean field and an isotropic Gaussian prior. This variational layer is meant to introduce the capability to predict epistemic uncertainty. The assumption that epistemic uncertainty is only present in the branch network was made to avoid having to deal with random variables when enforcing physical constraints associated with the discharge rate. If the physical constraint could be expressed as a probability density function, it would be desirable to include random variables in the trunk as well, but this was not the case here.

As the network predicts the discharge capacity rate, the residual of a multi-step integration evaluation is included to enforce mathematical consistency. The constraint is enforced via the following loss term,

$$\mathcal{L}_{num} = \| \sum_{j=0}^{s} \gamma_j C_{k+j} - h \sum_{j=0}^{s} \eta_j \left(\frac{dC}{dt}\right)_{k+j} \|_2^2, \tag{23}$$

where γ and η are coefficients from the Butcher tableau corresponding to the Adams-Bashforth linear multistep integration method, and h is the time step associated with the training data. The C_{k+j} term represents the $(k+j)_{th}$ time step in the C vector.

It is observed that the discharge capacity is a monotonically decreasing phenomenon. Based on this observation, a constraint enforcing negative self-acceleration is incorporated. The constraint is expressed mathematically by

$$\frac{d^2C}{dt^2} < 0. \tag{24}$$

As the network was constructed to model the discharge capacity, its second time derivative can be obtained by computing the gradient of the network via auto-differentiation. Furthermore, because the trunk is the only component with time dependence, the loss term associated with this physical constraint can be computed as

$$\mathcal{L}_{phys} = \left[\mathbf{ReLU}\left(\frac{d\mathcal{F}(t)}{dt}\right)\right]^2 \tag{25}$$

where **ReLU** is the rectified linear unit function which is non-zero only for positive values.

Lastly, the reconstruction loss was computed by evaluating the negative log-likelihood of the network with respect to the supervised discharge capacity rate as

$$\mathcal{L}_{recon} = -\sum_i^N \log p\left(\left(\frac{dC}{dt}\right)_i | \lambda_i, t\right). \tag{26}$$

where N is the number of samples in a batch. Note that this term corresponds to the third term in the variational loss term shown in Eq. (21). In addition to the reconstruction term, a complexity term is also computed as

$$\mathcal{L}_{complex} = \sum_i^N \log q(w^{(i)}|z) - \log P(w^{(i)}), \tag{27}$$

where $q(w^{(i)}|z)$ is the variational distribution as described in Section 2.3.1 and $P(w^{(i)})$ denotes the non-trainable isotropic Gaussian prior distribution. Because a Gaussian distribution was also assumed for the variational posterior, the parameter vector z contains the mean and standard deviation of the distribution.

The total loss was obtained by adding the four individual loss components as

$$\mathcal{L} = \zeta\mathcal{L}_{recon} + \beta\mathcal{L}_{phys} + \kappa\mathcal{L}_{num} + \rho\mathcal{L}_{complex}, \tag{28}$$

where ζ, β, κ, and ρ are used to weight the contribution of each loss term and are determined manually by inspecting the relative magnitude of the different loss terms. The loss is minimized using the Adam optimizer with a learning rate set to 10^{-4}.

Once trained, the discharge capacity is obtained by time-integrating the capacity rate as

$$C(t) = \int_0^t \frac{dC}{dt}dt, \tag{29}$$

where $\frac{dC}{dt}$ is sampled from the Bayesian neural network as shown in Eq. (22). The time T_e at which the discharge capacity will reach its limit state (i.e., its end of life), can then be predicted from which the RUL, $\hat{L}(t)$, is estimated as

$$\hat{L}(t) = T_e - t, \tag{30}$$

where t denotes the cycle at which the RUL is being estimated.

3. Application for model comparison

3.1. Dataset and pre-processing

The method was evaluated using the metrics described in Section 3.2 by predicting the RUL of the battery dataset provided in [45]. For consistency, the same 80/20 training/test random split was used for all methods. The capacity degradation curves were pre-processed to remove outliers and extend them beyond their endpoint by linearly extrapolating them to zero discharge capacity by following the instruction given in Ref [64]. Fig. 4 shows the battery capacity as a function of cycles for the original data and after removing outliers.

3.2. Performance metrics

To evaluate the performance of the algorithms, the following performance metrics were considered.

3.2.1. Error metric

The error was computed simply as the difference between the actual RUL curve and the prediction as

$$\epsilon_i = L_i - \hat{L}_i, \tag{31}$$

where i indicates the i-th test case.

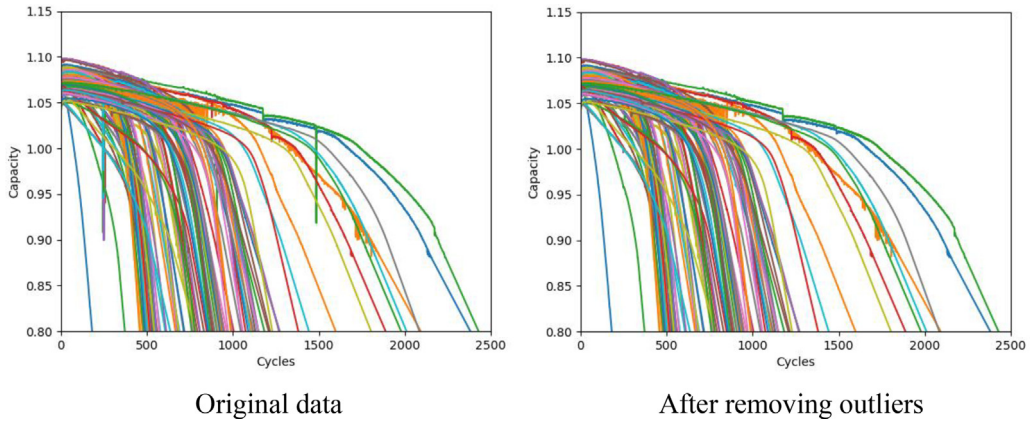

Fig. 4. Capacity data used in this example (Before and after removing outliers).

3.2.2. Root mean squared error

The root mean squared error (RMSE) is a simple way to evaluate the accuracy of the prediction. The RMSE was computed by comparing the mean RUL predictions to the actual RUL curve as

$$RMSE = \frac{1}{N}\sum_{i}^{N}(\epsilon_i)^2, \tag{32}$$

where the sum is performed over the N test cases.

3.2.3. S-Metric

The S-metric is based on the prognosis metric suggested by the authors of [65]. This score is an asymmetric metric that penalizes late predictions more heavily than early predictions. The metric is defined as

$$s = \begin{cases} \sum_i^N e^{\left(-\frac{\epsilon_i}{a_1}\right)} - 1, & \text{for } \epsilon_i < 0 \\ \sum_i^N e^{\left(\frac{\epsilon_i}{a_2}\right)} - 1, & \text{for } \epsilon_i \geq 0, \end{cases} \tag{33}$$

where s is the computed score, N is the number of test cases, and the parameters were defined as $a_1 = 1000$ and $a_2 = 1300$ following the guidance provided in [65] but scaling the parameters to be appropriate for the task at hand.

3.2.4. F1-Score

The F1-score was computed to test the robustness of the algorithms. To this end, the definition of true positives, false positives, and false negatives was taken from [66]. False positives were defined to be early predictions while false negatives were defined to be late predictions. These quantities were computed as

$$FP = \begin{cases} 1, & \text{if } \epsilon_i > t_{FP} \\ 0, & \text{otherwise} \end{cases}, \tag{34}$$

$$FN = \begin{cases} 1, & \text{if } -\epsilon_i > t_{FN} \\ 0, & \text{otherwise} \end{cases}, \tag{35}$$

where t_{FP} and t_{FN} are threshold values that are used to determine the acceptable region over which the prediction is considered a false prediction. For this paper, these two threshold were set to 50 cycles. Predictions that were within 50 cycles of the experimental data were considered true positives (TP). Lastly, the F1-score was computed as

$$F_1 = \frac{2TP}{2TP + FP + FN}. \tag{36}$$

3.3. Results

The four methods described in Section 2 were exercised to predict the battery RUL. Test 0 has an unusually high RUL while test 62 has one of the lowest RUL values of all the test cases observed. On the other hand, test 18 and 108 are typical of the battery RUL observed in most of the cells. For this reason, these four cases were chosen to exemplify the performance of each of the approaches. Fig. 5 and Fig. 6 show the mean RUL prediction and associated 95% confidence interval for all the methods considered.

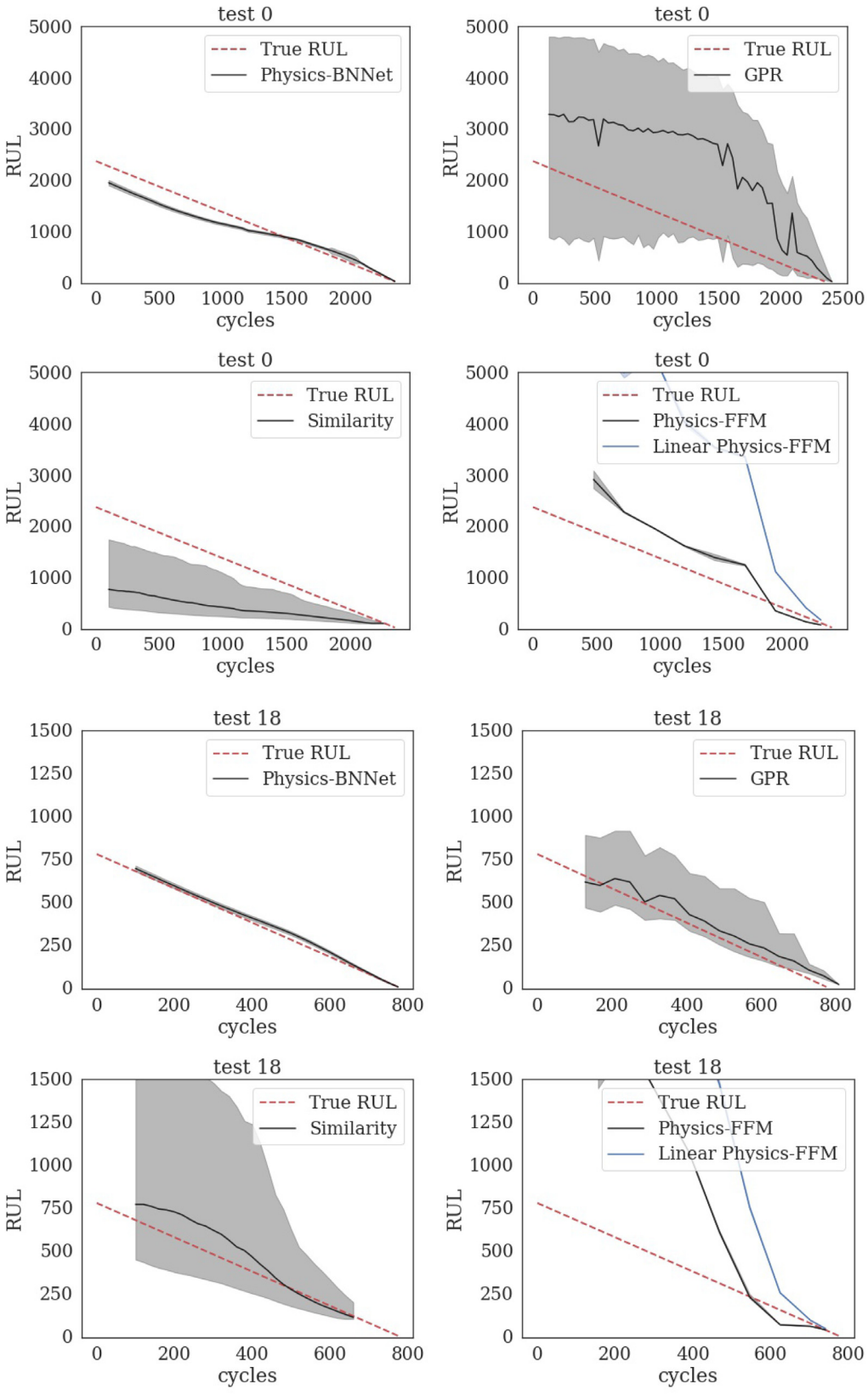

Fig. 5. RUL predictions of test 0 and 18 for all the approaches considered.

