Wu *et al. J. Mater. Inf.* **2025**, *5*, 18
DOI: 10.20517/jmi.2024.84

Journal of
Materials Informatics

Review

Open Access

Advancements in the estimation of the state of charge of lithium-ion battery: a comprehensive review of traditional and deep learning approaches

Yunhao Wu[1], **Dongxin Bai**[2], **Kai Zhang**[2], **Yong Li**[1,*], **Fuqian Yang**[3,*]

[1]School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai 201209, China.
[2]School of Aerospace Engineering and Applied Mechanics, Tongji University, Shanghai 200092, China.
[3]Materials Program, Department of Chemical and Materials Engineering, University of Kentucky, Lexington, KY 40506, USA.

[*]**Correspondence to:** Prof. Yong Li, School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, No. 2360 Jinhai Road, Shanghai 201209, China. E-mail: yongli@sspu.edu.cn; Prof. Fuqian Yang, Materials Program, Department of Chemical and Materials Engineering, University of Kentucky, 101 Main Building Lexington, Lexington, KY 40506, USA. E-mail: fuqian.yang@uky.edu

How to cite this article: Wu, Y.; Bai, D.; Zhang, K.; Li, Y.; Yang, F. Advancements in the estimation of the state of charge of lithium-ion battery: a comprehensive review of traditional and deep learning approaches. *J. Mater. Inf.* **2025**, *5*, 18. https://dx.doi.org/10.20517/jmi.2024.84

Received: 4 Dec 2024 **First Decision:** 8 Jan 2025 **Revised:** 20 Jan 2025 **Accepted:** 11 Feb 2025 **Published:** 28 Feb 2025

Academic Editor: Xingjun Liu **Copy Editor:** Ting-Ting Hu **Production Editor:** Ting-Ting Hu

Abstract

Accurately estimating the state of charge (SOC) of lithium-ion batteries is essential for optimizing battery management systems in various applications such as electric vehicles and renewable energy storage. This study explores advancements in data-driven approaches for SOC estimation, focusing on both conventional machine learning and deep learning techniques. While traditional machine learning methods offer reliable performance, they often encounter challenges with high-dimensional data and adaption to complex operational conditions. In contrast, deep learning models provide enhanced capabilities in nonlinear modeling and automated feature extraction, leading to improved accuracy and robustness. Through comprehensive evaluations across diverse scenarios, this research identifies key technical challenges and outlines future directions, including distributed training, incorporation of physical data, development of dynamic neural networks, and the establishment of standardized benchmarking protocols. These insights aim to guide the creation of more precise, efficient, and adaptive SOC estimation models, thereby advancing the reliability and effectiveness of battery management systems.

Keywords: State-of-charge estimation, lithium-ion batteries, machine learning, deep learning

INTRODUCTION

Lithium-ion batteries (LIBs), valued for their high energy density, long lifespan, and low self-discharge rate, are integral to modern technologies, powering portable electronic devices, electric vehicles (EVs), and renewable energy storage systems, propelling technological advancements and sustainable development. Despite these advantages, the widespread adoption of LIBs has also produced increasing challenges such as safety risks, capacity degradation, and the need for precise health monitoring. These challenges affect battery overall performance and pose significant safety hazards. Accurate estimation of state of charge (SOC) is essential for managing and optimizing battery performance under complex conditions and is considered one of the key steps in addressing these issues. As a key metric for determining charge levels, the accuracy of the SOC estimation directly influences the decision-making and control efficiency of the battery management system (BMS). Precise SOC estimation enhances battery usage efficiency, prolongs lifespan, and mitigates potential overcharging or over-discharging risks, thereby improving system safety and reliability. Consequently, achieving accurate SOC estimation remains a critical challenge in battery management.

Conventional SOC estimation methods, such as the open-circuit voltage (OCV) method[1], Coulomb counting[2], and equivalent circuit models (ECMs)[3], estimate the battery's SOC using voltage and current measurements combined with predefined models or formulas, but these approaches face significant limitations due to complex chemical processes, temperature variations, and aging effects in LIBs. For example, the OCV method requires long periods of battery inactivity, making real-time SOC estimation impractical[4]; Coulomb counting is susceptible to sensor inaccuracies and cumulative errors caused by battery aging[5]; and ECMs, despite considering dynamic battery behavior, involve complex model development and parameter identification, which hinder their accuracy under changing conditions[3]. Additionally, internal physical and chemical changes within LIBs, such as stress variations, charge distribution imbalance, and cyclic damage in composite electrodes, further impact SOC estimation by causing localized damage and uneven utilization of active material, as revealed by computational models[6], while experimental studies demonstrate that heterogeneous damage, including crack propagation and structural breakdown, exacerbates the challenges of accurate SOC estimation over prolonged operation[7]. These issues underscore the necessity for robust estimation techniques that consider the evolving internal states of batteries.

Recent advances in data-driven machine learning (ML) methods have enhanced SOC estimation[8,9], offering notable advantages. These methods leverage historical data to train models that learn the mapping relationship between battery characteristics/performance and SOC, bypassing the need for a complex physical modeling process. Traditional ML techniques, such as k-nearest neighbors (KNNs)[10], decision trees[11], support vector machines (SVMs)[12], extreme learning machines (ELMs)[13], and Gaussian process regression (GPR), are effective in SOC estimation with limited computational resources. These methods rely on known variables such as voltage, current, and temperature, with manually crafted voltage, current, and temperature, as well as manually crafted features for model training, providing good accuracy in specific conditions. For instance, the KNN method[14] uses charging and discharging data, decision tree algorithms and Kalman filter (KF) to enable dynamic SOC estimation[15], while the SVM, ELM and GPR methods improve nonlinear fitting and real-time performance by utilizing kernel functions and optimization algorithms[16,17]. However, these methods often depend on extensive feature selection, involve complex feature engineering processes, and exhibit limited generalization capability in complex scenarios.

The rapid advancement of deep learning (DL) has created new opportunities for SOC estimation. DL models, such as multilayer perceptrons (MLPs)[18], convolutional neural networks (CNNs)[19], and recurrent neural networks (RNNs)[20], excel in nonlinear fitting capabilities and can automatically learn feature representations, demonstrating strong performance in SOC estimation. MLPs map input variables and SOC by stacking fully connected (FC) layers[21], while CNNs capture local features in temporal signals through convolutional layers, making them effective for analyzing battery voltage and current variations over time[22]. RNNs and their variants, including long short-term memory (LSTM)[23] and gated recurrent units (GRUs)[24], are effective for sequence modeling, capturing the impact of historical measurements on the current SOC[25]. LSTM and GRUs address the vanishing and exploding gradient problems in standard RNNs through gating mechanisms, making them particularly suited for processing long sequences in SOC estimation. Despite progress in DL-based SOC estimation, challenges remain that must be addressed. A primary issue is the lack of standardized data collection, model training, and evaluation methods, which impedes comparison across different studies and the establishment of a unified benchmark. Additionally, as batteries age and their characteristics evolve, the modeling performance for SOC estimation may decline[26], highlighting the need for adaptive models.

The remainder of the paper is organized as follows. