

# Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach

Gae-won You\*, Sangdo Park, Dukjin Oh

Samsung Advanced Institute of Technology, Samsung Electronics, Electronic Materials Research Complex, 130 Samsung-ro, Gyeonggi-do, 443-803, Republic of Korea

## HIGHLIGHTS

- We developed a data-driven framework to estimate SOH.
- We proposed an algorithm to recognize time-varying distributions.
- We collected sensed data by cycling battery cells non-statically.
- We estimated SOH effectively with less than 2.18% average error.

## ARTICLE INFO

### Article history:

Received 25 February 2016

Received in revised form 2 May 2016

Accepted 5 May 2016

**Keywords:**  
Batteries  
State-of-health  
Electric vehicles  
Clustering  
Neural networks

## ABSTRACT

As electric vehicles (EVs) have been popularized, research on battery management systems (BMSs) in their core technology has drawn considerable attention. Among the various functions of a BMS, estimating the state-of-health (SOH) of the battery is crucial; this estimation is used to determine the replacement time of the battery or to assess driving mileage. While most studies utilize capacity fading or resistance growth as SOH metrics, they all define SOH using fairly constrained assumptions, e.g., full cycling with constant current. Unfortunately, those assumptions cannot be applied to EV batteries that are, for the most part, cycled partially and dynamically. In clear contrast, this paper studies how SOH can be estimated in more practical environments where the batteries must support real-world driving patterns. In particular, this paper proposes a data-driven approach to trace SOH on the fly by using sensible BMS data such as current, voltage, and temperature while leveraging their historical distributions. We validated that our approach provides highly accurate results under actual EV driving conditions, with an average error less than 2.18%.

© 2016 Elsevier Ltd. All rights reserved.

## 1. Introduction

As electric vehicles (EVs) have been popularized, research on battery management systems (BMSs) in their core technology has drawn considerable attention [1]. Among the various functions of a BMS, estimating the state-of-health (SOH) of the battery (as a metric of its aging) is the most crucial factor in determining the replacement time of the battery or predicting driving mileage.

Most studies have measured SOH based on (1) the equivalent-circuit (EC) model or (2) capacity fading because EC model parameters change and capacity decreases as the batteries age and are

utilized. However, they all define SOH metrics under fairly strict conditions; thus their SOH measurements cannot be applied to EV batteries, which require real-time computation and are usually charged and discharged partially and dynamically as follows:

- *Real-time computation:* In EV environments, SOH must be traced on the fly, because SOH can be leveraged as a key argument for determining state-of-charge (SOC) (which means a fuel gauge for the battery pack) and assessing mileage precisely.
- *Partial and dynamic cycling:* Owing to the lengthy charging times of EV batteries, they would be charged whenever they are not fully charged. In addition, the batteries will discharge non-statically owing to ever-changing velocities caused by various driving patterns, traffic conditions, road conditions, and other factors.

\* Corresponding author at: Samsung Advanced Institute of Technology, Samsung Electronics, Electronic Materials Research Complex, 130 Samsung-ro, Gyeonggi-do, 443-803, Republic of Korea.

E-mail address: [gaewon.you@samsung.com](mailto:gaewon.you@samsung.com) (G.-w. You).  
URL: <http://www.samsung.com/sec/> (G.-w. You).

First, the EC model's internal resistance or capacitance can be measured. By applying electrochemical impedance spectroscopy (EIS) measurements, complex impedance spectra are collected by sweeping high to low current frequencies, and then obtaining them from the Nyquist curve. However, for exact EIS measurement, we should allow sufficient time for the batteries to stabilize under very restricted conditions. To overcome this problem, several statistical approaches [2,3] have been proposed; however, they are still restrictive or wait until the battery converges to specific conditions, e.g., temperature 25 °C and SOC 80%, which makes it difficult to support real-time estimations.

Second, to measure the capacity fading, the ratio between the degraded capacity and the initial capacity is evaluated [4]. Researchers iterate fully charging and discharging a battery (called cycling) within a fixed range, e.g., from 2.7 V to 4.2 V, and vice versa for the battery cells. They then integrate the total current for each iteration under some constraints such that its temperature and C-rate<sup>1</sup> remain constant. However, when analyzing EV utilization patterns, such static cycling is too strong an assumption. For instance, Fig. 1 shows two datasets (including current, voltage, temperature, and capacity) sensed from 18650 battery cells in two different cycling modes: (Case 1) full and static, and (Case 2) partial and dynamic. While the present capacity can be measured for each cycle by such clear cycle definition (red<sup>2</sup> numbers annotated top) in (Case 1), it is ambiguous in (Case 2). In addition, the degradation curves vary depending on their utilization patterns. Observe that Fig. 1 (Case 2) depicts more erratic patterns compared to Fig. 1 (Case 1) in all settings, which indicates that it is more difficult to estimate SOH in such EV environments. However, to the best of our knowledge, no studies have analyzed battery aging datasets that contain data representing actual driving patterns.

In clear contrast, this paper suggests a data-driven approach to trace SOH on the fly; in this approach, SOH is estimated by exploiting sensed data such as current ( $I$ ), voltage ( $V$ ), and temperature ( $T$ ). According to EV driving conditions, such  $I/V/T$  patterns are very diverse. For example, Fig. 2 shows sets of three-dimensional points ( $x$ -,  $y$ -,  $z$ -axis correspond to  $I/V/T$ , respectively) where each point represents  $I/V/T$  values at a certain time. Two different datasets are used, which represent EV driving patterns on two different types of roads: (a) ECE (urban driving) and (b) US06 (highway driving). To distinguish such types and recognize their ever-changing patterns, we build a framework based on a neural network (NN) [5].

Our key contributions are summarized as follows:

- We developed a data-driven framework to estimate SOH on the fly. In particular, we leveraged an NN to distinguish historical distributions of the signals sensible from a BMS.
- We built various sensed datasets including current, voltage, and temperature by cycling 18650 battery cells partially and dynamically with more than ten driving profiles (including UDDS, US06, and NYCC) to extensively validate our framework in a more realistic fashion.
- We validated that our framework was very effective in very dynamic environments, i.e., the average error less than 2.18% in all our settings.

The remainder of this paper is organized as follows: Section 2 investigates the existing approaches. Section 3 develops our

data-driven framework, and Section 4 reports our experimental settings and results. Section 5 concludes the paper.

## 2. Related work

Thus far, research on SOH estimation seems insufficient compared to SOC or other BMS functions; this is probably the result of difficulties caused by the time-consuming process of battery aging, in which batteries are subject to charging cycles until they are no longer usable. Because a battery is a complicated analogue system that consists of various materials that generate energy by their chemical reactions, it has numerous aging factors [4,6–8] and there is no clear consensus on its aging indices. Nevertheless, the metrics of SOH can be narrowed down into two mainstream methodologies: (1) using the equivalent-circuit (EC) model and (2) using capacity degradation.

### 2.1. Using EC model

Because a battery can be represented as a corresponding equivalent-circuit (EC) model, one of the SOH metrics is defined using its internal resistance or capacitance. In general, as the battery ages and its chemical reactions such as the growth of the solid electrolyte interface (SEI) disrupt the migration of ions, the internal resistance or capacitance irreversibly changes. To reflect this concept, Haifeng et al. [2] formally defined SOH as follows:

$$\text{SOH} = \frac{R_{\text{EOL}} - R_i}{R_{\text{EOL}} - R_0} \times 100(\%), \quad (1)$$

where  $R_0$  is the initial resistance when the battery is fresh,  $R_{\text{EOL}}$  is the resistance at end-of-life (EOL), and  $R_i$  is the resistance at a certain time  $i$ . Similar to this, Chen et al. [9] used the diffusion capacitance of the EC model.

While those studies leveraged statistical techniques [3] such as Kalman Filter [2] and Genetic Algorithm [9] to identify resistance to estimate the diffusion capacitance, they all are sensitive to internal and external conditions such as SOC and ambient temperature. Thus, for precise estimation, they must compensate those effects and wait until the battery converges to some restricted status, which is not acceptable in our dynamic EV scenarios.

