Applied Energy 278 (2020) 115646

Contents lists available at ScienceDirect

Applied Energy

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

Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning

Joonki Hong [a,1,2], Dongheon Lee [a,1,2], Eui-Rim Jeong [b,3], Yung Yi [a,*]

[a] KAIST, Daehakro 291, Daejeon, South Korea
[b] Hanbat University, Dongseo-daero 125, Daejeon, South Korea

ARTICLE INFO

Keywords:
Lithium-ion battery
Remaining useful life
End-to-end deep learning
Dilated convolutional neural networks
Prediction uncertainty

ABSTRACT

This paper presents the first full end-to-end deep learning framework for the swift prediction of lithium-ion battery remaining useful life. While lithium-ion batteries offer advantages of high efficiency and low cost, their instability and varying lifetimes remain challenges. To prevent the sudden failure of lithium-ion batteries, researchers have worked to develop ways of predicting the remaining useful life of lithium-ion batteries, especially using data-driven approaches. In this study, we sought a higher resolution of inter-cycle aging for faster and more accurate predictions, by considering temporal patterns and cross-data correlations in the raw data, specifically, terminal voltage, current, and cell temperature. We took an in-depth analysis of the deep learning models using the uncertainty metric, t-SNE of features, and various battery related tasks. The proposed framework significantly boosted the remaining useful life prediction (25X faster) and resulted in a 10.6% mean absolute error rate.

1. Introduction

The ongoing development of high energy density lithium-ion (Li-ion) batteries has led to their application as mainstream energy sources in electric vehicles (EVs), energy storage systems (ESSs), and in information technology (IT) systems such as smartphones and laptops. The market for Li-ion batteries is expected to expand with the exponentially growing demand for eco-friendly energy harvesting and EVs. However, Li-ion cell instability is widely considered a significant obstacle to their wider commercial adoption [1]. For example, ESSs are often connected to energy harvesting grids to form GWh-level energy storage, but without preventative protections. An unexpected single Li-ion cell failure in such systems could ignite the entire ESS, resulting in a catastrophic explosion.

To ensure the more predictable operation and to protect Li-ion batteries, an array of research has been conducted on ways to predict the remaining useful life (RUL) of Li-ion batteries (see Section 2 for details). To be effective, the RUL prediction method needs to tackle the following two challenges.

- *Hardness:* First, the degradation modes of Li-ion batteries are complex and contain highly coupled nonlinear characteristics. Because of this complex inter-coupling, degradation in the Li-ion battery is not directly reflected in the capacity fade. Indications of Li-ion battery degradation are manifested in its operating patterns. The complexity of the operating patterns makes it hard for humans to develop a good rule-based algorithm to analyze them.
- *Swiftness:* The swift RUL prediction, i.e., the prediction that can be accomplished using only a few data cycles of the target cell, would allow new possible methods of prognostic protection during both manufacturing and operation. Without such prognostic protection, cell manufacturers are forced to utilize massive warehouses to minimize the risk of sudden cell death, and to verify the safety of new batteries by running dozens of charge–discharge cycles. Swift RUL prediction would enable the reliable detection of abnormal cells in a short period of time, resulting in a significant reduction in production costs. The method could also provide a quick diagnosis of Li-ion batteries being used in the field, and minimize the chances of unexpected failure in advance.

Main contributions. In this paper, we seek to provide a prediction of inter-cycle aging with higher resolution and swiftness by considering

* Correspondence to: Korea Advanced Institute of Science and Technology, Robovolt, Co., Ltd., South Korea.
E-mail address: yiyung@kaist.edu (Y. Yi).
[1] The first two authors contributed equally to this work.
[2] Korea Advanced Institute of Science and Technology, Robovolt, Co., Ltd.
[3] Hanbat University, Robovolt, Co., Ltd.

https://doi.org/10.1016/j.apenergy.2020.115646
Received 12 April 2020; Received in revised form 23 July 2020; Accepted 29 July 2020
Available online 14 August 2020

temporal patterns and cross-data correlations in the primary battery measurements. We summarize our main contributions in what follows:

- **C1.** We propose a new end-to-end deep learning framework for the swift prediction of the RUL of Li-ion batteries, which enables a more accurate prediction (10.6% error) from much shorter measurements (25X faster) than the SOTA (State-Of-The-Art) algorithm. By end-to-end we mean that prediction is made from the raw input data to target decision. The end-to-end deep learning framework is capable of learning to automatically extract the appropriate features necessary for RUL prediction from the complicated operating patterns. The key component of our framework is a *dilated* CNN based neural network architecture to analyze the long-term temporal patterns in the battery data.
- **C2.** We designed the neural networks and the training algorithm to output *the uncertainty of the prediction* as well as the prediction itself. The uncertainty estimate enables users to quantify how much we can trust the prediction results, which is highly useful in applying our prediction model to practice. We also seek for an *interpretable explanation* for the trained neural networks. We visualize the features drawn from the neural networks to compare the correlation with the human-defined features. We then tested our framework with two other battery-related tasks and verified that our framework could be extended to various applications.

Organization. This paper is organized as follows. Section 2 summarizes the related works and highlights the position of our work. Section 3 describes the end-to-end deep learning framework and details of the neural network architecture. In Section 4, we provide performance evaluation results, followed by interpretable explanations on the decision-making algorithm of our deep neural networks in Section 5. Finally, we conclude in Section 6.

2. Related work

Physical model-based. A number of early studies have attempted to estimate the state of degradation of Li-ion batteries. Approaches include the loss of lithium inventory (LLI), the loss of active material (LAM) [2,3], and the increase in internal resistance [4–6], based on the development of the physical models of several degradation mechanisms. For example, in [7,8] the physical model of the solid electrolyte interface (SEI) was used for the LLI and internal resistance estimation, and structural cracking and chemical deterioration models were used to appraise LAM. Comprehensive reviews of physical model-based RUL predictions can be found in [9]. These mechanism-specific and chemistry-specific models have achieved successful predictions, but their performance is known to be sensitive to battery operating conditions, because of the complicated thermal and electrical coupling between degradation mechanisms [10].

Data-driven. One emerging research area is data-driven RUL prediction which involves the use of statistical learning or machine learning with large amounts of data. Rather than building a complex physical model, those methods attempt to find a function between the electrical measurements in the charge–discharge cycles of the batteries and the RUL [11,12]. These approaches require immense amounts of data for chemistry-agnostic and operating condition-free predictions. The data-driven RUL prediction methods can be largely divided into two major categories: autoregressive capacity fade prediction and feature-based RUL prediction.

Autoregressive capacity fade prediction. Most existing data-driven studies have focused on the capacity degradation curve to predict the cycle life of the battery. Taking the capacity degradation pattern in the first hundreds of cycles, these studies have tried to predict the future degradation by auto-regressive model. Most commonly, exponential empirical models have been combined with advanced filter approaches such as Particle filter [13] or Kalman filter. Other works have employed machine learning methods such as Box–Cox transformation [14–16], relevance vector machine (RVM) [17], and Gaussian process regression (GPR) [18] to predict capacity degradation.

Autoregressive capacity fade prediction via deep learning. Recently, a number of works have adopted deep learning algorithms for better prediction. These works employ a recurrent neural network (RNN) [19,20] or its improved version, long short-term memory (LSTM) [21], to predict time sequential capacity degradation. However, these approaches have exhibited under-fitting problems. To overcome both overfitting and under-fitting problems, the authors in [22] built a hybrid model by combining the RNN and a convolutional neural network (CNN). *However, in general, works based on the capacity degradation curve can only capture a coarse level of degradation, since they abstract patterns from the charge–discharge cycle into a single discharge capacity value.* It is known that Li-ion batteries in their initial state (or new batteries) have negligible differences in discharge capacity, so that the degradation curve based techniques requires charge–discharge data from at least 25% of the entire lifetime of the target battery to predict the RUL accurately [23].

Feature-based RUL prediction. Several studies have tried to draw other aspects of features from the voltage, current and temperature profile. [24] extracted 21 features including the fluctuation, curvature, and kurtosis index of voltage, temperature, current curves and so on, and utilized support vector regression (SVR) to predict the RUL. [25] defined more complex features by quantifying the time of notable events in the charging and discharging process. Then, deep neural networks were employed to make a prediction. [26] utilized GPR to predict the RUL from their human-defined features. Recent groundbreaking research has revealed that when the cycle-to-cycle evolution of the discharge voltage curve is used as a feature, an elastic net can predict RUL with data from just the first 100 cycles (about 12.5% of the entire lifetime) [23,27].