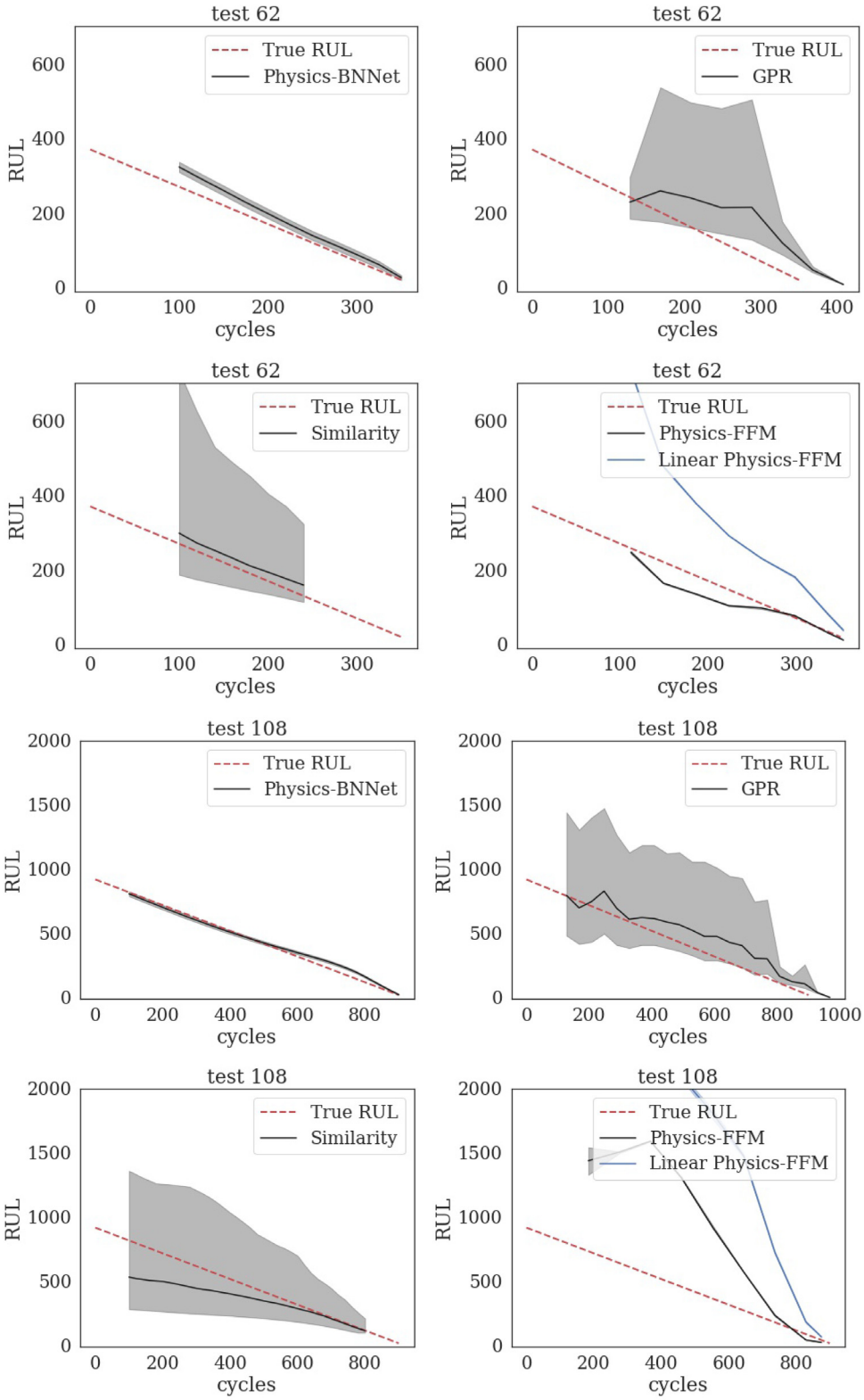

Fig. 6. RUL predictions of test 62 and 108 for all the approaches considered.

Table 1
Computational cost of a single RUL curve for each method.

Method	Training	Prediction
Physics-BNNet	5 hours	0.23s
GPR	0.42 hours	142s
Similarity	N/A	0.47s
Physics-FFM	N/A	60s

For the physics-based FFM method, a linear and nonlinear version of the model is shown. The FFM is a nonlinear regressor guided by a failure forecasting model equation that accounts for varying rate of capacity degradation over time, as observed in the battery's decline in capacity as well as other failure mechanisms such as fracture and crack propagation. By using the FFM, the remaining useful life of the battery can be probabilistically inferred. A specific version of the general nonlinear FFM is a linear FFM, which uses a forecasting model that assumes a constant degradation rate throughout the battery's life, as determined by the data used for forecasting.

Because all four methods are different, a direct comparison of computational cost is not too meaningful but runtimes for computing a single RUL curve are reported for completeness in Table 1. For the GPR, Similarity-based, and the FFM methods, a single CPU was used for computation of the RUL curve. For the BNNet approach, a single GPU was used to train the network for 50,000 epochs without early-stopping criteria. As shown in Table 1, even though the physics-constrained BNNet approach takes a very long time for training, it is the most efficient one among the four methods for RUL prediction. Because the GPR-based method implements a recursive uncertainty propagation scheme to quantify the uncertainty in the RUL prediction (see description in Section 2), it is the slowest. In addition, the similarity-based method and physics-based FFM method do not require any training time, due to the fact that these two methods use a sliding window to either check the similarity between the data of the window and the available run-to-failure data or to infer and update the model parameters of the FFM model using Bayesian method for RUL prediction. The computational time of these two methods for prediction is therefore mainly caused by similarity checking or Bayesian inference.

