Energy Reports 9 (2023) 2993–3021

Contents lists available at ScienceDirect

Energy Reports

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

Review article

A review of machine learning state-of-charge and state-of-health estimation algorithms for lithium-ion batteries

Zhong Ren, Changqing Du *

Hubei Key Laboratory of Advanced Technology for Automotive Components (Wuhan University of Technology), Wuhan 430070, China
Foshan Xianhu Laboratory of the Advanced Energy Science and Technology Guangdong Laboratory, Foshan 528200, China
Hubei Research Center for New Energy & Intelligent Connected Vehicle, Wuhan University of Technology, Wuhan 430070, China

ARTICLE INFO

Article history:
Received 7 March 2022
Received in revised form 24 October 2022
Accepted 23 January 2023
Available online 10 February 2023

Keywords:
Lithium-ion batteries
Machine learning techniques
State-of-charge
State-of-health

ABSTRACT

Vehicle electrification has been proven to be an efficient way to reduce carbon dioxide emissions and solve the energy crisis. Lithium-ion batteries (LiBs) are considered the dominant energy storage medium for electric vehicles (EVs) owing to their high energy density and long lifespan. To maintain a safe, efficient, and stable operating condition for the battery system, we must monitor the state of the battery, especially the state-of-charge (SOC) and state-of-health (SOH). With the development of big data, cloud computing, and other emerging techniques, data-driven machine learning (ML) techniques have attracted attention for their enormous potential in state estimation for LiBs. Therefore, this paper reviews the four most studied types of ML algorithms for SOC and SOH estimation, including shallow neural network (NN), deep learning (DL), support vector machine (SVM), and Gaussian process regression (GPR) methods. The basic principles and uniform flowcharts of different ML algorithms are introduced. Then, the applications of each ML algorithm for state estimation within recent years are comprehensively reviewed and compared in terms of used datasets, input features, hyperparameter selection, performance metrics, advantages, and disadvantages. Based on the investigation, this review discusses the current challenges and prospects from four aspects, aiming to provide some inspiration for developing advanced ML state estimation algorithms.

Contents

* Corresponding author at: Hubei Key Laboratory of Advanced Technology for Automotive Components (Wuhan University of Technology), Wuhan 430070, China.
E-mail address: cq_du@whut.edu.cn (C. Du).

https://doi.org/10.1016/j.egyr.2023.01.108

1. Introduction

To reduce carbon dioxide emissions and the harm caused by climate change, such as the greenhouse effect, sea level rise, and reduced biodiversity, the development of energy-saving and emission-reduction technologies has become the direction of joint efforts of all countries. Another emergency is the energy crisis, which has made a considerable difference in the worldwide energy system. According to the COP26 held in Glasgow, United Kingdom, the representatives of governments, businesses, and other organizations are committed to rapidly accelerating the transition to 100% zero-emission vehicles to achieve the goals of the Paris Agreement by 2040 or earlier or by no later than 2035 in leading markets (COP26, 2022). Vehicle electrification has been proven to be one of the most promising directions to reduce carbon dioxide emissions and solve the energy crisis, because electric vehicles (EVs) use lithium-ion batteries (LiBs) as the power source rather than diesel or petrol. Moreover, global EV sales reached 6.75 million units in 2021, 108% more than the sales volume in 2020 (EV-Volumes, 2022). Despite the high power and energy density, high energy efficiency, and relatively long life cycle of LiBs, it is critical to keep a safe, efficient, and stable operating condition for a battery storage system (BSS) to reduce overdesign costs and improve the whole vehicle's efficiency and performance (Mahmoudzadeh Andwari et al., 2017). Thus, the battery management system (BMS) is designed to protect the BSS by the functions of cell monitoring, cell balancing, fault diagnosis, thermal management, charge and discharge control, health management, and state estimation (Han et al., 2020; Wang et al., 2020; Dai et al., 2021). Among these functions, state estimation is the most fundamental task. Furthermore, the battery state-of-charge (SOC) and state-of-health (SOH) are two of the most critical indices used in state estimation.

SOC is a metric that represents the battery's remaining capacity that can be used under the current working state. Although SOC serves the same function for EVs as the fuel gauge for petrol-fueled vehicles, it is not a directly measurable value, making SOC estimation a difficult task in practical applications (Ren et al., 2021a). Therefore, numerous researchers have made significant efforts in the past few decades to develop secure and reliable approaches for SOC estimation. The existing SOC estimation methods are summarized in Fig. 1, and include look-up table methods, the ampere-hour integral method, filter-based methods, observer-based methods, and data-driven methods (Wang et al., 2020; Hu et al., 2019). The look-up table and ampere-hour integral methods are the most straightforward methods. However, they have some serious disadvantages, such as low accuracy in practical applications and poor robustness in dealing with sensor errors. The filter-based and observer-based methods have the advantages of higher estimation accuracy, self-corrective ability, and better robustness in dealing with noises compared with the above two methods (Shrivastava et al., 2019). However, to guarantee sufficient accuracy, practitioners must perform a large number of time-consuming and complicated battery tests to build a well-parameterized model, such as the equivalent circuit model (ECM) or electrochemical model (EM) (Wu et al., 2022). Data-driven methods require limited pre-knowledge about the electrochemical characteristics of LiBs because these methods consider the battery as a black box and focus on building the relationship between the input and output using machine learning (ML) techniques (Tian et al., 2021). Nevertheless, it is easy to encounter overfitting or underfitting problems, which largely depend on the quality of the training datasets and the settings of the algorithm. In addition, the onboard implementation of data-driven methods remains a great challenge.

Because LiBs are extremely complicated physical–chemical systems, their performance is subject to degradation during the long-term operation. It is necessary to know the actual health state of the battery during regular operation to maintain its efficiency, safety, and stability. Therefore, SOH serves as a critical state index for LiBs to evaluate the batteries' degradation (i.e., capacity fade or power decrease). In general, the existing SOH estimation approaches are usually divided into two categories: direct measurements methods (or experimental methods) and indirect analytical methods (Chen et al., 2018a), as depicted in Fig. 2. Direct measurement methods, like the capacity measurement method, internal resistance measurement test, and impedance measurement method, are suitable for laboratory conditions but not practical in real applications. Indirect analytical methods, including model-based and data-driven methods, are usually multi-step methods that first establish a mapping relationship between SOH and measured battery signals, and then estimate the SOH based on real-time measurements in practical applications. The core idea of model-based SOH estimation methods is to build high-fidelity battery models to describe the nonlinearity of LiBs, such as ECM, EM, and the empirical model. The ECM does not consider the internal chemical reactions of LiBs and uses basic electronic components to describe the nonlinearity, while EM focuses on the electrochemical phenomenon inside the LiBs. The basic principle of the empirical model is to build the relationship between the battery SOH and different aging factors without detailed knowledge about battery characteristics. The computation simplicity is the main advantage of the empirical-model-based method. However, to obtain rich and complete data, the experiments are usually time-consuming and costly. As for the data-driven methods, the battery is considered as a black box to estimate SOH without a specific battery model compared to model-based methods. Typical data-driven methods are based on ML techniques, such as artificial neural network

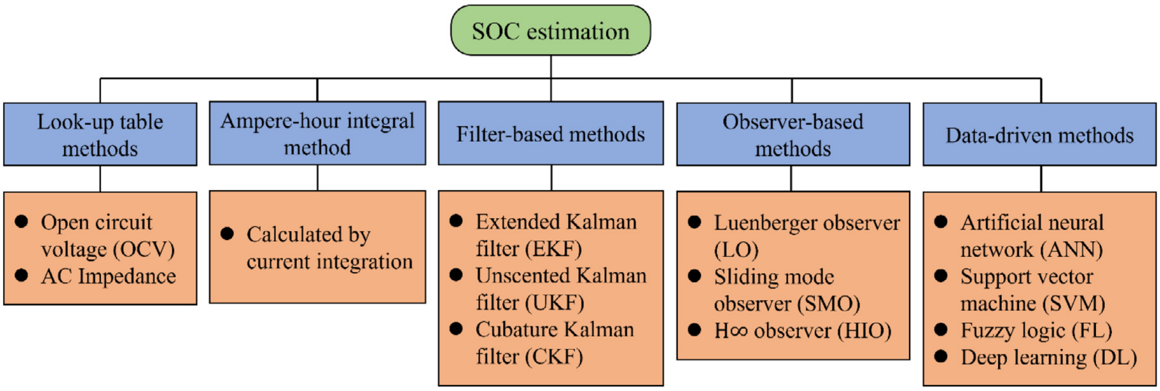
SOC estimation
Look-up table methods
Ampere-hour integral method
Filter-based methods
Observer-based methods
Data-driven methods
● Open circuit voltage (OCV)
● AC Impedance
● Calculated by current integration
● Extended Kalman filter (EKF)
● Unscented Kalman filter (UKF)
● Cubature Kalman filter (CKF)
● Luenberger observer (LO)
● Sliding mode observer (SMO)
● H∞ observer (HIO)
● Artificial neural network (ANN)
● Support vector machine (SVM)
● Fuzzy logic (FL)
● Deep learning (DL)

Fig. 1. Classification of SOC estimation methods.

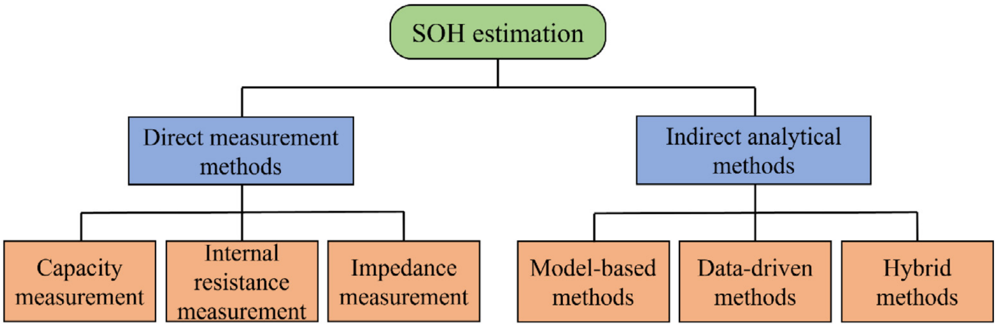

Fig. 2. Classification of SOH estimation methods.

(ANN), support vector machine (SVM), and deep learning (DL) algorithms. The battery is a dynamic system in which multiple states are coupled with each other, similar to the relationship between SOC and SOH. Their coupling relationship allows for the mutual improvement of SOC and SOH estimation accuracy.

Overall, with the development of big data, cloud computing, Internet of Things (IoT), and other emerging techniques, the data-driven methods have been successfully applied in different aspects of BSS and BMS, such as the manufacturing of LiB anodes and cathodes (Niri et al., 2022), material design (Mao et al., 2021b), battery safety (Finegan et al., 2021), fast-charging strategies (Attia et al., 2020; Jiang et al., 2022), and state estimation (Xiong et al., 2018). Because SOC and SOH estimations have always been a research hotspot in the field of BMS, review papers regarding these two topics have been published continually. Table 1 summarizes the main contents of several recent published reviews related to SOC and SOH estimation algorithms. Some of these review articles (Wang et al., 2020; Finegan et al., 2021) mainly focus on the key functions of BMS and simply review SOC and SOH estimation methods. In those reviews that focus on general methods for SOC or SOH estimation (Ge et al., 2021; Wang et al., 2021a; Li et al., 2017; Tian et al., 2020b), the discussion of data-driven methods is not in depth enough, as these articles usually involve the introduction of broad categories of SOC or SOH estimation methods. Even some recent reviews (Li et al., 2019d; Gao and Lu, 2021; Ng et al., 2020; Cui et al., 2022; Liu et al., 2022) focusing on data-driven methods are limited to a few methods or are not covered in depth. For example, one review (Cui et al., 2022) introduces the SOC estimation methods based on the ANN, while the SVM-based methods are not mentioned. Another review (Liu et al., 2022) focuses on the DL algorithms, while other ML algorithms are not introduced. In summary, most reviews mainly focus on the general approaches applied in SOC and SOH estimation. Even though data-driven methods are introduced as one of the general approaches, the details of the data-driven methods, including basic principles, application steps, data used, hyperparameter selection, and comparison, are not explained in depth.

To bridge these research gaps, this paper reviews recently published papers related to data-driven methods applied in SOC and SOH estimation. Note that this review mainly focuses on the four most studied ML data-driven methods, including shallow neural network (NN), DL, SVM, and Gaussian process regression (GPR) methods. The principles of each ML method, the comparison between different ML methods, current challenges, and prospects are comprehensively reviewed. With this information, we aim to provide some inspiration for the development of advanced ML state estimation algorithms. The main contributions of this paper are as follows:

- The basic principles and uniform flowcharts of different categories of ML algorithms, including shallow NN, DL, SVM, and GPR methods, are systematically introduced. In addition, the advantages and disadvantages of each algorithm are given.
- The applications of different ML algorithms for SOC and SOH estimation in the existing papers are comprehensively reviewed and compared. Detailed information, including used datasets, input features, hyperparameter selection, and performance metrics, are summarized in tables for easy reference and comparison.
- Current challenges and prospects are given from four aspects: data quality, structure selection and hyperparameter tuning, hybrid methods, and execution of the ML algorithms.

The remainder of this paper is organized as follows: Section 2 introduces the principles, applications, and comparison of the

Table 1
An overview of the published literature related to battery SOC and SOH estimation.

Ref.	Topic	Main content
Dai et al. (2021)	Battery management strategies	General review of different layers of advanced battery management strategies, including foundation layer, algorithm layer, application layer, and problems and challenge layer, as well as the future trends for next-generation BMS
Shen and Gao (2019)	BMS technologies	General review of the research and development of multiphysics model simulation and multifunction integrated BMS technology, including battery modeling, state estimation, energy equalization, thermal management, fault diagnosis, etc.
Ge et al. (2021)	SOH estimation and RUL prediction	General review of SOH estimation methods, including direct measurements and indirect analysis methods, as well as RUL prediction approaches, including data-driven and hybrid methods.
Wang et al. (2021a)	SOC and SOH estimation	General review of the state-of-the-art online SOC and SOH evaluation methods published within the recent five years, mainly including model-based, data-driven methods, and joint estimation strategy.
Li et al. (2017)	SOC estimation	General review of different issues about SOC, from fundamental principles for definition of SOC to methodologies to estimate SOC.
Tian et al. (2020b)	SOH estimation	General review of aging reasons for LiBs and four categories of SOH estimation methods, including model-based, data-driven, hybrid, and other methods.
Gao and Lu (2021)	Energy storage devices and systems (ESD & ESS)	General review of approaches and applications of ML techniques for ESD, including LiBs, supercapacitors, fuel cell, as well as ESS, including battery ESS, hybrid ESS, grid, and microgrid.
Ng et al. (2020)	SOC and SOH estimation, RUL prediction	General review of the limitations of the equivalent circuit and physics-based models and various ML techniques for battery state prediction, including SOC, SOH, and RUL.
Cui et al. (2022)	SOC estimation	General review of SOC estimation algorithms based on ANN, including feed-forward neural networks and DL networks.
Liu et al. (2022)	SOC estimation	General review of SOC estimation algorithms based on DL algorithms.

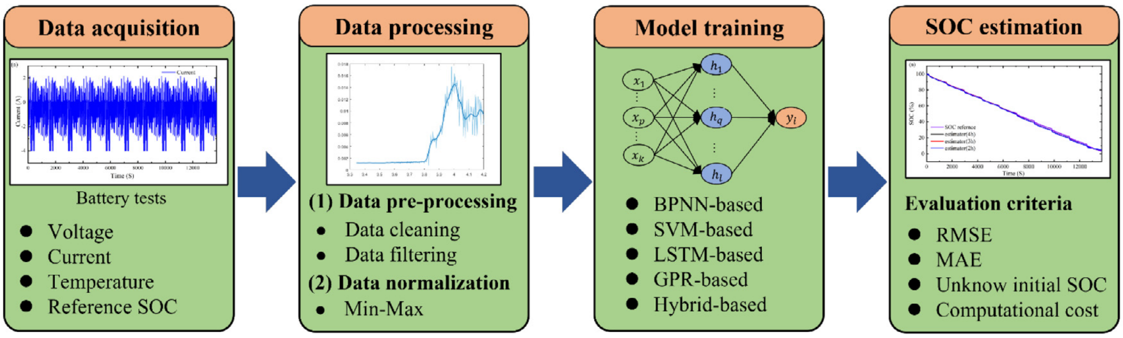

Fig. 3. The procedures for developing ML SOC estimation algorithms.

ML SOC estimation algorithms. Similarly, the principles, applications, and comparison of the ML SOH estimation algorithms are summarized in Section 3. Section 4 describes the challenges and prospects of ML algorithms applied in SOC and SOH estimation. Section 5 provides the conclusion.

2. Review of ML SOC estimation algorithms

The typical procedures of building an ML SOC estimation method are depicted in Fig. 3. Specifically, the first step is data acquisition by means of conducting different types of battery tests in laboratory conditions or collecting raw data from real-world EV operations. Subsequently, raw data must be processed before being used to train ML models. Third, an appropriate ML algorithm is selected, considering model accuracy and complexity. After we fully train and validate the selected algorithm, we can use it to estimate SOC using new data. In the remainder of this section, the applications of different ML algorithms, including shallow NN, DL, SVM, and GPR methods for SOC estimation, are detailed.

2.1. *Shallow neural network*

The ANN, which is inspired by real neurons in the human brain (Jain et al., 1996), is the most attractive and powerful ML algorithm currently. Generally, based on the internal structure, the ANN can be roughly categorized into traditional NN and DL algorithms (Shrestha and Mahmood, 2019). The traditional NN, also called the shallow NN (Gao et al., 2016), only consists of one input layer, one hidden layer, and one output layer. Typical shallow NNs include the backpropagation neural network (BPNN), radial basis function neural network (RBFNN), extreme learning machine (ELM), and wavelet neural network (WNN). In this section, the basic principles and typical applications of various kinds of shallow NNs in SOC estimation are introduced in detail, while the DL algorithm is discussed in the next section.

2.1.1. *Backpropagation neural network*

The feed-forward BPNN is the most classic NN, which only contains one hidden layer, as shown in Fig. 4. The training dataset

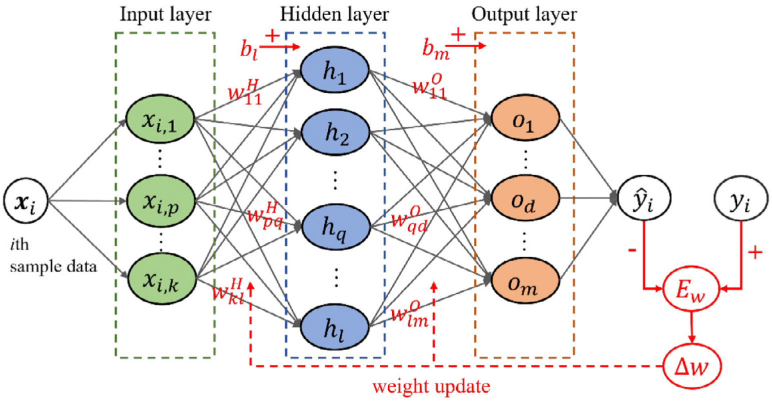

Fig. 4. The structure of the BPNN.

consists of k-dimensional N input vectors $X = (\{x_i\}_{i=1}^{N}, x_i < \mathbb{R}^k)$ and corresponding m-dimensional output vectors $Y = (\{y_i\}_{i=1}^{N}, y_i < \mathbb{R}^m)$. The feed-forward process, which describes the propagation process of the information of the input neurons, is explained as follows. First, the outputs of the hidden layer neurons are expressed in Eqs. (1) and (2):

$$net_q = \sum_{p=1}^{k} w_{pq}^{H} x_{i,p} + b_q, q = 1, \ldots, l \tag{1}$$

$$f_h\left(net_q\right) = \frac{1}{1 + e^{-(net_q)}} \tag{2}$$

where $x_{i,p}$ is the pth feature of the ith data sample in the input layer; w_{pq}^{H} is the weight connecting the pth input neuron and the qth hidden neuron; b_q is the bias of the qth hidden neuron; net_q is the qth component in the input vector of the hidden layer; $f_h(\cdot)$ is the activation function of the hidden layer neurons where the sigmoid function is the most widely used one as shown in Eq. (2); and l is the number of neurons in the hidden layer. Other popular activation functions, including hyperbolic tangent (tanh), rectified linear node (ReLU), and leaky ReLU, can be selected according to the different demands of the specific application. Second, from the hidden layer to the output layer, the expressions are as follows:

$$net_d = \sum_{q=1}^{l} w_{qd}^{O} h_q + b_d, d = 1, \ldots, m \tag{3}$$

$$f_o\left(net_d\right) = \frac{1}{1 + e^{-(net_d)}} \tag{4}$$

where h_q is the qth output of the hidden layer, net_d is the dth component of the output layer, w_{qd}^{O} is the weight connecting the qth hidden neuron and the dth output neuron, $f_o(\cdot)$ is the activation function of the output layer neurons, and m is the number of neurons in the output layer. Note that when SOC is the only output of the model, the value of m is 1.

The training process of BPNN aims to find the optimal values of the weights and biases to minimize the loss function. So far, the most widely used algorithm is backpropagation. The process of the backpropagation algorithm can be divided into two phases: the forward phase and the backward phase. Specifically, the forward phase is the same as in Eqs. (1) to (4) and calculates the estimated output. The backward phase uses the following equations to update the weights in every iteration by computing the gradient of an arbitrary loss function (E_w):

$$\frac{\partial E_w}{\partial w_{ij}} = \frac{\partial E_w}{\partial s_i} \frac{\partial s_i}{\partial net_i} \frac{\partial net_i}{\partial w_{ij}} \tag{5}$$

where w_{ij} represents the weight, such as w_{pq}^{H} and w_{qd}^{O}; s_i represents the output of the neuron, such as $f_h\left(net_q\right)$ and $f_o\left(net_d\right)$; and net_i is the input of the neuron, such as net_q and net_d. Once the partial derivative for each weight is known, the weights at the current step $(n+1)$ can be updated using the previous step (n), as follows:

$$w_{ij(n+1)} = w_{ij(n)} - \alpha \frac{\partial E_w}{\partial w_{ij(n)}} \tag{6}$$

where α is the learning rate. Depending on the specific settings of the backpropagation algorithm, different conditions, such as the number of iterations and the improvement rate of the error, can be regarded as the stopping criterion of the learning process.

Because the current, voltage, and temperature signals can be directly measured, and these signals are highly related to the battery SOC, numerous works of literature utilize current, voltage, and temperature as inputs to train the BPNN. However, in addition to current, terminal voltage, and temperature, other factors can influence the characteristics of LiBs, like the polarization states, SOH, and internal resistance. Considering more factors as inputs can theoretically improve the performance of the algorithm. Hu and Wang (2019) added the internal resistance as an input feature along with the current, voltage, and temperature to train a three-layer BPNN to estimate SOC. The experimental results show that considering the influence of internal resistance could improve the estimation accuracy. In another work (Chen et al., 2019), the polarization state was selected as an essential input feature. Generally, the number of the hidden layer neurons directly influences the overall performance of the ANN. Hannan et al. (2018) proposed an improved BPNN with the backtracking search algorithm (BSA). The BSA optimization was employed to improve the accuracy and robustness of the BPNN model by finding the optimal value of hidden layer neurons and learning rate under different driving cycles and temperatures. Compared with the trial-and-error method applied in the conventional BPNN, the BSA-BPNN could achieve a good trade-off between the desired accuracy and model complexity. Additionally, the proposed model could achieve higher estimation accuracy under different operations and temperatures in comparison to the RBFNN-BSA and ELM-BSA.

2.1.2. Radial basis function neural network

The RBFNN is another classic NN. Unlike the BPNN using the nonlinear monotonic single-valued activation function (Vidal et al., 2020), the RBFNN usually uses the radial basis function (RBF) as the activation function where the Gaussian kernel function is the most widely used one. The typical structure of RBFNN

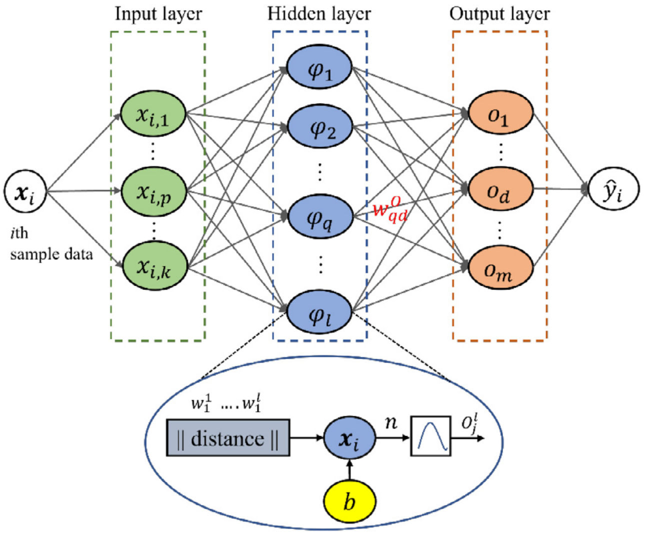

Fig. 5. The structure of the RBFNN.

is shown in Fig. 5. The outputs of the qth hidden layer neuron and the dth output can be formulated as follows:

$$\varphi_q(x) = G\left(\left|\left|x - w_q\right|\right|\right) = \exp\left(-\frac{\left|\left|x - w_q\right|\right|^2}{\sigma_q^2}\right), \quad q = 1, \ldots, l \tag{7}$$

$$o_d = \sum_{q=1}^{l} w^O_{qd}\varphi_q, \ d = 1, \ldots, m \tag{8}$$

where $\varphi_q(x)$ and o_d are the outputs of the qth hidden layer neuron and the dth output layer neuron, respectively; x is the input vector; w_q and σ_q are the center and width of qth hidden neuron in the Gaussian kernel function, respectively; l and m are the numbers of hidden neurons and output neurons, respectively; and $\| \sim \|$ means the Euclidean distance between the input vector and centroid vector. Note that when the SOC is the only output of the model, the value of m is 1. Fast training and good capability of interpolation are two advantages of the RBFNN.

Sun et al. (2020) compared the so-called direct and indirect methods among the ML SOC estimation methods, in which the direct method meant that the SOC was used as the network output, while the indirect method represented the method that used voltage as the network output. Specifically, the indirect method was a combination of an RBFNN and the unscented Kalman filter (UKF) in which the RBFNN was used to train a battery model, and the UKF was used to estimate the SOC. The direct method was a nonlinear autoregressive exogenous neural network (NARXNN) (introduced in Section 2.2.2.1). The comparison results show that both methods could achieve accurate SOC estimation, but the direct NARXNN method was incapable of dealing with sensor noises, while the RBFNN-UKF method was promising in onboard BMS applications owing to its superiority in uncertainty augmentation and feedback compensation. Instead of comparing the direct method and indirect method, Zhang et al. (2020b) proposed a compact RBFNN method to directly estimate the SOC of the battery pack, which utilized a fast recursive algorithm (FRA) to select strongly correlated input variables to improve accuracy. After selection, the inputs consisted of the average voltage of the battery pack; the mean voltage, current, and temperature of the past ten measurements; and the SOC value at timestep k-1. Furthermore, the FRA was used to determine the structure of the proposed RBFNN, while Particle Swarm Optimization (PSO) was used to optimize the kernel parameters (w and σ). The comparison results show that although the improved RBFNN method took more training time than the conventional RBFNN, the root mean square error (RMSE) of the improved RBFNN method was half that of the conventional method. In order to quantify the number of hidden layer neurons that is adequate to fit the nonlinear relationship, Chang (2013) utilized an orthogonal least-squares (OLS) algorithm to determine the optimal number of neurons in the hidden layer. With the confirmed structure, the centers, widths, and weights of the RBFNN were trained using an adaptive genetic algorithm (AGA) rather than the conventional stochastic gradient descent approach. Compared with the BPNN and Coulomb integration method under several different current discharging experiments, the proposed RBFNN-based hybrid SOC estimation method performed well. However, charging and dynamic conditions were not compared.

2.1.3. Extreme learning machine

Compared with the BPNN and RBFNN, ELM has several advantages, including lower computation complexity, a faster convergence speed, a simpler learning process, and better generalization performance (Ding et al., 2015). The structure of ELM is similar to that of BPNN and RBFNN, which has three layers, including one input layer, one hidden layer, and one output layer, as shown in Fig. 6. However, what makes ELM different from BPNN and RBFNN is that it has a more concise learning process that does not use the error backpropagation algorithm to update weights and bias. The whole learning process can be divided into three steps:

Step (1): The input features are selected arbitrarily. Then, the input weight matrix and the hidden layer bias are assigned randomly. The number of hidden neurons is determined flexibly to achieve acceptable accuracy.

Step (2): The feed-forward propagation algorithm is used to calculate the output of the hidden layer and output layer as follows:

$$net_q = \sum_{p=1}^{k} w^H_{pq} x_{i,p} + b_q, \quad q = 1, \ldots, l \tag{9}$$

$$f_h\left(net_q\right) = \frac{1}{1 + e^{-(net_q)}} \tag{10}$$

$$o_d = \sum_{q=1}^{l} w^O_{qd} h_q, \ d = 1, \ldots, m \tag{11}$$

where $x_{i,p}$ is the pth feature of the ith data sample in the input layer; w^H_{pq} is the weight connecting the pth input neuron and the qth hidden neuron; b_q is the bias of the qth hidden neuron; net_q is the qth component in the input vector of the hidden layer; $f_h(\cdot)$ is the activation function of the hidden layer neurons; h_q is the qth output of the hidden layer; w^O_{qd} is the weight connecting the qth hidden neuron and the dth output neuron; o_d is the dth component of the output layer; and l and m are the numbers of neurons in the hidden layer and output layer, respectively. Note that when the SOC is the only output of the model, the value of m is 1.

Step (3): The output weights can be analytically determined through the generalized inverse operation of the hidden layer matrix. Therefore, the least squares solution is used to calculate the output weight as follows:

$$W^O = H^+ \cdot Y = (H^T H)^{-1} H^T \cdot Y \tag{12}$$

where W^O is the weight matrix connecting the hidden layer and the output layer; Y is the output matrix; H is the output matrix of the hidden layer; and H^+ is the Moore–Penrose generalized inverse of H. It is evident from the above steps that there is no

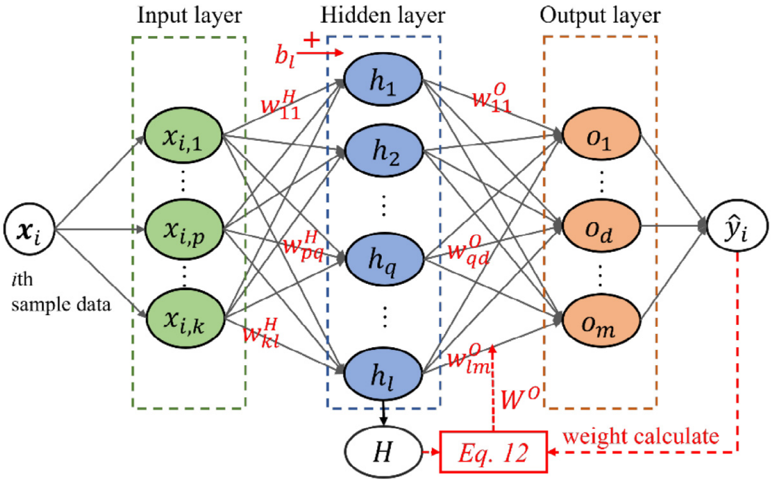

Fig. 6. The structure of ELM.

iteration in the ELM algorithm, because it randomly assigns the input weights and hidden layer bias and then utilizes the Moore–Penrose generalized inverse of the pseudoinverse matrix to train the model (Bin Huang et al., 2006).

To overcome the difficulty of limited training samples in the initial training stage, Chin and Gao (2018) utilized the adaptive online sequential extreme learning machine (AOS-ELM) to estimate the SOC of the battery pack. Using the AOS-ELM could avoid the time-consuming process of obtaining sufficient data in the initial stage of model training. The performance of the AOS-ELM was compared with the extended Kalman filter (EKF)-based methods and the ELM-based methods using a 12-cell pack, and the RMSE of the proposed method was 0.0529%. In another study (Hossain Lipu et al., 2019), an ELM-based SOC estimation algorithm that used voltage, current, and temperature as inputs was compared with a BPNN and RBFNN at different ambient temperatures (e.g., 25 °C and 45 °C) using different driving cycles. The main contribution of the proposed method is that the gravitational search algorithm (GSA) is employed to improve the computational intelligence of ELM by searching for the optimal value of hidden neurons. Compared with PSO, the GSA performed well in obtaining the lowest fitness function, thus ensuring a higher accuracy for SOC estimation. Specifically, the RMSE of the proposed ELM-based algorithm was reduced by 21.8% and 29.8% compared to the BPNN and RBFNN algorithms at 45 °C under the Beijing Dynamic Stress Test (BJDST) cycle, respectively.

2.1.4. Wavelet neural network

Another feed-forward NN applied in SOC estimation is a WNN, which integrates the wavelet decomposition property with the nonlinear relationship approximation ability of ANNs to avoid the local optimum problem and improve the convergence rate (Chen and Liang, 2017). Moreover, unlike the three NNs mentioned above, WNN uses the Morlet wavelet function as the activation function of the hidden layer neuron. The expression is as follows:

$$\psi\,(x) = \cos(1.75x)\exp(-0.5x^2) \tag{13}$$

Therefore, the feed-forward propagation process is as follows:

First, the output of the hidden layer is formulated as

$$h_q = \psi_{a_q,b_q}\left(net_q\right) \tag{14}$$

$$net_q = \frac{\sum_{p=1}^{k} w^H_{pq} x_{i,p} - b_q}{a_q}, \quad q = 1, \ldots, l \tag{15}$$

where a_q and b_q are the wavelet dilation and translator parameters from the input layer to the hidden layer, respectively.

Second, the output of the output layer is calculated as

$$o_d = \psi *_{a_d,b_d}\left(net_d\right) \tag{16}$$

$$net_d = \frac{\sum_{q=1}^{l} w^O_{qd} h_q - b_d}{a_d}, \quad d = 1, \ldots, m \tag{17}$$

where a_d and b_d are the wavelet dilation and translator parameters from the hidden layer to the output layer, respectively.

To estimate SOC, Zhou et al. (2013) proposed an adaptive WNN (AWNN) containing 12 hidden layer neurons. The learning algorithm to gain the parameters of the AWNN is the widely used steepest descent method. Compared with the conventional BPNN, the proposed method had a faster convergence speed and higher accuracy. In addition to the typical steepest descent algorithm, Cui et al. (2018) proposed a hybrid WNN model that applies the discrete wavelet transform (DWT) method to preprocess the input data and the Levenberg–Marquardt (L–M) algorithm to train the model instead of using the steepest descent algorithm. The proposed method was thoroughly compared with other SOC estimation methods such as BPNN, L-M-based BPNN, L-M-based WNN, DWT-based LMBPNN, and EKF in terms of the mean absolute error (MAE), maximum error, and training time, yielding values of 0.59%, 3.13%, and 13.7539 s, respectively. Furthermore, the proposed method performed better under unknown EV driving cycles and noise tests owing to the DWT method. In order to further improve the L-M algorithm, Xia et al. (2018) utilized PSO to make the L-M-based WNN more suitable for practical applications. The optimal process from the standard WNN to PSO-based L–M algorithm optimized multi-hidden-layer WNN was comprehensively given and compared. Similarly, the proposed method was validated under untrained driving cycles and measurement noise tests, demonstrating its excellent robustness against measurement noises.

2.2. Deep learning algorithm

The development of big data and cloud computation platforms makes it possible to apply DL algorithms in the state estimation of BMS. The so-called DL algorithms employ the adjective "deep" to describe the use of multiple hidden layers or other different types of architectures (Schmidhuber, 2015). Compared with the shallow NNs, DL algorithms can map complex and nonlinear functions efficiently, and usually have higher accuracy. According to the architectures of different DL algorithms, in this paper, typical

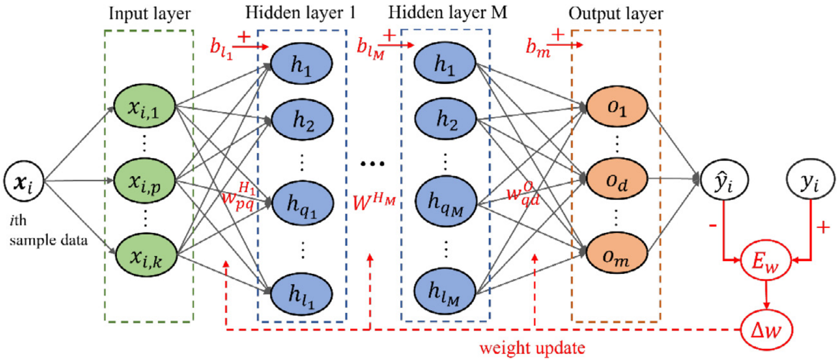

Fig. 7. The structure of the DNN.

DL algorithms are divided into three categories: the deep neural network (DNN), the recurrent neural network (RNN), and the convolutional neural network (CNN). In this section, the basic principles and applications of these three algorithms for SOC estimation are explained in detail.

2.2.1. Deep neural network

A DNN can be regarded as a direct extension of a BPNN, but it contains multiple hidden layers, as depicted in Fig. 7. Different hidden layers may have a different number of hidden neurons, represented by $l_1, \ldots, l_M$, respectively. Additionally, the learning algorithm and the principles of information transformation of a DNN are the same as those of a BPNN.

In order to solve the common over-fitting or under-fitting problems for data-driven methods in SOC estimation, He et al. (2014) built a multilayer feed-forward NN. One of the contributions of this paper was that the number of the input vectors and the structure of the NN were determined by a constructive method, which meant that the NN was constructed starting from the simplest version with a low-dimension input vector to a complicated structure with a high-dimension input vector. Although this constructive method is essentially a trial-and-error method, it represents an important step in training DNNs. Another innovation was that the outputs of the DNN were filtered by the UKF to improve its accuracy further. After noise reduction, the RMSE of the proposed method was 1.4%, while the maximum error was only 1.9%, smaller than 2% under Federal Urban Driving Schedule (FUDS) driving cycles. However, combining UKF would significantly increase the execution time of the algorithm. Therefore, Chemali et al. (2018) were committed to building a DNN-based SOC estimation method without the Kalman filter (KF) that can achieve sufficient accuracy. A comparison of the computational times between a well-trained two-layer DNN and an EKF-based method were conducted, yielding values of 0.07 s and 0.66 s, respectively. Obviously, the execution time of the trained DNN is much shorter than that of the EKF-based algorithm. To maintain a better trade-off between the computational efficiency and memory requirements, the authors selected the voltage, temperature, current, average current, and voltage as the input features of the proposed DNN. After training the model under different driving cycles, they proved that 400 historical data points could achieve better performance than 100 historical data points. In addition, the robustness of the trained DNN was verified against measurement offsets, gains, and noise under Highway Fuel Economy Driving Schedule (HWFET) and Supplemental Federal Test Procedure Driving Schedule (US06) driving cycles at 25 °C. Motivated by Chemali, How et al. (2020) developed a series of DNN models with different numbers of hidden layers to investigate the optimal number of hidden layers on unseen driving cycles. The details of the hyperparameter selection for the proposed DNN are provided in Table 2, which is worth advocating to guarantee reproducibility (Campestrini et al., 2016). By means of verifying a series of DNN models with a different number of hidden layers under the training dataset (e.g., dynamic stress test (DST) driving cycles) and testing dataset (e.g., unseen FUDS, US06, and BJDST driving cycles), they validated a four-hidden-layer DNN with 64 neurons per hidden layer as the best-performing model. However, apart from the number of hidden layers, the number of hidden neurons in every hidden layer is also important for the computational burden and accuracy. As concluded in Chemali et al. (2018), an increase in the number of hidden layers neurons results in the growth of floating-point operation per second (FLOPs) exponential, not to mention an increase in the numbers of hidden layers and hidden neurons at the same time. Therefore, all other hyperparameters need to be optimized further, especially the number of hidden neurons.

Table 2
Hyperparameter selection of the proposed DNN in How et al. (2020).

Hyperparameters	Selection options
Number of hidden neurons	64 per hidden layer
Number of hidden layers	Variable
Learning rate	Adaptive learning rate
Neuron initialization	He. Initialization
Optimization algorithm	Adam
Dropouts	No
Batch size	256
Batch normalization	After non-linearity
Non-linearity	ReLU for hidden layer neurons Sigmoid for output layer neurons

2.2.2. Recurrent neural network

With respect to the structure only, there are some similarities between the RNN and BPNN because the architecture of RNN is derived from a feed-forward NN to some extent. Historically, typical RNNs have two structures: the Jordan NN (Jordan, 1986) and the Elman NN (Elman, 1990). The main differences between these two NNs are that the context units of the Jordan NN are from the output layer, while the context units of the Elman NN are from the hidden layer. In other words, the hidden layer is fed from the output layer instead of the hidden layer in the Jordan NN. In this section, the basic principles and recent publications of RNNs for SOC estimation are reviewed in detail.

2.2.2.1. Nonlinear autoregression with exogenous input neural network. Nonlinear autoregression with exogenous input neural network (NARXNN) can learn to predict one time series by giving the past values of the same time series, the feedback input, and another time series called the external or exogenous time series. The structure of the NARXNN is depicted in Fig. 8. According to the feedback mechanism of NARXNN, it can be regarded as

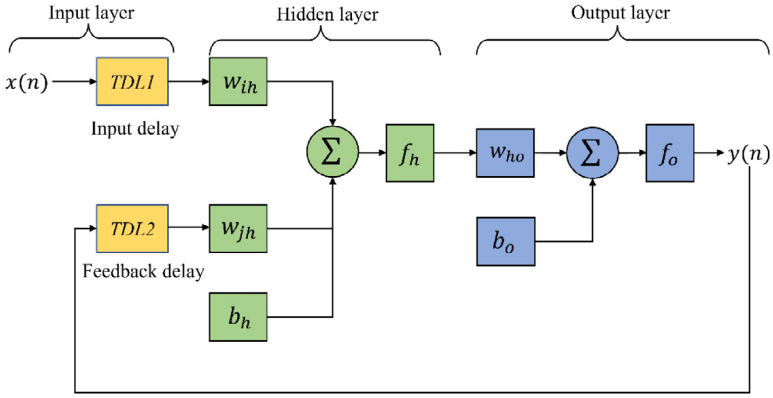

Fig. 8. The structure of the NARXNN.

a variant of the Jordan NN. The expression of NARXNN is as follows:

$$y\left(n\right)=f_o[b_o+\sum_{h=1}^{l}w_{ho}f_h(b_h+\sum_{i=0}^{d_u}w_{ih}x(n-i)+\sum_{j=0}^{d_y}w_{jh}y(n-j))] \quad (18)$$

where $f_o(\cdot)$ and $f_h(\cdot)$ are activation functions of the output layer and hidden layer, respectively; $[w_{ih}, w_{jh}, w_{ho}]$ and $[b_h, b_o]$ are weights and biases between the corresponding layers, respectively; d_u and d_y represent the input and output delays, respectively; and l is the number of hidden neurons.