Position of our work. We propose a deep learning framework to predict the RUL of a Li-ion battery directly from time-series measurements from the battery operation: terminal voltage, current and cell temperature. We try to extract various aspects of features just like feature-based RUL prediction, but the features are automatically extracted by the deep learning models in our framework. The proposed end-to-end deep learning model captures informal and diverse temporal patterns and learns to extract the ideal features for RUL prediction.

3. Swift prediction of the RUL of Li-ion batteries

We tackle the problem of RUL prediction of Li-ion batteries using end-to-end deep learning. In our framework, the measurements from the battery management system, terminal voltage, current, and cell temperature time-series data are given as inputs, and the neural network directly predicts the RUL of the target Li-ion battery. Instead of using the abstracted human-crafted features in existing works, we train the deep neural networks to process the raw data and capture complex temporal patterns and cross-data correlations between voltage, current, and temperature. This enables faster and more accurate RUL prediction. Fig. 1 illustrates our entire framework.

Unfortunately, one of the innate traits of the deep learning model is that it some-times makes arbitrary decisions when the test data distribution has inductive bias, which means that the test data distribution is disjointed from the training data distribution has inductive bias, which means that the test data distribution is disjoint from the training data distribution. The fundamental solution to this problem is to gather a wide range of training data. However, even with large amounts of data, there is still a chance of making an arbitrary decision, which can lead to critical failure in industrial applications. To provide a basis for determining whether to trust or not to trust the decision of the neural network, this paper proposes a method to estimate the uncertainty of the neural network output.

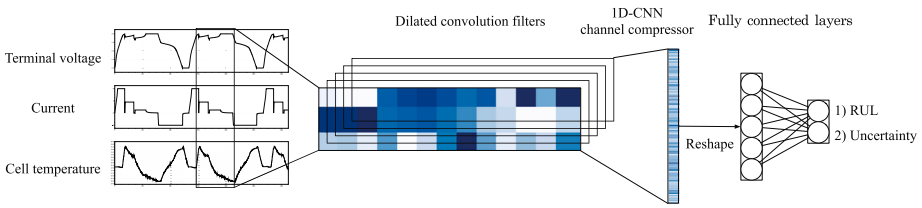

Fig. 1. Proposed end-to-end RUL prediction framework via deep learning. Here, end-to-end means raw data to target prediction. The neural networks get the time-series data of terminal voltage, current, and cell temperature to predict the RUL of Li-ion battery and state prediction uncertainty.

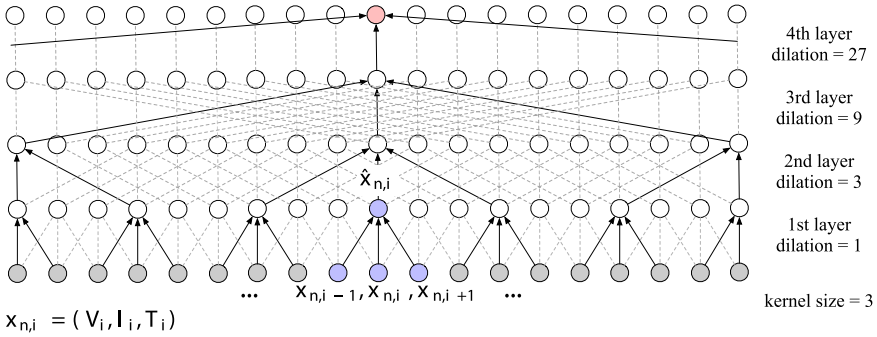

Fig. 2. Illustration of the dilated convolution operation. The inputs to the first layer correspond to the time-series data of terminal voltage, current, and cell temperature. The figure shows the typical architecture of dilated CNN with kernel size 3 and the tripping dilation size of each layer.

3.1. End-to-end RUL prediction of Li-ion batteries

The capacity of the Li-ion battery decreases as it is used. Unfortunately, even identical types of Li-ion battery cells may have different degradation curves, due to deviations in the production process and different operation conditions, which can make RUL prediction particularly difficult. Typically, when the Li-ion battery degrades to 80% of its initial capacity, we determine that the Li-ion battery has reached its end-of-life (EOL). Then, the RUL of the Li-ion battery can be defined as

$$RUL = C_{EOL} - C_M, \tag{1}$$

where C_{EOL} is the number of charge–discharge cycles before the Li-ion battery reaches EOL, and C_M is the number of cycles when the measurement ends.

The proposed framework utilizes the L seconds of the terminal voltage, current, and cell temperature measured every second, where L is the observation window or in other words, sample length. The input to the neural network is given by

$$\mathbf{x}_{i,j} = \begin{bmatrix} V_{i,c_j} & V_{i,c_j+1} & \cdots & V_{i,c_j+L-1} \\ I_{i,c_j} & I_{i,c_j+1} & \cdots & I_{i,c_j+L-1} \\ T_{i,c_j} & T_{i,c_j+1} & \cdots & T_{i,c_j+L-1} \end{bmatrix}_{3\times L}, \tag{2}$$

where V_{i,c_j}, I_{i,c_j}, and T_{i,c_j} are the terminal voltage, current, and cell temperature, respectively, of the ith battery cell at time $c_j \in [0, \text{EOL} - L + 1]$. All measurements are given to the neural networks as real values. Then, the neural networks are trained to learn the function mapping $F : \mathbf{X} \rightarrow Y$, where $\mathbf{X}$ is the set of the raw data $\mathbf{x}$ and Y which is the set of ground-truth RUL y of the data $\mathbf{x}$. The entire dataset $\mathbf{D} = \{\mathbf{d}_{i,j} := (\mathbf{x}_{i,j}, y_{i,j}) | i \in [0, I-1], j \in [0, J-1]\}$ is divided into 3 parts: the training part $\mathbf{D}_{train} = \{\mathbf{d}_{i,j} | i \in B_{train}\}$, the validation part $\mathbf{D}_{valid} = \{\mathbf{d}_{i,j} | i \in B_{valid}\}$, and the test part $\mathbf{D}_{test} = \{\mathbf{d}_{i,j} | i \in B_{test}\}$, where B_{train}, B_{valid}, and B_{test} are the disjointed indices of the batteries for training, validation, and testing, respectively.

3.2. Neural network architecture for end-to-end RUL prediction

Dilated CNN. This paper employs dilated CNN to predict the battery RUL. To analyze the inter-cycle aging of the Li-ion battery, we need to take data that is longer than one charge–discharge cycle as inputs. Even for a small cell, it takes hundreds of seconds or more to fully charge and discharge. RNN and its variants, such as long short-term memory (LSTM) and gated recurrent unit (GRU), are well known for their ability to interpret sequence inputs [28]. These architectures provide state-of-the-art performance for various sequence tasks such as language modeling, sentiment analysis, speech recognition, and time-series classification. However, their ability to capture considerably long term relationships is limited.

On the other hand, dilated CNN has proven capable of capturing long term relationships in time series data. The authors in [29] showed that a dilated CNN can generate natural human voice in raw audio levels, which requires the understanding of tens of thousands of steps in time-series data. Dilated convolution is a convolution applied to an input signal with definite gaps in the kernels. The dilated convolution for multivariate inputs can be formulated as

$$\hat{x}_{n,i} = \sum_{m=1}^{M} \sum_{t=-\lfloor (K-1)/2 \rfloor}^{\lfloor (K-1)/2 \rfloor} k_{n,t} x_{m,i+dt}, \tag{3}$$

where K is the length of the kernel, M is the number of input channels, $\hat{x}_{n,i}$ is the output for channel n at timestep i, $k_{n,t}$ is the tth kernel coefficients, and $x_{m,i}$ is the input for channel m at timestep i. A large value of dilation d causes dilated convolution to cover inputs longer than the kernel length, while dilated convolution with dilation size $d = 1$ reduces to a standard convolution operation. Fig. 2 shows how we designed a dilated CNN to extract long term relationships in the input sequence. While a conventional CNN has linearly growing receptive fields with the number of layers, a dilated CNN has exponentially growing receptive fields. As we can observe in Fig. 2, the exponentially increasing dilation size offers a wide receptive field of dilated CNN. We carefully control both kernel size and dilation size to build neural networks optimized for the RUL task.

Full architecture. The dilated CNN is the primary building block of the proposed neural network architecture. The first five layers of dilated CNN analyze the temporal patterns in multivariate input sequences. We double the dilation size for each layer to create sufficient receptive fields for the RUL prediction. The output of each dilated CNN layer is normalized before being fed to the next layer. This process is a well-known technique called batch normalization (BN). BN reduces the

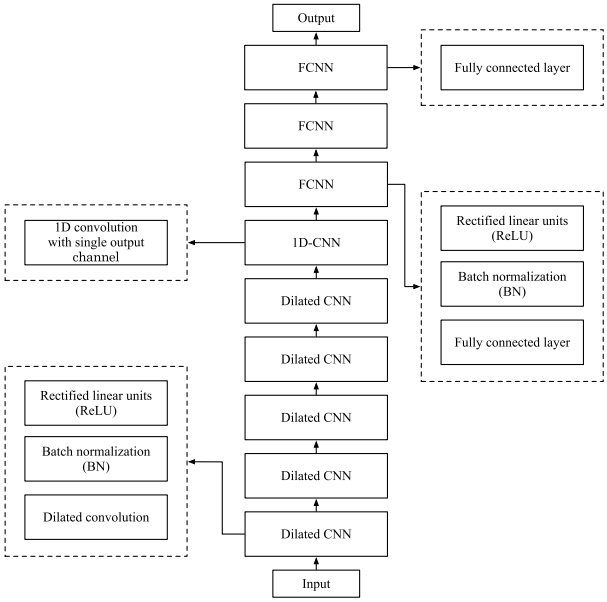

Fig. 3. Proposed dilated CNN based neural network architecture. The proposed dilated CNN consists of five layers of dilated CNN, one channel-compressing CNN layer, and three fully connected layers.

internal covariance shift, i.e., each layer of neural networks experiences similar input domains, making the training process fast and stable. The rectified linear unit (ReLU) is used as an activation function. We kept the length of each output sequence of the dilated CNN to maintain the temporal information while increasing the number of channels. Then, we compressed the multi-channel output sequence of the final dilated CNN layer with 1D-CNN to get the essential features. The final three concatenated layers of the fully connected neural networks (FCNN) allowed nonlinear mapping from the extracted features to the RUL prediction. Fig. 3 shows the full architecture of the proposed neural networks. The mean square error (MSE) loss is employed to train the neural network:

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(F_\theta(\mathbf{x}_n) - y_n)^2, \tag{4}$$

where N is the total number of training data, F is the function of the designed neural networks, θ is the parameters of the neural networks, $\mathbf{x}_n$ is a multivariate input data of index n, and y_n is a ground-truth RUL of the Li-ion battery. We use a stochastic gradient descent (SGD) optimizer with momentum to find the neural networks parameters to predict the RUL of Li-ion batteries [30].

3.3. Uncertainty estimates of the RUL predictions

Deep learning with artificial neural networks is a non-probabilistic approach to approximate a function mapping. The Bayesian neural network is one class of approaches used to estimate the uncertainty in neural networks using a Bayesian framework. In a Bayesian framework, we infer the posterior distribution with the following Bayes' rule:

$$p(\theta|\mathbf{D}) = \frac{p(\mathbf{D}|\theta)p(\theta)}{p(\mathbf{D})}, \tag{5}$$

where θ is the weight parameters of neural networks and $\mathbf{D}$ is a training dataset. The primary goal of a Bayesian neural network is to learn the true weight distribution $p(\theta|\mathbf{D})$ given the training data. Thus, unlike the conventional neural networks, the Bayesian neural network has a distribution of weight parameters rather than fixed weights. Since obtaining true posterior $p(\theta|\mathbf{D})$ is intractable, various works on Bayesian neural networks propose an explicit or implicit variational approximation of $q(\theta) \approx p(\theta|\mathbf{D})$. Then, predictive distribution of the Bayesian neural network can be formulated as

$$p(y^*|\mathbf{x}^*, \mathbf{D}) = \int p(y^*|\mathbf{x}^*, \theta)p(\theta|\mathbf{D})d\theta \tag{6}$$

$$\approx \int p(y^*|\mathbf{x}^*, \theta)q(\theta)d\theta, \tag{7}$$

where $\mathbf{x}^*$ is the test input data and y^* is the corresponding prediction of the neural networks. The distribution of weight parameters yields the distribution over the distribution of predictions, and the expected predictive distribution given the training data can be obtained by marginalizing the θ. Finally, how widespread the predictive distribution is becomes the uncertainty that the Bayesian neural network estimates about the data. The entropy of the predictive distribution is often defined as the uncertainty measure [31].

Unfortunately, Bayesian neural networks have high computational cost, and their performance depends significantly on the form of approximation used to calculate the posterior distribution, and their assumptions on prior weight distribution. Recent advanced works on probabilistic approaches to neural networks have theoretically proved that the predictive distribution of Bayesian neural networks can be approximated via implicit sampling, such as MC-dropout, or explicit ensembling, such as deep ensembles [32,33]. The following equation formulates the approximated predictive distribution via sampling:

$$p(y^*|\mathbf{x}^*, \mathbf{D}) \approx \frac{1}{H}\sum_{i=1}^{H} p(y^*|\mathbf{x}^*, \theta^i), \theta^i \sim q(\theta), \tag{8}$$

where H is the number of samples drawn and θ^i is the sampled weight parameters from the estimated weight distribution. In this paper, we use explicit ensembling to estimate the uncertainty in the RUL prediction. We sample H weight parameters by training H differently initialized neural networks independently. To infer the distribution of predictions $p(y^*|\mathbf{x}^*, \theta^i)$, we train the neural networks to output the mean and variance of the distribution, assuming that it follows a Gaussian distribution. We employ the negative log likelihood (NLL) loss instead of the MSE loss for uncertainty estimation:

$$NLL = \frac{\log V_\theta(\mathbf{x})}{2} + \frac{(y - M_\theta(\mathbf{x}))^2}{2V_\theta(\mathbf{x})}, \tag{9}$$

where M_θ and V_θ are the functions of the neural networks that output the mean and variance of the predictive distribution given input data $\mathbf{x}$, respectively. Next, using the same training dataset, we independently train randomly initialized multiple neural networks. The expected predictive distribution calculated by Eq. (8) is a Gaussian

Algorithm 1: Training algorithm for estimating the uncertainty of the RUL prediction

Input : Test data $\mathbf{x}^*$
Output: Predictive distribution $p(y|\mathbf{x}^*)$, uncertainty estimate $U(\mathbf{x}^*)$
Data : Training dataset $\mathbf{D}_{train}$, validation dataset $\mathbf{D}_{valid}$

Initialize the neural networks parameters $\theta_1, \theta_2, ...\theta_H$ randomly
for $i = 1$ *to* H **do**
  Loop until the terminal condition is met. One epoch:
    Calculate the NLL loss defined in **Eq. (9)** with the data from $\mathbf{D}_{train}$
    Optimize the neural networks parameter θ_i with stochastic gradient optimizer (SGD)
    terminal condition: validation accuracy on $\mathbf{D}_{valid}$ saturates
  end
end
Calculate the predictive distribution $p(y|\mathbf{x}^*)$ with **Eq. (8)**
Calculate the uncertainty estimate $U(\mathbf{x}^*)$ with **Eq. (10)**

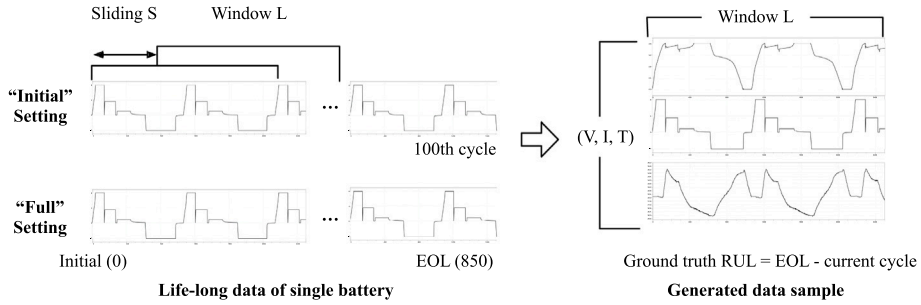

Fig. 4. Overall data augmentation process. The data augmentation process outputs the data samples with $3 \times L$ dimension.

mixture distribution. We define the uncertainty of the RUL prediction in the data $\mathbf{x}^*$ as follows

$$U(\mathbf{x}^*) = \frac{1}{M^2}\sum_{i=1}^{M}(\sigma_i^2 + \mu_i^2 - \mu^2), \tag{10}$$

where μ is the mean of the expected predictive distribution, σ_i is the variance, and μ_i is the mean of the predictive RUL distribution of the ith seed, respectively. The algorithm to train and calculate the uncertainty is described in Algorithm. 1.

4. Experiments

4.1. Data augmentation

We used the Li-ion battery dataset from MIT-Stanford [23], which is available online, to assess the proposed framework and neural network architecture. The dataset is generated from 124 commercial high-power LFP/graphite A123 APR18650M1A cells, which have a nominal capacity of 1.1 Ah and a nominal voltage of 3.3 V. These cells were cycled with various candidate fast-charging scenarios but discharged identically until the EOL in a temperature controlled environmental chamber (30 ° C). The Li-ion batteries were charged from 0% to 80% state-of-charge (SoC) with one of 72 different one-step or two-step charging policies. The charging time from 0% to 80% SoC ranged from 9 to 13.3 min. The terminal voltage, current, and cell temperature were gathered every second.

The proposed end-to-end framework utilizes the ground-truth label of the RUL, and the consecutive raw charge–discharge data of terminal voltage, current, and cell temperature. In general, the bigger the size and the greater the diversity in the dataset, the more performance and robustness we can achieve in deep learning. Thus, we augmented the data with an overlapped window slicing. First, we concatenated the consecutive cycle data from a single battery to make life-long time series data of the terminal voltage, current, and cell temperature. Then, we sliced the life-long data with the window size L, which is the input length to the neural networks, and with the sliding size S. This process generates the jth data $\mathbf{d}_{i,j}$ with at most $L - S$ overlap from the battery i. Finally, each sliced sample was labeled with the number of cycles left from the slicing point to the end of life. Since our model requires at least three cycles of information to make the best prediction, the typical values of L and S were 2500 and 500, respectively. The above process was performed for the data from all the cells.

The MIT-Stanford dataset consists of three battery batches separated by measurement trials. Fig. 5 shows the cycle lives of the cells in the Batches 1, 2 and 3. As we can observe in the figure, the different batches cover different cycle ranges. When the distribution of training data and test data is different, machine learning models can induce a large bias in the prediction, which is the well-known problem of inductive bias. In this paper, to minimize the effect of inductive bias, we created the training dataset by sampling a particular portion of batteries from Batches 1, 2, and 3. We used the same policy when creating the validation dataset and the test dataset, so all the datasets

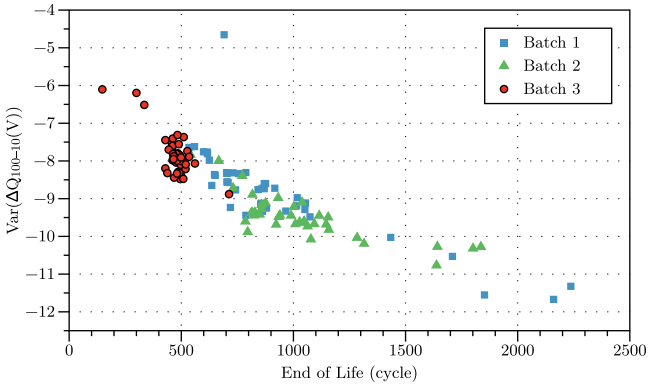

Fig. 5. Illustration of the MIT-Stanford Li-ion battery dataset. Each data point corresponds to a single battery. $Var(\Delta Q_{100-10}(V))$ implies the variance in the cycle-to-cycle evolution of the discharge voltage curve. We calculated the variance of the difference between the discharge voltage curve of the 100 th cycle and 10th cycle.

covered the entire cycle range. Approximately 70% of the total data was assigned to the training dataset, and the remaining data was split in half, into the validation dataset and test dataset. We did not allow the data from one battery to belong to two different datasets. Refer to Section 3.1. We trained the neural networks using the training dataset, and we used the validation dataset to perform early stopping. Finally, we evaluated the trained neural networks on the test dataset. For fair comparison the performance of the various models was averaged over randomly generated combinations of training, validation, and test datasets. The overall process of generating data for training and test is described in Fig. 4.

4.2. RUL prediction performance of the deep neural networks

We evaluated the RUL prediction performance of the various deep learning models and human-crafted feature-based baseline models, and present the results in this section. For the deep learning models, the 2,500-second-long data corresponding to approximately three charge–discharge cycle was given as an input, while the 100-cycle data was given to the previous state-of-the-art baseline models [23]. The multi-layer perceptron (MLP), CNN, CNN-LSTM were selected as the control groups for the deep learning models, since these models have shown superior performance in numerous time-series regression tasks [22,34,35]. We designed the neural network architecture and hyperparameters of the MLP, CNN, and CNN-LSTM, such as the number of layers, the kernel size, and the channel size, following the conventional rules found in many previous works, and then fine-tuned them empirically.

Table 1 summarizes the performance of the baseline models and the proposed dilated CNN model. The baseline methods focused on the RUL prediction using only the first 100 cycle data. For fair comparison, we trained our dilated CNN model to predict the RUL for any continuous 2,500-second data from the first 100 cycle data. In the remainder of this

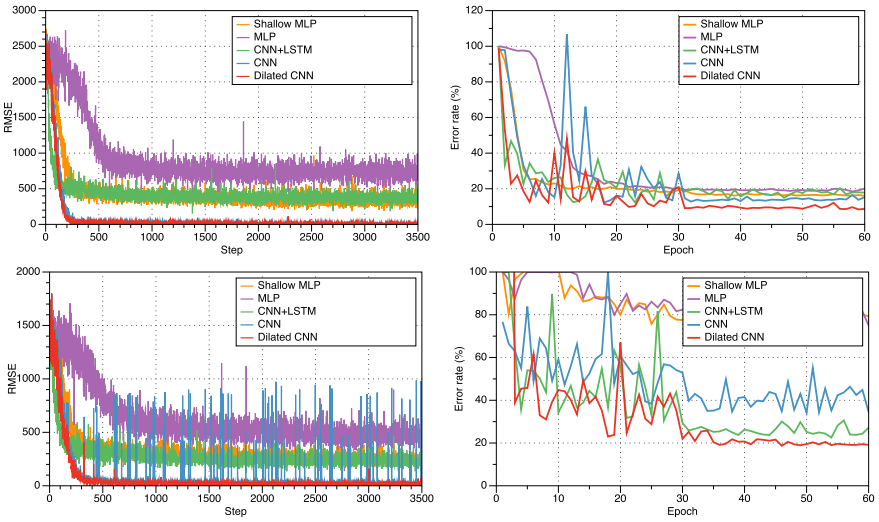

Fig. 6. Training losses and test errors of various deep neural network architectures over training epochs. The first row represents the experiment in the "Initial setting" and the second row shows the experiment in the "Full" setting.

Table 1
Performance evaluation of the human-crafted feature-based models and the proposed neural networks in the "Initial" setting. Here "Required cycle" means the length of measurement length of the target battery needed to make the RUL prediction. The error rate implies the percentage error ratio between the prediction and the ground-truth label averaged over five random seeds. The cycle error implies the absolute cycle error between the prediction and the ground-truth label averaged over five random seeds.

Models	Error rate (%)	Cycle error	Required cycles
Full model [23]	14.6 (3.2)	99 (14)	100
Variance [23]	15.7 (3.4)	112 (20)	100
SVR	33.1 (4.1)	245 (35)	100
Dilated CNN	**10.6 (1.4)**	**76 (11)**	**4**

paper, we will refer to this setting as "Initial". The dilated CNN showed superior performance, achieving an 10.6% error rate with much shorter cycle data of the target battery. In contrast, a single feature (cycle-to-cycle evolution of the discharge voltage curve) based baseline model achieved a 15.7% error rate, and the model with eight additional features achieved a 14.6% error rate.

Table 2 summarizes the performance of various deep learning models, and Fig. 6 shows the training loss and test error over epochs. Our framework was designed to predict the RUL regardless of the location of the data we received from the entire battery life. Thus, we also evaluated the ability to learn to predict the RUL given 2,500-second data from any location in the life cycle. We will refer to this setting as "Full" in this paper. We designed the shallow MLP model to have a similar number of parameters as the dilated CNN model and the MLP model, to ensure it had many more parameters. Nevertheless, both the shallow MLP and deep MLP models showed a higher error rate than the proposed architecture. Since the CNN and CNN-LSTM architectures were more specialized for the time-series analysis, they showed better performance. Still, the dilated CNN achieved the best error rate. In the "Full" setting, performance differences were more pronounced. Since Li-ion batteries have nonlinear and complex degradation behavior, the RUL prediction algorithm with the early-stage data and the RUL prediction algorithm with the last-stage data can be very different. The neural networks in the "Full" setting should be able to detect which stage the data came from and also be able to predict the RUL with different algorithms in relation to the stages. Since the "Full" setting was a harder task, the shallow MLP and the MLP model failed to learn to solve the task. However, the dilated CNN achieved even lower cycle errors than that in the "Initial" setting. Fig. 7 shows the prediction of the proposed dilated CNN model.