### 2.2. Using capacity degradation

Capacity fading is another phenomenon caused by battery aging. Several approaches [10] defined this concept as an index as follows:

$$\text{SOH} = \frac{C_i}{C_0} \times 100(\%), \quad (2)$$

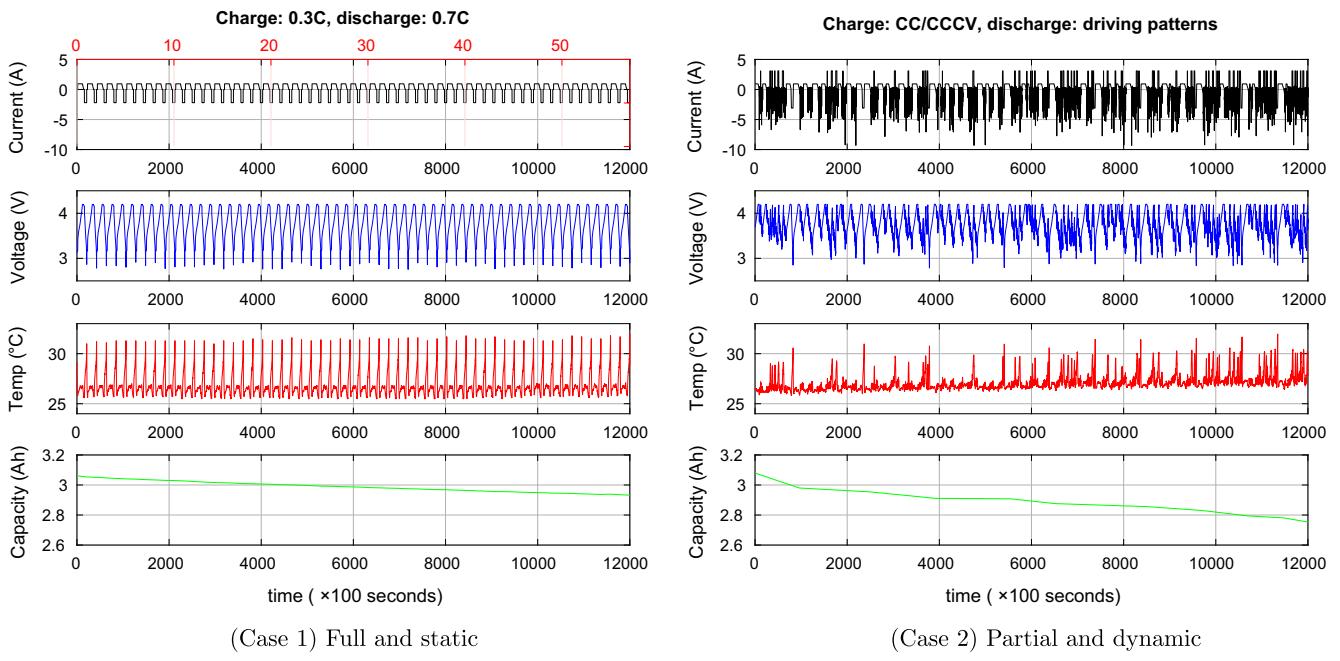
where  $C_0$  is the nominal capacity when the battery is fresh and  $C_i$  is the capacity at a certain time  $i$ .

In previous studies, it was generally assumed that the battery is fully cycled with a constant current within fixed ranges [10–12]; unfortunately, this assumption is also not realistic in our EV scenarios, which inherently require the battery to charge partially and discharge dynamically.

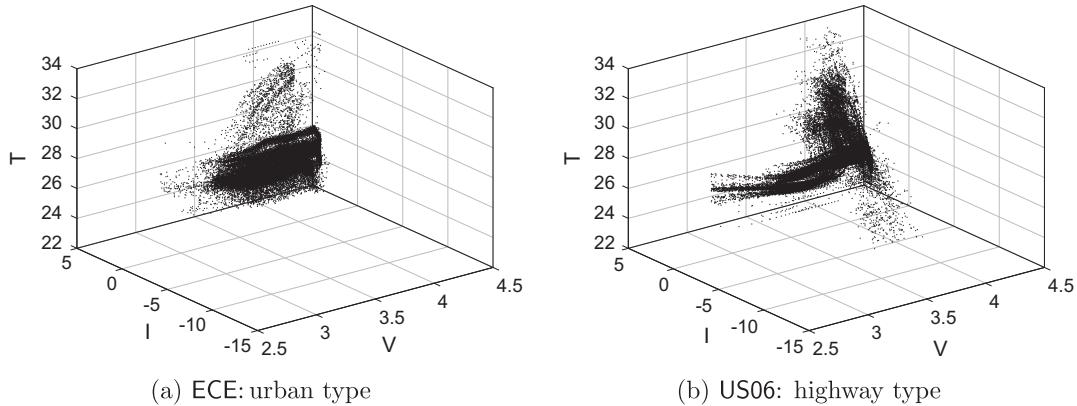
For the purpose of prognosis, several approaches using this SOH definition focused on predicting the remaining-useful-life (RUL), i.e., predicting how much longer the battery will be able to function, using statistical techniques such as Gaussian process regression [13], Kalman filter [14], particle filter [8,15,16], and machine learning techniques such as a naive Bayes model [17], support vector regression [18], and neural networks [19]. They also assumed the capacity measurement to be available by fully cycling batteries every time. Although these approaches are useful for evaluating

<sup>1</sup> C-rate is a unit to be used to scale charge and discharge current of a battery, e.g., to fully discharge a battery with a capacity of 3.2 Ah, it takes an hour to impose 1 C (3.2 A) current to the battery.

<sup>2</sup> For interpretation of color in Figs. 1, 3, 4, 7, 8, 11, 13 and 14, the reader is referred to the web version of this article.



**Fig. 1.** Charge and discharge modes (+ current: charge, – current: discharge).



**Fig. 2.** Historical  $I/V/T$  patterns on two different types of profiles.

retainability of the battery, they are inappropriate for diagnosing battery aging on the fly, because such measurements cannot determine the capacity directly when the battery is partially cycled. Meanwhile, besides the literatures on the battery studies, these prognostic technologies were exploited for anomaly detection, e.g., based on Mahalanobis distance [20,21] or particle filter [22].

In recent years, to diagnose EV battery states, several data-driven studies were proposed using support vector machine (SVM) [23–25], extended Kalman filter (EKF) [26], etc. While, the most similar to our approach, Nuhic et al. [23] exploited the historical distributions of the EV signals, their approach is more cumbersome to formulate inputs from the distributions. In clear contrast, our approach minimizes such effort with the unsupervised learning such as  $k$ -means clustering, and then improves the estimation accuracy. In Section 4, we will directly compare two approaches with our driving-pattern datasets. Klass et al. [24,25] designed an SVM-based model to estimate the voltage and then calculate SOH by virtually testing the model with hypothetical inputs for the capacity or resistance tests, e.g., HPPC [26]. However, as this work assumed certain SOC ranges, their solution cannot apply to our scenarios, which cover total SOC range (0–100%).

### **3. Framework for SOH estimation**

While batteries have a multitude of internal and external aging factors, including SOC, imposed load, ambient temperature, humidity, and their chemical reactions, it is not possible to comprehensively reflect all factors in our estimation model. However, we argue that our goal is not to discover all such factors, but to propose a framework that handles them effectively and is seamlessly extensible, even augmented with other sensible data. In this paper, we address three signals (current, voltage, and temperature) generally sensible from the BMS:

$$\mathcal{D} = \{(I_1, \dots, I_{\text{EOL}}), (V_1, \dots, V_{\text{EOL}}), (T_1, \dots, T_{\text{EOL}})\}, \quad (3)$$

where  $I_t$ ,  $V_t$ ,  $T_t$  are current, voltage, and temperature, respectively, at time  $t$ . In particular, we analyze its three-dimensional joint distribution  $D_{\text{joint}}$  where the  $x$ -,  $y$ -, and  $z$ -axes are current, voltage, and temperature, respectively.

$$\mathcal{D}_{\text{joint}} = \{(I_1, V_1, T_1), \dots, (I_t, V_t, T_t), \dots, (I_{\text{EOL}}, V_{\text{EOL}}, T_{\text{EOL}})\}, \quad (4)$$

where  $p_t = (I_t, V_t, T_t)$  jointly represents sensed data at time  $t$  as a point. For example, Fig. 2(a) and (b) shows two different sets of such

three-dimensional points, respectively representing urban and highway driving conditions.

We handle those signals in joint distribution because they are not all independent of each other. Observe that in Fig. 3, the three signals are considerably correlated; in this figure, the  $x$ -axis represents time and the  $y$ -axes represent current (top), voltage (middle), and temperature (bottom). When the battery is discharged (minus current), its corresponding voltage decreases and its temperature increases. On the other hand, when the battery is charged (plus current), its voltage increases and its temperature decreases to the ambient temperature, which is approximately 25 °C.

As the sensed points  $(I_t, V_t, T_t)$ 's are accumulated to the distribution  $\mathcal{D}_{\text{joint}}$  while the battery is being utilized, its density always increases and its pattern is very diverse according to its utilization history (thus it is called a *historical distribution*). For example, Fig. 4 shows the historical distributions of three different utilization patterns, where  $x$ -axis is time and  $y$ -axis is capacity ((c)–(e) ECE: low-speed driving and frequent stopping in city conditions, (b) HWY: high-speed driving and less stopping in highway conditions, (a) LA92DDS: mild-speed driving). As their distributions become denser according to their different utilizations, those patterns change toward more distinguishable directions. For instance, the capacity fading of LA92DDS (marked with red circle) is the steepest while that of HWY (marked with blue square) is the most gradual.