Owing to the feedback mechanism, the previous information of inputs and outputs can help to improve the accuracy in time-sequence problems, which is extremely helpful in SOC estimation and has gained interest in this field. Jiménez-Bermejo et al. (2018) proposed a dynamic NN model based on the NARX structure for SOC estimation. Note that the proposed model was trained offline using a real EV driving cycles dataset extracted from 54 different journeys with 800 km under different unknown initial SOC. The proposed method could achieve accurate results in all situations. In one study (Lipu et al., 2018; Hannan et al., 2020), Lipu utilized the lighting search algorithm (LSA) to search for the best values of the input delay, output delay, and hidden layer neurons. In another study (Hossain Lipu et al., 2018), they used PSO to do the same optimization work. Both optimization algorithms, namely the PSO and LSA algorithms, improved the performances of NARXNN in SOC estimation in terms of accuracy and robustness, but the comparison between these two optimization algorithms was deficient. Using heuristic optimization algorithms to determine the hyperparameters is considered a promising way to avoid the time-consuming and difficult hyperparameter tuning process. Because using NARXNN as a direct method to estimate SOC may produce noisy estimation results owing to the measurement noises (Sun et al., 2020), Qin et al. (2019) combined the NARXNN and UKF algorithm to improve the adaptability, efficiency, and robustness of the overall SOC estimation method, where the inputs of the UKF algorithm were the measured current and the output of the NARXNN, similar to He et al. (2014). It is well known that the RNNs usually encounter gradient vanishing or gradient explosion problems. Wang et al. (2021b) used a moving window method that is based on the traditional finite element principle to solve the gradient disappearance and explosion problem, especially when the voltage and current change periodically. The size of the moving window can be changed with the change in the discharge rate. Thus, the improved NARXNN can expand or reduce the window at an appropriate time to change the amount of historical information. By means of this moving window method, the gradient vanishing or explosion problems were avoided, and the estimation accuracy was improved.

2.2.2.2. Gated recurrent neural network. The Elman NN is another popular variant of the RNN. The typical structure of the Elman NN is shown in Fig. 9. Specifically, the function of the context units is to store the outputs of the hidden layer during the propagation. Then, during the next time step, the hidden layer neurons not only receive information from the input layer but also from the context units. Owing to these additional context units, the Elman NN can deal with sequential data powerfully. The output of the hidden layer is expressed as:

$$H\left(t\right)=f_h(W^H\left(t\right)x\left(t\right)+V\left(t-1\right)H\left(t-1\right)+b_l(t)) \quad (19)$$

where t is the current time step; H is the output vector of the hidden layer; x is the input vector; W^H is the weight matrix connecting the input layer and hidden layer; V is the weight matrix connecting the context layer and hidden layer; b_l is the bias vector; l is the number of hidden neurons; and $f_h(\cdot)$ is the activation function. The weights and bias can be optimized by means of gradient-descent-based algorithms.

Although a conventional RNN can use the previous information to solve nonlinear problems more efficiently, the performance of RNN is limited owing to the exploding or vanishing gradient during training. Therefore, some variants of RNN have been created to solve this limitation, such as the long–short term memory (LSTM) network, bidirectional LSTM (BiLSTM) network, and the gated recurrent unit (GRU) network. Because these kinds of RNNs are structured with gates, they are collectively referred to as gated RNNs in this paper.

The LSTM is the most widely used variant of the RNN, which adds memory cells that can carry information across multiple time steps. According to Fig. 10(a), the computation process of LSTM can be expressed as follows:

$$\begin{aligned}
f_t &= \sigma\left(W_f x_t + U_f H_{t-1} + b_f\right)\\
i_t &= \sigma\left(W_i x_t + U_i H_{t-1} + b_i\right)\\
o_t &= \sigma\left(W_o x_t + U_0 H_{t-1} + b_o\right)\\
c_t &= f_t \odot c_{t-1} + i_t \odot tanh\left(W_c x_t + U_C H_{t-1} + b_c\right)\\
H_t &= o_t \odot tanh\left(c_t\right)
\end{aligned} \quad (20)$$

where x_t is the input of the LSTM unit layer at time step t; H_{t-1} is the hidden stat; W and U are the weight matrices; b is the bias; i_t, f_t, o_t, and c_t are the input gate, forget gate, output gate, and memory cell, respectively; σ is the sigmoid activation function; and $\odot$ represents the element product. The characteristics of the LSTM are the gates with distinct functions. Specifically, the input gate (i_t) determines what new information can pass the memory cell, while the forget gate (f_t) determines which information should be ignored from the previous memory cell (c_{t-1}). Moreover, the output gate is responsible for computing the

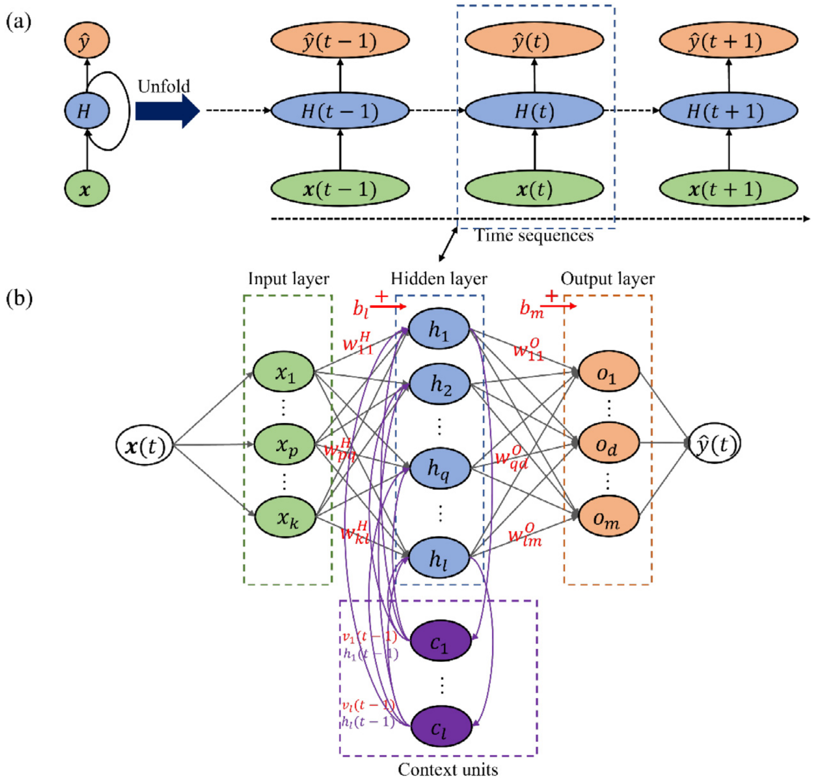

Fig. 9. (a) The time-sequence structure of the Elman NN, (b) The structure of the Elman NN.

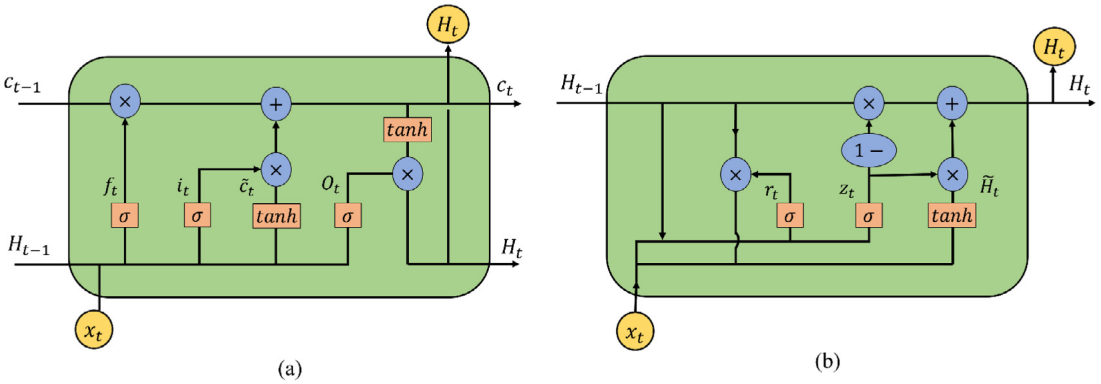

Fig. 10. The memory cell structure of RNN (a) LSTM, (b) GRU.

outputs and the hidden neurons. For more details, refer to the literature (Understanding LSTM Networks, 2021).

Xu et al. (2020) utilized LSTM to learn the relationship between SOC and OCV based on the charging–discharging data considering the hysteresis characteristics of LFP. Then, the proposed model was combined with the Sigma-point Kalman filter (SPKF) to estimate SOC. In one study (Ma et al., 2021), Ma et al. developed an LSTM model, that was trained and validated on public datasets provided by Phillip Kollmeyer (Kollmeyer, 2018), to estimate SOC and state of energy (SOE) simultaneously. The proposed approach was fully verified with different driving cycles and materials under various temperature conditions. In addition, the results show that the RMSE in terms of SOC estimation dropped from 1.8% to 1% when the depth of time increased from 10 s to 50 s but remained stable when the depth of time continued to increase. To match the data characteristics of LiBs with the network topology, the authors optimized the key parameters of LSTM, including the number of hidden units, learning rate, and iteration times, by the PSO algorithm in Ren et al. (2021b). The anti-interference ability of the developed PSO-LSTM algorithm was validated under two different driving cycles, the constant current test and the Urban Dynamometer Driving Schedule (UDDS), both with random noise. The results show that the proposed algorithm outperformed the other algorithms, including EKF and LSTM without PSO, in all cases. Instead of only using LSTM to estimate SOC, researchers have made contributions to improve the performance of LSTM by means of combining other ML algorithms. In one study (Fasahat and Manthouri, 2020), the autoencoder NN, which had one hidden layer with 22 hidden neurons, was utilized to extract the features. Subsequently, the outputs of the autoencoder NN were considered as the inputs of the LSTM. To reduce the complexity of the hyperparameter tuning process of ML algorithms and avoid the tedious battery modeling process of model-based algorithms, Tian et al. (2020a) applied adaptive cubature Kalman filter (ACKF) to smooth the outputs of the LSTM, thus achieving a more accurate and stable

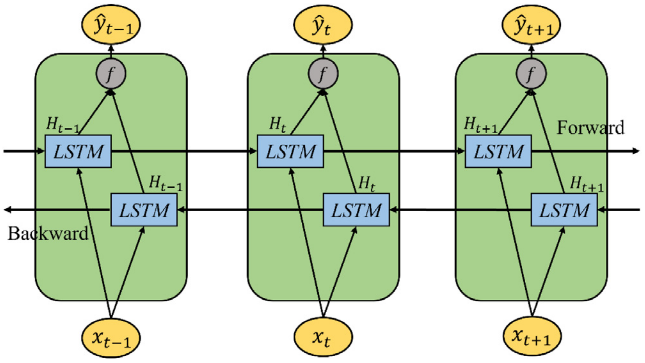

Fig. 11. The structure of the BiLSTM.

SOC estimation. The results show that the RMSE of the LSTM-based algorithm decreased from 3.2% to 1.2% at 30 °C under FUDS driving cycles when combined with ACKF. To tackle the variation of temperature and degradation, researchers utilized the transfer learning and rolling learning algorithms to optimize the LSTM-based SOC estimation algorithm (Liu et al., 2021). In addition, the computation burden can be dramatically lessened when conducting new battery SOC estimation. The BiLSTM, which generally consists of the forward and backward "unidirectional" LSTM layers, can be regarded as a bidirectional version of the LSTM, as shown in Fig. 11. In one study (Bian et al., 2020), multiple parameters that affect the performance of a stacked BiLSTM, including the number of hidden neurons, different merge mode, and recurrent blocks, were compared and analyzed in detail to find the best structure. Two stacked BiLSTM layers of the model were finally confirmed under two public datasets and had the best estimation accuracy. However, the total learnable parameters of the proposed structure were over 13 000, making the network harder to train and more prone to overfitting.

The GRU is another variant of the RNN. Compared with LSTM, the GRU only uses one gate unit to simultaneously control the information flow and the decision to update the state. Therefore, the GRU exhibits a simple structure, needs fewer parameters, and has been proved to outperform LSTM on smaller datasets (Chung et al., 2014). Theoretically, the accuracy of GRU is usually lower than that of LSTM but still in an acceptable bound. The equations of GRU can be expressed as follows.

$$\begin{aligned} z_t &= \sigma\left(W_z x_t + U_z H_{t-1} + b_z\right) \\ r_t &= \sigma\left(W_r x_t + U_r H_{t-1} + b_r\right) \\ \widetilde{H}_t &= tanh\left(W_h x_t + U_h\left(H_{t-1} \odot r_t\right) + b_h\right) \\ H_t &= z_t \odot \widetilde{H}_t + \left(1 - z_t\right) \odot H_{t-1} \end{aligned} \tag{21}$$

where z_t and r_t are the gate updating and resetting vectors, respectively; $\widetilde{H}_t$ is the candidate state; σ is the sigmoid activation function; W and U are the weight matrices; b is the bias; and $\odot$ represents the element product. The comparison between the LSTM and GRU is shown in Fig. 10.

In one study (Li et al., 2019c), Li et al. trained a GRU-RNN using the whole training datasets at 0 °C, 10 °C, and 25 °C to prove that the proposed model can achieve reasonable SOC estimation accuracy under various operating conditions with only a set of network parameters. The statistical results demonstrate that the MAEs were all less than 1.5% under any operating conditions. Note that not only two public datasets (the Panasonic NCR18650PF (Kollmeyer, 2018) and the Samsung 18 650-20R datasets from the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland University of Maryland, 2021) but also the self-made datasets (e.g., high-rate pulse discharge conditions) were used to train and evaluate the proposed method. In addition, the influences of different hyperparameters (e.g., time step and iteration) and the size of training data on the performance of the GRU-RNN were fully analyzed. A more comprehensive analysis of the impact of different network parameters (e.g., epoch number, hidden neurons, GRU layers) on SOC estimation performance was demonstrated in another study (Yang et al., 2019a). In addition to the hyperparameter selection, the optimization algorithm is equally important for the training phase in terms of stability, convergence rate, and accuracy. Xiao et al. (2019) proposed an ensemble optimization algorithm that combines the Nadam and AdaMax algorithms as a backpropagation optimization algorithm for parameter self-learning. Compared with the LSTM-Adam and GRU-Adam, which utilize the most commonly used optimization method (Adam Kingma and Ba, 2014), the proposed GRU-Ensemble algorithm has a simpler structure and few parameters compared with LSTM-Adam, so it achieved a more efficient and rapid training phase compared with GRU-Adam.

2.2.3. Convolutional neural network

Another popular DL algorithm is the CNN, which is widely used in computer vision and image recognition (Shrestha and Mahmood, 2019). More researchers have applied the CNN in the field of state estimation of LiBs. The typical structure of a CNN is several stacks of convolutional layers and pooling layers, fully connected layers, and an output layer, as shown in Fig. 12. Unlike the fully connected network, in the CNN, neurons in each convolution layer are not connected to all neurons. This characteristic is called sparse connectivity, which is performed by sliding a filter (e.g., a weights matrix) over the input space. Specifically, in the convolution layers, the subsets of the input space conduct element-wise multiplication with the filter, and this calculation process is called convolution. The size of the filter, the number of filters, and the step when sliding the filter are three parameters that determine the convolution process. Owing to the existence of the convolution process, the learning speed can be faster, and the memory requirements can be reduced. Thereafter, the pooling layers are used to reduce the size of feature maps further. Two commonly used methods are max pooling and average pooling. Finally, as the name suggested, the neurons between the fully connected layers are fully connected, similarly to how the shallow NNs are connected. Usually, the ReLU activation function is used in a CNN.

The CNN and RNN are usually combined to estimate SOC for their advantages, namely the automatic feature extraction capabilities of a CNN and the time-sequences prediction capabilities of an RNN. In one study (Song et al., 2019), a hybrid CNN-LSTM network was built to estimate SOC, where the one-dimensional (1D) convolution was utilized to capture the spatial features within the measured variables while the LSTM was used to obtain the features between the current output and past inputs. In another work (Huang et al., 2019), the same research team proposed a hybrid CNN-GRU network. However, unlike the work in Song et al. (2019), they did not directly use the average current and voltage as inputs. Instead, they used the 1D convolution layer to extract similar information adaptively. Therefore, the input features of the hybrid network were the measured voltage, current, and temperature. Furthermore, the hyperparameter selection, including the size of the convolutional layer, the number of GRU layers, and the number of neurons in the GRU layers, were determined by comparing different structures. The effectiveness and superiority of the proposed hybrid network were verified by comparing it with other ML algorithms (e.g., GRU, SVM, and ELM) under FUDS and DST driving cycles.

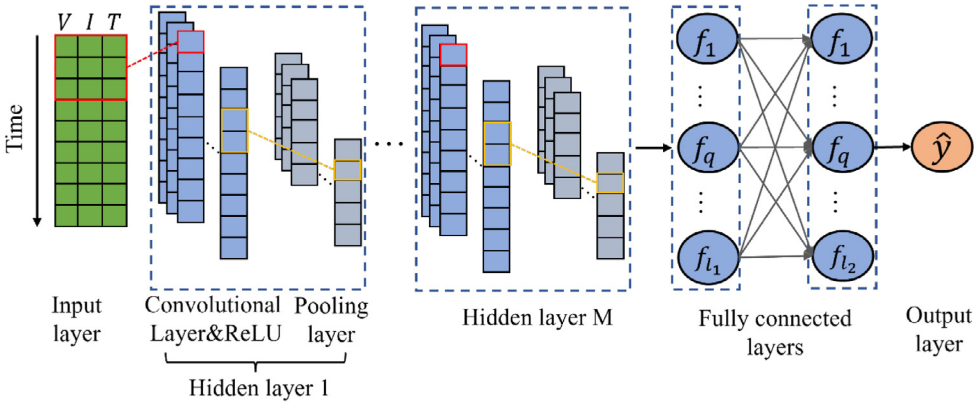

Fig. 12. The structure of the CNN (Crocioni et al., 2020).

In addition, a CNN can be used alone to estimate SOC. Bhattacharjee et al. (2021) used a 1D CNN alone to estimate SOC for xEV batteries for the first time. Unlike the standard two-dimensional (2D) CNN, where the filters stride across both the spatial dimensions of an image, filters in 1D CNN layers slide only in one dimension, the temporal dimension, as shown in Fig. 12. The proposed 1D CNN uses the measured voltage, current, and temperatures from timestep $k-t_w+1$ to k as inputs and predicts the SOC at timestep k. Compared with state-of-the-art SOC estimation methods, the proposed model had better performance and less offline training time. In another work (Hannan et al., 2021), Hannan et al. proposed a fully convolutional network (FCN) that consists of four temporal convolutions to estimate SOC. Instead of using a fully connected layer, they used a global average pooling (GAP) to avoid overfitting. Furthermore, learning rate optimization strategies were employed to improve the error rate of the model. According to the results, the model could achieve an RMSE of 0.85% and an MAE or 0.7% at room temperature.

2.3. Support vector machine

SVM, proposed by Vapnik (1998), is another popular and widely used ML tool suitable for both classification and regression problems, especially for small datasets. The SVM was initially proposed to solve classification problems, and then they were generalized to solve regression problems, called support machine regression (SVR) (Fletcher, 2009). Generally, many regression problems cannot be linearly regressed in the space of the inputs. Therefore, the advantages of SVM are that they can map feature vectors to a higher-dimensional space that can be linearly separated. The brief principles are as follows; for more details about the theory of SVM, refer to the literature (Vapnik, 1998; Fletcher, 2009). Given a training dataset $\mathcal{D}=\{(\mathbf{x}_i,y_i)\,,i=1,\ldots,L\}$, where $\mathbf{x}_i$ is the input vector and y_i is the output, the SVM model is defined as:

$$f\left(\mathbf{x}_i\right)=\mathbf{w}^T\cdot\phi\left(\mathbf{x}_i\right)+b,\mathbf{w}\in\mathbb{R}^n,\mathbf{x}_i\in\mathbb{R}^m,b\in\mathbb{R} \tag{22}$$

where $\mathbf{w}$ and b are the parameters that need to be determined; $\phi(\cdot)$ is a mapping function from an input space ($\mathbb{R}^m$) to a high-dimensional space ($\mathbb{R}^n$). In the new feature space, the input data can be linearly separated. Unlike simple linear regression models, the SVM model uses more sophisticated principles to calculate the loss function, which is called the ε-insensitive loss function. In short, if the predicted value is larger than a distance (ε) away from the true value, the error is deemed unacceptable. To make the optimization feasible, slack variables (ξ_i^+ and ξ_i^-) are introduced to create a so-called soft margin in comparison to the hard margin, as shown in Fig. 13. Then, the primal model of SVM is expressed as follows:

$$\min_{\mathbf{w},b,\xi_i^+,\xi_i^-}\frac{1}{2}\parallel\mathbf{w}\parallel^2+C\sum_{i=1}^{L}\left(\xi_i^++\xi_i^-\right) \tag{23}$$

subject to

$$\left\{\begin{array}{c}\mathbf{w}^T\cdot\phi\left(\mathbf{x}_i\right)+b-y_i\leqslant\varepsilon+\xi_i^+\\ y_i-\mathbf{w}^T\cdot\phi\left(\mathbf{x}_i\right)-b\leqslant\varepsilon+\xi_i^-\\ \xi_i^+,\xi_i^-\geqslant0\end{array}\right\}i=1,\ldots,L$$

where C determines the trade-off between the margin and the magnitude of the slack variables; the term $\|\mathbf{w}\|^2$ is used to measure the function flatness, and L is the number of training data samples. By introducing the Lagrangian function, we can obtain the regression problem as follows:

$$f\left(\mathbf{x}\right)=\sum_{i=1}^{L}\left(\alpha_i^*-\alpha_i\right)\cdot K\left(\mathbf{x}_i,\mathbf{x}\right)+b \tag{24}$$

where α_i^* and α_i are the Lagrange multipliers, and $K\left(\mathbf{x}_i,\mathbf{x}\right)=\phi\left(\mathbf{x}_i\right)\cdot\phi\left(\mathbf{x}\right)$ is the kernel function that meets the Mercer's condition (Vapnik, 1998). Commonly used kernel functions in SVM are polynomial kernel and radial basis kernel functions.