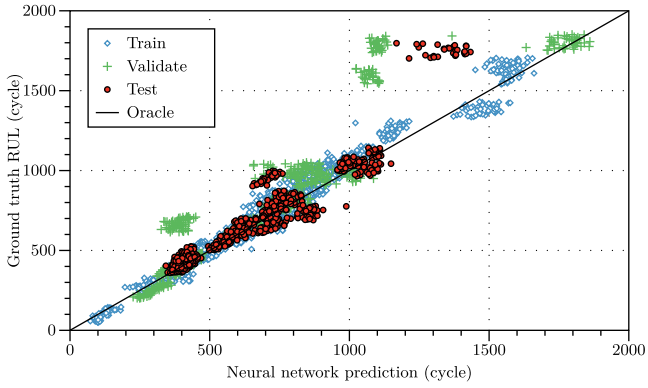

Fig. 7. Illustration of the predictions of the proposed dilated CNN for the training, validation, and test datasets. Each data point is from the sliced 2500-second time-series data. The oracle line represents the perfect prediction. The proposed neural networks for the training, validation, and test achieved 5%, 12%, 7% error rates, respectively.

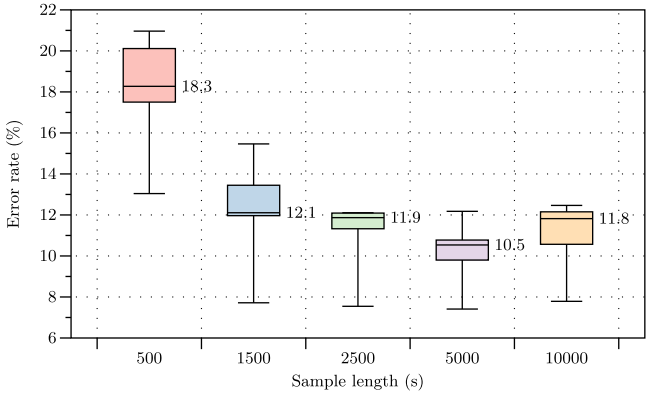

Fig. 8. Error rates over different sample lengths. The performance was evaluated in the "Initial" setting.

Finally, we evaluated the prediction performance of the proposed dilated CNN model concerning the input sample length. Fig. 8 shows the average prediction performance in the "Initial" setting with regard to the sample length. We tested 500-second (1 cycle), 1500-second (2~3 cycle), 2500-second (3~4 cycle), 5000-second (6~7 cycle), and

Table 2
Performance evaluation of the various deep neural networks in both "Initial" and "Full" setting. The CNN architecture follows the design of the dilated CNN architecture except that it uses a conventional convolution. More detailed comparisons between various CNN and dilated CNN designs are provided in Appendices A and B.

Deep learning models	Error rate (Initial, %)	Cycle error (Initial)	Error rate (Full, %)	Cycle error (Full)	Number of param.
Shallow MLP	21.9 (5.9)	159 (42)	73.3 (3.0)	150 (25)	2,643,751
MLP [34]	20.9 (3.2)	175 (46)	74.8 (5.0)	174 (19)	12,243,491
CNN [35]	14.0 (1.8)	102 (12)	35.1 (3.4)	82 (13)	2,393,468
CNN + LSTM [22]	14.1 (4.7)	106 (34)	22.3 (3.2)	72 (15.6)	5,369,345
Dilated CNN	**10.6 (1.4)**	**76 (11)**	**19.7 (1.8)**	**65 (14)**	**2,393,468**

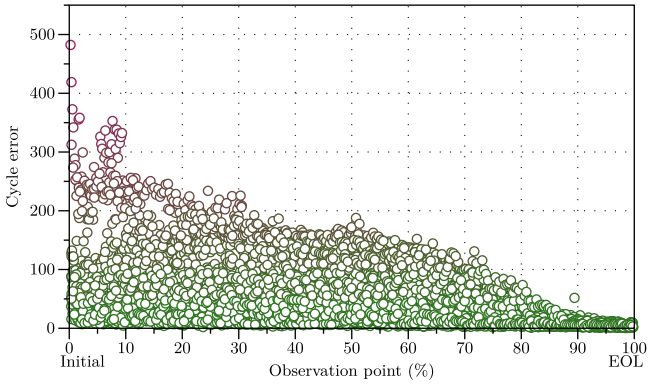

Fig. 9. Cycle errors in relation to observation points in the "Full" setting. Each data point corresponds to a 2500-second sample. The observation point implies where the sample is sliced between the initial cycle and EOL.

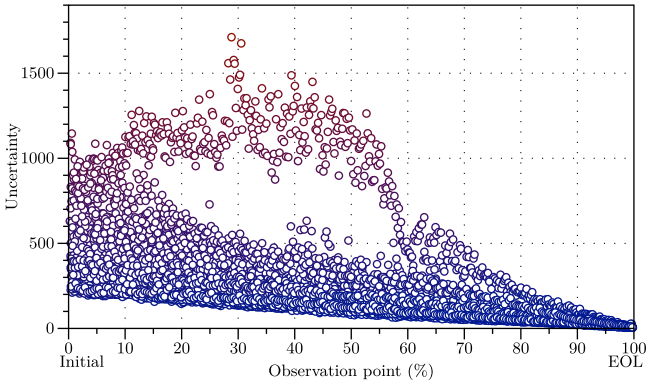

Fig. 10. Uncertainty estimates over observation points. Each data point corresponds to a 2500-second sample. We trained neural networks with Algorithm 1 and calculated the uncertainty estimate using Eq. (10).

10000-second (12~13 cycle) data as the input. The average error rate decreased as the sample length increased until the sample length reached 2,500 s, but interestingly, after that, increasing the length of the sample did not improve performance. In general, catching the cycle-to-cycle difference was the key to the accurate RUL prediction. In the feature-based method, the cycle-to-cycle difference was abstracted into a single feature variable so that it required longer data to capture the difference. However, the deep learning model is capable of analyzing the raw time-series data so that it can capture the minor differences in consecutive battery cycles. The result here implies that four data cycles was enough to capture the cycle-to-cycle evolution for the proposed neural networks.

4.3. Reliable use of deep neural network predictions using uncertainty estimates

In the previous subsection, we demonstrated that the proposed dilated CNN model exhibited superior performance compared to the human-crafted feature-based models and various deep neural network models. However, as mentioned above, neural networks can make arbitrary decision when the test data has inductive bias, which means that the test data distribution is far from the training data distribution. This means there is always a question of whether to trust the neural network output. Since conventional neural networks cannot tell the confidence of the prediction, sometimes the human-crafted feature-based model can be more trustworthy in critical situations. In this section, we verified that a theoretically proposed uncertainty metric can provide appropriate confidence for both the "Full" and "Initial" settings. Five structurally identical but differently initialized neural networks were trained under Algorithm 1 to calculate the uncertainty metric.

We first took a look into the predictions made in the "Full" setting by comparing the cycle errors and the uncertainty estimates of the predictions. Fig. 9 illustrates the error rate over the measurement point in the "Full" setting. Since there is negligible degradation in the

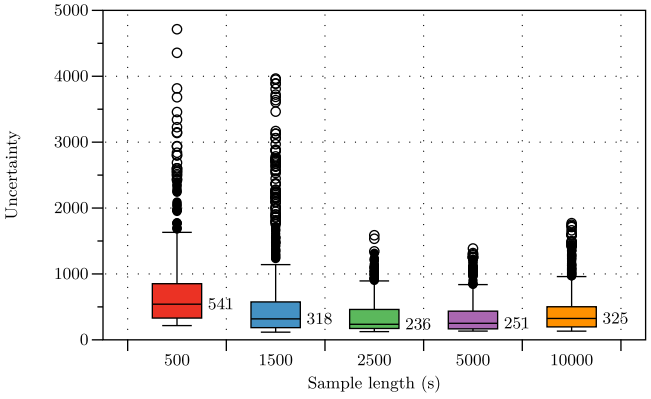

Fig. 11. Uncertainty estimates over different sample lengths.

early stage of the cycle life, RUL prediction is challenging in the early stages. As the measurement point moves closer to the battery's EOL, the capacity degradation of the cell accelerates, and this makes the prediction easier. Fig. 9 shows the results that matched our intuition. The prediction made in the early stages exhibited a higher error rate than the prediction made near EOL. While the predictions made in the initial stages had cycle errors up to 200, the cycle errors in the final stages were less than 50.