The results of the GPR-based method show that the GPR method can effectively quantify the uncertainty in the RUL predictions. As shown in Fig. 5 and Fig. 6, the true RUL curves lie within the 95 confidence bounds of the RUL prediction. The GPR-based method can accurately predict the RUL for tests 18, 62, and 108. During the construction of the GPR model, we only rely on the training data and did not leverage any physics knowledge. As a result, the prediction of the GPR model could have large prediction errors when the degradation pattern of the test is quite different from that of the training. For instance, the predicted RUL for test 0 from GPR-based method has large errors while it is pretty accurate for the other test cases. This is because the degradation of test 0 is unusually slow as mentioned above, which is quite different from that of tests used for training. The similarity-based method has a similar pattern as the GPR-based method. It can also effectively quantify the RUL prediction uncertainty and is pretty accurate for most cases. Very similar to the GPR-based method, the similarity-based method also has large prediction errors for test 0. This is attributed to the fact that both the GPR-based method and similarity-based method are interpolation-based methods. For interpolation-based methods, they could have good prediction accuracy when the prediction is within the training domain and may produce large errors if the prediction is outside the domain of the training data. The results of the FFM-based method shows that it overshoots the prediction in the early phase and the prediction gradually converges to the true value towards the end. This is because the FFM model is not suitable for forecasting battery capacity discharge because it is a nonlinear regression model designed for forecasting failure of a damage or fracture type of failure where degradation accelerates towards the end. The capacity degradation of battery behaves differently and can be divided into three stages: initial, mid and final phase. In the initial phase, the capacity degrades at a constant rate, in the mid phase the rate of degradation accelerates and in the final phase, the slope becomes constant. Unlike the fracture type problem, the degradation rate of battery capacity does not change much in the final phase. As a result, the FFM model form does not fully capture this degradation characteristics, resulting in overshooting of the prediction until the data window comes closer to the failure threshold. We observe that the general nonlinear FFM, which is capable of capturing changes in slope, leads to better predictions than the linear FFM. In contrast, the BNNet approach is fully nonlinear and does not enforce a specific form of nonlinearity, making it more flexible while still able to enforce the monotonically decreasing discharge capacity. By leveraging this physics knowledge within the neural network training, the BNNet approach performs the best among the presented four methods for the four test cases.

A qualitative and quantitative comparison of the RUL predictions for all the methods is presented in the next section.

3.4. Comparison and discussions

The performance was evaluated by comparing the RUL curves to the actual data. Fig. 7 illustrates the distribution of errors calculated as shown in Eq. 31. A positive error indicates early end-of-life prediction while a negative error indicates late end-of-life. For this reason, a positive value can be considered conservative. As illustrated, the residuals from the physics-constrained BNNet are narrower and centered around zero, whereas the physics-based FFM and the GPR predictions are generally broader. The similarity-based approach has similarly narrower distributions than the other GPR approach, but broader than the physics-constrained BNNet method. One of the trends that can be appreciated from this plot is the fact

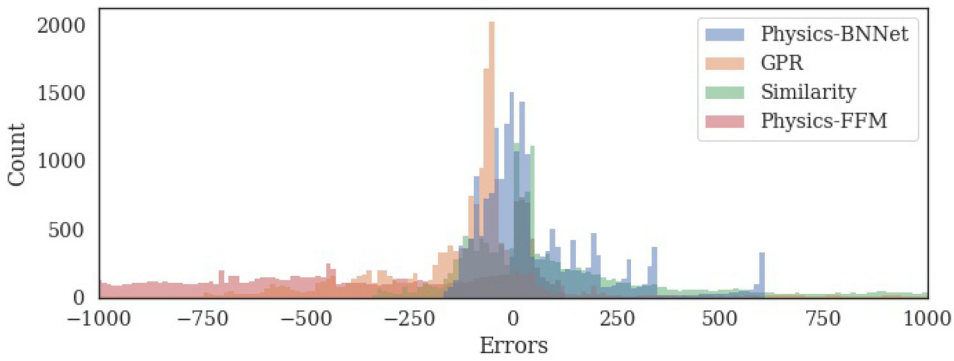

Fig. 7. Histogram of error distributions over the test set for the different techniques considered.

Table 2
Summary of RUL Prediction Performance.

Method	RMSE	S-metric	F1-score
Physics-BNNet	166	87	0.62
GPR	733	302	0.28
Similarity	425	101	0.46
Physics-FFM	5643	982	0.33

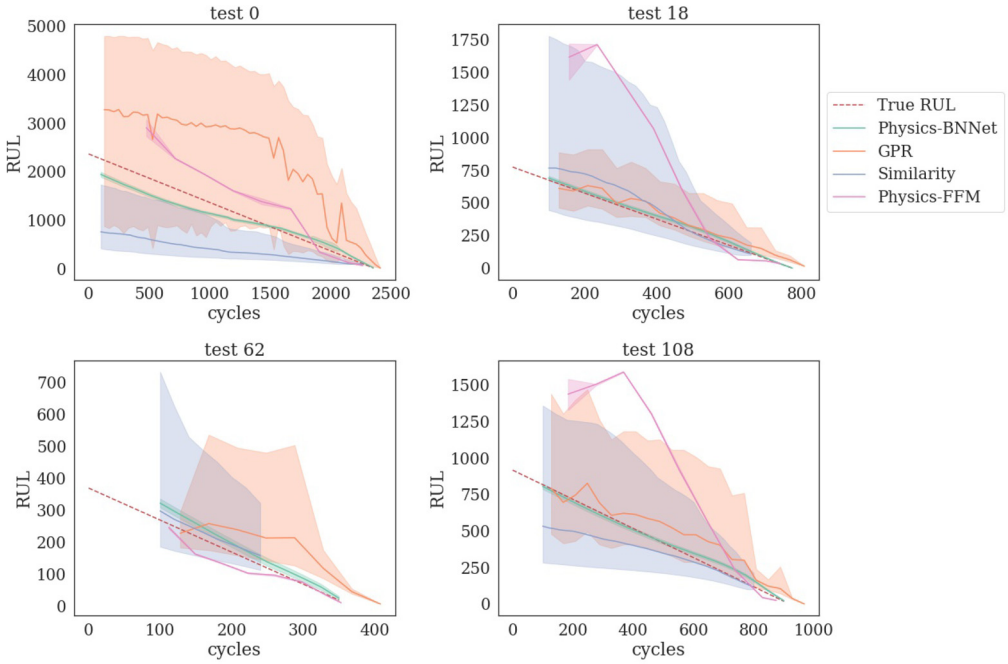

Fig. 8. RUL curve predictions for four representative cells.

that the BNNet and similarity-based approach are generally conservative (i.e., early prediction of the RUL) whereas the other methods are unconservative (i.e., late predictions). To quantify the trade-off between precision and accuracy of these methods, the F1-score is reported in Table 2. As shown, the computed F1-score is consistent with the visual observations about the BNNet method being more precise while having a good recall rate.

To illustrate the type of predictions obtained with the different methods considered, the RUL curves are plotted together in Fig. 8. Only four representative RUL curves are shown for easiness of visualization (these are the same curves shown in Fig. 5 and Fig. 6). As illustrated, both similarity-based approach and the GPR-based method have large 95% intervals, whereas the physics-based FFM and physics-constrained BNNet methods have generally narrower bounds. The similarity-based approach and the GPR-based method convey the uncertainty of their predictions through this large confidence interval which is useful because even when the predictions are not accurate, the model is able to express this uncertainty in the

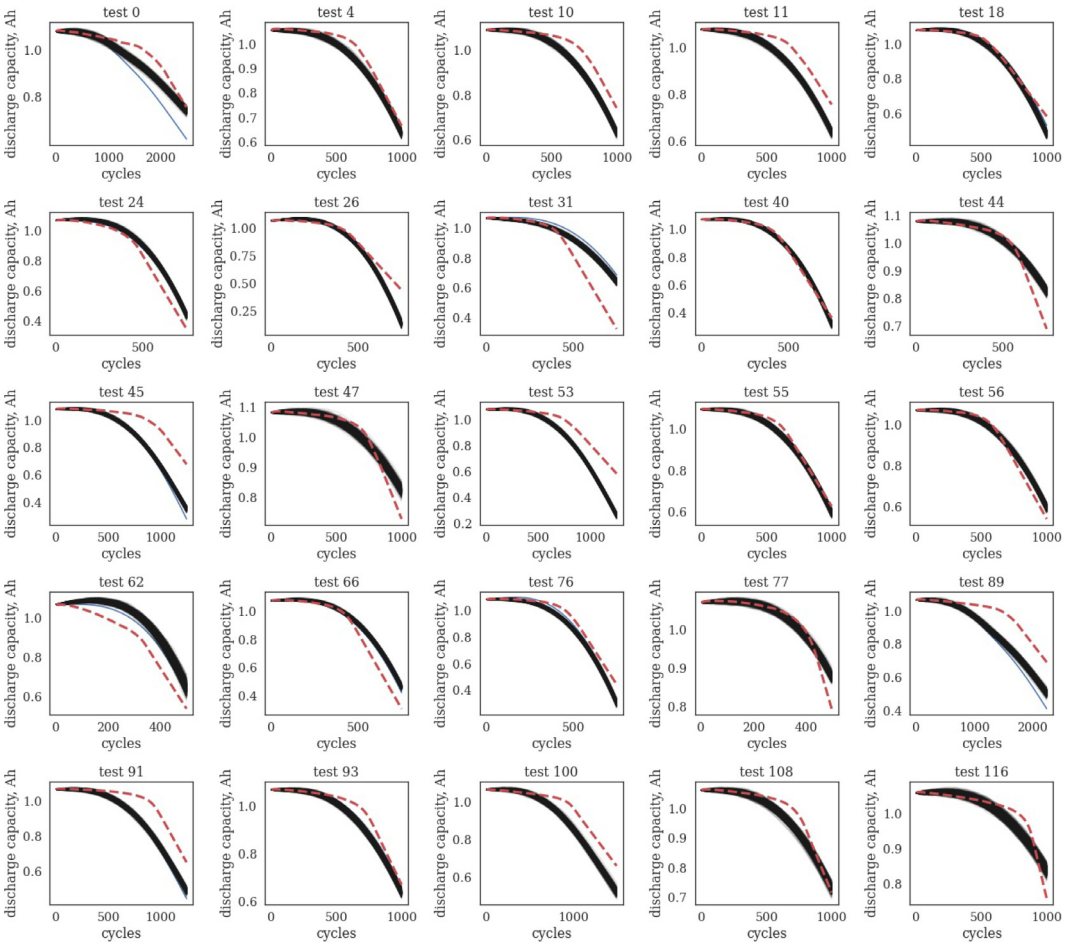

Fig. 9. Random discharge capacity curve samples generated by the BNNet (black) and actual experimental curve (dashed red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

prediction. The physics-constrained BNNet method has significantly narrower bands but it is also more accurate, as judged by the performance metrics shown in Table 2. The BNNet's narrow 95% intervals seem to correspond to increased accuracy. The prediction uncertainty for time of failure using the FFM model, which is a nonlinear regressor, is dependent on the variability of the battery capacity data that is used to probabilistically infer the model parameters for the FFM regressor equation. In the case of battery capacity curves shown in Fig. 4, the variability is low, leading to a narrow prediction interval for the time of failure and the remaining useful life (RUL). Additionally, the prediction uncertainty is highest in the early phase of the prediction as uncertainty propagates during extrapolation for future capacity, and becomes narrower as the data window nears the failure threshold (see Fig. 1). As mentioned, quantitative performance metrics are shown in Table 2. The metrics show a consistent trend, with the physics-constrained BNNet method being the most accurate followed by the similarity-based approach method, then the GPR-based method, and lastly the physics-based FFM.