Because SVM has the advantage of good regression ability under small samples, it has been applied to SOC estimation by many researchers. Alvarez Anton et al. (2013) proposed an SVM-based method to estimate SOC using voltage, current, and temperature data collected from DST driving cycles as well as charging and discharging cycles at different C-rates. Tenfold cross-validation was not only used to validate the performance of the SOC prediction but also to choose the optimal hyperparameters, including the regularization factor (C), the hyperparameter (v) that defined the SVM type regression, and the kernel parameters of the Gaussian kernel function (r). The results demonstrate that the prediction errors were stable within 6% in all tests, and the RMSE was only 0.71% during the whole test. Similar work can be found in Álvarez Antón et al. (2013) from the same research team, where 100-Ah LFP batteries were used to demonstrate the validation of the proposed SVM method. Several strategies (Hu et al., 2014; Li et al., 2020c) were used to accelerate the training process and find the optimal parameters, thus improving the performance of SVM. Hu et al. (2014) utilized a double-step search strategy to reduce the searching time for hyperparameter optimization, where the strategy first applies a rough search step to find the optimal pairs of hyperparameters and then further uses a smaller searching step to find the best pairs of hyperparameters in a smaller region. In another work (Li et al., 2020c), the PSO algorithm was

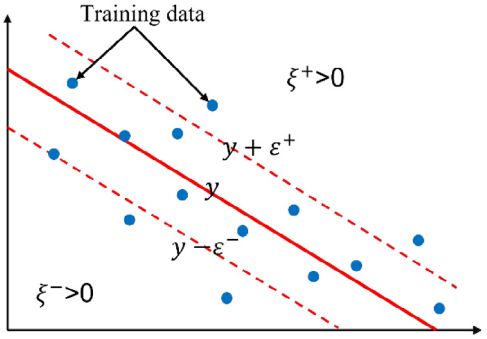

Fig. 13. The illustration of the SVM.

used to find the hyperparameters of the SVM model, namely the penalty factor (γ) and kernel width (σ) of the RBF. Compared with the SVM model with a normal optimization algorithm, the PSO-SVM model exhibited the advantages of faster processing speed and higher estimation accuracy. Moreover, the generalization ability of the proposed method was validated under four states of LiBs fault, namely the terminal voltage in excess of the upper or lower cut-off voltage, excess temperature, and high internal resistance. The average errors of the proposed method under these four battery fault conditions were all less than 2.5%.

2.4. Gaussian process regression

GPR is a probabilistic and non-parametric ML method to obtain the predictive probability distribution of the estimated value rather than just a point (Deng et al., 2020). Compared with other non-Bayesian ML methods (e.g., SVM, ANN), GPR has several advantages. First, the GPR model can quantify the confidence intervals around the estimates and thus provides more reliable estimation outputs because of the Bayesian framework. Second, the hyperparameter tuning process is also simpler by means of maximizing a marginal loglikelihood function rather than using grid-search or trial-and-error methods. Third, the impact of each input feature on output can be easily determined owing to the Bayesian learning in hyperparameter estimation. A brief overview of the theory of GPR applied in SOC estimation is as follows. For more details, the reader is referred to the literature (Sahinoglu et al., 2018).

Considering a training dataset, $\mathcal{D}=(\mathbf{X},\mathbf{y})$, which consists of D-dimensional N input vectors $\mathbf{X}=(\{\mathbf{x}_n\}_{n=1}^{N},\mathbf{x}_n<\mathbb{R}^{D})$ and the corresponding outputs $\mathbf{y}=(\{y_n\}_{n=1}^{N},y_n<\mathbb{R})$. The aim of GPR is to find a latent function $(f(\cdot))$ that maps the inputs $(\mathbf{x}_n)$ to their corresponding outputs (y_n). The key assumption in GPR is that the function values $(\mathbf{f}=[f(\mathbf{x}_1),f(\mathbf{x}_2),\ldots,f(\mathbf{x}_n)]^{T})$ are distributed according to a multivariate Gaussian distribution, as follows:

$$p(\mathbf{f}\mid\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n)=\mathcal{N}(\mathbf{0},\mathbf{K}) \tag{25}$$

where $\mathbf{f}=[f(\mathbf{x}_1),f(\mathbf{x}_2),\ldots,f(\mathbf{x}_n)]^{T}$ is the output vector; and $\mathbf{0}$ is an N x 1 vector whose elements are 0. $\mathbf{K}$ is a kernel matrix whose elements $(\mathbf{K}_{ij}=k_s(\mathbf{x}_i,\mathbf{x}_j))$ correspond to the values of the kernel function. Typical kernel functions are the squared exponential (SE) kernel, Matern5 kernel, Rational quadratic (RQ) kernel, Polynomial kernel, and Spectral mixture (SM) kernel. The optimal values of the hyperparameters of the kernel function can be determined by means of gradient-based optimization methods. Then, when given a new pair of inputs and outputs, denoted as $\mathbf{x}_*$ and y_* respectively, the joint distribution of the training outputs $(\mathbf{y})$ and new output (y_*) is as follows:

$$p(\mathbf{y},y_*\mid\mathbf{X},\mathbf{x}_*,\Theta)=\mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\0\end{bmatrix},\begin{bmatrix}\mathbf{K}+\sigma_n^2\mathbf{I} & \mathbf{k}_*\\ \mathbf{k}_*^{T} & k_{**}+\sigma_n^2\end{bmatrix}\right) \tag{26}$$

where $\mathbf{k}_*=[k(\mathbf{x}_1,\mathbf{x}_*),\ldots,k(\mathbf{x}_N,\mathbf{x}_*)]^{T}$ is the covariance vector between the training data $(\mathbf{X})$ and the new input; $k_{**}=k_s(\mathbf{x}_*,\mathbf{x}_*)$ is the variance of the new input $(\mathbf{x}_*)$; Θ is the hyperparameters of the kernel function; and σ_n^2 is the variance of a zero-mean additive Gaussian noise. By marginalizing the joint distribution above, we can obtain the predictive distribution of the new output as follows:

$$p(y_*\mid\mathbf{X},\mathbf{y},\mathbf{x}_*,\Theta)=\mathcal{N}(\mu_*,\Sigma_*) \tag{27}$$

Finally, the mean (μ_*) and variance (Σ_*) of the predictive distribution above denote the SOC estimation and the confidence interval of the SOC estimation, respectively. The expressions are as follows:

$$\begin{aligned}\mu_*&=\mathbf{k}_*^{T}\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{y}\\ \Sigma_*&=\sigma_n^2+k_{**}-\mathbf{k}_*^{T}\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{k}_*\end{aligned} \tag{28}$$

where $\mathbf{I}$ is the identity matrix.

In a study (Chandran et al., 2021), Chandran et al. comprehensively compared six commonly used ML algorithms for SOC estimation, including the ANN, SVM, linear regression (LR), GPR, ensemble bagging (EBa), and ensemble boosting (EBo). The results show that ANN and GPR could achieve the best performance. The kernel function plays a significant role in the GPR model because it encodes the prior assumptions about the properties of the function being modeled. In order to analyze the impact of kernel function selection on SOC estimation performance, Ozcan et al. (2016) built four GPR models, with an SE kernel, RQ kernel, Matern5 kernel, and Quasi-periodic kernel, respectively. According to the validation results under simulated data and experimental data, the GPR models with the SE kernel and RQ kernel both could obtain a reasonable estimation accuracy, with an RMSE of less than 1.5%. To further improve the accuracy of the GPR model, Sahinoglu et al. (2018) optimized the regular GPR and built the Recurrent GPR model as well as the autoregression Recurrent GPR model. The Recurrent GPR model combines the previously estimated SOC value with the measurements of the voltage, current, and temperature to form the input vectors, while the autoregression Recurrent GPR model further considers the impact of the present and past values of voltage, current, and temperature. The validation was implemented under the constant charge–discharge test and dynamic driving cycle test. Compared with SVM, RVM, and NN, the autoregression Recurrent GPR could achieve the best estimation performance, with the RMSE and MAE were below 0.3% and 0.9%, respectively. Because the temporal dependence of the sequential data was ignored, Xiao et al. (2021) proposed a new framework called GPR with a gated recurrent unit kernel (GRU-GPR). The proposed method integrates the non-parametric flexibility of kernel methods as well as the structural properties of the DL network to capture ordering matters and recurrent structures in sequential data. The public dataset (The Panasonic NCR18650PF) and the self-made dataset based on 18-Ah LiBs were used to validate the proposed method. The results show that the GRU-GPR method delivered better results than state-of-the-art techniques such as SVM, RNN, ECM-EKF, and GPR with conventional kernels and other methods.

2.5. Summary of the existing ML SOC estimation algorithms

Table 3 summarizes four main aspects of the ML SOC estimation algorithms reviewed in the above section, including the data profiles used to train and validate the models, input and output features, part of the hyperparameters of the proposed algorithms, and performance metrics. In addition, Fuzzy Logic (FL) and Genetic Algorithm (GA) are also summarized in Table 3. Although it is difficult to draw a direct conclusion as to which

Table 3
Summary of ML SOC estimation algorithms.

Algorithms	Refs.	Data profiles	Input and output	Hyperparameters	Performance metric (at 25 °C)
BPNN	Hannan et al. (2018)	CALCE dataset (DST, FUDS)	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Hidden neurons = 24 - Sigmoid function	-RMSE = 0.91% (FUDS) -MAE =0.59% (FUDS)
RBFNN	Chang (2013)	Constant current discharge test	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Hidden neurons = 24	-MAPE = 1.82%
ELM	Hossain Lipu et al. (2019)	BJDST, US06, DST, FUDS	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Hidden neuron s = 328/220/158/147	-RMSE = 0.76% (BJDST) -MAE = 0.55% (BJDST)
WNN	Cui et al. (2018)	NEDC, UDDS, UKBC	- Inputs = [V(t), I(t)] - Output = [SOC(t)]	- Levenberg–Marquardt (L-M) algorithm - Hidden neurons = 10	-MAE = 0.59% (NEDC)
WNN	Xia et al. (2018)	NEDC, EUDC, UDDS, UKBC	- Inputs = [V(t), I(t)] - Output = [SOC(t)]	- Levenberg–Marquardt (L-M) algorithm - Hidden neurons = 10	-MAE = 0.8% (UDDS)
DNN	He et al. (2014)	CALCE dataset (DST, FUDS, US06)	- Inputs = [I(t), I(t−4), …, I(t−4k), V(t), V(t−4), …, V(t−4k), T(t), T(t−4), …, T(t−4k)] - Output = [SOC(t)]	- Two hidden layers - Hidden neurons = 5 per hidden layer - k = 30	-RMSE = 1.4% (FUDS) -RMSE = 1.5% (US06)
DNN	Chemali et al. (2018)	Panasonic NCR18650PF dataset (US06, HWFET)	- Inputs = [V(t), I(t), V_avg(t), I_avg(t)] -Output = [SOC(t)]	- Four hidden layers - Hidden neurons = 50 per hidden layer	- RMSE = 0.78% (US06) - MAE = 0.61% (US06)
DNN	How et al. (2020)	CALCE dataset (DST, FDUS, US06, BJDST)	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Four hidden layers - Hidden neurons = 64 per hidden layer	- RMSE = 3.68% (DST, FDUS, US06, and BJDST)
NARX	Lipu et al. (2018)	FDUS, US06	- Inputs = [V(t), V(t−1), V(t−2), V(t−3), I(t), I(t−1), I(t−2), I(t−3), T(t), SOC(t−1), SOC(t−2)] - Output = [SOC(t)]	- Hidden neurons = 16 - Feedback delays = 2 - Input delays = 3	- RMSE = 0.68% (FUDS) - MAE = 0.48% (FUDS)
NARX	Wang et al. (2021b)	CC-CV, DST, UDDS, FUDS	- Inputs = [V(t), V(t−1), V(t−2), I(t), I(t−1), I(t−2), T(t), SOC(t−1), SOC(t−2)] - Output = [SOC(t)]	- Window size: changes with the characteristics of data - Hidden neurons = 12 - Feedback delays = 2 - Input delays = 2	- RMSE = 0.48% (UDDS) - MAE = 0.24% (UDDS)
LSTM	Ma et al. (2021)	Panasonic NCR18650PF dataset (US06, LA92, UDDS)	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t), SOE(t)]	- Number of LSTM neurons = 256 - Time step = 50	- RMSE = 1.71% (UDDS) - MAE = 1.39% (UDDS)
LSTM	Tian et al. (2020a)	CALCE dataset (DST, FDUS, US06)	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Number of LSTM neurons = 32 - Time step = 30	- RMSE = 0.9% (FUDS) - MaxE = 2.7% (FUDS)
BiLSTM	Bian et al. (2020)	Panasonic NCR18650PF & CALCE datasets	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Hidden neurons = 64 - Numbers of BLSTM layers = 2	- MAE = 0.84% (FUDS) - MaxE = 3.46% (FUDS)
GRU	Li et al. (2019c)	Panasonic NCR18650PF & CALCE & High-rate pulse discharge condition datasets	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Hidden neurons = 1000 - Time step = 100	- MAE = 0.86% (FUDS) - MaxE = 3.13% (FUDS)

(continued on next page)

Table 3 (*continued*).

GRU	Yang et al. (2019a)	DST, FUDS	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Hidden neurons = 150 - GRU layer = 1	-RMSE = 1.05% (FUDS) -MAE = 0.77% (FUDS)
GRU	Xiao et al. (2019)	CACLE dataset (CC-CV, DST, FUDS, US06)	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Hidden neurons = 260 - Optimizer = Nadam & AdaMax	-RMSE = 0.64% (FUDS) -MAE = 0.49% (FUDS)
CNN+LSTM	Song et al. (2019)	Private dataset (DST, US06, FUDS)	- Inputs = [V(t), I(t), T(t), V_avg(t), I_avg(t)] - Output = [SOC(t)]	- Neurons in LSTM layer = 300 - Filters in convolutional layer = 6 - Neurons in fully connected layer = 80;	-RMSE = 1.31% -MAE = 0.92% (DST, US06, and FUDS)
CNN+GRU	Huang et al. (2019)	DST, FUDS	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Two GRU layers - Hidden neurons =150 & 80 per GRU layers - Filters in convolutional layer = 8	-RMSE = 1.54% (FUDS) -MAE = 1.26% (FUDS)
CNN	Bhattacharjee et al. (2021)	Panasonic NCR18650PF and LG 18650HG2 datasets (UDDS, US06, HWFET, LA92)	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Number of convolutional layers = 2 - Number of filters in conv layer = 8	-MAE = 0.8% (US06) -MaxE = 4.72% (US06)
SVM	Alvarez Anton et al. (2013)	DST, CC-CV	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Number of support vectors = 903 - Penalty factor $\gamma = 0.125$ - Kernel width $\sigma = 1$	- RMSE = 0.4% (DST) - MaxE < 4% (DST)
SVM	Li et al. (2020c)	CC-CV	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Penalty factor $\gamma = 1.41$ - Kernel width $\sigma = 1.18$	- AE = 1.2%
GPR	Sahinoglu et al. (2018)	CC-CV, FUDS, US06	- Inputs = [V(t), V(t−1), I(t), I(t−1), T(t), T(t−1), SOC(t−1)] - Output = [SOC(t)]	- Time delay unit = 1 - SE kernel function	-RMSE = 0.24% -MAE = 0.81% (FUDS&US06)
GPR	Xiao et al. (2021)	Panasonic NCR18650PF & High-rate pulse discharge test datasets	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Batch size = 64 - Deep recurrent kernel function	-MAE = 0.79% -MaxE = 7.92% (US06, HWFET, UDDS, and LA92)
FL	Hu et al. (2016)	FUDS	- Inputs = [V_avg(t), I_avg(t), Power(t)] - Output = [SOC(t)]	- Six two-sided Gaussian membership function	-RMSE = 1.68%
GA	Chen et al. (2018c)	NASA dataset & NEDC, JP1015	- Inputs = [V(t), I(t), T(t)] - Output = [SOC(t)]	- Window size = 8 - Sliding size = 1 - Crossover rate = 0.8 - Initial population = 20	-MAE < 1% -MaxE < 1.04%

Notes: In the column of "Data profiles", the Panasonic NCR18650PF dataset, NASA dataset, LG 18650HG2, and CALCE dataset are public datasets, and others are private datasets.

algorithm is better just from the table (different studies utilized different training, testing, and validating datasets, different hyperparameters), it is still worth finding some commonalities.

From the perspective of input features, most of the studies used voltage, current, and temperature as input features, while a few utilized average values, previous values, or polarization states as additional inputs. It is reasonable that the voltage, current, and temperature were directly considered as input features owing to their easy access. However, adding other parameters related to the electrochemical characteristics of LiBs can improve the overall performance of the algorithms to some extent. From the perspective of the output, most studies took the SOC as the only output, while a few took other internal states (e.g., Ma et al., 2021) along with the SOC as the output. For the data profiles that are used to train and validate the proposed methods, most of the studies used datasets collected from dynamic driving cycles (e.g., FUDS, US06, UDDS, and other profiles), which is more convincing compared with the constant current–constant voltage (CC–CV) test or constant current pulse test datasets. However, more studies used public datasets (e.g., CALCE dataset, Panasonic NCR18650PF) to build their algorithms. This is worth advocating because researchers can validate and compare their proposed algorithms with the published algorithms in a relatively fair way. In the column of hyperparameters, different studies utilized various settings to achieve reasonable accuracy. The most important hyperparameters are listed to offer a reference for readers. In addition, providing the hyperparameters in as much detail as possible is the premise for reproducibility. As for the performance metrics, the most widely used indices, such as the RMSE, MAE, average error (AE), mean absolute percentage (MAPE), and max error (MaxE), are chosen for comparison. Note that only the performance metrics at 25 °C or room temperature are chosen, because almost every study validated its algorithm at 25 °C. Although it is impossible to obtain a direct conclusion from the error comparison, as they used different datasets and hyperparameter settings, the table can provide an easy reference for other researchers. Finally, a more detailed comparison between the above ML algorithms is given in Section 3.5.

3. Review of ML SOH estimation algorithms

It is inevitable for LiBs to degrade during long-term cycling or storage. The information on battery SOH is important for the energy management system of EVs to maintain a safe and highly efficient working condition. The study and analysis of the battery aging mechanisms and their consequences are critical to determining the performance of health estimation and the accuracy of predictions. Generally, the battery aging mechanism analysis is carried out on several levels, including the influence factors, internal side reactions, degradation modes, and external effects (Han et al., 2019). For external effects, capacity fade and power fade are the two most intuitive characteristics. For degradation modes, the aging mechanism can be summarized as loss of lithium-ion inventory (LLI), loss of active materials (LAM), internal resistance increase (RI), and loss of electrolyte (LE). As the battery charging and discharging processes are related to the lithium-ion intercalation and deintercalation of the anode and cathode active materials, the amount of the active materials and the available lithium ions determine the battery capacity directly. In one study (Dubarry et al., 2012), the dual tank model was used to describe the aging mechanism. The LAM is like the change of the tank itself, while the LLI is like the loss of water in the tank. In addition, the power fade is related to the RI, while the LE may lead to the capacity diving at the end of battery life. As for internal side reactions, they are the main reasons for the abovementioned degradation modes. Note that the internal side reactions are extremely complicated and may even be coupled. Typical internal side reactions include graphite exfoliation, metal dissolution coupled with electrolyte decomposition, solid electrolyte interface (SEI) film formation, cathode electrolyte interface (CEI) formation, and lithium deposition. However, for the onboard applications, the loss of capacity and the increase in resistance are two widely used indicators to reflect the battery SOH, expressed as follows:

$$SOH_c = \frac{C_t}{C_0} \times 100\%, \tag{29}$$

$$SOH_r = \left| \frac{R_{EOL} - R_{act}}{R_{EOL} - R_{NEW}} \right| \times 100\%, \tag{30}$$

where SOH_c means the battery SOH is reflected from the perspective of capacity loss; C_t and C_0 are the actual and nominal capacity of the battery, respectively; SOH_r means the battery SOH is reflected from the perspective of resistance increase; R_{act} represents the current resistance of the battery; and R_{EOL} and R_{NEW} are the resistance of the end-of-life (EOL) battery and new battery, respectively.

Therefore, SOH estimation can be regarded as capacity-based or resistance-based estimation to some extent. Similar to the SOC of LiBs, SOH is also not directly measurable with the commonly used sensors, which makes it a challenging task in BMS. This section focuses on ML algorithms for SOH estimation. As depicted in Fig. 14, the ML SOH estimation algorithms mainly consist of four steps. Specifically, the first step is to conduct battery aging tests and collect raw data, such as voltage, current, and temperature. As the raw data cannot provide sufficient information to reflect SOH directly, it cannot guarantee an accurate and robust SOH estimation. The second step is hence to extract highly related features from the raw data using different techniques, such as model-based analysis, Incremental Curve Analysis (ICA), and Differential Voltage Analysis (DVA). Before choosing health features, we must analyze the battery aging mechanisms and degradation modes, and make a connection with the health features. Then, correlation analysis is applied to analyze whether the extracted features are valuable. Thereafter, the features, together with the true SOH values, constitute the training dataset. Subsequently, different ML methods are used to learn and validate the nonlinear relationship between the input features and output based on the training dataset. Finally, the established ML algorithms can be used to estimate SOH for new data. Usually, the training process is performed offline, and then the trained model is used online.

In order to thoroughly learn the characteristics of battery degradation modes, it is evident that training an ML SOH estimation algorithm usually needs more datasets (e.g., aging data covering the entire life cycle) compared with SOC estimation. For the sake of simplicity, this section mainly reviews the applications of different ML algorithms for SOH estimation in recently published papers. As the main principles of these algorithms have been given in Section 2, only the complementary principles are provided.

3.1. Shallow neural network

3.1.1. Backpropagation neural network & Radial basis function neural network

As discussed in Sections 2.1 and 2.2, the BPNN and RBFNN can map a nonlinear relationship between an arbitrary number of inputs and outputs. Thus, these two kinds of NNs have been commonly used in SOH estimation. To consider the environments of EVs in a more practical way, won You et al. (2016) made great efforts to collect datasets from extensive dynamic driving profiles for more than 1 year at various temperatures. The measured data, including current, voltage, and temperature, as well as their historical distributions, which were based on three-dimensional points clouds, were then used as inputs to train a BPNN-based SOH estimation algorithm. In another study (Kashkooli et al., 2019), the calendar-aged data for training and validating a BPNN-based SOH estimation model were obtained from the calendar-life experiment of eight commercial 15-Ah LiBs. Different storage temperatures (e.g., 35 °C, 60 °C) and storage conditions (e.g., fully charged or fully discharged) were fully considered to investigate their effect on calendar life. To track the degradation trend, the researchers performed reference performance tests (RPTs) at 1-month intervals for 10 consecutive months. The results demonstrate that the proposed methods could not only predict the battery SOH accurately but also improve the SOC estimation accuracy by calibrating the available capacity at certain aging conditions. To improve the quality of training datasets, researchers

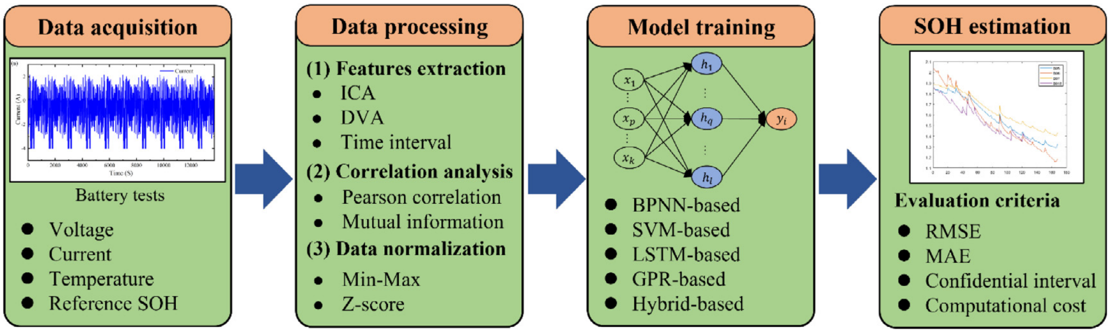

Fig. 14. The procedures for developing ML SOH estimation algorithms.

employed important sampling (IS) rather than traditional uniform in one study (Wu et al., 2016) to select more important samples from terminal voltage curves. The voltage difference between two adjacent samples was defined as the importance function in reflecting larger voltage variation. Finally, 11 samples were selected as input features, and a BPNN with 40 hidden layer neurons was trained for health prognostication.

It is well known that the model performance is highly dependent on the quality of the input features. Thus, principal component analysis (PCA) was used in one study (Mao et al., 2021a) to select important features to reduce computational complexity. Then, the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) was applied to further decompose and denoise the collected features. Combined with the logistic regression (LR) and KF, which were used to predict the trend and minor fluctuation, an RBFNN-based method was proposed to estimate SOH. In another work (Lin et al., 2021), Lin et al. developed an adaptive tunable hybrid RBF network, which also possesses the majority of the features of the autoregressive model. The main advantage of the proposed RBF network is that the structural parameters are updated by Brownian motion (BM) and PF. Moreover, a continuous hidden Markov model (CHMM) and kernel density estimation (KDE) method were used to calculate Kullback–Leibler distance as input features. Compared with the RBF network with a fixed structure, the proposed tunable RBF network could achieve better estimation accuracy in the NASA dataset (Saha et al., 2009) and CALCE dataset. Furthermore, when adding Gaussian white noise to dynamic discharging profiles, the proposed method could still maintain robust estimation results.

3.1.2. Extreme learning machine

Owing to the advantages of a faster convergence speed and acceptable model accuracy, ELM has been applied in SOH estimation (Wei et al., 2021). Considering the more realistic operating conditions of EVs, Tian and Qin (2021) extracted three features from random discharge data and conducted the Pearson correlation analysis to approve the rationality of the features. The three features were the discharge time to reach the cut-off voltage, the voltage upward appreciation after discharge, and the variance of a random walk (RW) cycle voltage curve, respectively. Furthermore, the online sequential ELM (OSELM) was proposed for online learning and optimized by the Bernstein inequality (BI-DD) algorithm to guide the update mechanism. The proposed method has the advantage of fast training, high accuracy, and great generalization capability. In one study (Pan et al., 2018), the internal ohmic resistance and polarized internal resistance of the Thevenin model identified by the recursive least square (RLS) method were selected as health indicators because the ANOVA results show that the different loading files had no significant impact on the battery model parameters. This conclusion meant that no matter the battery's condition, the battery model parameters could be identified online by the RLS method with sufficient accuracy. Subsequently, an ELM was utilized to build the SOH estimation model where the internal ohmic resistance and polarized internal resistance were employed as input features. Compared with the BPNN model using identical activation function and the number of neurons, the results show that the ELM-based model not only had higher accuracy (e.g., 1.72% MAE for ELM and 1.79% for BPNN), but also a faster training speed (e.g., 0.0136 s for ELM and 0.3420 s for BPNN).

Apart from using ELM alone, combining other methods to improve the overall performance was proposed in two studies (Jia et al., 2019; Chen et al., 2021). To reduce the computational complexity of BPNN and use at least the previous 30% of the aging data to train a model, Jia et al. (2019) combined transfer learning and ELM to build a SOH estimation method. Owing to the difficulty in applying the SOH estimation algorithm for different types of batteries with unknown conditions, Chen et al. (2021) proposed a novel metabolic ELM (MELM) framework to realize this aim. In the online estimation phase of the proposed MELM framework, the metabolic mechanism and the grey-model-based error compensator were fused into the ELM-based degradation model to guarantee accurate SOH estimation. The maximum absolute error of the proposed method was less than 1.93% for other types of batteries. Furthermore, the proposed method could also guarantee high estimation accuracy for batteries at different usage levels using five historical data points.

3.2. Deep learning algorithm

With the availability of large-scale data and powerful chips, DL algorithms have been increasingly considered for SOH estimation owing to their advantages of higher accuracy, stronger robustness, and better generalization, as well as the capability to process large-scale data. This section focuses on the application of DL-based SOH estimation algorithms.

3.2.1. Deep neural network

Compared with shallow NNs, a DNN can map more complex and nonlinear functions efficiently owing to the multiple hidden layers. In one study (Zhang et al., 2019a), five different features were extracted from partial IC curves first, which were processed by the Gaussian filter. According to the Spearman correlation analysis between the extracted features and SOH, not all five features were used as the input vector of a four-layer NN. Only two features value that had the highest correlation with SOH were selected as the input of the NN model. In addition, the final structure of the DNN is four-layered, with the number of neurons being 2-5-6-1, according to the analysis of training accuracy and speed. In order to fully cover the corresponding battery usage scenarios as much as possible, Xia and Qahouq (2021) extracted five features from the whole self-made aging dataset, such as the voltage decrease rate during the constant current discharging mode, voltage increase rate during CC charging mode, CC charging time duration, current decrease rate during the CV charging mode, and temperature decrease rate during the CV charging mode. The results of the grey relational analysis show that all five features had a strong correlation with SOH values. Furthermore, different SOH estimators that used different numbers of input features were trained using a four-layer DNN with 128 neurons per layer. The results demonstrate that using all five features could obtain the highest estimation accuracy. To compare the detailed application of the model-based EKF method and ML DNN method, Shi et al. (2021) proposed an estimation system that combines these two methods together. Over 20 different structures of DNNs were compared to find the optimal model. Note that the higher accuracy of using a DNN also comes with the drawback of a higher computational time and the need for more computing resources.

3.2.2. Recurrent neural network

3.2.2.1. Nonlinear autoregression with exogenous input neural network. Because the change in battery SOH is much slower than that of SOC, employing an RNN that contains former information to track the degradation state of LiBs is naturally a better choice. The accuracy and robustness of ML algorithms highly rely on the availability of a comprehensive battery degradation dataset that covers all kinds of battery aging patterns as much as possible. Khaleghi et al. (2021) collected the whole dataset from 21 NCM batteries that were cycled in various operating conditions for more than two years. This extensive dataset was used to train a NARX-based SOH estimation model to guarantee high estimation accuracy. The input features were directly extracted from partial charging voltage curves (voltage boundary of 3.6–3.8 V), which were easily accessible in practical applications. Another contribution of this work is that for the purpose of improving the generalization capability of the proposed method, the NARX-based model was not only validated on untrained data but also on the entire unseen test regime. Therefore, 12 cells were used to train the model, while another nine cells were opted to validate the model. The results show that compared with GPR and BPNN, the proposed model had the lowest computational cost and the highest estimation accuracy. Rather than use the traditional back-propagation algorithm to learn the weights of the NARX, Cui et al. (2021) applied the Bayesian regularization algorithm to improve the convergence speed and avoid the local extremum. Eight health indicators were extracted from the discharge process to form the input features. Furthermore, the current output was also used as internal input, together with the historical outputs, which were used as external inputs. Compared with the BPNN and CNN, the MAEs of the proposed method were 53.63% and 36.29% lower, respectively.

3.2.2.2. Gated recurrent neural network. Compared with NARX-based methods, the gated RNN can solve the problem of the exploding or vanishing gradient during training. Thus, gated RNNs are widely used for SOH estimation. Li et al. (2019b) proposed a co-estimation scheme for capacity and SOC estimation. First, the accelerated aging tests were conducted to investigate the relationship between the partial voltage curves during charging and the relevant aging levels. Then, the traditional Elman NN was utilized to predict the capacity in real time, which provided the foundation for SOC estimation. The proposed strategy not only could provide an accurate capacity estimation but also improve the performance of SOC estimation after the capacity calibration.

Kaur et al. (2021) comprehensively compared three different NNs, the BPNN, LSTM, and CNN, for SOH estimation. The hyperparameters of each network were tuned using six test sets obtained in a six-fold manner before the performance comparison. Various aspects of the three algorithms, including the model complexity, sampling rate of the input, and type of signals, were analyzed and compared. Overall, the LSTM-based algorithm, which contained one hidden layer and 50 hidden layer neurons and used voltage, current, and charging capacity as input data, was validated and found to be the best SOH estimation algorithm. However, the average computational time of the LSTM-based model was much longer than that of the other two models. Furthermore, the results prove that the measured temperature of the battery made a slight difference in capacity estimation. To fully consider the distribution discrepancy between the training and testing datasets when developing an ML SOH algorithm, Kim et al. (2021) applied three public datasets (i.e., one source and two target datasets) with different electrochemical characteristics to validate their proposed LSTM-based methods. In detail, the CALCE dataset was used to build the model, and the NASA and Cavendish Laboratory datasets (Zhang et al., 2020a) were used to validate the generalization capability of the proposed model by means of the transfer learning algorithm. In addition, the degradation patterns of the batteries were predicted using the variational inference of the Monte Carlo dropout. Overall, the main advantage of the proposed model was that it could accurately estimate SOH and predict RUL for different LiBs types while only using less than 20% of data cycles. In Shu et al. (2021), Shu et al. also combined the LSTM and transfer learning to predict the SOH of battery packs. To address the problem of inconsistency among battery cells, a cell mean model (CMM) and a cell difference model (CDM) were used to determine the SOH of the battery packs. The results demonstrate that using only 40% of the whole dataset to train the model could maintain an estimation error of less than 3%. Furthermore, the adaptability and scalability of the proposed method were verified using the other two battery datasets where the proposed algorithm was transplanted from NCM batteries to LCO and LFP batteries.

Compared with LTSM, GRU may have slightly higher estimation errors, but GRU requires fewer parameters which makes it more suitable for embedded applications. Ungurean et al. (2020) observed that the number of parameters for GRU was 25% less than the number of the parameters of LSTM. In the training phases, the MAE of GRU and LSTM were close, indicating 0.0264 and 0.0270, respectively. However, in the testing phase, it could be observed that the GRU-based estimation errors were slightly higher than the ones based on LSTM in most of the cases. Excluding the regeneration points, which would cause large estimation errors, the absolute errors of GRU and LSTM were 2.91% and 2.45% on the NASA dataset. However, there remain challenges in dealing with the regeneration points. Fan et al. (2020) proposed a hybrid network, called GRU-CNN, as shown in Fig. 15, to reduce the computational burden of the number of weights. At the same time, the proposed network can also manipulate the advantages of GRU to learn features and time dependencies from time-sequential data. Two public datasets, the NASA and Oxford datasets (Christoph, 2017), were utilized to verify the performance of the proposed hybrid network work. Compared with the individual use of CNN or GRU, the hybrid mode achieved higher accuracy in both datasets but also needed more training time.

3.2.3. Convolutional neural network

For traditional ML algorithms applied in capacity estimation, the input features are usually extracted manually. However, extracting useful information from a large amount of data is time-consuming and easily neglects representative characteristics. As for the CNN, the features can be extracted automatically from raw data with the aid of multi-level layers. Based on the directly measured voltage, current, and calculated capacity, Shen et al. (2019)

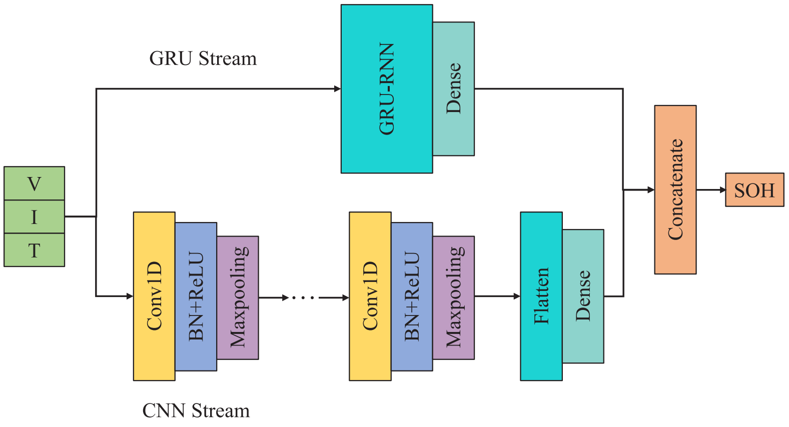

Fig. 15. The hybrid GRU-CNN architecture (Fan et al., 2020).

Table 4
Summary of the configuration of the proposed DCNN model (Shen et al., 2019).

Layer name	Filter size	Number of kernels	Stride size	Number of weights	Number of biases
Input	25 × 3×1	–	–	–	–
Conv.1	1 × 2×1	16	(1,1)	32	16
Max-pooling	3 × 1×1	16	(2,1)	–	–
Conv.2	3 × 1×1	32	(1,1)	96	32
Conv.3	3 × 1×1	40	(1,1)	120	40
Conv.4	3 × 1×1	40	(1,1)	120	40
Conv.5	3 × 1×1	40	(1,1)	120	40
FC.1	40 × 1	–	–	25 600	40
FC.2	40 × 1	–	–	1600	40
FC.2	1 × 1	–	–	40	1

Notes: “Conv” means convolutional layer, and “FC” means fully connected layer.

constructed a deep CNN (DCNN) to estimate SOH for the first time. The proposed DCNN consisted of five convolution stages and three fully connected stages with over 28,000 parameters and about 6000 neurons. The specific configuration of the DCNN model is summarized in Table 4. A private dataset comprising 16 LiBs cycling for 10 years, as well as the NASA dataset, was used to test and validate the proposed method. Compared with the RVM model and ANN model, the proposed DCNN could achieve better estimation accuracy. In another work (Yang, 2021), Yang constructed a hybrid CNN (HCNN) that was a fusion of a three-dimensional (3D) CNN and a 2D CNN to deal with the problems in the early prediction of battery cycle life and RUL. The 3D CNN was used to fuse the voltage, current, and temperature curves and their differences between cycles, while the 2D CNN was utilized to extract the features from the voltage, current, and temperature curves automatically. In another study (Gong et al., 2022), a 2D CNN was combined with an ultra-lightweight subspace attention mechanism (ULSAM) module and a simple recurrent unit (SRU) module to encode the sampling data from charging curves and generate the input sequences of a BPNN model.

Another common application of a CNN is to combine it with an RNN to take full advantage of both algorithms. Li et al. (2022) proposed a novel active-state-tracking LSTM and 1D CNN (CNN-ASTLSTM) to extract the aging features and learn the time dependencies between these features. An improved Bayesian optimization algorithm was utilized to tune the hyperparameters automatically during SOH estimation. A comprehensive comparison between the proposed method and the state-of-art methods was conducted to evaluate its superiority in terms of accuracy, FLOPs, the number of parameters, storage time, and training time. To capture the regeneration points of the LiBs mentioned in Ungurean et al. (2020) and Zhou et al. (2020) built a temporal convolutional network (TCN) where causal convolution and dilated convolution are used to separate the global capacity degradation and local capacity regeneration. The proposed model framework was fully validated using two public datasets, namely the NASA and CALCE datasets.

3.3. Support vector machine

As SVM is one of the most used ML methods because of its great trade-off between usability and performance, the standard SVM model and other variants of SVM have been applied in SOH estimation. It is well known that the kernel function of SVM can implicitly map the input features to a high-dimensional feature space where the samples can be linearly separated. Therefore, various optimization methods have been utilized to modify the kernel function to improve the performance of SVM models. Liu et al. (2020) modified the form of the Gaussian kernel function to adapt to the volatility of the SOH curve. Specifically, the original Gaussian kernel function was divided into two terms; the first term was used to represent the overall degradation trend of SOH, while the other term was utilized to simulate small fluctuations. Furthermore, a differential evolution (DE) algorithm was used to optimize the hyperparameters of the SVM model. Compared with other ML estimation methods using the NASA dataset, the proposed method outperformed in terms of both dynamic working conditions and the constant current discharging process. Feng et al. (2018) investigated the influence of the standard deviation (σ) in the Gaussian kernel on estimation accuracy, and then the principle for selecting the proper σ was proposed.

Using more relevant input features is another common way to optimize the SVM model. Widely used features includes IC peak values and the IC position (Li et al., 2020b; Guo et al., 2021), constant current charging time (Liu et al., 2020), sample entropy (Widodo et al., 2011), energy signal (Chen et al., 2018b), terminal voltage response under current pulse test (Meng et al., 2018; Cai et al., 2019), mean and standard deviation of

Table 5
Summary of the input features of ML SOH estimation algorithms.

Extracted features	Methods	Examples
Direct features	Collected by sensors directly	Voltage, current, temperature, charge, and discharge capacity
Model-based features	Calculated by EKF, RLS towards battery model	Internal resistance, capacitance, capacitance resistance, and SOC
Preprocessed features	Processed by ICA, DVA, DTM	IC peak, IC position, and other geometric features

the measured current and voltage (Ma et al., 2019), and parameters of ECM (Yang et al., 2018). In order to optimize the feature extraction and hyperparameters tuning process simultaneously, Cai et al. (2019) utilized a hybrid encoding technology in one chromosome. In another work (Ma et al., 2019), a sliding-window-based feature extraction strategy (SWBFE) was designed to extract the mean and standard deviation of the measured current and voltage from different discharge profiles to form multiple feature sets. Furthermore, the based SVRs were trained on multiple feature sets, and two ensemble SVR models were generated by using the Adaboost and Stacking algorithms to incorporate multiple-based SVRs.

Compared with the standard SVM model, the least-squares support vector machine (LS-SVM) is an important variant that has the advantages of a faster solving speed and simpler solving process. In one work (Xiong et al., 2021), a weighted LS-SVM-based SOH estimation method was proposed based on a weighting function and linear equations for retired LiBs. Compared with the standard SVM, the weighted LS-SVM could improve the SOH estimation performance by weighting the error variables, even when the test datasets involve different battery materials and different conditions. Further, in another work (Yang et al., 2018), the PSO algorithm was used to improve the LSSVR ability of global optimization. Compared with the LSSVR method without PSO optimization, the PSO-LSSVR method showed better performance, with an RMSE of less than 2%. Note that the real-time parameters of the Thevenin model identified by RLS and the SOC estimated by EKF were used as the inputs of the LSSVR.

3.4. Gaussian process regression

As a non-parametric ML method, GPR can obtain the predictive probability distribution of the estimated value, which is suitable for SOH estimation. Using the same aging dataset and aging features, researchers compared a data-driven GPR model with BPNN and SVR to verify its accuracy, generality, and robustness (Lyu and Gao, 2020). Similar to Yang et al. (2018), this work combined the model-based method and GPR-based method to estimate SOH, where the model-based method was used to identify the parameters of ECM, and then the parameters were used as inputs to train the GPR-based model. ICA and DVA have been shown to reflect the degradation state of LiBs from different perspectives, such as the area, position, peak amplitude, and gradient. In addition, differential thermal voltammetry (DTV) curves (Wang et al., 2021c) can provide sufficient information to reflect the aging state. Similar to one study (Hannan et al., 2021), in order to capture the features of peak amplitude in IC curves, which was submerged by measurement noises easily, Li et al. (2020a, 2019a) first applied the Gaussian filter algorithm to obtain smooth IC curves. They then extracted features between the voltage range from 3.8 and 4.1 V at the interval 30 mV, resulting in 11 features in total. After that, a GPR-based battery degradation model was trained based on the first 55% of data and verified based on the residual data. The validation results demonstrate that the proposed GPR model could achieve high accuracy, with the MAE and RMSE being less than 1.2% for all tested batteries. Rather than extracting features from IC curves, Jia et al. (2020) extracted eight features from the discharge profile directly. After the grey relational analysis between these eight features and SOH, three features (e.g., the time needed to reach the lowest discharge point, the time decrement from 3.8 V to 3.5 V, and the battery temperature rising to the highest point) were used to train a GPR-based SOH estimation model. The SOH estimation results on the NASA dataset show that the RMSE of the proposed model was lower than 0.4% for all tested batteries. To further improve the accuracy of the single GPR model, the multiple GPR (MGPR) method was proposed in one study (Zheng and Deng, 2019), where three single GPR models trained by three different datasets were combined together to form a weighted MGPR model. The prediction certainty of every single GPR was utilized to design the model weight for the MGPR model. However, the prediction uncertainty of the MGPR was not solved. One of the difficult tasks in SOH estimation is the capacity regenerations and fluctuations during the aging process, as mentioned in another study (Ungurean et al., 2020). Therefore, Yu (2018) applied empirical mode decomposition (EMD) to extract the local regenerations and various fluctuations in the SOH time sequence. Then, multiscale LR and GPR were integrated based on the residuals and a series of intrinsic mode functions. Compared with other methods (e.g., SVR, GPR without EMD), the proposed method had better estimation performance.

3.5. Summary of the existing ML SOH estimation algorithms

Table 6 summarizes five main aspects of the ML SOH estimation algorithms reviewed in the above section, including the data profiles used to train, test, and validate the models; the number of the input features; the correlation analysis techniques; the hyperparameters; and the performance metrics. As discussed in Section 2.5, it is challenging to conclude which algorithm is better simply based on the table because of the differences between datasets, hyperparameter selection, and battery types. It is still worth finding some commonalities.

First, ML methods are sort of data-driven algorithms. Therefore, the quality of the dataset used to train, test, and validate is undoubtedly one of the most important aspects to be considered. As summarized in Table 4, more studies use the public datasets (summarized in Section 4.1) to develop their algorithms. Compared with the self-made private datasets, public datasets are usually easier to access and can cover more realistic conditions for EV applications. It is recommended for future research to use both private and public datasets to develop algorithms as well as share datasets on the public platform.

Second, for the data processing, the feature extraction and selection method, correlation analysis between the extracted features and outputs, and data cleaning methods are common aspects to be considered. Specifically, according to the types of input features, they can be categorized into three groups, as shown in Table 5. Most of the studies used the model-based features or preprocessed features as inputs to train their models, while DL algorithms could use direct features because the spatial features and temporal features can be reflected by convolutional layers in CNN and memory states in RNN, respectively. Note that analyzing the battery aging mechanism and degradation modes is the premise of choosing suitable model-based and preprocessed features. Thereafter, correlation analysis between the extracted features (especially for the model-based features and preprocessed features) and the aging state is necessary to abandon the

Table 6
Summary of ML SOH estimation algorithms.

Algorithms	Refs.	Data profiles (Public/ Private)	Number of input features	Correlation analysis	Hyperparameters	Performance metric
BPNN	won You et al. (2016)	Private dataset (UDDS, US06, and NYCC)	80 (groups 1&3)	None	- Hidden neurons = 80	-RMSE: 0.66%
BPNN	Wu et al. (2016)	Private dataset (CC-CV test)	11 (group 1)	None	- Hidden neurons = 40 - Hyperbolic tangent sigmoid function	-MAE = 29.42 (RUL estimation)
RBFNN	Mao et al. (2021a)	NASA dataset (5#, 6#, and 7#)	3 (group 3)	Pearson correlation analysis	- Single hidden layer	-RMSE = 0.7% -MAE = 0.53%
RBFNN	Lin et al. (2021)	NASA dataset (5#, 6#, and 7#) and CACLE dataset (35#, 36#, 37#, and 38#)	3 (group 2)	Pearson correlation analysis	- Structural parameters are adaptively modulated by BM modeling and PF predictor	-RMSE = 0.7% -MAE = 0.61%
ELM	Tian and Qin (2021)	NASA dataset	3 (group 3)	Pearson correlation analysis	- Hidden neurons = 10 - Data block length = 50	-MAE = 0.64%
ELM	Pan et al. (2018)	Private dataset (CC discharge test, NEDC, and pulsed discharge test)	2 (group 2)	Pearson correlation analysis & Spearman correlation analysis	- Hidden neurons = 20 - Sigmoid function	-RMSE = 1.09% -MAE = 1.72%
DNN	Zhang et al. (2019a)	NASA dataset (5#, 6#, 7#, and 18#)	2 (group 3)	Spearman correlation analysis	- Two hidden layers - Hidden neurons= 6 & 5	-SOH error < 3% -RMSE = 5.41 (RUL estimation)
DNN	Xia and Qahouq (2021)	Private dataset (CC-CV test)	5 (group 3)	Grey relational analysis	- Two hidden layers - Hidden neurons = 128 per layer	-RMSE = 1.576% -MAPE = 5.022%
NARX	Khaleghi et al. (2021)	Private dataset	5 (group 1)	None	- Hidden neurons = 10 - Feedback delays = 2	-RMSE = 0.26%
NARX	Cui et al. (2021)	NASA dataset (5#, 6#, 7#, and 18#)	8 (group 3)	None	- Hidden neurons = 50 - Feedback delays = 8	-MAE = 0.72% -MaxE = 4.69%
LSTM	Kaur et al. (2021)	Private dataset (CC charge and discharge test)	3 (group 1)	None	- One hidden layer - Hidden neurons = 50 - Sampling rate = 10	-RMSE = 0.042 -MAE = 0.0216 (Capacity estimation)
LSTM	Kim et al. (2021)	CALCE, Cavendish Laboratory, and NASA datasets	1 (group 2)	None	- Sliding windows = 100 - Dropout masks = 10 000	-MAPE = 1.95% (Capacity estimation)
GRU	Ungurean et al. (2020)	NASA dataset	5 (group 2)	None	- Two hidden layers - Hidden neurons = 50 per layer	-AE = 2.91%
GRU	Fan et al. (2020)	NASA and Oxford datasets	3 (group 1)	None	- Hidden neurons = 256 (GRU) - Convolution number = 64 - Size of each convolution layer = 32 × 1	-MAE = 1.03% -MaxE = 4.11%

(continued on next page)

Table 6 (*continued*).

CNN	Shen et al. (2019)	NASA and private datasets	3 (group 1)	None	- 5 convolution stages - 3 fully connected stages - ReLU activation function - Optimizer: Stochastic gradient descent (SGD)	-RMSE = 1.477% -MaxE = 9.479% (Capacity estimation)
CNN	Zhou et al. (2020)	NASA dataset (5#, 6#, and 18#) and CALCE dataset	1 (group 1)	None	– Number of convolution kernels = 256 - Size of the kernel = 3 × 1;	-RMSE = 1.1% -MAE = 0.9%
SVM	Liu et al. (2020)	NASA dataset (5#, 6#, and 7#) and Private dataset (NCR18650PF)	3 (group 3)	Grey relational analysis	–Mutation factor of the DE algorithms = 0.6	-MAPE = 0.23%
SVM	Li et al. (2020b)	Private dataset (CC-CV test)	3 (group 3)	Pearson correlation analysis	- Iteration = 100 - Standard deviation = 1e−4	-RMSE = 0.94% -MAE = 0.769%
SVM	Yang et al. (2018)	Private dataset (CC-CV test, HPPC test)	4 (group 2)	none	- Number of particles in PSO = 30	-RMSE = 1.92%
GPR	Li et al. (2020a)	NASA dataset (5#, 6#, 7#, and 18#)	11 (group 3)	Pearson correlation analysis	- SE kernel function	-RMSE = 0.78% -MAE = 0.4%
GPR	Jia et al. (2020)	NASA dataset (5#, 6#, 7#, and 18#)	3 (group 3)	Grey relational analysis	- SE kernel function	-RMSE = 0.37%

Notes: In the column of "Number of input features", group1, 2, and 3 represent the direct features, model-based features, and processed features summarized in Table 4, respectively. In the column of "Data profiles", the NASA dataset (5#, 6#, 7#, and 8#) means the four tested batteries labeled 5#, 6#, 7#, and 18# were used. All else follows.

low-related features. Commonly used correlation analysis techniques include Pearson correlation analysis, Spearman correlation analysis, and grey relational analysis.

Third, selecting an appropriate ML algorithm is another important task. As discussed in Section 2.5, different ML algorithms have unique advantages and disadvantages. Choosing an appropriate ML algorithm according to the specific application is more important than pursuing model complexity and accuracy blindly. Generally, the comparison between the reviewed ML algorithm is as follows. Shallow NNs have been approved to learn any nonlinear relationship between an arbitrary number of inputs and outputs. However, the training process may be slow and can be trapped in a local minimum. DL algorithms can perform better in dealing with a large amount of data because of their unique structures. They can also achieve higher accuracy and greater generalization capability. However, the large computational burden is the main drawback of DL algorithms. Regarding the SVM method, it has the advantage of good generalization capability and reasonable estimation accuracy, but it is not suitable for large datasets, which would greatly increase the execution time. GPR methods can provide the uncertainty of the estimated value, which is valuable for practical applications. However, they have low efficiency in high-dimensional spaces, and their calculation is relatively complex.

4. Challenges and prospects

According to the above introduction of different ML algorithms, it can be concluded that the ML algorithms have become a research hotspot and have great potential in SOC and SOH estimation for LiBs. However, sustainable progress is still required, because the implementations of ML SOC and SOH estimation algorithms in real applications still face many problems. In this section, the challenges and prospects of ML state estimation algorithms are discussed.

4.1. *Data quality*

Overall, ML state estimation algorithms are based on a large quantity of data. Therefore, collecting high-quality datasets is the foundation for these algorithms.

First, high-quality datasets rely on data precision. Usually, the battery dataset is acquired in a well-designed battery test platform that contains a controller, programmable load, and a host computer with integrated software. Typical battery test platforms include Arbin BT2000 (Sahinoglu et al., 2018), NEWARE BTS series, including BTS 4000 (Hu et al., 2020; Hannan et al., 2020) and BTS 600 (Hu et al., 2020; Feng et al., 2020). Different battery test platforms differ in terms of data acquisition, response time, and especially precision, which greatly influence the data quality and further affect the overall performance of the ML algorithms. Therefore, advanced battery test platforms are essential to eliminating the measurement noise, thus building a solid foundation for model training and validating.

Second, high-quality datasets rely on data abundance. Theoretically, the increasing amount of training data would solve the overfitting problems, especially for DL algorithms, and improve their generalization capability. Particularly, the LiBs are complicated electrochemical systems that would be seriously influenced by the parasitic side reactions inside the LiBs as well as the harsh operating environment of EVs. Therefore, the overall performance of ML algorithms in terms of accuracy, robustness, and generalization capability would be improved by using training data covering different temperatures, aging states, positive

Table 7
Summary of the publicly available battery datasets.

Sources	Reference
Panasonic NCR18650PF LiBs	Kollmeyer (2018)
LG 18650HG2 LiBs	Naguib et al. (2020)
LFP A123(APR18650M1A) LiBs	Severson et al. (2019)
Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland	University of Maryland (2021)
Research & development data repository from Sandia National Labs	Barkholtz et al. (2017)
NASA Ames Prognostic Data Repository	Saha and Goebel (2007)
Oxford battery degradation dataset from Howey research group	Christoph (2017)

Table 8
Summary of the heuristic optimization techniques in ML algorithms.

Algorithms	Refs.	Optimal hyperparameters
BSA + BPNN	Hannan et al. (2018)	The number of hidden layer neurons and learning rate
PSO + RBFNN	Zhang et al. (2020b)	Kernel parameters (w and σ)
OLS + RBFNN	Chang (2013)	The number of hidden layer neurons and kernel parameters ($w and \sigma$)
GSA + ELM	Hossain Lipu et al. (2019)	The number of hidden layer neurons
LSA + NARXNN	Lipu et al. (2018)	The values of input delays, output delays and hidden layer neurons
FA + DRNN	Lipu et al. (2019)	The number of hidden layers and hidden neurons

materials, and working conditions. However, collecting wide-ranging datasets, including abundance and variety, would be laborious, difficult, and time-consuming. In addition, it is not accessible for everyone to have a laboratory to conduct battery tests or purchase data (dos Reis et al., 2021). Fortunately, with the development of the energy storage field and all the effort of researchers, more public datasets are available, which can be used to build, validate, modify, and optimize the algorithm. In addition, using the same dataset is the premise of comparing different algorithms (Manoharan Aaruththiran et al., 2021). Several publicly available datasets are summarized in Table 7, including the datasets of two types of 18 650 LiBs provided by Phillip Kollmeyer at McMaster University (Kollmeyer, 2018; Naguib et al., 2020), the dataset from William C. Chueh and Richard D. Braatz (Severson et al., 2019), the dataset from the CALCE at the University of Maryland (University of Maryland, 2021), the dataset form Research & development data repository from Sandia National Labs (Barkholtz et al., 2017), the dataset from NASA Ames Prognostic Data Repository (Saha and Goebel, 2007), and the Oxford battery degradation dataset from the Howey research group (Christoph, 2017). For more details, refer to the literature (dos Reis et al., 2021). In addition, other researchers are committed to improving the data shortage issues by means of alternative solutions, such as migration-based ML methods (Tang et al., 2021) and DNN-based methods (Herle et al., 2021).

Third, high-quality datasets depend on data variety. Although simulating experiments under various conditions in the laboratories can result in a large amount of data, this does not contribute substantially to the diversity of data. Most of the studies only used the datasets collected from laboratories, such as the constant charge or discharge test, and other widely used dynamic driving cycles, like FUDS, UDDS, and US06. Only limited studies (Jiménez-Bermejo et al., 2018; Zhao et al., 2020) used the datasets collected from real EV applications. These standard driving cycles are simulated data, and their regular patterns are significantly different from real-world operations. From this point of view, algorithms that are trained only using the datasets from laboratories may show poor performance under real EV applications, especially under extreme conditions. Therefore, collecting and using the datasets from real EV applications to train an ML algorithm is a promising way to improve and guarantee the performance of the state estimation algorithm. In addition, most ML algorithms use the measured current, voltage, and temperature as inputs. Other kinds of measurements are ignored. With the existing and emerging sensing techniques, such as the optical Fiber Bragg Grating (FBG) sensor (Peng et al., 2019) to measure volumetric change, the resistance temperature detector (RTD) (Zhu et al., 2020), and thermocouples (Parhizi et al., 2017) to measure internal temperature, so the data variety can be enlarged considerably.

In conclusion, improving the precision, abundance, and variety of training data is a basic but important task for developing ML algorithms in the future.

4.2. Structure selection and hyperparameter tuning

Optimal structure selection and hyperparameter tuning are other challenging tasks for ML algorithms because the performance of the ML algorithm is sensitive to network architecture and hyperparameter selection. Take the NN as an example; the more complex the structure, usually, the higher the number of parameters that need to be trained. The more hidden layers, the higher the computational burden required to train and implement the model. So far, most of the architecture selection has been constrained to the trial-and-error method, which is inefficient and wastes time and energy. For example, the number of hidden layer neurons could be determined by a singular value decomposition approach (Teoh et al., 2006). For hyperparameter tuning, which includes the learning rate, neuron initialization, optimization algorithm, dropouts, batch size, batch normalization, and activation function, there is still no standardization strategy that can be used to guide how to choose suitable hyperparameters for differently specific problems. Therefore, the hyperparameters are also tuned by means of the trial-and-error approach until reasonable results are achieved. Some researchers (Cui et al., 2018; Song et al., 2019; Li et al., 2019c) demonstrated that different hyperparameter settings would significantly influence the performance of the algorithm, and they also compared the performance of the same algorithm under different hyperparameter settings.

Because architecture selection and hyperparameter tuning are the most laborious works for ML algorithms, it is necessary to combine other optimization algorithms to tune the hyperparameters quickly and efficiently. Recently, ML algorithms combined with heuristic optimization techniques have been proposed and recognized as an effective strategy to optimize hyperparameters. Widely used heuristic optimization techniques such as the BSA, LSA, PSO, OLS, GSA, and firefly algorithm (FA) are all combined with different ML algorithms to estimate SOC and SOH, as summarized in Table 8. Nevertheless, heuristic optimization techniques have the disadvantages of slow convergence, poor search capability, and time-consuming parameter tuning, which may degrade the performance of the algorithms. In addition, the comparison between the above heuristic optimization techniques applied in hyperparameter tuning is insufficient. Therefore, the optimal structure selection and hyperparameter tuning of the ML algorithm needs to be improved in future work.

4.3. Hybrid algorithms and ensemble learning

As reviewed in the above sections, most of the studies used a single algorithm for SOC or SOH estimation. Recently, hybrid algorithms have attracted great attention for producing performance enhancement in terms of accuracy, robustness, and efficiency. Compared with those single methods, hybrid algorithms, which are formed by integrating multiple algorithms, can make full use of the advantages of different algorithms to obtain better performance. The hybrid algorithms can be categorized into two types: the fusion of model-based and ML methods and the fusion of different ML methods, summarized in Table 9. Shallow NNs, such as the BPNN, RBFNN, and ELM, are commonly combined with the EKF (Chen et al., 2019), UKF (Sun et al., 2020), and adaptive UKF (AUKF) (Du et al., 2014) to improve the accuracy and robustness of the proposed SOC estimation algorithms. DL algorithms, such as the DNN, NARXNN, and LSTM, can be optimized by means of the UKF (He et al., 2014; Qin et al., 2019) and PF (Zhang et al., 2019b). Moreover, integrating different ML methods is a widely used strategy. For example, a CNN is usually combined with a GRU or LSTM to make full use of their advantages, namely the automatic feature extraction capabilities of a CNN and the time-sequences prediction capabilities of an RNN. In addition, LR was integrated with the RBFNN (Mao et al., 2021a) and GPR (Yu, 2018) to obtain better SOH estimation results.

Another promising strategy to improve the training accuracy, training efficiency, and model robustness is ensemble learning (EL). The main objective of EL is to reduce the risk of choosing a single learning algorithm with a poor performance and improve upon the performance of one algorithm by using an intelligent ensemble of several individual algorithms (Shen et al., 2020). One of the most popular EL algorithms is random forest (RF), which is based on bagging algorithm and cart decision tree. Bootstrap is used to generate different training datasets from the original dataset, and normally, the size of each training dataset is about two-thirds that of the original dataset. The samples that are not selected are included as part of another subset called out-of-bag (OBB) samples. Then, multiple cart decision trees are established based on these training datasets, while the OOB sample is used to evaluate the performance of the decision trees. Finally, combining the results of different decision tree, a random forest is obtained. Li et al. (2018) proposed an RF model with 500 trees for battery capacity estimation. The collected raw data, including measured current, voltage and time, can be directly fed into the trained model without any pro-processing, leading to a low computational cost. The trained RF could achieve good estimation results with an RMSE of less than 1.3%. In another study (Xu et al., 2018), four time-related features were extracted from charge–discharge profiles and temperature profiles to training the least squares regression tree model and the RF. The comparison results demonstrate that the RF performed better. Another category of EL is boosting, which improves the prediction accuracy through serial learning, such as AdaBoost and gradient boosting decision tree (GBDT). AdaBoost learns multiple weak learning machines by changing the weights of training samples based on the results of the previous study, and finally combines these weak leaning machines by voting strategy (Qin et al., 2022). Compared with single algorithm, EL has the advantages of high prediction accuracy, strong generalization ability, and less vulnerability to falling into over-fitting. However, it is time-consuming to train multiple base learners. As a result, substantial research efforts are still required to improve the overall performance of state estimation algorithms using hybrid algorithms or EL.

4.4. Evaluation and implementation of the algorithm

Overall, multi-dimensional evaluation and onboard implementation of the ML methods will be important tasks in future studies. First, when evaluating an ML algorithm, especially compared with other algorithms, it is necessary to use the same training, validating, and testing datasets. The public datasets summarized in Section 4.1 can be utilized to build SOC and SOH estimation algorithms. Furthermore, it is fair to match the model complexity (e.g., number of learnable parameters, training algorithm) between the compared algorithms as much as possible because the computational burden and memory storage are highly related to the model complexity. In addition to the model accuracy, other aspects, including the model robustness under various uncertainties (Dineva et al., 2021), training time (Feng et al., 2021), memory occupation (Bonfitto et al., 2019), FLOPs (Yang et al., 2019b), and number of parameters (Ungurean et al., 2020), are required to consider. Finding the right balance between model complexity and model accuracy is one of the main challenges in developing ML algorithms. Therefore, future studies should evaluate the proposed ML methods from a multi-dimensional perspective.

Second, most of the studies were performed using the same battery type for model training and testing. However, when applied to other types of batteries (e.g., rated capacity or positive electrode material), the offline trained ML algorithms could have bad performance. Furthermore, it is unpractical to recollect the training data and train the models for each type of battery. To address this issue, researchers have used the transfer learning technique (Bhattacharjee et al., 2021; Kim et al., 2021) in combination with ML methods for different batteries recently. Thus, further investigation is required to improve the generalization capability of ML methods for different batteries.

Third, the onboard implementation of ML state estimation algorithms has been scarcely researched owing to the computational burden. With the development of big data, 5G technology, cloud computation platforms, and other emerging techniques, the onboard implementation of ML algorithms is a promising candidate for the next-generation BMSs. Big data and IoT techniques can provide a large amount of available data, which is the premise of ML algorithms. Moreover, cloud computing provides the possibility to handle such large-scale data. In addition, the popularity of GPUs lessens the computational burden of training ML algorithms offline. From another point of view, online retraining of ML state estimation algorithms is important for the following reasons. First, as LiBs age, the performance of the state estimation algorithms that are trained offline degrades. Second, the ML state estimation algorithms can obtain good estimation results under specific working conditions, but the working conditions of EVs are greatly different from training conditions. Furthermore, online retraining of ML state estimation algorithms is challenging, as it requires the specific computing power of the platform and a certain amount of data to guarantee the accuracy and stability of the algorithms. In short, future work in this field should focus more on the onboard implementation of ML state estimation algorithms.

5. Conclusion

State estimation has always been a research hotspot in the field of BMS applications, especially in SOC and SOH estimation. Review papers regarding these two topics have been published continually over time. However, most of the reviews mainly focus on the general approaches for SOC and SOH estimation, while ML methods are not explained in depth. Therefore, to bridge the research gap, this paper comprehensively and systematically

Table 9
Summary of the applications of hybrid algorithms for state estimation.

Fusion types	Refs.	Hybrid methods
The fusion of model-based and ML methods	Chen et al. (2019)	BPNN + EKF
	Sun et al. (2020)	RBFNN + UKF
	Du et al. (2014)	ELM + AUKF
	He et al. (2014)	DNN + UKF
	Qin et al. (2019)	NARX + UKF
	Xu et al. (2020), Tian et al. (2020a) and Zhang et al. (2019b)	LSTM + SPKF, LSTM + ACKF, LSTM + PF
	Lyu and Gao (2020)	GPR + RLS
The fusion of different ML methods	Mao et al. (2021a)	LR + RBFNN
	Huang et al. (2019) and Fan et al. (2020)	CNN + GRU
	Song et al. (2019) and Li et al. (2022)	CNN + LSTM
	Yang (2021)	3D CNN + 2D CNN
	Yu (2018)	LR + GPR

reviews the detailed applications of different ML methods for SOC and SOH estimation. As a first contribution, this review explains the typical procedures of the ML SOC and SOH estimation methods. As a second contribution, the basic principles of four commonly used categories of ML algorithms, that is, shallow NN, DL, SVM, and GPR methods, are introduced with mathematical equations, uniform illustrations, and diagrams to give readers a clear understanding. Typical algorithms of shallow NNs and DL, such as BPNN, ELM, LSTM, and CNN, are also comprehensively introduced. As a third contribution, typical applications of these four categories of ML algorithms applied to SOC and SOH estimation are systematically reviewed following the explanation of principles. Furthermore, the details, such as the used datasets, input features, hyperparameters selection, and performance metrics, are summarized in two tables for easy reference and comparison. In addition, the advantages and disadvantages of each algorithm, as well as the commonalities between different applications, are analyzed. As a fourth contribution, this review explores the key challenges and prospects of the ML state estimation methods including data quality, structure selection and hyperparameters tuning, hybrid framework and EL, and the execution of the algorithms. Several important future directions are provided regarding improving data quality from the aspects of precision, abundance, and variety, optimizing structure selection and hyperparameter tuning, applying advanced learning strategy to improve model accuracy and robustness, evaluating the ML methods from a multi-dimension, and investigating onboard implementation of the ML state estimation methods.