An ideal uncertainty metric should show the extent to which predictions are out of order. With a reasonable estimate of the uncertainty, we can afford much more reliable use of the RUL predictions of the neural networks by not relying on the neural network outputs when the accompanied uncertainty estimate is high. Fig. 10 shows the uncertainty score estimated over the measurement point. We observed that the predictions in the early stages yielded higher uncertainty, and the predictions near the EOL yielded lower uncertainty, which is consistent with the cycle errors in Fig. 9. We calculated the relationship

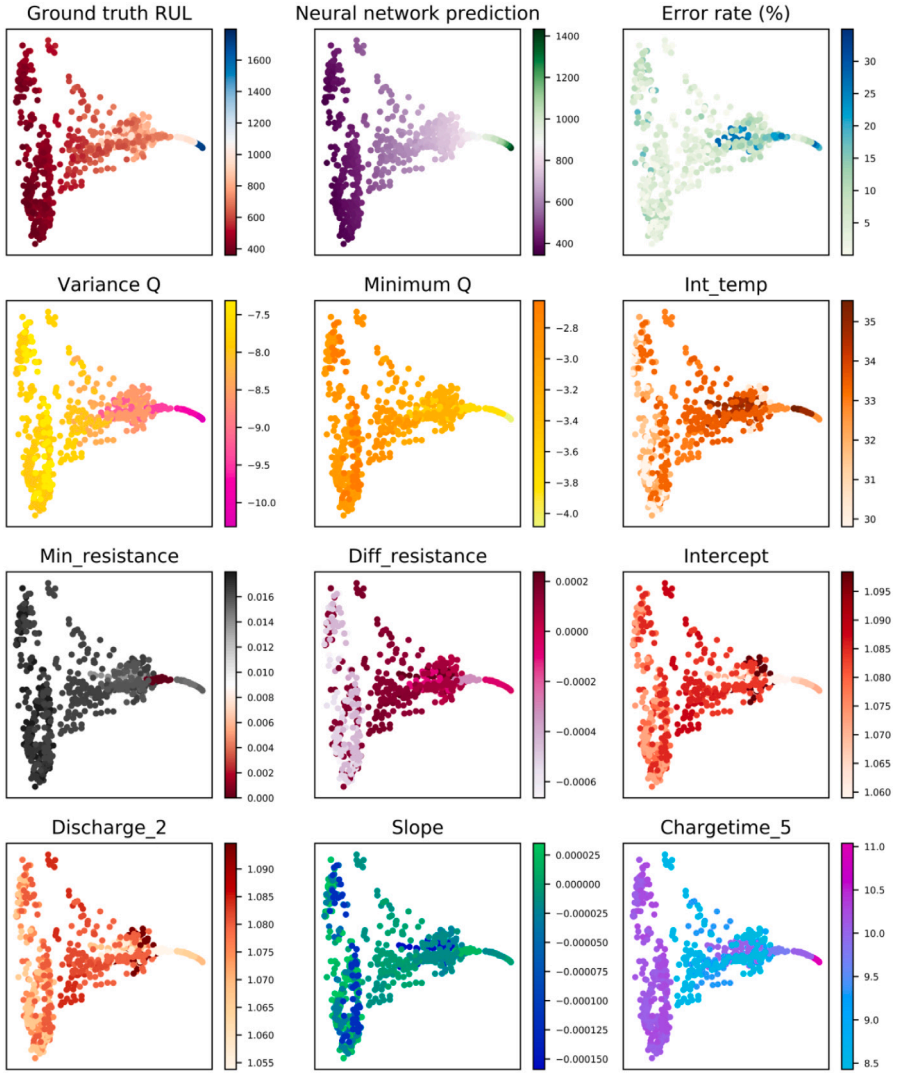

Fig. 12. Deep visualization of hidden layer representation using t-SNE. The mapped features are colored with the heatmap of three predictions performance and nine different features. The nine features were presented in descending order of correlation with model predictions. The Pearson Correlation Constants of the nine features were −0.94, −0.92, 0.47, −0.36, 0.31, −0.32, 0.22, 0.04, 0.001, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

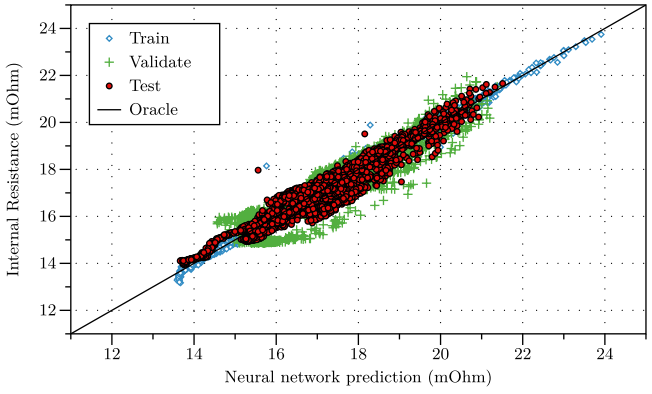

Fig. 13. Estimations of the internal resistance of the training, validation, and test dataset. The oracle line corresponds to the perfect estimation.

between the cycle error and the estimated uncertainty using the Pearson Correlation Coefficient [36]. The correlation factor was 0.62, which means strong positive correlation between the cycle error and the estimated uncertainty. This implies that we can estimate the average cycle error before we get the ground-truth label using a proper linear transformation. Fig. 11 shows the uncertainty over the input sample length in the "Initial" setting, which also shows a result consistent with the cycle error analysis in Fig. 8. Here, we conclude that the suggested uncertainty metric provides extra information about the neural network outputs, which establishes trustworthiness with the users.

5. Discussion

In the previous section, we validated the performance and reliability of the end-to-end deep learning-based RUL prediction method using various experiments. Still, the decision-making algorithm of the neural networks is unexplainable to humans. In this section, we make a more in-depth analysis of the features which the neural networks more strongly focus on. The analysis can be divided into two main questions: (1) how do neural networks make predictions; (2) what are the proposed neural networks capable of?

5.1. Visualization of deep neural network features

The data fed to the neural networks are radically transformed, are added and multiplied together over successive layers until it reaches the final layer. The output of each layer is a high dimensional tensor

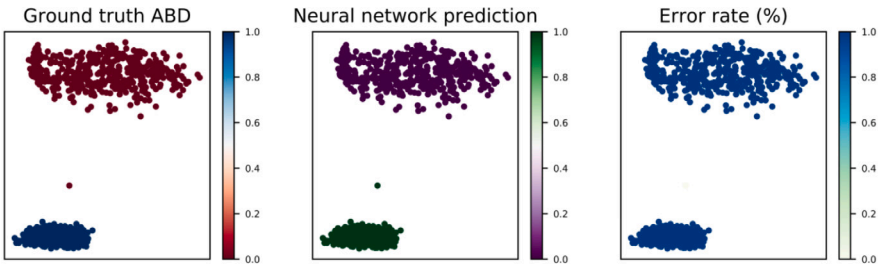

Fig. 14. Deep visualization of the feature representation of the proposed neural networks trained for the abnormal battery detection task using t-SNE.

that contains critical information on the RUL prediction task. In other words, these tensors are features that are automatically extracted from the raw charge–discharge cycle data. Still, the extracted features are not interpretable to humans.

To understand the implicit meaning of the features, we mapped the high dimensional features into a 2-dimensional space using t-stochastic neighbor embedding (t-SNE) [37]. t-SNE creates a low dimensional mapping while maintaining the local relationships between the data points in the high dimensional space. In this analysis, the dilated CNN model was trained in the "Initial" setting, and the features were drawn from the second to last layer of the neural networks. Fig. 12 illustrates the mapped features with the heatmap of the human-defined critical features overlaid on them. We calculated nine features from the first 100 cycle data of each battery and colored the data points of the t-SNE plot in relation to the value of each feature. Each data point in Fig. 12 corresponds to one data sample in the test dataset. The data augmentation process generates a few data samples from one battery. Since the feature is calculated per battery, we colored the data points from the same battery the same color. The heatmap on the first row of the figure shows the neural network predictions and the ground-truth labels of the data points. We can see that the data points with the high RUL tend to be on the right side of the figure, and the data points with the low RUL tend to be on the left side. The predictions of the neural networks show a consistent tendency, which results in low error rates and low uncertainty estimates over all the distributions. The last three rows in the figure represent the heatmap of the human-defined critical features used in the baseline model of Table 2 [23]. Among nine features, the "Variance Q" is a human-defined primary feature, the quantized metric of the cycle-to-cycle evolution of the discharge voltage curve. We measured the Pearson Correlation Constant and compared the model prediction and the nine human-defined features. The "Variance Q", "Minimum Q", and "Int_temp" showed the high correlations, of −0.94, −0.92, 0.47. In this way, we verified that the proposed neural networks also capture the features that humans defined as critical and by complicatedly combining them, the proposed neural networks achieved beyond-human performance.

5.2. Neural networks as multi-task learner

The proposed end-to-end framework showed superior performance, reliability, and interpretability in the RUL prediction of the Li-ion batteries. Even though we did not utilize any prior knowledge about the RUL prediction of Li-ion batteries, the trained neural networks automatically captured the human-defined primary features and their correlations. In this section, we test the learning capability of the proposed neural networks for two different Li-ion battery diagnosis tasks: (1) internal resistance estimation, and (2) abnormal battery detection [23,38].

Internal resistance estimation. The internal resistance of the Li-ion battery, which is a critical indicator of battery degradation, grows as the battery is used. Another canonical definition of the EOL of Li-ion batteries is the time it takes for the internal resistance to double from its initial value. Measuring the internal resistance is not a trivial task because of the nonlinear characteristics of Li-ion batteries and non-instantaneous voltage drop under a load. Thus, we need a specific load pattern to measure the internal resistance of the Li-ion cells. Here, the internal resistance was measured while charging at 80% SoC by averaging ten pulses of $\pm 3.6C$ with a pulse width of 30 or 33 ms. This subtask estimates the current internal resistance of the target battery based on the unconstrained, run-time, past few-cycle raw data.

Abnormal battery detection. Some Li-ion batteries lose their capacity much faster than the normal batteries. Practically, Practically, its helpful just knowing whether the battery is too bad or not. The subtask here is to detect the batteries that have significantly shorter EOL using the early-cycle raw data. Since the battery dataset used in this paper has an average cycle life of 806, we set the threshold of "bad" as 550.

Since the internal resistance estimation is a regression problem, like the RUL prediction, we used the same neural network architecture and loss function proposed for the RUL prediction. On the other hand, we modified the final layer of the dilated CNN architecture to output the softmax probability for binary classification. The binary cross-entropy (BCE) loss used for abnormal battery detection can be formulated as:

$$BCE = -\frac{1}{N}\sum_{n=1}^{N} y_n \log(F_\theta(x_n)) + (1-y_n)\log(1-F_\theta(x_n)), \tag{11}$$

where y_n represents the Boolean value of abnormality.

Fig. 13 compares the estimations of internal resistance and the ground-truth labels. The average error rate over the five randomly generated training, validation, and test dataset was 3%. Next, Fig. 14 illustrates the t-SNE plot of the extracted features for the abnormal battery detection. The features were drawn from the output of the second to last layer. We observed two well-divided clusters, the cluster of the abnormal battery and normal battery, respectively. With well-divided clusters, the proposed neural network achieved 95% accuracy in abnormal battery detection. Finally, we concluded that the end-to-end deep learning-based framework can be generalized to various battery related tasks.

6. Conclusions

In this work, we have presented an end-to-end deep learning framework for predicting the remaining useful life of lithium-ion batteries. The framework addresses the challenge of predicting battery remaining useful life from short measurements. The proposed framework significantly boosted remaining useful life prediction by making predictions with less than four data cycles of the target battery, while the previous models required 100 cycles of data. Among various deep learning models, the proposed dilated CNN demonstrated a 10.6% error rate, while the previous model achieved a 14.6% error rate. We have also described a training algorithm and metric to quantify the uncertainty in the neural network predictions. We verified that the proposed uncertainty estimate exhibited high value for a particular data, which had a considerable prediction error or which had features not found in the training dataset. Our in-depth analysis of the neural networks

Table A.3
Performance of various CNN architectures according to the choice of design parameters.

Modifications made from the CNN architecture in Table 2		Error rate (Initial, %)	Cycle error (Initial)
Number of CNN layers	5	14.0 (1.4)	102 (11)
	7	13.7 (3.1)	94 (18)
	10	12.4 (2.0)	84 (14)
	15	11.8 (1.0)	86 (9)
CNN kernel size	25	12.8 (1.7)	89 (8)
	49	11.1 (1.4)	81 (14)
	81	11.6 (1.7)	83 (10)
	121	11.6 (1.7)	85 (17)
Sequence length reduction	1/4	13.9 (2.0)	96 (12)
	1/8	11.6 (1.8)	84 (11)
	1/16	11.4 (1.7)	86 (15)
	1/32	15.7 (5.6)	106 (25)
Dilated CNN		10.6 (1.4)	76 (11)

Table B.4
Performance of dilated CNN architectures according to the choice of design parameters.

Hyper parameter effect of the proposed model in Table 2		Error rate (Initial, %)	Number of parameters
Number of Dilated layers	7	10.9 (2.1)	2,819,388
	10	10.2 (2.5)	4,738,236
	15	12.1 (3.8)	16,671,932
Dilated kernel size	25	10.2 (2.8)	4,481,980
	49	11.8 (3.5)	8,486,556
	81	11.3 (3.8)	14,232,252
	121	12.6 (4.4)	21,196,732
Size of dilation	2	10.6 (1.4)	2,392,636
	3	9.7 (1.6)	2,392,636
	4	12.1 (3.7)	2,392,636
	5	10.4 (2.1)	2,392,636

for battery-related tasks revealed the possible use of end-to-end deep learning approaches to analyze the lithium-ion battery data.

Current lithium-ion battery industries are willing to endure hundreds of full charge and discharge cycles just for the diagnostic purposes. Our framework significantly improved the prognostics of lithium-ion batteries and made the prognostic procedure much more affordable. Also, we believe that our framework can respond to dynamic conditions, which will be helpful in industries where batteries are not always fully charged and discharged. In our approach, the neural networks have the power to capture features from unconstrained and unorganized data, but we leave this as our future research. These swift battery life forecasts can be used for meaningful life forecasts not only in academia but also in industry.

CRediT authorship contribution statement

Joonki Hong: Conceptualization, Methodology, Experiments, Writing. **Dongheon Lee:** Conceptualization, Methodology, Experiments, Writing. **Eui-Rim Jeong:** Supervision, Writing - review & editing. **Yung Yi:** Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Performance comparison of various CNN architectures

Table A.3 summarizes the performance of CNN architectures according to the choice of design parameters. We tested three canonical designs to increase the receptive fields in the CNN architecture: (1) deeper layers of CNN, (2) wider kernel, and (3) reduced sequence length of intermediate features. The deeper layers and the wider kernels yielded extended receptive fields, by increasing the number of parameters. In contrast, reducing the sequence length of intermediate features decreased the number of parameters. We observed a substantial increase in performance by controlling the depth of the layer and the size of the kernel, but the dilated CNN showed superior performance compared to the CNN with the optimal choice of parameters. Likewise, reducing the sequence length of intermediate features resulted in a performance increase, but the dilated CNN showed superior performance.

Appendix B. Performance comparison of various dilated CNN architectures

Table B.4 summarizes the performance of dilated CNN architectures according to the choice of design parameters. We explored through major hyperparameters in the dilated CNN architecture: (1) the number of dilated CNN layers, (2) the convolution kernel size, and (3) dilation size. Increasing the number of dilated CNN layers, convolution kernel size, and dilation size did not result in significant improvement in the performance. In contrast, they produced an exponential increase in trainable parameters. We followed the principle of Occam's Razor to

Table D.5
The nine human-defined critical features for RUL prediction calculated from the first 100 cycle data.

Feature	Definition
Variance Q	Variance of cycle-to-cycle evolution of discharge voltage curve
Minimum Q	Minimum value of cycle-to-cycle evolution of discharge voltage curve
Int_temp	Integral of temperature over time, cycles 2 to 100
Min_resistance	Minimum internal resistance cycles 2 to 100
Diff_resistance	Internal resistance difference between cycles 2 to 100
Intercept	Intercept of the linear fit to capacity fade curve, cycles 2 to 100
Discharge_2	Discharge capacity at cycle 2
Slope	Slope of the linear fit to the capacity fade curve, cycles 2 to 100
Chargetime_5	Average charge time, first 5 cycles

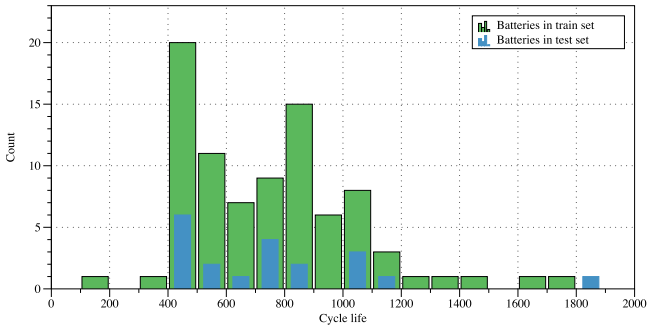

Fig. C.15. The cycle life distribution of the training dataset and test dataset in Fig. 10.

choose the hyperparameters. All of our experiments in the text were performed with 5 dilated CNN layers, convolution kernel size 13, and dilation size 2.