As the BNNet exhibits superior accuracy, its performance merits further exploration. One way to evaluate the performance of the proposed method is to perform hypothesis testing. We obtain the predicted RUL, $\hat{L}$ from the probabilistic model as described in Section 2.3.2. We can then test $H_0 : \hat{L}_\mu = L$ versus $H_1 : \hat{L}_\mu \neq L$, that is, whether the actual RUL curve matches the mean of the samples of $\hat{L}$ generated from the physics-constrained BNNet. To that end, a T-test was performed by generating 500 samples from the BNNet model, and then computing the RUL curve and comparing it to the experimental data. The test was performed at each cycle along the RUL curve to obtain a distribution of T-tests. The proportion of times the null hypothesis was accepted (i.e., the proportion of cases for which the experimental mean agrees with the predicted mean) with a 95% confidence level (two-sided test with p-value less than 0.05) is plotted in Fig. 10 for all the test cases. As illustrated, the lowest acceptance rate is 90% which indicates that the distribution sampled from the BNNet generative process agrees well with the experimental data most of the time. To further illustrate this result, Fig. 9 shows 500 random discharge capacity curve samples generated by the BNNet plotted alongside the actual experimental curve (shown in red). The prediction of each curve takes on average 0.23 seconds. Note that the uncertainty does not grow significantly with time

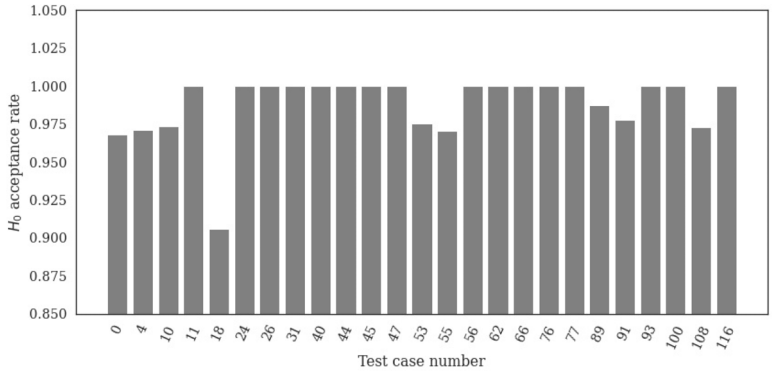

Fig. 10. Proportion of cycles for which the null hypothesis $H_0 : \hat{L}_\mu = L$ was accepted.

because the predictions are not recursive due to the network learning the capacity discharge rate and forward computation achieved through time integration (with the RK45 integrator).

The superior performance of the BNNet compared to the other approaches may be attributed to the fact that the BNNet approach combines many of the attributes of the other methods. Due to the Bayesian architecture of the BNNet method, this approach is similar to GPR, as explored in [67] which allows the BNNet to generate good predictions even when the data exhibit variability. Furthermore, the physics constraints of the BNNet approach were inspired by the constraints utilized in the FFM approach, namely the fact that the battery degradation process exhibits monotonic behavior and thus can be constrained to have a concave down trend through its second derivative. In addition to these features, the BNNet algorithm incorporated ideas from operator inference in its architecture which presumably increased its expressiveness. All of these features may have contributed to the good performance of the BNNet method.

4. Conclusions

This paper presented a comparison of four distinct approaches to perform RUL prediction of a lithium-ion battery dataset. As described in the original paper ([45]), the battery dataset introduces several challenges due to the broad variability in capacity discharge trends for nominally identical battery cells. For this reason, this paper was focused on the application of probabilistic methods that can account for the variability inherent in this dataset. Three of the four methods considered are well-established techniques in the field of RUL forecasting. It was shown that the FFM approach was too restrictive in the assumptions it makes about the degradation trends, and that it may not be appropriate to capture the physics of the battery degradation phenomenon. On the other hand, the data-driven approaches (GPR and Similarity-based) lacked the physical constraints to achieve high accuracy but they were able to express this epistemic shortcoming via large uncertainty bounds. Lastly, the BNNet approach combined features from the physics-based and data-driven approaches to achieve higher accuracy while predicting confidence intervals that were generally wide enough to capture the behavior of the actual dataset. The BNNet outperformed the other approaches in all the metrics considered which implies that the BNNet method is more robust, precise, and accurate than the other methods. Future opportunities for research include extending the network architecture to be recurrent or memory-aware for cases in which the physical phenomenon considered is not monotonic and to include additional physical constraints by leveraging insights from physical modeling and simulation.

Data availability

Data will be made available on request.

Acknowledgement

This work was funded by Sandia National Laboratories, Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. The University of California San Diego acknowledges Sub-Contract Agreement 2,169,310 from Sandia National Laboratories for its participation in this work.

References

[1] D.E. Acuña, M.E. Orchard, Particle-filtering-based failure prognosis via sigma-points: application to lithium-ion battery state-of-charge monitoring, Mech. Syst. Signal Process. 85 (2017) 827–848.
[2] T. Aggab, M. Avila, P. Vrignat, F. Kratz, Unifying model-based prognosis with learning-based time-series prediction methods: application to li-ion battery, IEEE Syst. J. 15 (4) (2021) 5245–5254.

[3] J. Zhang, J. Lee, A review on prognostics and health monitoring of li-ion battery, J. Power Sources 196 (15) (2011) 6007–6014.
[4] J. Guo, Z. Li, M. Pecht, A bayesian approach for li-ion battery capacity fade modeling and cycles to failure prognostics, J. Power Sources 281 (2015) 173–184.
[5] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B.D. Youn, M.D. Todd, S. Mahadevan, C. Hu, Z. Hu, A comprehensive review of digital twin-part 1: modeling and twinning enabling technologies, Struct. Multidiscip. Optim. 65 (12) (2022) 354.
[6] Y.H. Lui, M. Li, A. Downey, S. Shen, V.P. Nemani, H. Ye, C. VanElzen, G. Jain, S. Hu, S. Laflamme, et al., Physics-based prognostics of implantable-grade lithium-ion battery for remaining useful life prediction, J. Power Sources 485 (2021) 229327.
[7] S.S. Ng, Y. Xing, K.L. Tsui, A naive bayes model for robust remaining useful life prediction of lithium-ion battery, Appl. Energy 118 (2014) 114–123.
[8] G. Ning, R.E. White, B.N. Popov, A generalized cycle life model of rechargeable li-ion batteries, Electrochim. Acta 51 (10) (2006) 2012–2022.
[9] C. Hu, B.D. Youn, J. Chung, A multiscale framework with extended kalman filter for lithium-ion battery soc and capacity estimation, Appl. Energy 92 (2012) 694–704.
[10] F. Cadini, C. Sbarufatti, F. Cancelliere, M. Giglio, State-of-life prognosis and diagnosis of lithium-ion batteries by data-driven particle filters, Appl. Energy 235 (2019) 661–672.
[11] E. Walker, S. Rayman, R.E. White, Comparison of a particle filter and other state estimation methods for prognostics of lithium-ion batteries, J. Power Sources 287 (2015) 1–12.
[12] A. Downey, Y.-H. Lui, C. Hu, S. Laflamme, S. Hu, Physics-based prognostics of lithium-ion battery using non-linear least squares with dynamic bounds, Reliab. Eng. Syst. Saf. 182 (2019) 1–12.
[13] B. Saha, K. Goebel, J. Christophersen, Comparison of prognostic algorithms for estimating remaining useful life of batteries, Trans. Inst. Meas. Control 31 (3–4) (2009) 293–308.
[14] Y. Zhang, R. Xiong, H. He, X. Qu, M. Pecht, Aging characteristics-based health diagnosis and remaining useful life prognostics for lithium-ion batteries, ETransportation 1 (2019) 100004.
[15] C. Chen, M. Pecht, Prognostics of lithium-ion batteries using model-based and data-driven methods, in: Proceedings of the IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing), IEEE, 2012, pp. 1–6.
[16] G. Bai, Y. Su, M.M. Rahman, Z. Wang, Prognostics of lithium-ion batteries using knowledge-constrained machine learning and kalman filtering, Reliab. Eng. Syst. Saf. 231 (2023) 108944.
[17] M. Mishra, J. Martinsson, M. Rantatalo, K. Goebel, Bayesian hierarchical model-based prognostics for lithium-ion batteries, Reliab. Eng. Syst. Saf. 172 (2018) 25–35.
[18] S. Son, S. Jeong, E. Kwak, J.-h. Kim, K.-Y. Oh, Integrated framework for soh estimation of lithium-ion batteries using multiphysics features, Energy 238 (2022) 121712.
[19] G. Dong, J. Wei, A physics-based aging model for lithium-ion battery with coupled chemical/mechanical degradation mechanisms, Electrochim. Acta 395 (2021) 139133.
[20] P. Li, Z. Zhang, Q. Xiong, B. Ding, J. Hou, D. Luo, Y. Rong, S. Li, State-of-health estimation and remaining useful life prediction for the lithium-ion battery based on a variant long short term memory neural network, J. Power Sources 459 (2020) 228069.
[21] C.-P. Lin, J. Cabrera, F. Yang, M.-H. Ling, K.-L. Tsui, S.-J. Bae, Battery state of health modeling and remaining useful life prediction through time series model, Appl. Energy 275 (2020) 115338.
[22] Y. Liu, G. Zhao, X. Peng, C. Hu, Lithium-ion battery remaining useful life prediction with long short-term memory recurrent neural network, in: Annual Conference of the PHM Society, volume 9, 2017, p. 7pages.
[23] Y. Liu, Z. Hu, M. Todd, C. Hu, Data-driven remaining useful life estimation using gaussian mixture models, in: AIAA Scitech 2021 Forum, 2021, p. 1487.
[24] A. Thelen, M. Li, C. Hu, E. Bekyarova, S. Kalinin, M. Sanghadasa, Augmented model-based framework for battery remaining useful life prediction, Appl. Energy 324 (2022) 119624.
[25] M. Catelani, L. Ciani, R. Fantacci, G. Patrizi, B. Picano, Remaining useful life estimation for prognostics of lithium-ion batteries based on recurrent neural network, IEEE Trans. Instrum. Meas. 70 (2021) 1–11.
[26] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, L. Zhang, Remaining useful life prediction for lithium-ion battery: a deep learning approach, IEEE Access 6 (2018) 50587–50598.
[27] J. Hong, D. Lee, E.-R. Jeong, Y. Yi, Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning, Appl. Energy 278 (2020) 115646.
[28] J. Liu, Z. Chen, Remaining useful life prediction of lithium-ion batteries based on health indicator and gaussian process regression model, IEEE Access 7 (2019) 39474–39484.
[29] P. Tagade, K.S. Hariharan, S. Ramachandran, A. Khandelwal, A. Naha, S.M. Kolake, S.H. Han, Deep gaussian process regression for lithium-ion battery health prognosis and degradation mode diagnosis, J. Power Sources 445 (2020) 227281.
[30] V. Nemani, A. Thelen, C. Hu, S. Daining, Degradation-aware ensemble of diverse predictors for remaining useful life prediction, J. Mech. Des. 145 (3) (2023) 031706.
[31] S. Shen, M. Sadoughi, M. Li, Z. Wang, C. Hu, Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries, Appl. Energy 260 (2020) 114296.
[32] Y. Zhang, R. Xiong, H. He, M.G. Pecht, Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries, IEEE Trans. Veh. Technol. 67 (7) (2018) 5695–5705, doi:10.1109/TVT.2018.2805189.
[33] Y. Liu, G. Zhao, X. Peng, Deep learning prognostics for lithium-ion battery based on ensembled long short-term memory networks, IEEE Access 7 (2019) 155130–155142, doi:10.1109/ACCESS.2019.2937798.
[34] K. Liu, Y. Shang, Q. Ouyang, W.D. Widanage, A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery, IEEE Trans. Ind. Electron. 68 (4) (2021) 3170–3180, doi:10.1109/TIE.2020.2973876.
[35] M.A. Chao, C. Kulkarni, K. Goebel, O. Fink, Fusing physics-based and deep learning models for prognostics, Reliab. Eng. Syst. Saf. 217 (2022) 107961.
[36] A. Thelen, Y.H. Lui, S. Shen, S. Laflamme, S. Hu, H. Ye, C. Hu, Integrating physics-based modeling and machine learning for degradation diagnostics of lithium-ion batteries, Energy Storage Mater. (2022).
[37] Z. Xi, M. Dahmardeh, B. Xia, Y. Fu, C. Mi, Learning of battery model bias for effective state of charge estimation of lithium-ion batteries, IEEE Trans. Veh. Technol. 68 (9) (2019) 8613–8628.
[38] C. Jiang, M.A. Vega, M.D. Todd, Z. Hu, Model correction and updating of a stochastic degradation model for failure prognostics of miter gates, Reliab. Eng. Syst. Saf. 218 (2022) 108203.
[39] R. Jing, Z. Xi, X.G. Yang, E. Decker, A systematic framework for battery performance estimation considering model and parameter uncertainties, Int. J. Prognost. Health Manag. 5 (2) (2014).
[40] J. Shi, A. Rivera, D. Wu, Battery health management using physics-informed machine learning: online degradation modeling and remaining useful life prediction, Mech. Syst. Signal Process. 179 (2022) 109347.
[41] L. Liao, F. Köttig, Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction, IEEE Trans. Reliab. 63 (1) (2014) 191–207.
[42] Z. Tong, J. Miao, S. Tong, Y. Lu, Early prediction of remaining useful life for lithium-ion batteries based on a hybrid machine learning method, J. Clean. Prod. 317 (2021) 128265.
[43] S. Kohtz, Y. Xu, Z. Zheng, P. Wang, Physics-informed machine learning model for battery state of health prognostics using partial charging segments, Mech. Syst. Signal Process. 172 (2022) 109002.
[44] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, Nature Mach. Intell. 3 (3) (2021) 218–229, doi:10.1038/s42256-021-00302-5.

[45] 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 (5) (2019) 383–391, doi:10.1038/s41560-019-0356-8.
[46] B. Voight, A relation to describe rate-dependent material failure, Science 243 (4888) (1989) 200–203.
[47] B. Voight, A method for prediction of volcanic eruptions, Nature 332 (6160) (1988) 125–130.
[48] M. Todd, M. Leung, J. Corcoran, A probability density function for uncertainty quantification in the failure forecast method, in: Proc. 9th European workshop on structural health monitoring, BINDT, 2018, p. 12pages.
[49] N.M. O'Dowd, R. Madarshahian, M.S.H. Leung, J. Corcoran, M.D. Todd, A probabilistic estimation approach for the failure forecast method using bayesian inference, Int. J. Fatigue 142 (2021) 105943.
[50] A. Gelman, D. Simpson, M. Betancourt, The prior can often only be understood in the context of the likelihood, Entropy 19 (10) (2017) 555.
[51] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Rubin, Bayesian data analysis, Chapman and Hall/CRC, 1995.
[52] Y. Zhao, C. Jiang, M.A. Vega, M.D. Todd, Z. Hu, Surrogate modeling of nonlinear dynamic systems: a comparative study, J. Comput. Inf. Sci. Eng. 23 (1) (2023) 011001.
[53] T. Wang, J. Yu, D. Siegel, J. Lee, A similarity-based prognostics approach for remaining useful life estimation of engineered systems, in: 2008 international conference on prognostics and health management, IEEE, 2008, pp. 1–6.
[54] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (4) (1989) 303–314, doi:10.1007/bf02551274.
[55] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366, doi:10.1016/0893-6080(89)90020-8.
[56] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B.D. Youn, M.D. Todd, S. Mahadevan, C. Hu, Z. Hu, A comprehensive review of digital twin-part 1: modeling and twinning enabling technologies, Struct. Multidiscip. Optim. 65 (12) (2022) 1–55.
[57] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, doi:10.1016/j.jcp.2018.10.045.
[58] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural networks, 2015, 10.48550/ARXIV.1505.05424
[59] G.E. Hinton, D. van Camp, Keeping the neural networks simple by minimizing the description length of the weights, in: Proceedings of the sixth annual conference on Computational learning theory - COLT '93, ACM Press, 1993, pp. 5–13, doi:10.1145/168304.168306.
[60] A. Graves, Practical variational inference for neural networks, in: J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 24, Curran Associates, Inc., 2011. https://proceedings.neurips.cc/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf
[61] J.V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M.D. Hoffman, R.A. Saurous, Tensorflow distributions, (2017). arXiv:1711.10604
[62] D.A. Najera-Flores, M.D. Todd, A structure-preserving neural differential operator with embedded hamiltonian constraints for modeling structural dynamics, Comput. Mech. (2023), doi:10.1007/s00466-023-02288-w.
[63] T. Chen, H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, IEEE Trans. Neural Networks 6 (4) (1995) 911–917, doi:10.1109/72.392253.
[64] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B.D. Youn, M.D. Todd, S. Mahadevan, C. Hu, Z. Hu, A comprehensive review of digital twin-part 2: roles of uncertainty quantification and optimization, a battery digital twin, and perspectives, Struct. Multidiscip. Optim. 66 (1) (2023) 1–43.
[65] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: 2008 International Conference on Prognostics and Health Management, 2008, pp. 1–9, doi:10.1109/PHM.2008.4711414.
[66] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, M. Schwabacher, Metrics for evaluating performance of prognostic techniques, in: 2008 International Conference on Prognostics and Health Management, 2008, pp. 1–17, doi:10.1109/PHM.2008.4711436.
[67] J. Lee, Y. Bahri, R. Novak, S.S. Schoenholz, J. Pennington, J. Sohl-Dickstein, Deep neural networks as gaussian processes, 2017, 10.48550/ARXIV.1711.00165