In summary, the ML SOC and SOH estimation algorithms have shown enormous potential for future BMSs. However, in the current stage, there remain several challenges in their practical applications. Technologies that include advanced sensing to improve data variety, fusion models and multi-states joint estimation, cloud computation and online retraining, full life cycle prediction, and onboard implementation are expected to be hotspot topics of future research. We hope that through this paper, interested researchers will be inspired to improve the applications of ML algorithms in SOC and SOH estimation.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by Key R&D project of Hubei Province China (grant number 2020BAB132), Foshan Xianhu Laboratory of the Advanced Energy Science and Technology Guangdong Laboratory (grant number XHD2020-003), and the 111 Project (grant number B17034).

References

Alvarez Anton, J.C., Garcia Nieto, P.J., Blanco Viejo, C., Vilan Vilan, J.A., 2013. Support vector machines used to estimate the battery state of charge. IEEE Trans. Power Electron. 28 (12), 5919–5926. http://dx.doi.org/10.1109/TPEL.2013.2243918.

Álvarez Antón, J.C., García Nieto, P.J., de Cos Juez, F.J., Sánchez Lasheras, F., González Vega, M., Roqueñí Gutiérrez, M.N., 2013. Battery state-of-charge estimator using the SVM technique. Appl. Math. Model. 37 (9), 6244–6253. http://dx.doi.org/10.1016/j.apm.2013.01.024.

Attia, P.M., et al., 2020. Closed-loop optimization of fast-charging protocols for batteries with machine learning. Nature 578 (7795), 397–402. http://dx.doi.org/10.1038/s41586-020-1994-5.

Barkholtz, H.M., Fresquez, A., Chalamala, B.R., Ferreira, S.R., 2017. A database for comparative electrochemical performance of commercial 18650-format lithium-ion cells. J. Electrochem. Soc. 164 (12), A2697–A2706. http://dx.doi.org/10.1149/2.1701712jes.

Bhattacharjee, A., Verma, A., Mishra, S., Saha, T.K., 2021. Estimating state of charge for xEV batteries using 1D convolutional neural networks and transfer learning. IEEE Trans. Veh. Technol. 70 (4), 3123–3135. http://dx.doi.org/10.1109/TVT.2021.3064287.

Bian, C., He, H., Yang, S., 2020. Stacked bidirectional long short-term memory networks for state-of-charge estimation of lithium-ion batteries. Energy 191, 116538. http://dx.doi.org/10.1016/j.energy.2019.116538.

Bin Huang, G., Zhu, Q.Y., Siew, C.K., 2006. Extreme learning machine: Theory and applications. Neurocomputing 70 (1–3), 489–501. http://dx.doi.org/10.1016/J.NEUCOM.2005.12.126.

Bonfitto, A., Feraco, S., Tonoli, A., Amati, N., Monti, F., 2019. Estimation accuracy and computational cost analysis of artificial neural networks for state of charge estimation in lithium batteries. Batteries 5 (2), http://dx.doi.org/10.3390/batteries5020047.

Cai, L., Meng, J., Stroe, D., Luo, G., Teodorescu, R., 2019. An evolutionary framework for lithium-ion battery state of health estimation. J. Power Sources 412 (November 2018), 615–622. http://dx.doi.org/10.1016/j.jpowsour.2018.12.001.

Campestrini, C., Heil, T., Kosch, S., Jossen, A., 2016. A comparative study and review of different Kalman filters by applying an enhanced validation method. J. Energy Storage 8, 142–159. http://dx.doi.org/10.1016/j.est.2016.10.004.

Chandran, V., Patil, C.K., Karthick, A., Ganeshaperumal, D., Rahim, R., Ghosh, A., 2021. State of charge estimation of lithium-ion battery for electric vehicles using machine learning algorithms. World Electr. Veh. J. 12 (1), http://dx.doi.org/10.3390/wevj12010038.

Chang, W.Y., 2013. Estimation of the state of charge for a LFP battery using a hybrid method that combines a RBF neural network, an OLS algorithm and AGA. Int. J. Electr. Power Energy Syst. 53 (1), 603–611. http://dx.doi.org/10.1016/j.ijepes.2013.05.038.

Chemali, E., Kollmeyer, P.J., Preindl, M., Emadi, A., 2018. State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach. J. Power Sources 400, 242–255. http://dx.doi.org/10.1016/j.jpowsour.2018.06.104.

Chen, H.Y., Liang, J.W., 2017. Adaptive wavelet neural network controller for active suppression control of a diaphragm-type pneumatic vibration isolator. Int. J. Control Autom. Syst. 15 (3), 1456–1465. http://dx.doi.org/10.1007/s12555-014-0428-2.

Chen, L., Lü, Z., Lin, W., Li, J., Pan, H., 2018a. A new state-of-health estimation method for lithium-ion batteries through the intrinsic relationship between ohmic internal resistance and capacity. Measurement 116, 586–595. http://dx.doi.org/10.1016/J.MEASUREMENT.2017.11.016.

Chen, Z., Sun, M., Shu, X., Xiao, R., Shen, J., 2018b. Online state of health estimation for lithium-ion batteries based on support vector machine. Appl. Sci. 8 (6), http://dx.doi.org/10.3390/app8060925.

Chen, L., Wang, H., Liu, B., Wang, Y., Ding, Y., Pan, H., 2021. Battery state-of-health estimation based on a metabolic extreme learning machine combining degradation state model and error compensation. Energy 215, 119078. http://dx.doi.org/10.1016/J.ENERGY.2020.119078.

Chen, C., Xiong, R., Yang, R., Shen, W., Sun, F., 2019. State-of-charge estimation of lithium-ion battery using an improved neural network model and extended Kalman filter. J. Clean. Prod. 234, 1153–1164. http://dx.doi.org/10.1016/J.JCLEPRO.2019.06.273.

Chen, L., et al., 2018c. A novel state-of-charge estimation method of lithium-ion batteries combining the grey model and genetic algorithms. IEEE Trans. Power Electron. 33 (10), 8797–8807. http://dx.doi.org/10.1109/TPEL.2017.2782721.

Chin, C.S., Gao, Z., 2018. State-of-charge estimation of battery pack under varying ambient temperature using an adaptive sequential extreme learning machine. Energies 11 (4), http://dx.doi.org/10.3390/en11040711.

Christoph, R.B., 2017. Diagnosis and Prognosis of Degradation in Lithium-Ion Batteries (Ph. D. Thesis). Department of Engineering Science, University of Oxford.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. [Online]. Available: http://arxiv.org/abs/1412.3555.

2022. COP26 declaration on accelerating the transition to 100% zero emission cars and vans - GOV.UK. https://www.gov.uk/government/publications/cop26-declaration-zero-emission-cars-and-vans/cop26-declaration-on-accelerating-the-transition-to-100-zero-emission-cars-and-vans (accessed Jan. 24, 2022).

Crocioni, G., Pau, D., Delorme, J.M., Gruosso, G., 2020. Li-ion batteries parameter estimation with tiny neural networks embedded on intelligent IoT microcontrollers. IEEE Access 8, 122135–122146. http://dx.doi.org/10.1109/ACCESS.2020.3007046.

Cui, Z., Wang, C., Gao, X., Tian, S., 2021. State of health estimation for lithium-ion battery based on the coupling-loop nonlinear autoregressive with exogenous inputs neural network. Electrochim. Acta 393, 139047. http://dx.doi.org/10.1016/j.electacta.2021.139047.

Cui, Z., Wang, L., Li, Q., Wang, K., 2022. A comprehensive review on the state of charge estimation for lithium-ion battery based on neural network. Int. J. Energy Res. 46 (5), 5423–5440. http://dx.doi.org/10.1002/er.7545.

Cui, D., et al., 2018. A novel intelligent method for the state of charge estimation of lithium-ion batteries using a discrete wavelet transform-based wavelet neural network. Energies 11 (4), http://dx.doi.org/10.3390/en11040995.

Dai, H., Jiang, B., Hu, X., Lin, X., Wei, X., Pecht, M., 2021. Advanced battery management strategies for a sustainable energy future: Multilayer design concepts and research trends. Renew. Sustain. Energy Rev. 138, 110480. http://dx.doi.org/10.1016/j.rser.2020.110480, Elsevier Ltd.

Deng, Z., Hu, X., Lin, X., Che, Y., Xu, L., Guo, W., 2020. Data-driven state of charge estimation for lithium-ion battery packs based on Gaussian process regression. Energy 205, http://dx.doi.org/10.1016/j.energy.2020.118000.

Dineva, A., Csomós, B., Kocsis Sz, S., Vajda, I., 2021. Investigation of the performance of direct forecasting strategy using machine learning in State-of-Charge prediction of Li-ion batteries exposed to dynamic loads. J. Energy Storage 36, 102351. http://dx.doi.org/10.1016/j.est.2021.102351.

Ding, S., Zhao, H., Zhang, Y., Xu, X., Nie, R., 2015. Extreme learning machine: algorithm, theory and applications. Artif. Intell. Rev. 44 (1), 103–115. http://dx.doi.org/10.1007/s10462-013-9405-z.

dos Reis, G., Strange, C., Yadav, M., Li, S., 2021. Lithium-ion battery data and where to find it. Energy AI 5, 100081. http://dx.doi.org/10.1016/j.egyai.2021.100081, Elsevier.

Du, J., Liu, Z., Wang, Y., 2014. State of charge estimation for Li-ion battery based on model from extreme learning machine. Control Eng. Pract. 26 (1), 11–19. http://dx.doi.org/10.1016/J.CONENGPRAC.2013.12.014.

Dubarry, M., Truchot, C., Liaw, B.Y., 2012. Synthesize battery degradation modes via a diagnostic and prognostic model. J. Power Sources 219, 204–216. http://dx.doi.org/10.1016/j.jpowsour.2012.07.016.

Elman, J.L., 1990. Finding structure in time. Cogn. Sci. 14 (2), 179–211. http://dx.doi.org/10.1016/0364-0213(90)90002-E.

2022. EV-Volumes - The electric vehicle world sales database. https://www.ev-volumes.com/country/total-world-plug-in-vehicle-volumes/ (accessed Feb. 27, 2022).

Fan, Y., Xiao, F., Li, C., Yang, G., Tang, X., 2020. A novel deep learning framework for state of health estimation of lithium-ion battery. J. Energy Storage 32, 101741. http://dx.doi.org/10.1016/j.est.2020.101741.

Fasahat, M., Manthouri, M., 2020. State of charge estimation of lithium-ion batteries using hybrid autoencoder and Long Short Term Memory neural networks. J. Power Sources 469, 228375. http://dx.doi.org/10.1016/j.jpowsour.2020.228375.

Feng, X., Chen, J., Zhang, Z., Miao, S., Zhu, Q., 2021. State-of-charge estimation of lithium-ion battery based on clockwork recurrent neural network. Energy 236, 121360. http://dx.doi.org/10.1016/j.energy.2021.121360.

Feng, X., et al., 2018. Incremental capacity analysis on commercial lithium-ion batteries using support vector regression: A parametric study. Energies 11 (9), http://dx.doi.org/10.3390/en11092323.

Feng, F., et al., 2020. Co-estimation of lithium-ion battery state of charge and state of temperature based on a hybrid electrochemical-thermal-neural-network model. J. Power Sources 455, 227935. http://dx.doi.org/10.1016/J.JPOWSOUR.2020.227935.

Finegan, D.P., et al., 2021. The application of data-driven methods and physics-based learning for improving battery safety. Joule 5 (2), 316–329. http://dx.doi.org/10.1016/j.joule.2020.11.018.

Fletcher, T., 2009. Support vector machines explained. In: Tutor. Pap.. pp. 1–19, [Online]. Available: http://sutikno.blog.undip.ac.id/files/2011/11/SVM-Explained.pdf.

Gao, R.F., Ji, C.X., Qiang, X.J., Cheng, G.J., Liu, Y., 2016. Rock thin section image classification research from shallow network to deep neural network. In: Proceedings of the 2016 International Conference on Education, Management and Computer Science (ICEMC 2016), Vol. 129, no. International Conference on Education, Management and Computer Science (ICEMC). pp. 620–625.

Gao, T., Lu, W., 2021. Machine learning toward advanced energy storage devices and systems. iScience 24 (1), 101936. http://dx.doi.org/10.1016/j.isci.2020.101936, Elsevier.

Ge, M.F., Liu, Y., Jiang, X., Liu, J., 2021. A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries. Meas. J. Int. Meas. Confed. 174 (January), 109057. http://dx.doi.org/10.1016/j.measurement.2021.109057.

Gong, Q., Wang, P., Cheng, Z., 2022. An encoder–decoder model based on deep learning for state of health estimation of lithium-ion battery. J. Energy Storage 46 (December 2021), 103804. http://dx.doi.org/10.1016/j.est.2021.103804.

Guo, Y.F., Huang, K., Hu, X.Y., 2021. A state-of-health estimation method of lithium-ion batteries based on multi-feature extracted from constant current charging curve. J. Energy Storage 36, 102372. http://dx.doi.org/10.1016/j.est.2021.102372.

Han, W., Wik, T., Kersten, A., Dong, G., Zou, C., 2020. Next-generation battery management systems: Dynamic reconfiguration. IEEE Ind. Electron. Mag. 14 (4), 20–31. http://dx.doi.org/10.1109/MIE.2020.3002486.

Han, X., et al., 2019. A review on the key issues of the lithium ion battery degradation among the whole life cycle. eTransportation 1, 100005. http://dx.doi.org/10.1016/j.etran.2019.100005.

Hannan, M.A., Lipu, M.S.H., Hussain, A., Saad, M.H., Ayob, A., 2018. Neural network approach for estimating state of charge of lithium-ion battery using backtracking search algorithm. IEEE Access 6, 10069–10079. http://dx.doi.org/10.1109/ACCESS.2018.2797976.

Hannan, M.A., et al., 2020. Toward enhanced state of charge estimation of lithium-ion batteries using optimized machine learning techniques. Sci. Rep. 10 (1), 1–15. http://dx.doi.org/10.1038/s41598-020-61464-7.

Hannan, M.A., et al., 2021. SOC estimation of li-ion batteries with learning rate-optimized deep fully convolutional network. IEEE Trans. Power Electron. 36 (7), 7349–7353. http://dx.doi.org/10.1109/TPEL.2020.3041876.

He, W., Williard, N., Chen, C., Pecht, M., 2014. State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation. Int. J. Electr. Power Energy Syst. 62, 783–791. http://dx.doi.org/10.1016/j.ijepes.2014.04.059.

Herle, A., Channegowda, J., Prabhu, D., 2021. Overcoming limited battery data challenges: A coupled neural network approach. Int. J. Energy Res. http://dx.doi.org/10.1002/er.7081.

Hossain Lipu, M.S., Hannan, M.A., Hussain, A., Saad, M.H., Ayob, A., Uddin, M.N., 2019. Extreme learning machine model for state-of-charge estimation of lithium-ion battery using gravitational search algorithm. IEEE Trans. Ind. Appl. 55 (4), 4225–4234. http://dx.doi.org/10.1109/TIA.2019.2902532.

Hossain Lipu, M.S., Hussain, A., Saad, M.H.M., Ayob, A., Hannan, M.A., 2018. Improved recurrent NARX neural network model for state of charge estimation of lithium-ion battery using pso algorithm. In: ISCAIE 2018-2018 IEEE Symp. Comput. Appl. Ind. Electron.. pp. 354–359. http://dx.doi.org/10.1109/ISCAIE.2018.8405498.

How, D.N.T., Hannan, M.A., Lipu, M.S.H., Sahari, K.S.M., Ker, P.J., Muttaqi, K.M., 2020. State-of-charge estimation of li-ion battery in electric vehicles: A deep neural network approach. IEEE Trans. Ind. Appl. 56 (5), 5565–5574. http://dx.doi.org/10.1109/TIA.2020.3004294.

Hu, X., Feng, F., Liu, K., Zhang, L., Xie, J., Liu, B., 2019. State estimation for advanced battery management: Key challenges and future trends. Renew. Sustain. Energy Rev. 114 (April), 109334. http://dx.doi.org/10.1016/j.rser.2019.109334.

Hu, X., Jiang, H., Feng, F., Liu, B., 2020. An enhanced multi-state estimation hierarchy for advanced lithium-ion battery management. Appl. Energy 257 (October 2019), 114019. http://dx.doi.org/10.1016/j.apenergy.2019.114019.

Hu, X., Li, S.E., Yang, Y., 2016. Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles. IEEE Trans. Transp. Electrif. 2 (2), 140–149. http://dx.doi.org/10.1109/TTE.2015.2512237.

Hu, Y., Wang, Z., 2019. Study on soc estimation of lithium battery based on improved bp neural network. In: 2019 8th Int. Symp. Next Gener. Electron. ISNE 2019. pp. 2019–2021. http://dx.doi.org/10.1109/ISNE.2019.8896605.

Hu, J.N., et al., 2014. State-of-charge estimation for battery management system using optimized support vector machine for regression. J. Power Sources 269, 682–693. http://dx.doi.org/10.1016/j.jpowsour.2014.07.016.

Huang, Z., Yang, F., Xu, F., Song, X., Tsui, K.L., 2019. Convolutional gated recurrent unit-recurrent neural network for state-of-charge estimation of lithium-ion batteries. IEEE Access 7, 93139–93149. http://dx.doi.org/10.1109/ACCESS.2019.2928037.

Jain, A.K., Mao, J., Mohiuddin, K.M., 1996. Artificial neural networks: a tutorial. Comput. (Long Beach Calif). 29 (3), 31–44. http://dx.doi.org/10.1109/2.485891.

Jia, B., Guan, Y., Wu, L., 2019. A state of health estimation framework for lithium-ion batteries using transfer components analysis. Energies 12 (13), http://dx.doi.org/10.3390/en12132524.

Jia, J., Liang, J., Shi, Y., Wen, J., Pang, X., Zeng, J., 2020. SOH and RUL prediction of lithium-ion batteries based on Gaussian process regression with indirect health indicators. Energies 13 (2), 375, [Online]. Available: https://doi.org/10.3390/en13020375.

Jiang, B., et al., 2022. Fast charging design for Lithium-ion batteries via Bayesian optimization. Appl. Energy 307 (July 2021), 118244. http://dx.doi.org/10.1016/j.apenergy.2021.118244.

Jiménez-Bermejo, D., Fraile-Ardanuy, J., Castaño-Solis, S., Merino, J., Álvaro-Hermana, R., 2018. Using dynamic neural networks for battery state of charge estimation in electric vehicles. Procedia Comput. Sci. 130, 533–540. http://dx.doi.org/10.1016/j.procs.2018.04.077.

Jordan, M.I., 1986. Serial Order: A Parallel Distributed Processing Approach-Ies Report 8604, No. 667. Institute for Cognitive Science University of California, San Diego.

Kashkooli, A.G., Fathiannasab, H., Mao, Z., Chen, Z., 2019. Application of artificial intelligence to state-of-charge and state-of-health estimation of calendar-aged lithium-ion pouch cells. J. Electrochem. Soc. 166 (4), A605–A615. http://dx.doi.org/10.1149/2.0411904jes.

Kaur, K., Garg, A., Cui, X., Singh, S., Panigrahi, B.K., 2021. Deep learning networks for capacity estimation for monitoring SOH of Li-ion batteries for electric vehicles. Int. J. Energy Res. 45 (2), 3113–3128. http://dx.doi.org/10.1002/er.6005.

Khaleghi, S., et al., 2021. Online health diagnosis of lithium-ion batteries based on nonlinear autoregressive neural network. Appl. Energy 282, 116159. http://dx.doi.org/10.1016/j.apenergy.2020.116159.

Kim, S., Choi, Y.Y., Kim, K.J., Il Choi, J., 2021. Forecasting state-of-health of lithium-ion batteries using variational long short-term memory with transfer learning. J. Energy Storage 41, 102893. http://dx.doi.org/10.1016/j.est.2021.102893.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. In: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.. Accessed: Sep. 19, 2021. [Online]. Available: https://arxiv.org/abs/1412.6980v9.

Kollmeyer, P., 2018. Panasonic 18650PF li-ion battery data. In: Mendeley Data, Vol. 1. http://dx.doi.org/10.17632/WYKHT8Y7TG.1.

Li, Z., Huang, J., Liaw, B.Y., Zhang, J., 2017. On state-of-charge determination for lithium-ion batteries. J. Power Sources 348, 281–301. http://dx.doi.org/10.1016/j.jpowsour.2017.03.001.

Li, X., Wang, Z., Yan, J., 2019a. Prognostic health condition for lithium battery using the partial incremental capacity and Gaussian process regression. J. Power Sources 421 (March), 56–67. http://dx.doi.org/10.1016/j.jpowsour.2019.03.008.

Li, X., Wang, Z., Zhang, L., 2019b. Co-estimation of capacity and state-of-charge for lithium-ion batteries in electric vehicles. Energy 174, 33–44. http://dx.doi.org/10.1016/j.energy.2019.02.147.

Li, C., Xiao, F., Fan, Y., 2019c. An approach to state of charge estimation of lithium-ion batteries based on recurrent neural networks with gated recurrent unit. Energies 12 (9), http://dx.doi.org/10.3390/en12091592.

Li, X., Yuan, C., Li, X., Wang, Z., 2020a. State of health estimation for Li-Ion battery using incremental capacity analysis and Gaussian process regression. Energy 190, 116467. http://dx.doi.org/10.1016/j.energy.2019.116467.

Li, X., Yuan, C., Wang, Z., 2020b. State of health estimation for Li-ion battery via partial incremental capacity analysis based on support vector regression. Energy 203, 117852. http://dx.doi.org/10.1016/j.energy.2020.117852, Elsevier Ltd.