Appendix C. In-depth analysis of uncertainty estimates

The proposed uncertainty estimate includes three sources of uncertainty: model uncertainty, distributional uncertainty, and data uncertainty [31]. The model uncertainty quantifies how the model is trained given the training data, the distributional uncertainty quantifies whether the test data belongs to the training data distribution, and the data uncertainty quantifies whether data is hard to predict. In Fig. 10, some data points showed extraordinarily high uncertainty. It seems that the uncertainty estimates on these points were overestimated and did not correspond to prediction errors. However, by taking a deeper look into the distribution of the training and test datasets, we found out that those uncertainties were drawn from the distributional uncertainty. Fig. C.15 illustrates the cycle life of batteries in the training dataset and test dataset. The batteries in the training dataset have cycle lives of less than 1600 cycles. However, in the test dataset, there is a battery with an extraordinary high cycle life of 1800 cycles. We found that the data points which had extraordinary high uncertainty were generated from a battery with an extraordinary cycle life. This reveals that the proposed uncertainty estimate can be used to detect out-of-distribution data, in which our model is incapable of making predictions.

Appendix D. Definition of nine human-defined features for RUL prediction

The authors of [23] defined nine critical features for RUL predictions. "Variance Q" and "Minimum Q" are calculated from the discharge voltage curve while "Intercept", "Discharge_2", and "Slope" are extracted from the discharge capacity fade curve. "Int_temp" is calculated from the cell temperature time-series data, "Min resistance" and "Diff resistance" are calculated from the impedance history data, and "Chargetime_5" is measured during the experiments (see Table D.5).

Table E.6
Performance over combinations among different time series data under discharge condition.

Combinations			Error rate	Cycle error
V	I	T	(Initial, %)	(Initial, cycle)
–	✓	✓	17.1 (2.4)	126 (13)
✓	–	✓	15.8 (1.8)	118 (24)
✓	✓	–	18.9 (2.3)	146 (25)
✓	✓	✓	16.0 (2.7)	121 (15)

Appendix E. Ablation study of various combinations of time-series data under discharge cycle

Table E.6 reports the RUL prediction performance of the dilated CNN using only discharge patterns of Li-ion batteries over different combinations of inputs. To fully exclude the contribution of current patterns, we run the experiment only with discharge cycle data (The batteries are charged with various charge patterns but discharge with constant current). The result shows us that the cell temperature brought 2.9% performance increase compared to the prediction only with the terminal voltage and current. The terminal voltage brought 1.1% performance increase compared to the prediction only with the current and cell temperature.

References

[1] Schmuch R, Wagner R, Hörpel G, Placke T, Winter M. Performance and cost of materials for lithium-based rechargeable automotive batteries. Nat Energy 2018;3(4):267–78.
[2] Christensen J, Newman J. Cyclable lithium and capacity loss in Li-ion cells. J Electrochem Soc 2005;152(4):A818–29.
[3] Zhang Q, White RE. Capacity fade analysis of a lithium ion cell. J Power Sources 2008;179(2):793–8.
[4] Wright R, Christophersen J, Motloch C, Belt J, Ho C, Battaglia V, et al. Power fade and capacity fade resulting from cycle-life testing of advanced technology development program lithium-ion batteries. J Power Sources 2003;119:865–9.
[5] Ramadesigan V, Chen K, Burns NA, Boovaragavan V, Braatz RD, Subramanian VR. Parameter estimation and capacity fade analysis of lithium-ion batteries using reformulated models. J Electrochem Soc 2011;158(9):A1048–54.
[6] Cordoba-Arenas A, Onori S, Guezennec Y, Rizzoni G. Capacity and power fade cycle-life model for plug-in hybrid electric vehicle lithium-ion battery cells containing blended spinel and layered-oxide positive electrodes. J Power Sources 2015;278:473–83.
[7] Christensen J, Newman J. A mathematical model for the lithium-ion negative electrode solid electrolyte interphase. J Electrochem Soc 2004;151(11):A1977–88.
[8] Pinson MB, Bazant MZ. Theory of SEI formation in rechargeable batteries: capacity fade, accelerated aging and lifetime prediction. J Electrochem Soc 2013;160(2):A243–50.
[9] Waag W, Fleischer C, Sauer DU. Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. J Power Sources 2014;258:321–39.
[10] Waldmann T, Bisle G, Hogg B-I, Stumpp S, Danzer MA, Kasper M, et al. Influence of cell design on temperatures and temperature gradients in lithium-ion cells: an in operando study. J Electrochem Soc 2015;162(6):A921–7.
[11] Wu L, Fu X, Guan Y. Review of the remaining useful life prognostics of vehicle lithium-ion batteries using data-driven methodologies. Appl Sci 2016;6(6):166.

[12] Li Y, Liu K, Foley AM, Zülke A, Berecibar M, Nanini-Maury E, et al. Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review. Renew Sustain Energy Rev 2019;113:109254.
[13] Cadini F, Sbarufatti C, Cancelliere F, Giglio M. State-of-life prognosis and diagnosis of lithium-ion batteries by data-driven particle filters. Appl Energy 2019;235:661–72.
[14] Zhang Y, Xiong R, He H, Pecht MG. Lithium-ion battery remaining useful life prediction with Box–Cox transformation and Monte Carlo simulation. IEEE Trans Ind Electron 2018;66(2):1585–97.
[15] Liu D, Zhou J, Liao H, Peng Y, Peng X. A health indicator extraction and optimization framework for lithium-ion battery degradation modeling and prognostics. IEEE Trans Syst Man Cybern A 2015;45(6):915–28.
[16] Zhou Y, Huang M, Chen Y, Tao Y. A novel health indicator for on-line lithium-ion batteries remaining useful life prediction. J Power Sources 2016;321:1–10.
[17] Chang Y, Fang H, Zhang Y. A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery. Appl Energy 2017;206:1564–78.
[18] Richardson RR, Osborne MA, Howey DA. Gaussian process regression for forecasting battery state of health. J Power Sources 2017;357:209–19.
[19] Liu J, Saxena A, Goebel K, Saha B, Wang W. An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries. Tech. rep., National Aeronautics and Space Administration Moffett Field Ca Ames Research; 2010.
[20] Pang X, Huang R, Wen J, Shi Y, Jia J, Zeng J. A lithium-ion battery RUL prediction method considering the capacity regeneration phenomenon. Energies 2019;12(12):2247.
[21] Zhang Y, Xiong R, He H, Pecht MG. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. IEEE Trans Veh Technol 2018;67(7):5695–705.
[22] Ma G, Zhang Y, Cheng C, Zhou B, Hu P, Yuan Y. Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network. Appl Energy 2019;253:113626.
[23] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, et al. Data-driven prediction of battery cycle life before capacity degradation. Nat Energy 2019;4(5):383.
[24] Patil MA, Tagade P, Hariharan KS, Kolake SM, Song T, Yeo T, et al. A novel multistage support vector machine based approach for Li ion battery remaining useful life estimation. Appl Energy 2015;159:285–97.
[25] Ren L, Zhao L, Hong S, Zhao S, Wang H, Zhang L. Remaining useful life prediction for lithium-ion battery: A deep learning approach. IEEE Access 2018;6:50587–98.
[26] Richardson RR, Osborne MA, Howey DA. Battery health prediction under generalized conditions using a Gaussian process transition model. J Energy Storage 2019;23:320–8.
[27] Attia PM, Grover A, Jin N, Severson KA, Markov TM, Liao Y-H, et al. Closed-loop optimization of fast-charging protocols for batteries with machine learning. Nature 2020;578(7795):397–402.
[28] Cho K, van Merrienboer B, Gülçehre Ç, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014, arXiv:1406.1078, CoRR abs/1406.1078, URL http://arxiv.org/abs/1406.1078.
[29] Oord Avd, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, et al. Wavenet: A generative model for raw audio. 2016, arXiv preprint arXiv:1609.03499.
[30] Ruder S. An overview of gradient descent optimization algorithms. 2016, arXiv preprint arXiv:1609.04747.
[31] Malinin A, Gales M. Predictive uncertainty estimation via prior networks. In: Advances in neural information processing systems. 2018, p. 7047–58.
[32] Gal Y, Ghahramani Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. 2015, arXiv preprint arXiv:1506.02142.
[33] Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in neural information processing systems. 2017, p. 6402–13.
[34] Chaoui H, Ibe-Ekeocha CC, Gualous H. Aging prediction and state of charge estimation of a LiFePO4 battery using input time-delayed neural networks. Electr Power Syst Res 2017;146:189–97.
[35] Shen S, Sadoughi M, Chen X, Hong M, Hu C. A deep learning method for online capacity estimation of lithium-ion batteries. J Energy Storage 2019;25:100817.
[36] Benesty J, Chen J, Huang Y, Cohen I. Pearson correlation coefficient. In: Noise reduction in speech processing. Springer; 2009, p. 1–4.
[37] Maaten Lvd, Hinton G. Visualizing data using t-SNE. J Mach Learn Res 2008;9(Nov):2579–605.
[38] Laig-Hoerstebrock H. Method for prediction of the internal resistance of an energy storage battery, and a monitoring device for energy storage batteries, US Patent 7, 098, 665. 2006.