To recognize those distributional differences and estimate SOH, our overall framework consists of two parts (see Fig. 5: (1) an off-line learner that trains an SOH estimation model from some collected datasets and provides its model parameters to BMS and (2) an online SOH estimator of the BMS that receives signals from the battery modules and traces SOH on the fly; the feature extractor quantifies the historical patterns for the pattern recognizer to compute the SOH).

Therefore, Section 3.1 briefly introduces neural networks (NNs), algorithms widely used for recognizing patterns in various scientific areas; we utilize NN techniques to complete our data-driven framework. In particular, Section 3.2 proposes a method to formulate the historical distributions as inputs to the NN, which is our main contribution. We argue that our method for handling the historical distributions is orthogonal to the learning methods we can leverage.

### 3.1. Preliminary: learning neural network

The neural network is one of the most popular algorithms for recognizing patterns and has been addressed in a variety of applications [5,27,28]. An NN, whose structure consists of several neurons and their internal connections, is designed to mathematically mimic the genetic activity of the brain. Fig. 6(a) shows an oversimplified structure to depict its concept. Neurons, known as perceptrons, are a fundamental building block of NNs, and are represented as circles. An enlarged example is shown in Fig. 6(b). It takes some inputs  $p = (p_1, \dots, p_i, \dots)$ , their corresponding weights  $w = (w_1, \dots, w_i, \dots)$  and a bias  $b$ , and passes the sum of the inner-product and the bias  $q = wp^T + b$  through an activation function  $f_a$  (which may be a linear, sigmoid, or tanh function), and produces a scalar output  $a = f_a(q)$ . Several perceptrons comprise each layer, and their scalar outputs transfer to inputs of the perceptrons in the next layer to the right. They are forwardly (or optionally backwardly) fed from left to right until reaching the output layer (this is called a FeedForward network).

The network of such multilayer perceptrons can mainly be divided into three layers: (1) input layer, (2) output layer, and (3) hidden layer. First, the input layer takes the data that are already formulated from the raw data; this formulation makes it easy to recognize the patterns. (This formulation is our main inter-

est, and is generally referred to as “feature extraction.” We will focus on that in the next section.) Second, the output layer can produce two different results depending on target problems, classification and regression. While the former classifies the input data into several discrete categories, the latter is exploited to produce continuous values from the input data. To be specific, in this study, as we aim to estimate SOH, we utilize NNs for the regression. Last, the hidden layers manipulate the data transferred from the input layer to render the correct results. They can be constructed as multiple layers whose structures, e.g., the number of hidden layers, the number of neurons in each layer, and the type of activation function, should be optimized differently depending on the application.

To configure the network to produce accurate SOH results, we leverage some training data  $\mathcal{D}_{\text{train}}$  and thus optimize parameter  $\theta$  mainly containing the weights of interconnections between neurons, which are represented by  $w$ 's. More formally, we employ a given training dataset

$$\mathcal{D}_{\text{train}} = \{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_n, y_n)\}, \quad (5)$$

where  $x_i$  and  $y_i$  represent sensed data and labeled SOH values, respectively. We aim to find a  $\theta$  such that minimizes the estimation error  $J(\theta)$  and minimizes the difference between the actual SOH  $y_i$  and its corresponding estimation value  $f(x_i; \theta)$ :

$$J(\theta) = \frac{1}{2} \sum_{i=1}^n (f(x_i; \theta) - y_i)^2 \quad (6)$$

$$\hat{\theta} = \arg \min_{\theta} J(\theta) \quad (7)$$

There are various numerical algorithms to optimize  $\theta$ , such as Levenberg–Marquardt and stochastic gradient descent (SGD) [5,27], which are the following partial derivative formula:

$$\theta := \theta - \alpha \frac{\partial J(\theta)}{\partial \theta} \quad (8)$$

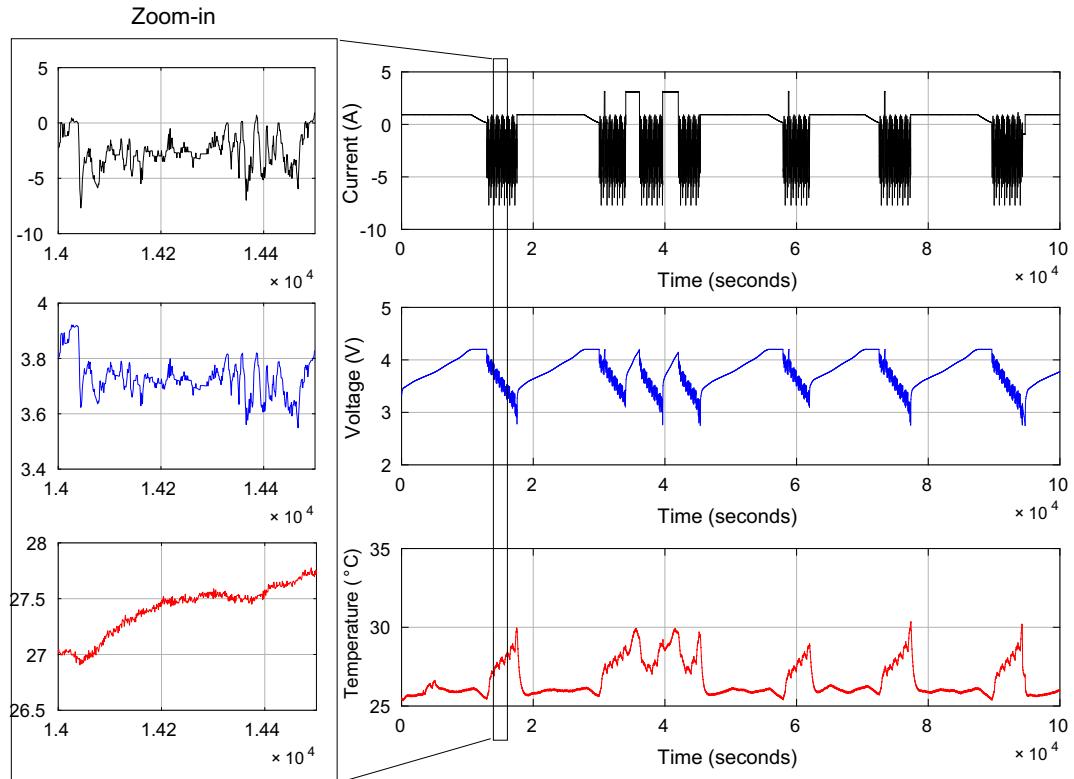
Some  $\theta$  is initialized<sup>3</sup> and repeatedly updated until its convergence; here,  $\alpha$  is called the *learning rate*. They almost leverage the gradient with respect to the network weights  $\theta$  to minimize the estimation error in Eq. (7) using a “backpropagation” algorithm, which computes backwardly through the network based on the chain rule of calculus. However, the goal of this study is not to develop the NN or its learning algorithms “as they are” but to apply them to our problem properly. Thus, we must determine how to formulate the data  $x_i$ 's from the raw sensed data to correctly estimate SOH  $y_i$ 's.

### 3.2. Formulation of historical distributions

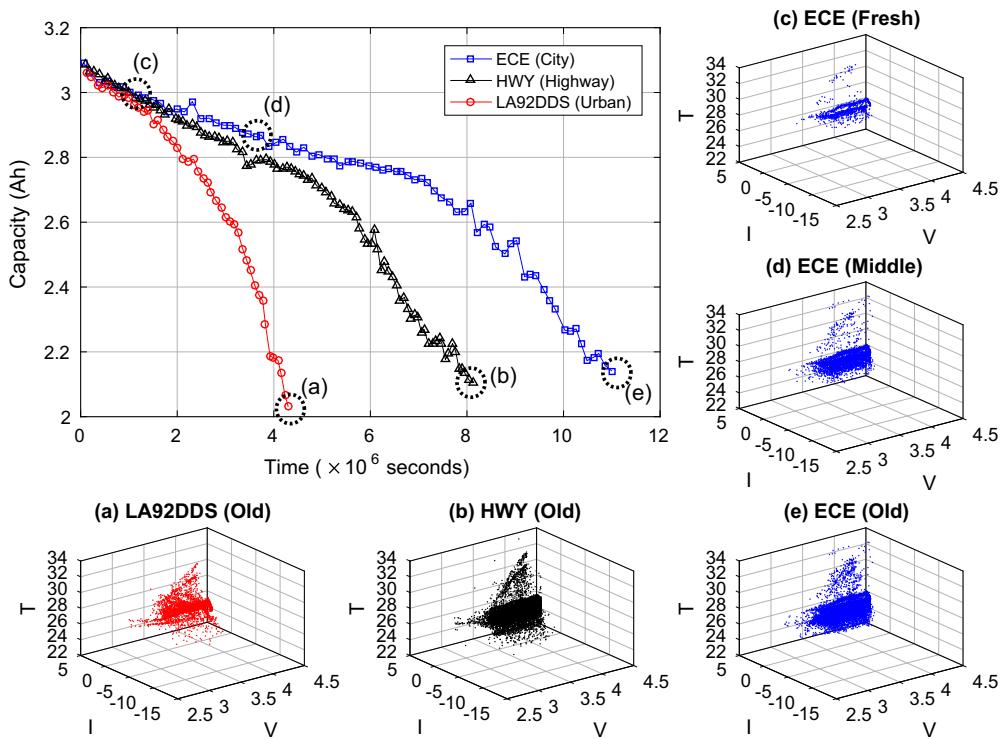
It is common knowledge that the aging effect is highly dependent on usage of certain signal ranges. For instance, persistently using the batteries in low voltage, high C-rate, and high temperature situations accelerates their aging. To reveal the different effects caused by such usage ranges, we segment the entire historical distribution region into smaller subregions and then assess each subregion separately.

A naive solution is to divide each dimension into several disjoint subranges and adopt corresponding subregions as their combinations. For example, if we divide three I/V/T dimensions into  $m$  disjoint ranges, the entire region is partitioned to  $m^3$  cubic subregions. (Although  $m$  can vary for each dimension, for the sake of explanation, we set  $m$  to the same value for all dimensions.)

<sup>3</sup> Typical NNs heuristically initialize  $\theta$  with some random values, but DNNs (deep neural networks) such as DBN (deep belief network) leverage a probabilistic method, e.g., stacking layers with RBMs (restricted Boltzmann machines) bottom to top, which alleviates the information vanishing problem.



**Fig. 3.** Dependency of current (black), voltage (blue), and temperature (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 4.** Various  $I/V/T$  historical patterns (a)–(e), and their capacity degradation.

However, this solution has two disadvantages. First, as the size of  $m$  increases, the number of subregions (combinations) increases cubically. As the types of signals increase in number, this problem

becomes more severe. This property also makes it difficult to adjust the number of subregions with a fine granularity. Second, because it does not consider correlation between signals (described in

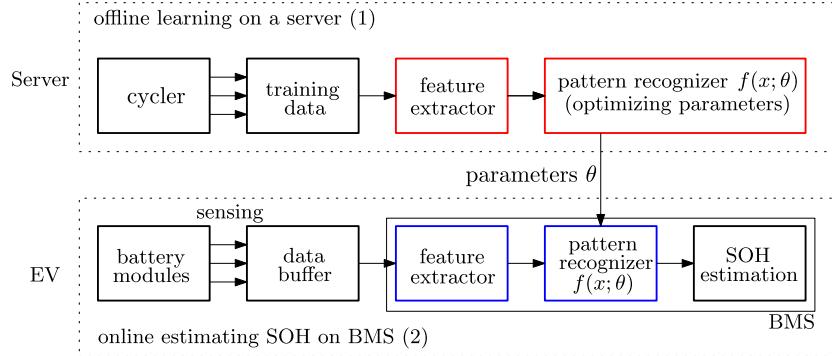


Fig. 5. Our framework (top: offline learning, bottom: online processing).

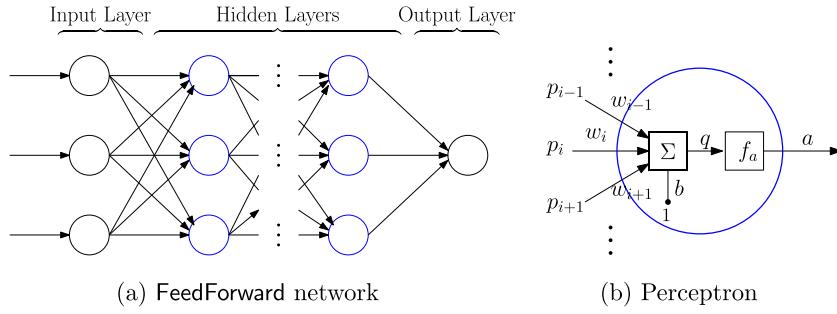


Fig. 6. Structure of neural network.

Fig. 1), many unnecessary subregions to which no point belongs can be generated.

To overcome these problems, we decide  $k$  subregions by clustering the data collected a priori into a desirable size of  $k$  groups. In particular, we use the  $k$ -means clustering algorithm [5,29] because of its simplicity and popularity. More specifically, given a dataset  $\mathcal{D}_{\text{joint}} = \{p_1, \dots, p_N\}$ , it divides  $\mathcal{D}_{\text{joint}}$  into  $k$  disjoint subsets  $D_1, \dots, D_k$  such that  $\forall i \neq j, D_i \cup D_j = \emptyset$ . As its byproducts, all subsets  $D_1, \dots, D_k$  have their corresponding centroids  $\mathcal{C} = \{c_1, \dots, c_k\}$ , where every point  $p_j \in D_i$  has the shortest distance to  $c_i$  out of all the centroids  $\mathcal{C}$ :

$$c_i = \arg \min_{c_i \in \mathcal{C}} d(p_j, c_i), \quad (9)$$

where  $d(\cdot, \cdot)$  is a distance between two points, e.g., L1 (Manhattan) and L2 (Euclidean) distance [5].

Using these centroids, we can divide the entire region into  $k$  subregions more flexibly than by using the above naive solution, in which every point  $p_i$  is assigned to the subregion whose centroid is the closest to  $p_i$ . While the naive solution only separates the region orthogonally to every axis, the proposed solution can separate the region regardless of such directions. In particular, the  $k$  subregions look like the cells of a Voronoi diagram [5]. For example, Fig. 7 describes a two-dimensional toy example of a  $k$ -means clustering algorithm. There are 26 points in the entire region of Fig. 7(a); these points are partitioned into the six subregions by the six centroids  $\mathcal{C} = \{c_1, \dots, c_6\}$  (marked with red 'x's) in Fig. 7(b). For instance, a blue box point  $p_i$  belongs to the subregion of centroid  $c_3$  with the minimum distance among all centroids  $\mathcal{C}$ .

We can now formulate the training data  $x_i$ 's to learn the parameters  $\theta$  (Eq. (7)) of the SOH estimation model  $f(x_i; \theta)$ . The training data  $x_i$ 's are represented as density values of the subregions, i.e., the number of points assigned to the subregions corresponding to  $k$  centroids. For example, Fig. 8(a) displays the 10 historical

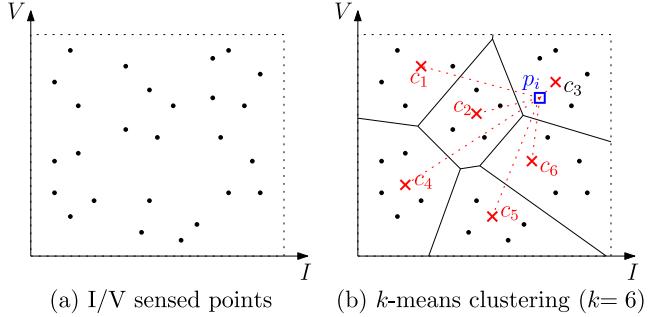
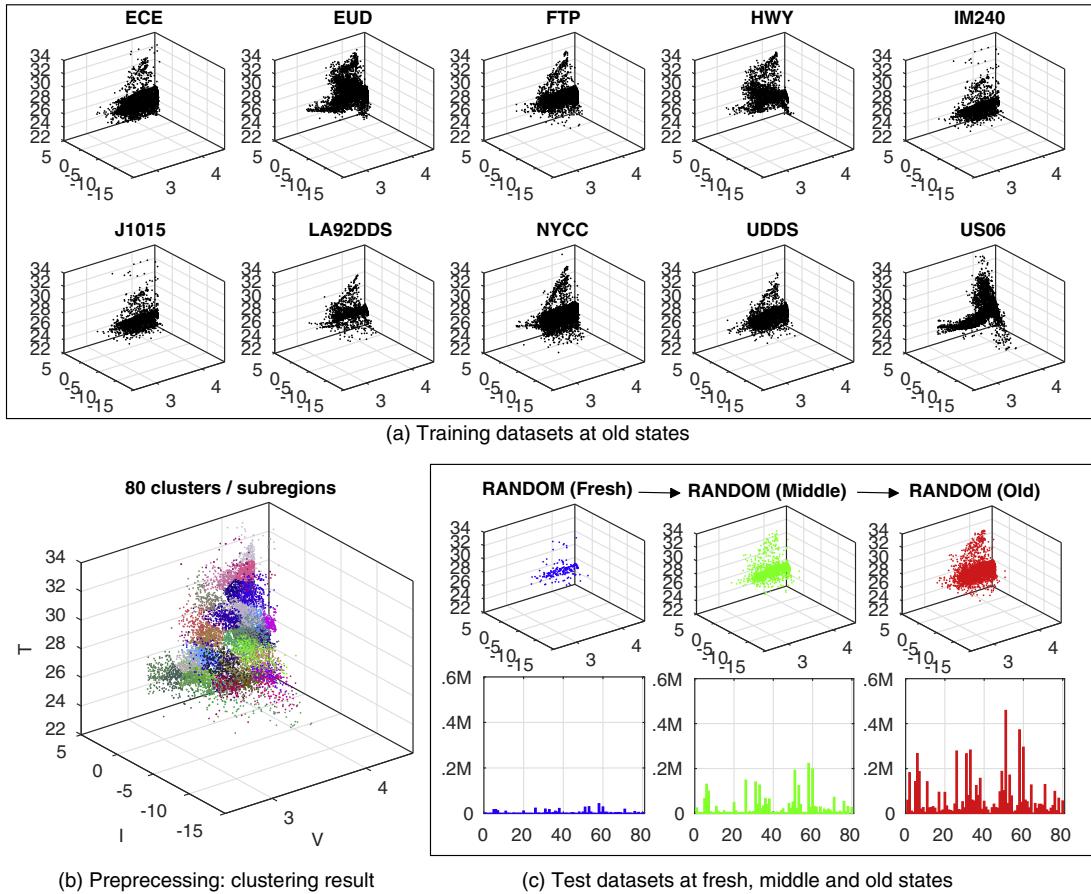


Fig. 7. Toy example of  $k$ -means clustering algorithm.

distributions accumulated until the batteries' end to life. We merge all the distributions and separate the entire region into 80 subregions by finding the 80 centroids with a  $k$ -means clustering algorithm when  $k = 80$  in Fig. 8(b). The points in the same subregions are drawn with the same color. Based on this separation, for RANDOM, Fig. 8(c) shows the density values of the 80 subregions when the cell is at fresh (blue), middle (green) and old (red) states, respectively, where 80 bars represent the numbers of points that belong to the corresponding subregions. The three 80-dimensional vectors can be used as the inputs to test our model NN. Here, we stress that (a)-(b) are executed offline (server part in Fig. 5) and (c) is only processed online to estimate SOH (EV part in Fig. 5). The centroids produced in (b) are built in BMS a priori and those are used to assess the density values as in (c). Although we depict the ten distributions at EOL in (a), we periodically assess the density values to collect training datasets offline like (c) online.

More specifically, by using this formulation method, we incrementally collect training data  $\mathcal{D}_{\text{train}}$  of Eq. (5) (in the same manner as the test data  $\mathcal{D}_{\text{test}}$ ) as follows:



**Fig. 8.** Historical distributions and their formulation.

1. Given a cell dataset  $\mathcal{D}_{\text{joint}} = \{p_1, \dots, p_{\text{EOL}}\}$  of Eq. (4) during its lifetime and centroids  $\mathcal{C} = \{c_1, \dots, c_k\}$ , let  $x_i = (x_i^1, \dots, x_i^k)$  with  $\forall j, x_i^j = 0$ , and set  $\text{first} = 1, \text{end} = \Delta$  where  $\Delta$  is a tunable size for evaluation frequency.
2. For  $\text{first} \leq l \leq \text{end}$ , find a centroid  $j$  to which  $p_l$  belongs using Eq. (9), and set  $x_i^j = x_i^j + 1$ .
3.  $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{train}} \cup \{(x_i, y_i)\}$  such that  $y_i$  is an actual SOH. Note that  $y_i$  should be measured precisely, e.g., by fully cycling the battery to assess the capacity (refer to the details in Section 4.1).
4.  $\text{first} = \text{first} + \Delta, \text{end} = \text{end} + \Delta$ , and goto 2.

To summarize, our approach is a data-driven approach to recognize the density of the time-varying distributions. Battery usage history shows that density patterns can be diverse. To address the differences between those patterns, we propose a method to formulate the battery signals by assessing the densities of the subregions.

## 4. Experiments

This section reports our experimental results to validate our proposed framework. Section 4.1 reports how we acquired our evaluation datasets, and Section 4.2 extensively validates the effectiveness of our framework.

### 4.1. Experimental settings: data acquisition

To our knowledge, there are no battery cycling datasets that reflect public driving conditions. Because we could not obtain the

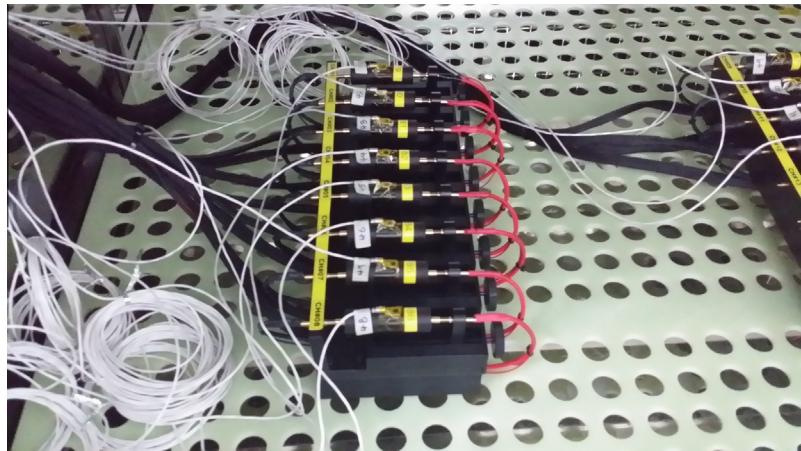
sensed data from EVs directly, we first collected such data by exploiting a cycler, a piece of simulation equipment that can charge and discharge the batteries by imposing varying current. We used an 18650 form-factor battery cell with a capacity of approximately 3.1 Ah at room temperature. We performed cycling in various chambers with four ambient temperatures (10/25/45/60 °C). In particular, Fig. 9 shows the configuration of our experimental equipment in the chamber, where our interior setting included batteries, jigs, sensing wires, and channels connected to the cycler.

Sections 4.1.1 and 4.1.2 explain how we used this setting to generate discharging and charging profiles, respectively. Based on these explanations, Section 4.1.3 describes how we completed total cycling profiles by combining discharging and charging together.

#### 4.1.1. Discharging profile generation

To emulate the batteries as realistically as possible, we downloaded 10 dynamometer driving profiles, including UDDS and LA92DDS, from the website of US Environmental Protection Agency (EPA).<sup>4</sup> These profiles include speed information for emission testing of gasoline vehicles (called dynamometer tests). Those profiles vary considerably depending on their road conditions (downtown, highway, etc). Ever-changing acceleration causes the batteries to be discharged quite dynamically; as a result, the batteries age at different rates. For example, in the harsher profiles with frequent changes of speed (e.g., LA92DDS) and high speeds (e.g., HWY), the batteries age faster. To observe these patterns more intuitively, we calculated several statistical values, as Table 1 summarizes. To adopt various

<sup>4</sup> Refer to the site: <http://www.epa.gov/nvfel/testing/dynamometer.htm>.



**Fig. 9.** Configuration for our experimental settings.

**Table 1**  
Statistics of 10 dynamometer driving profiles.

Name	Length (s)	Distance (km)	Avg. speed (kph)	Max. speed (kph)	Std. speed (kph)	Type
ECE	196	0.99	18.26	50.00	17.06	City
EUD	401	6.95	62.44	120.00	31.57	Highway
FTP	1875	17.77	34.12	91.25	25.66	Urban
HWY	766	16.51	77.58	96.40	16.48	Highway
IM240	241	3.15	47.08	91.25	25.42	Urban
J1015	892	6.34	25.58	70.00	23.82	City
LA92DDS	1436	15.80	39.60	108.15	31.74	Urban
NYCC	599	1.90	11.41	44.58	12.88	City
UDDS	1370	11.99	31.51	91.25	23.65	Urban
US06	601	12.89	77.20	129.23	39.56	Highway

driving scenarios, we categorize them into three types: (1) *city*: low average speed, low maximum speed, and frequent stops, (2) *highway*: high average speed, high maximum speed, and few stops, and (3) *urban*: others conditions.

Given a profile, we imposed the load corresponding to its speed to the battery cell in the cycler by scaling and transforming the speed sequence of the profile into the current sequence; this was accomplished reflecting several influential factors, such as the mass of the car, rolling and wind resistance, power, and self-recharging ratio. For instance, Fig. 10 shows the speed variation of the UDDS profile in (a) and its transformation to current in (b). Note that, for the sake of representation, we flip over current signs in Fig. 10 (+: charge, -: discharge). The slowing-down speed causes a self-recharging with a plus value.

#### 4.1.2. Charging profile generation

In general, a standard battery charging procedure consists of two phases: (1) constant current (CC) phase and (2) constant voltage (CV) phase. More specifically, the battery is charged with a constant current (e.g.,  $0.3 \text{ C} \approx 0.93 \text{ A}$  in Fig. 11) until it reaches its upper bound voltage (e.g.,  $4.2 \text{ V}$  in Fig. 11); it maintains constant voltage until the current drops to its lower bound current (e.g.,  $1/20 \text{ C} \approx 155 \text{ mA}$  in Fig. 11), where the CV phase helps to mitigate an over-potential by the charging speed. Because we were also unable to obtain a charging profile based on this procedure, we divided the charging into three fast modes (CC20, CC40, and CC60) and one slow mode (CCCV) as follows:

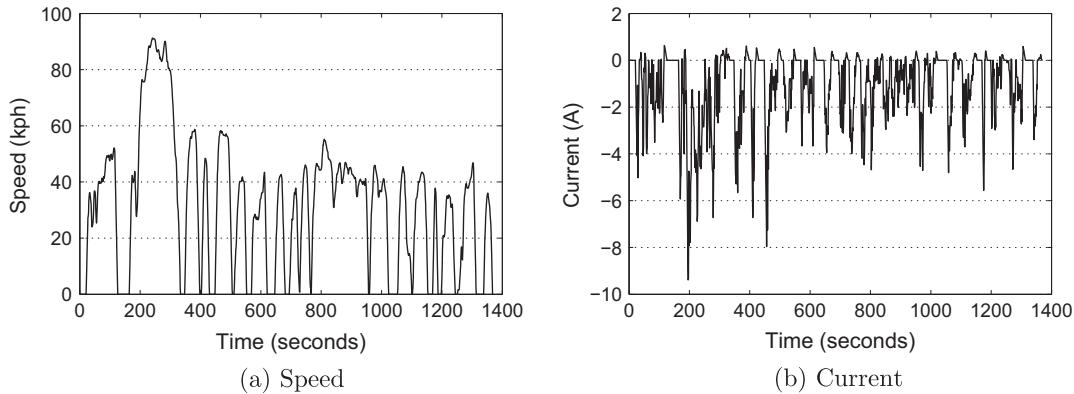
- CC20: A fast mode of  $1 \text{ C}$  ( $\approx 3.1 \text{ A}$ ) CC 20 min in duration
- CC40: A fast mode of  $1 \text{ C}$  CC 40 min in duration
- CC60: A fast mode of  $1 \text{ C}$  CC 60 min in duration
- CCCV: A slow mode of  $0.3 \text{ C}$  ( $\approx 0.93 \text{ A}$ ) CC and CV

#### 4.1.3. Whole profile generation

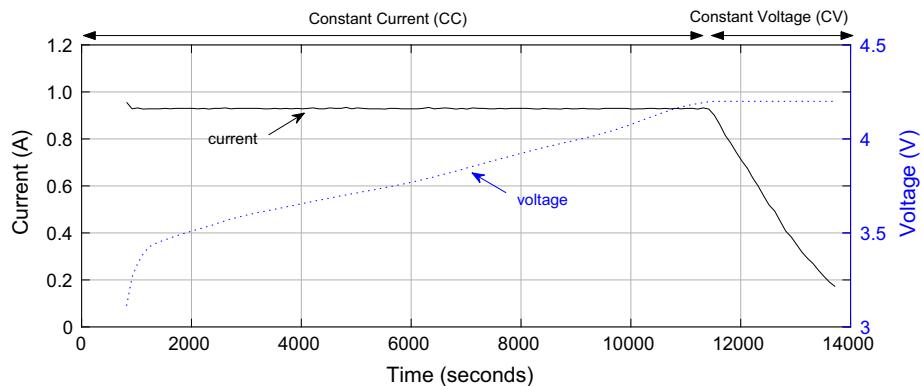
By combining the strategies described in the previous two sections, we generated our complete profiles by randomly mixing discharge and charge profiles. We collected a total of 44 datasets (11 datasets for each temperature  $10/25/45/60^\circ\text{C}$ ). For each temperature, we generated 10 datasets corresponding to 10 discharging profiles to train our model. We then generated a profile combining those profiles at random to test the model. As their representatives, Fig. 12 depicts the  $I/V/T$  historical distributions for the four random profiles when the cells are in the almost end states. Observe that, in  $z$ -axes, they cover various temperature ranges ( $5$ – $70^\circ\text{C}$ ). As the cells are more rapidly aged in both extreme temperatures, the distribution in  $45^\circ\text{C}$  is the densest among the four ambient temperatures because of its long lifespan. Additionally, to emulate time-varying ambient temperature, by weekly switching the battery cells between room- and high-temperature chambers, we collected a total of eight datasets (four sets between  $45$  and  $25^\circ\text{C}$ , and four sets between  $65$  and  $25^\circ\text{C}$ ) by cycling four random profiles similar to previous experiments.

Note that, on a regular basis, we exhausted some remaining energy to assess a reference capacity by performing a CCCV mode charge. Although this is unnecessary in a realistic environment, we could not help but interleave this step to identify an actual capacity, which is necessary to optimize our model or test its effectiveness. To obtain the reference values of the entire lifetime for each cell, we interpolate these intermittent points as the SOH gradually decreases. We also argue that our approach aims at the applications wherein the available driving mileage is the most important, and thus we optimize our model to estimate the capacity. However, we can apply our approach to the other objectives, e.g., on internal resistance or power, in the same way.

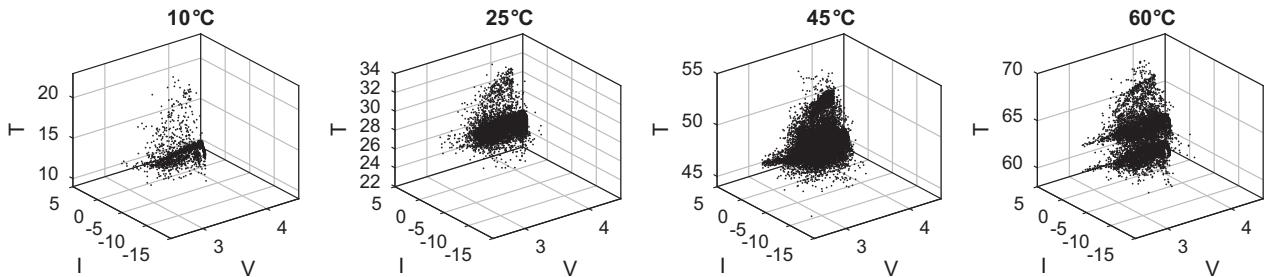
These profiles repeat cycling until the capacity of the batteries is less than approximately 80%, which is their available scope because, thereafter, their capacity steeply deteriorates. It took more than a year to collect these datasets; in retrospect, we should have let the batteries age until their capacities were less than the threshold. Although we did not reflect the rest effect of the batteries and the aging was accelerated, the process was very costly and time-consuming.



**Fig. 10.** Transformation from driving speed to current.



**Fig. 11.** Two charging phases (constant current (CC) and constant voltage (CV)).



**Fig. 12.** Historical distributions for 10/25/45/60 °C.

#### 4.2. Experimental results: SOH estimation accuracy

This section reports the experimental results obtained using the cycling datasets described in Section 4.1. We define two metrics to measure the effectiveness in Section 4.2.1 and report the results in Section 4.2.2.

#### *4.2.1. Effectiveness measures*

This section defines two metrics to validate our framework. First, the root mean squared error (RMSE) is widely used to measure the accuracy of the estimation model by computing the average error. Given a test dataset  $\mathcal{D}_{\text{test}} = \{(x_1, y_1), \dots, (x_m, y_m)\}$  and an estimation model  $f(x; \theta)$  where  $\theta$  is a set of the tuned parameters, it is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - f(x_i; \theta))^2} \quad (10)$$

Second, because we should continue to trace the SOH, which can be utilized to estimate SOC or calculate driving mileage on the fly, the estimation accuracy can be important at every moment. To handle it, we define the maximum absolute error (MAE) as follows:

$$\text{MAE} = \max_{1 \leq i \leq m} |y_i - f(x_i; \theta)| \quad (11)$$

#### *4.2.2. Effectiveness results*

We first found the best number of subregions to optimize our estimation results. As mentioned in Section 3.1, depending on the initial parameters, NN is optimized differently. Thus, we chose the best model among the ten models trained after different random initialization. Fig. 13 shows the ten test errors (y-axis) over the designated number of the subregions (x-axis). Observe that, when the number of subregions is 80, the average error (red) is minimized and the deviation is also minimal. Note

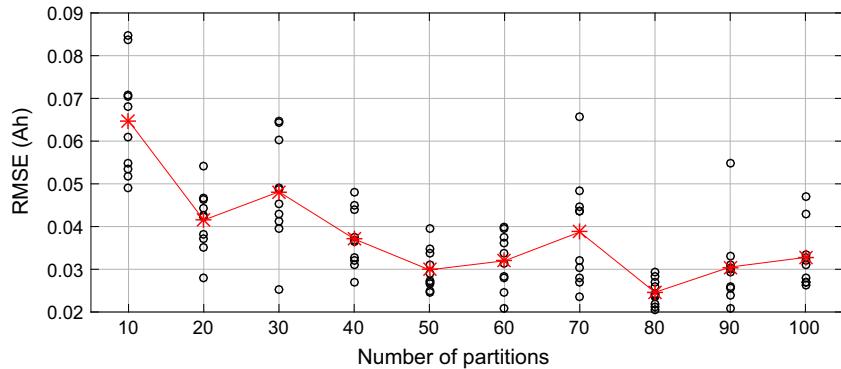


Fig. 13. Optimizing the number of subregions  $k$  with 25 °C datasets.

that, these experiments were carried out with 25 °C datasets and these optimized numbers were applied to the other experimental settings.

In addition, when we optimized our model NN, we adopted various optimization technologies which were well-known in the machine learning area. First, to prevent our model from biasing to certain dimensions, we normalized all datasets to have mean zero and scaled to have standard deviation one (called Z-score). Second, when we built our NN model, we stacked only single hidden layer with 80 nodes because more complex model (e.g., multiple hidden layers and more nodes) can overfit our datasets. This simple structure also allows extremely fast computation for each estimation less than hundreds milliseconds with a commodity CPU. Third, we adopt SGD as a parameter optimization algorithm with miscellaneous techniques, e.g., *momentum* for improving the optimization speed and *dropout* for regularization.

We validated our proposed approach using the historical distributions of  $I/V/T$  signals. We implemented another data-driven approach introduced in Section 2 as a baseline approach. Our approach has a similar philosophy in a sense that both employ some combinatorial frequencies of  $V/I/T$ . However, while they handle 2-dimensional combinations called *collective* in their paper (denoted CL), e.g.,  $I/T$  and  $V/T$  [23], we manipulate 3-dimensional ones of  $I/V/T$  (denoted HS). In particular, by applying the  $k$ -means clustering algorithm, we can automatically find more flexible regions (as suggested in Section 3.2) while they only consider uniform rectangular regions. In summary, we ran the following three algorithmic variations:

- SVM(CL) [23]: This is the baseline using an SVM model to estimate SOH where the inputs are formulated as two combinatorial matrices between two different signals, i.e.,  $I/T$  and  $V/T$ , and a matrix generated the rainflow counting method.
- SVM(HS): This is an SVM model using our  $I/V/T$  historical distributions.
- NN(HS): This is our suggested approach based on an NN model using our  $I/V/T$  historical distributions.

Table 2 summarizes the effectiveness of the results using RMSE and MAE values over the above three variations, where the values in parenthesis are normalized by their initial capacity 3.1(Ah) and the bold numbers mark the best performance. All numbers are evaluated with the test datasets generated by RANDOM profiles. Observe that our approach shows the best performance for all temperature conditions with average RMSE = 0.0329 Ah (1.06%) and average MAE = 0.1183 (3.81%). Interestingly, observe that SVM (HS) outperforms SVM(CL), which indirectly shows the effectiveness of our automatic partitioning method using the  $k$ -means clustering.

To schematically show how our proposed approach runs, Fig. 14 (a)–(d) plot the estimated capacities of the three approaches and their corresponding actual capacities at 10, 25, 45, and 60 °C, where the gray ranges represent 5% error boundaries from the actual capacities. Although the degradation trajectories vary considerably with temperature, only our estimation results (blue lines) are always within these ranges and trace very well in all temperatures.

Finally, we evaluated those approaches over two cases of the datasets reflecting time-varying temperature between 45 and 25 °C, and 60 and 25 °C. Rather than the above experiments, collecting these datasets was more labor-intensive and time-consuming because we should periodically move the cells form one chamber to another retaining different temperature. Since we have only few datasets compared to the above, we tested the approaches with 4-fold cross validations. More specifically, given four disjoint datasets  $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_4\}$ , we then used  $\mathcal{D} - \mathcal{D}_i$  for optimizing SVM or NN, and the other  $\mathcal{D}_i$  for testing the results. Table 3 shows the average results of the four folds over the two cases of datasets. Although the gap between our approach and the other approaches narrows down for RMSE, our approach still remains the winner. For MAE, although SVM(HS) is the best, the gap among the three methods is slight. This results show our partitioning method is still working effectively as SVM(HS) uses it as well.

## 5. Conclusion and future work

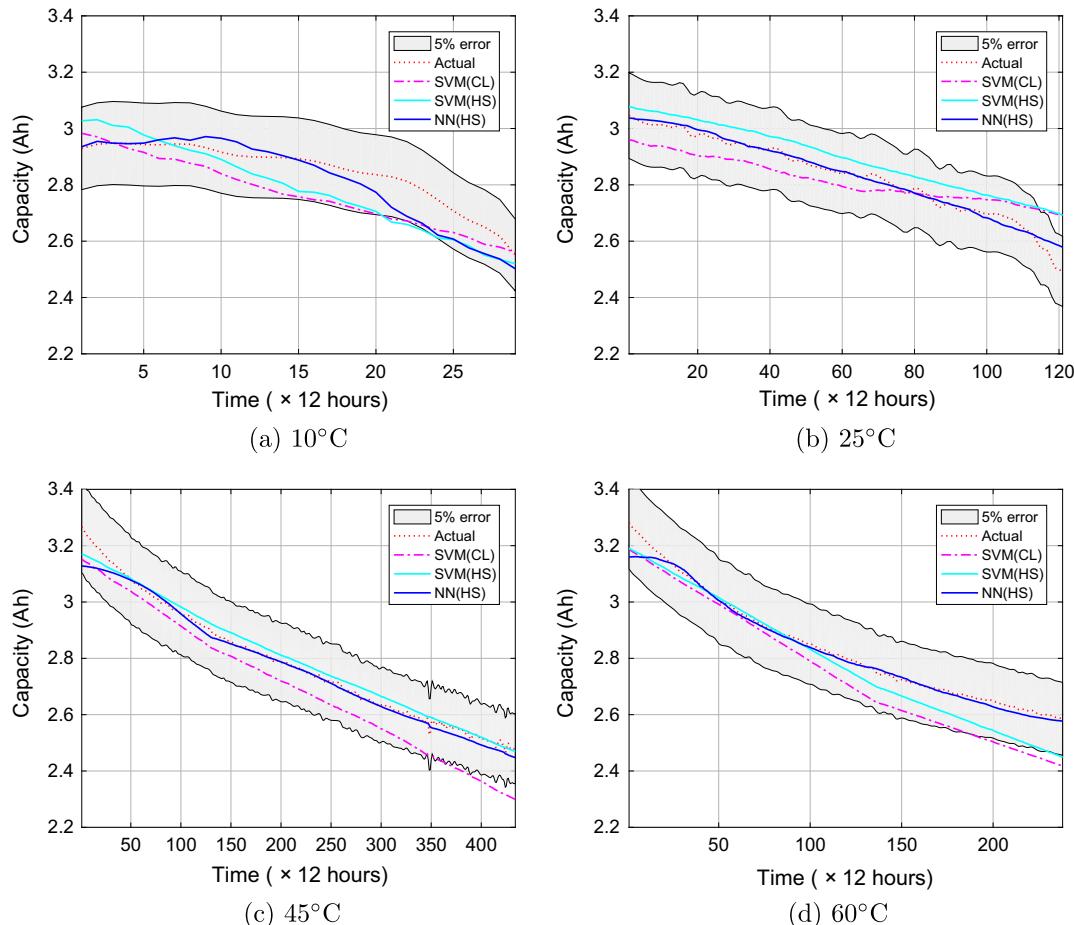
This paper proposed a method to trace the SOH of EV batteries using sensible data (current, voltage, and temperature) from a BMS where batteries are cycled dynamically according to various driving patterns. To the best of our knowledge, this paper is the first effort to estimate SOH using extensive driving profiles to consider EV environments in a more practical way. We struggled to collect those datasets for more than a year in four ambient temperatures (10/25/45/60 °C) and time-varying temperature (45–25 °C and 60–25 °C). We also proposed a data-driven framework and validated that our framework worked effectively with those datasets in all settings, i.e., RMSE = 0.0677 (average error less than 2.18%).

Meanwhile, in future work, we will consider the following issues. First, we will consider the rest effect in the aging process. While this paper accelerates the aging process by continuously cycling our profiles without rest, interleaving the rest with our cycling patterns makes the experiments more realistic. However, as such experiments are too time-consuming, we are seeking a solution to minimize the interleaved rest time. Second, so far, we addressed the SOH of the cells composing an EV's battery pack. Because the pack consists of serial and parallel cells, based on this

**Table 2**

Effectiveness on four temperatures (()) normalized by initial capacity 3.1(Ah)).