Li, Y., et al., 2018. Random forest regression for online capacity estimation of lithium-ion batteries. Appl. Energy 232 (September), 197–210. http://dx.doi.org/10.1016/j.apenergy.2018.09.182.

Li, Y., et al., 2019d. Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review. Renew. Sustain. Energy Rev. 113, 109254. http://dx.doi.org/10.1016/j.rser.2019.109254, Elsevier Ltd.

Li, R., et al., 2020c. State of charge prediction algorithm of lithium-ion battery based on PSO-SVR cross validation. IEEE Access 8, 10234–10242. http://dx.doi.org/10.1109/ACCESS.2020.2964852.

Li, P., et al., 2022. An end-to-end neural network framework for state-of-health estimation and remaining useful life prediction of electric vehicle lithium batteries. Renew. Sustain. Energy Rev. 156 (July 2021), 111843. http://dx.doi.org/10.1016/j.rser.2021.111843.

Lin, M., Zeng, X., Wu, J., 2021. State of health estimation of lithium-ion battery based on an adaptive tunable hybrid radial basis function network. J. Power Sources 504, 230063. http://dx.doi.org/10.1016/J.JPOWSOUR.2021.230063.

Lipu, M.S.H., Hannan, M.A., Hussain, A., Saad, M.H.M., Ayob, A., Blaabjerg, F., 2018. State of charge estimation for lithium-ion battery using recurrent NARX neural network model based lighting search algorithm. IEEE Access 6, 28150–28161. http://dx.doi.org/10.1109/ACCESS.2018.2837156.

Lipu, M.S.H., Hannan, M.A., Hussain, A., Saad, M.H.M., Ayob, A., Muttaqi, K.M., 2019. Lithium-ion battery state of charge estimation method using optimized deep recurrent neural network algorithm. In: 2019 IEEE Ind. Appl. Soc. Annu. Meet. IAS 2019. pp. 1–9. http://dx.doi.org/10.1109/IAS.2019.8912322.

Liu, Y., He, Y., Bian, H., Guo, W., Zhang, X., 2022. A review of lithium-ion battery state of charge estimation based on deep learning: Directions for improvement and future trends. J. Energy Storage 52 (PA), 104664. http://dx.doi.org/10.1016/j.est.2022.104664.

Liu, Z., Zhao, J., Wang, H., Yang, C., 2020. A new lithium-ion battery SOH estimation method based on an indirect enhanced health indicator and support vector regression in PHMs. Energies 13 (4), 830. http://dx.doi.org/10.3390/en13040830.

Liu, Y., et al., 2021. State of charge prediction framework for lithium-ion batteries incorporating long short-term memory network and transfer learning. J. Energy Storage 37 (March), http://dx.doi.org/10.1016/j.est.2021.102494.

Lyu, Z., Gao, R., 2020. Li-ion battery state of health estimation through Gaussian process regression with thevenin model. Int. J. Energy Res. 44 (13), 10262–10281. http://dx.doi.org/10.1002/er.5647.

Ma, L., Hu, C., Cheng, F., 2021. State of charge and state of energy estimation for lithium-ion batteries based on a long short-term memory neural network. J. Energy Storage 37, 102440. http://dx.doi.org/10.1016/j.est.2021.102440.

Ma, C., et al., 2019. State of health prediction for lithium-ion batteries using multiple- view feature fusion and support vector regression ensemble. Int. J. Mach. Learn. Cybern. 10 (9), 2269–2282. http://dx.doi.org/10.1007/s13042-018-0865-y.

Mahmoudzadeh Andwari, A., Pesiridis, A., Rajoo, S., Martinez-Botas, R., Esfahanian, V., 2017. A review of Battery Electric Vehicle technology and readiness levels. Renew. Sustain. Energy Rev. 78, 414–430. http://dx.doi.org/10.1016/j.rser.2017.03.138, Pergamon.

Manoharan Aaruththiran, D.S., Begam, K.M., Aparow, Vimal Rau, 2021. Performance analysis on artificial neural network based state of charge estimation for electric vehicles. In: IEEE Int. Conf. Internet Things Intell. Syst. (IoTaIS 2021).

Mao, L., Hu, H., Chen, J., Zhao, J., Qu, K., Jiang, L., 2021a. Online state of health estimation method for lithium-ion battery based on CEEMDAN for feature analysis and RBF neural network. IEEE J. Emerg. Sel. Top. Power Electron. 6777 (c), http://dx.doi.org/10.1109/JESTPE.2021.3106708.

Mao, J., Miao, J., Lu, Y., Tong, Z., 2021b. Machine learning of materials design and state prediction for lithium ion batteries. Chinese J. Chem. Eng. http://dx.doi.org/10.1016/j.cjche.2021.04.009.

Meng, J., Cai, L., Luo, G., Stroe, D.I., Teodorescu, R., 2018. Lithium-ion battery state of health estimation with short-term current pulse test and support vector machine. Microelectron. Reliab. 88–90 (September), 1216–1220. http://dx.doi.org/10.1016/j.microrel.2018.07.025.

Naguib, M., Kollmeyer, P., Skells, M., 2020. LG 18650hg2 li-ion battery data and example deep neural network xEV SOC estimator script. In: Mendeley Data, V3. Mendeley Data, http://dx.doi.org/10.17632/CP3473X7XV.3.

Ng, M.-F., Zhao, J., Yan, Q., Conduit, G.J., Seh, Z.W., 2020. Predicting the state of charge and health of batteries using data-driven machine learning. Nat. Mach. Intell. 2 (3), 161–170. http://dx.doi.org/10.1038/s42256-020-0156-7.

Niri, M.F., et al., 2022. Quantifying key factors for optimised manufacturing of li-ion battery anode and cathode via artificial intelligence. Energy AI 7, 100129. http://dx.doi.org/10.1016/J.EGYAI.2021.100129.

Ozcan, G., Pajovic, M., Sahinoglu, Z., Wang, Y., Orlik, P.V., Wada, T., 2016. Online state of charge estimation for lithium-ion batteries using Gaussian process regression. In: IECON Proc. (Industrial Electron. Conf.. pp. 998–1003. http://dx.doi.org/10.1109/IECON.2016.7793002.

Pan, H., Lü, Z., Wang, H., Wei, H., Chen, L., 2018. Novel battery state-of-health online estimation method using multiple health indicators and an extreme learning machine. Energy 160, 466–477. http://dx.doi.org/10.1016/j.energy.2018.06.220.

Parhizi, M., Ahmed, M.B., Jain, A., 2017. Determination of the core temperature of a li-ion cell during thermal runaway. J. Power Sources 370, 27–35. http://dx.doi.org/10.1016/J.JPOWSOUR.2017.09.086.

Peng, J., Zhou, X., Jia, S., Jin, Y., Xu, S., Chen, J., 2019. High precision strain monitoring for lithium ion batteries based on fiber Bragg grating sensors. J. Power Sources 433, 226692. http://dx.doi.org/10.1016/j.jpowsour.2019.226692.

Qin, X., Gao, M., He, Z., Liu, Y., 2019. State of charge estimation for lithium-ion batteries based on NARX neural network and UKF. In: IEEE Int. Conf. Ind. Informatics, Vol. 2019-July. pp. 1706–1711. http://dx.doi.org/10.1109/INDIN41052.2019.8972319.

Qin, P., Zhao, L., Liu, Z., 2022. State of health prediction for lithium-ion battery using a gradient boosting-based data-driven method. J. Energy Storage 47 (February 2021), 103644. http://dx.doi.org/10.1016/j.est.2021.103644.

Ren, Z., Du, C., Wu, Z., Shao, J., Deng, W., 2021a. A comparative study of the influence of different open circuit voltage tests on model-based state of charge estimation for lithium-ion batteries. Int. J. Energy Res. 45 (9), 13692–13711. http://dx.doi.org/10.1002/er.6700.

Ren, X., Liu, S., Yu, X., Dong, X., 2021b. A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM. Energy 234, 121236. http://dx.doi.org/10.1016/j.energy.2021.121236.

Saha, K., Goebel, B., 2007. Battery Data Set. NASA Ames Prognostics Data Repository, NASA Ames, Moffett Field, CA, [http://ti.arc.nasa.gov/project/prognostic-data-repository].

Saha, B., Goebel, K., Poll, S., Christophersen, J., 2009. Prognostics methods for battery health monitoring using a Bayesian framework. IEEE Trans. Instrum. Meas. 58 (2), 291–296. http://dx.doi.org/10.1109/TIM.2008.2005965.

Sahinoglu, G.O., Pajovic, M., Sahinoglu, Z., Wang, Y., Orlik, P.V., Wada, T., 2018. Battery state-of-charge estimation based on regular/recurrent Gaussian process regression. IEEE Trans. Ind. Electron. 65 (5), 4311–4321. http://dx.doi.org/10.1109/TIE.2017.2764869.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Netw. 61, 85–117. http://dx.doi.org/10.1016/j.neunet.2014.09.003, Pergamon.

Severson, K.A., et al., 2019. Data-driven prediction of battery cycle life before capacity degradation. Nat. Energy 4 (5), 383–391. http://dx.doi.org/10.1038/s41560-019-0356-8.

Shen, M., Gao, Q., 2019. A review on battery management system from the modeling efforts to its multiapplication and integration. Int. J. Energy Res. 43 (10), 5042–5075. http://dx.doi.org/10.1002/er.4433.

Shen, S., Sadoughi, M., Chen, X., Hong, M., Hu, C., 2019. A deep learning method for online capacity estimation of lithium-ion batteries. J. Energy Storage 25 (June), 100817. http://dx.doi.org/10.1016/j.est.2019.100817.

Shen, S., Sadoughi, M., Li, M., Wang, Z., Hu, C., 2020. Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries. Appl. Energy 260 (July 2019), 114296. http://dx.doi.org/10.1016/j.apenergy.2019.114296.

Shi, Y., et al., 2021. The optimization of state of charge and state of health estimation for lithium-ions battery using combined deep learning and Kalman filter methods. Int. J. Energy Res. 45 (7), 11206–11230. http://dx.doi.org/10.1002/er.6601.

Shrestha, A., Mahmood, A., 2019. Review of deep learning algorithms and architectures. IEEE Access 7, 53040–53065. http://dx.doi.org/10.1109/ACCESS.2019.2912200.

Shrivastava, P., Soon, T.K., Bin Idris, M.Y.I., Mekhilef, S., 2019. Overview of model-based online state-of-charge estimation using Kalman filter family for lithium-ion batteries. Renew. Sustain. Energy Rev. 113 (2018), 109233. http://dx.doi.org/10.1016/j.rser.2019.06.040.

Shu, X., Shen, J., Li, G., Zhang, Y., Chen, Z., Liu, Y., 2021. A flexible state-of-health prediction scheme for lithium-ion battery packs with long short-term memory network and transfer learning. IEEE Trans. Transp. Electrif. 7 (4), 2238–2248. http://dx.doi.org/10.1109/TTE.2021.3074638.

Song, X., Yang, F., Wang, D., Tsui, K.L., 2019. Combined CNN-LSTM network for state-of-charge estimation of lithium-ion batteries. IEEE Access 7, 88894–88902. http://dx.doi.org/10.1109/ACCESS.2019.2926517.

Sun, W., Qiu, Y., Sun, L., Hua, Q., 2020. Neural network-based learning and estimation of battery state-of-charge: A comparison study between direct and indirect methodology. Int. J. Energy Res. 44 (13), 10307–10319. http://dx.doi.org/10.1002/er.5654.

Tang, X., Liu, K., Li, K., Widanage, W.D., Kendrick, E., Gao, F., 2021. Recovering large-scale battery aging dataset with machine learning. Patterns 2 (8), 100302. http://dx.doi.org/10.1016/j.patter.2021.100302.

Teoh, E.J., Tan, K.C., Xiang, C., 2006. Estimating the number of hidden neurons in a feedforward network using the singular value decomposition. IEEE Trans. Neural Netw. 17 (6), 1623–1629. http://dx.doi.org/10.1109/TNN.2006.880582.

Tian, Y., Lai, R., Li, X., Xiang, L., Tian, J., 2020a. A combined method for state-of-charge estimation for lithium-ion batteries using a long short-term memory network and an adaptive cubature Kalman filter. Appl. Energy 265 (March), 114789. http://dx.doi.org/10.1016/j.apenergy.2020.114789.

Tian, H., Qin, P., 2021. State of health prediction for lithium-ion batteries with a novel online sequential extreme learning machine method. Int. J. Energy Res. 45 (2), 2383–2397. http://dx.doi.org/10.1002/er.5934.

Tian, H., Qin, P., Li, K., Zhao, Z., 2020b. A review of the state of health for lithium-ion batteries: Research status and suggestions. J. Clean. Prod. 261, 120813. http://dx.doi.org/10.1016/j.jclepro.2020.120813.

Tian, J., Xiong, R., Shen, W., Lu, J., 2021. State-of-charge estimation of LiFePO4 batteries in electric vehicles: A deep-learning enabled approach. Appl. Energy 291, 116812. http://dx.doi.org/10.1016/j.apenergy.2021.116812.

2021. Understanding LSTM Networks – colah's blog. http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed Oct. 11, 2021).

Ungurean, L., Micea, M.V., Cârstoiu, G., 2020. Online state of health prediction method for lithium-ion batteries, based on gated recurrent unit neural networks. Int. J. Energy Res. 44 (8), 6767–6777. http://dx.doi.org/10.1002/er.5413.

University of Maryland, 2021. Battery data | center for advanced life cycle engineering (CALCE). In: Battery Data. https://calce.umd.edu/battery-data (accessed Sep. 21, 2021).

Vapnik, V., 1998. The support vector method of function estimation. In: Nonlinear Modeling. Springer, Boston, MA, pp. 55–85. http://dx.doi.org/10.1007/978-1-4615-5703-6_3.

Vidal, C., Malysz, P., Kollmeyer, P., Emadi, A., 2020. Machine learning applied to electrified vehicle battery state of charge and state of health estimation: state-of-the-art. IEEE Access 8, 52796–52814. http://dx.doi.org/10.1109/ACCESS.2020.2980961, Institute of Electrical and Electronics Engineers Inc..

Wang, Z., Feng, G., Zhen, D., Gu, F., Ball, A., 2021a. A review on online state of charge and state of health estimation for lithium-ion batteries in electric vehicles. Energy Rep. 7, 5141–5161. http://dx.doi.org/10.1016/j.egyr.2021.08.113.

Wang, Q., Gu, H., Ye, M., Wei, M., Xu, X., 2021b. State of charge estimation for lithium-ion battery based on NARX recurrent neural network and moving window method. IEEE Access 9, 83364–83375. http://dx.doi.org/10.1109/ACCESS.2021.3086507.

Wang, Z., Yuan, C., Li, X., 2021c. Lithium battery state-of-health estimation via differential thermal voltammetry with Gaussian process regression. IEEE Trans. Transp. Electrif. 7 (1), 16–25. http://dx.doi.org/10.1109/TTE.2020.3028784.

Wang, Y., et al., 2020. A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems. Renew. Sustain. Energy Rev. 131 (March), 110015. http://dx.doi.org/10.1016/j.rser.2020.110015.

Wei, M., Ye, M., Wang, Q., Wu, C., Ma, Y., 2021. State-of-health estimation and remaining useful life prediction of lithium-ion batteries based on extreme learning machine. J. Phys. Conf. Ser. 1983 (1), http://dx.doi.org/10.1088/1742-6596/1983/1/012058.

Widodo, A., Shim, M.C., Caesarendra, W., Yang, B.S., 2011. Intelligent prognostics for battery health monitoring based on sample entropy. Expert Syst. Appl. 38 (9), 11763–11769. http://dx.doi.org/10.1016/j.eswa.2011.03.063.

Wu, L., Pang, H., Geng, Y., Liu, X., Liu, J., Liu, K., 2022. Low-complexity state of charge and anode potential prediction for lithium-ion batteries using a simplified electrochemical model-based observer under variable load condition. Int. J. Energy Res. 46 (9), 11834–11848. http://dx.doi.org/10.1002/er.7949.

Wu, J., Zhang, C., Chen, Z., 2016. An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks. Appl. Energy 173, 134–140. http://dx.doi.org/10.1016/j.apenergy.2016.04.057.

Xia, Z., Qahouq, J.A.A., 2021. Lithium-ion battery ageing behavior pattern characterization and state-of-health estimation using data-driven method. IEEE Access 9, 98287–98304. http://dx.doi.org/10.1109/ACCESS.2021.3092743.

Xia, B., et al., 2018. State of charge estimation of lithium-ion batteries using optimized Levenberg–Marquardt wavelet neural network. Energy 153, 694–705. http://dx.doi.org/10.1016/J.ENERGY.2018.04.085.

Xiao, F., Li, C., Fan, Y., Yang, G., Tang, X., 2021. State of charge estimation for lithium-ion battery based on Gaussian process regression with deep recurrent kernel. Int. J. Electr. Power Energy Syst. 124 (June 2020), 106369. http://dx.doi.org/10.1016/j.ijepes.2020.106369.

Xiao, B., Liu, Y., Xiao, B., 2019. Accurate state-of-charge estimation approach for lithium-ion batteries by gated recurrent unit with ensemble optimizer. IEEE Access 7, 54192–54202. http://dx.doi.org/10.1109/ACCESS.2019.2913078.

Xiong, R., Li, L., Tian, J., 2018. Towards a smarter battery management system: A critical review on battery state of health monitoring methods. J. Power Sources 405, 18–29. http://dx.doi.org/10.1016/j.jpowsour.2018.10.019.

Xiong, W., Mo, Y., Yan, C., 2021. Online state-of-health estimation for second-use lithium-ion batteries based on weighted least squares support vector machine. IEEE Access 9, 1870–1881. http://dx.doi.org/10.1109/ACCESS.2020.3026552.

Xu, H., Peng, Y., Su, L., 2018. Health state estimation method of lithium ion battery based on NASA experimental data set. IOP Conf. Ser. Mater. Sci. Eng. 452 (3), http://dx.doi.org/10.1088/1757-899X/452/3/032067.

Xu, Z., Wang, J., Fan, Q., Lund, P.D., Hong, J., 2020. Improving the state of charge estimation of reused lithium-ion batteries by abating hysteresis using machine learning technique. J. Energy Storage 32, 101678. http://dx.doi.org/10.1016/j.est.2020.101678.

Yang, Y., 2021. A machine-learning prediction method of lithium-ion battery life based on charge process for different applications. Appl. Energy 292, 116897. http://dx.doi.org/10.1016/j.apenergy.2021.116897.

Yang, F., Li, W., Li, C., Miao, Q., 2019a. State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network. Energy 175, 66–75. http://dx.doi.org/10.1016/j.energy.2019.03.059.

Yang, F., Song, X., Xu, F., Tsui, K.L., 2019b. State-of-charge estimation of lithium-ion batteries via long short-term memory network. IEEE Access 7, 53792–53799. http://dx.doi.org/10.1109/ACCESS.2019.2912803.

Yang, D., Wang, Y., Pan, R., Chen, R., Chen, Z., 2018. State-of-health estimation for the lithium-ion battery based on support vector regression. Appl. Energy 227 (May 2017), 273–283. http://dx.doi.org/10.1016/j.apenergy.2017.08.096.

won You, G., Park, S., Oh, D., 2016. Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach. Appl. Energy 176, 92–103. http://dx.doi.org/10.1016/j.apenergy.2016.05.051.

Yu, J., 2018. State of health prediction of lithium-ion batteries: Multiscale logic regression and Gaussian process regression ensemble. Reliab. Eng. Syst. Saf. 174 (February), 82–95. http://dx.doi.org/10.1016/j.ress.2018.02.022.

Zhang, Y., Tang, Q., Zhang, Y., Wang, J., Stimming, U., Lee, A.A., 2020a. Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning. Nature Commun. 11 (1), http://dx.doi.org/10.1038/s41467-020-15235-7.

Zhang, S., Zhai, B., Guo, X., Wang, K., Peng, N., Zhang, X., 2019a. Synchronous estimation of state of health and remaining useful lifetime for lithium-ion battery using the incremental capacity and artificial neural networks. J. Energy Storage 26, 100951. http://dx.doi.org/10.1016/j.est.2019.100951.

Zhang, C., Zhu, Y., Dong, G., Wei, J., 2019b. Data-driven lithium-ion battery states estimation using neural networks and particle filtering. Int. J. Energy Res. 43 (14), 8230–8241. http://dx.doi.org/10.1002/er.4820.

Zhang, L., et al., 2020b. State-of-charge estimation of lithium-ion battery pack based on improved RBF neural networks. Complexity 2020, http://dx.doi.org/10.1155/2020/8840240.

Zhao, L., Yao, W., Wang, Y., Hu, J., 2020. Machine learning-based method for remaining range prediction of electric vehicles. IEEE Access 8, 212423–212441. http://dx.doi.org/10.1109/ACCESS.2020.3039815.

Zheng, X., Deng, X., 2019. State-of-health prediction for lithium-ion batteries with multiple Gaussian process regression model. IEEE Access 7, 150383–150394. http://dx.doi.org/10.1109/ACCESS.2019.2947294.

Zhou, D., Li, Z., Zhu, J., Zhang, H., Hou, L., 2020. State of health monitoring and remaining useful life prediction of lithium-ion batteries based on temporal convolutional network. IEEE Access 8, 53307–53320. http://dx.doi.org/10.1109/ACCESS.2020.2981261.

Zhou, F., Wang, L., Lin, H., Lv, Z., 2013. High accuracy state-of-charge online estimation of EV/HEV lithium batteries based on Adaptive Wavelet Neural Network. In: 2013 IEEE ECCE Asia Downunder - 5th IEEE Annual International Energy Conversion Congress and Exhibition, IEEE ECCE Asia 2013. pp. 513–517. http://dx.doi.org/10.1109/ECCE-Asia.2013.6579145.

Zhu, S., et al., 2020. A novel embedded method for in-situ measuring internal multi-point temperatures of lithium ion batteries. J. Power Sources 456, 227981. http://dx.doi.org/10.1016/J.JPOWSOUR.2020.227981.