Temperature (°C)	RMSE (Ah)			MAE (Ah)		
	SVM(CL)	SVM(HS)	NN(HS)	SVM(CL)	SVM(HS)	NN(HS)
10	0.0954 (0.0308)	0.0936 (0.0302)	<b>0.0619</b> <b>(0.0200)</b>	0.1498 (0.0483)	0.1607 (0.0518)	<b>0.1251</b> <b>(0.0404)</b>
25	0.0699 (0.0225)	0.0703 (0.0227)	<b>0.0206</b> <b>(0.0066)</b>	0.1988 (0.0641)	0.2004 (0.0647)	<b>0.0863</b> <b>(0.0278)</b>
45	0.0924 (0.0298)	0.0270 (0.0087)	<b>0.0269</b> <b>(0.0087)</b>	0.1871 (0.0604)	<b>0.0960</b> <b>(0.0310)</b>	0.1394 (0.0450)
60	0.0989 (0.0319)	0.0665 (0.0215)	<b>0.0223</b> <b>(0.0072)</b>	0.1673 (0.0540)	0.1354 (0.0437)	<b>0.1222</b> <b>(0.0394)</b>
Average	0.0891 (0.0288)	0.0644 (0.0208)	<b>0.0329</b> <b>(0.0106)</b>	0.1758 (0.0567)	0.1481 (0.0478)	<b>0.1183</b> <b>(0.0381)</b>



**Fig. 14.** SOH estimation results over four ambient temperatures (dotted red: actual capacity, dotted dashed purple: SVM(CL), plain light blue: SVM(HS), plain blue: NN(HS), gray: 5% error boundary) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

**Table 3**

Effectiveness on 4-fold cross validation in time-varying ambient temperature.

Temperature (°C)	RMSE (Ah)			MAE (Ah)		
	SVM(CL)	SVM(HS)	NN(HS)	SVM(CL)	SVM(HS)	NN(HS)
45–25	0.1167 (0.0376)	<b>0.0586</b> <b>(0.0189)</b>	0.0599 (0.0193)	0.1742 (0.0562)	<b>0.1085</b> <b>(0.0350)</b>	0.1603 (0.0517)
60–25	0.0789 (0.0255)	0.1289 (0.0416)	<b>0.0755</b> <b>(0.0244)</b>	<b>0.1622</b> <b>(0.0523)</b>	0.2070 (0.0668)	0.1880 (0.0606)
Average	0.0978 (0.0316)	0.0938 (0.0303)	<b>0.0677</b> <b>(0.0218)</b>	0.1682 (0.0543)	<b>0.1577</b> <b>(0.0509)</b>	0.1741 (0.0562)

approach, we will study how we can extend it for application to the pack.

## References

- [1] Rahimi-Eichi Habiballah, Ojha Unnati, Baronti Federico, Chow Mo-Yuen. Battery management system: an overview of its application in the smart grid and electric vehicles. *IEEE Ind Electron Mag* 2013;7:4–16.
- [2] Haifeng Dai, Xuezhe Wei, Zechang Sun. A new SOH prediction concept for the power lithium-ion battery used on HEVs. In: Proceedings of vehicle power and propulsion conference, VPPC'09. IEEE; 2009. p. 1649–53.
- [3] Remminger Jürgen, Buchholz Michael, Meiler Markus, Bernreuter Peter, Dietmayer Klaus. State-of-health monitoring of lithium-ion batteries in electric vehicles by on-board internal resistance estimation. *J Power Sources* 2011;196:5357–63.
- [4] Williard Nick, He Wei, Osterman Michael, Pecht Michael. Comparative analysis of features for determining state of health in lithium-ion batteries. *Int J Progn Health Manage* 2013;1:1–7.
- [5] Duda Richard O, Hart Peter E, Stork David G. Pattern classification. 2nd ed. Wiley-Interscience; 2000.
- [6] Barre Anthony, Suard Frederic, Gerard Mathias, Montaru Maxime, Riu Delphine. Statistical analysis for understanding and predicting battery degradations in real-life electric vehicle use. *J Power Sources* 2014;245:846–56.
- [7] Rezvanianani Seyed Mohammad, Liu Zongchang, Chen Yan, Lee Jay. Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle (EV) safety and mobility. *J Power Sources* 2014;256:110–24.
- [8] Saha Bhaskar, Goebel Kai. Modeling Li-ion battery capacity depletion in a particle filtering framework. In: Proceedings of annual conference of the prognostics and health management society, PHM'09. p. 1–10.
- [9] Chen Zheng, Mi Chunting Chris, Fu Yuhong, Xu Jun, Gong Xianzhi. Online battery state of health estimation based on genetic algorithm for electric and hybrid vehicle applications. *J Power Sources* 2013;240:184–92.
- [10] Le Daniel, Tang Xidong. Lithium-ion battery state of health estimation using Ah-V characterization. In: Proceedings of annual conference of the prognostics and health management society, PHM'11. p. 1–7.
- [11] Eddahch Akram, Briat Olivier, Bertrand Nicolas, Delestage Jean-Yves, Vinassa Jean-Michel. Behavior and state-of-health monitoring of Li-ion batteries using impedance spectroscopy and recurrent neural networks. *Int J Electr Power Energy Syst* 2012;42:487–94.
- [12] Lin Ho-Ta, Liang Tsoring-Juu, Chen Shih-Ming. Estimation of battery state of health using probabilistic neural network. *IEEE Trans Ind Inform* 2013;9:679–85.
- [13] Liu Datong, Pang Jingyue, Zhou Jianbao, Peng Yu, Pecht Michael. Prognostics for state of health estimation of lithium-ion batteries based on combination gaussian process functional regression. *Microelectron Reliab* 2013;53:832–9.
- [14] He Wei, Williard Nick, Osterman Michael, Pecht Michael. Prognostics of lithium-ion batteries using extended kalman filtering. In: Proceedings of IMAPS advanced technology workshop on high reliability microelectronics for military applications. p. 1–4.
- [15] Goebel Kai, Saha Bhaskar, Saxena Abhinav, Celaya Jose R, Christoffersen Jon P. Prognostics in battery health management. *IEEE Instrum Meas Mag* 2008;11:33–40.
- [16] Liu Datong, Luo Yue, Liu Jie, Peng Yu, Guo Limeng, Pecht Michael. Lithium-ion battery remaining useful life estimation based on fusion nonlinear degradation AR model and RPF algorithm. *Neural Comput Appl* 2014;25:557–72.
- [17] Ng Selina SY, Xing Yinjiao, Tsui Kwok L. A naive Bayes model for robust remaining useful life prediction of lithium-ion battery. *Appl Energy* 2014;118:114–23.
- [18] Patil Meru A, Tagade Piyush, Hariharan Krishnan S, Kolake Subramanya M, Song Taewon, Yeo Taejung, et al. A novel multistage support vector machine based approach for Li ion battery remaining useful life estimation. *Appl Energy* 2015;159:285–97.
- [19] Wu Ji, Zhang Chenbin, Chen Zonghai. An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks. *Appl Energy* 2016;173:134–40.
- [20] Patil Nishad, Das Diganta, Pecht Michael. A prognostic approach for non-punch through and field stop IGBTs. *Microelectron Reliab* 2012;52:482–8.
- [21] Patil Nishad, Das Diganta, Pecht Michael. Anomaly detection for IGBTs using Mahalanobis distance. *Microelectron Reliab* 2015;55:1054–9.
- [22] Oh Hyunseok, Han Bongtae, McCluskey Patrick, Han Changwoon, Youn Byeng D. Physics-of-failure, condition monitoring, and prognostics of insulated gate bipolar transistor modules: a review. *IEEE Trans Power Electron* 2015;30:2413–26.
- [23] Nuhic Adhan, Terzimehic Tarik, Soczka-Guth Thomas, Buchholz Michael, Dietmayer Klaus. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *J Power Sources* 2013;239:680–8.
- [24] Klass Verena, Behm Marten, Lindbergh Goran. A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation. *J Power Sources* 2014;270:262–72.
- [25] Klass Verena, Behm Marten, Lindbergh Goran. Capturing lithium-ion battery dynamics with support vector machine-based battery model. *J Power Sources* 2015;298:92–101.
- [26] Zou Yuan, Hu Xiaosong, Ma Hongmin, Li Shengbo Eben. Combined state of charge and state of health estimation over lithium-ion battery cell cycle lifespan for electric vehicles. *J Power Sources* 2015;273:793–803.
- [27] Beale Mark Hudson, Hagan Martin T, Demuth Howard B. Neural network toolbox user's guide R2014a. Mathworks; 2014.
- [28] Hagan Martin T, Demuth Howard B, Beale Mark H. Neural network design. PWS Publishing; 1996.
- [29] Cormen Thomas H, Stein Clifford, Rivest Ronald L, Leiserson Charles E. Introduction to algorithms. 2nd ed. McGraw-Hill Higher Education; 2001.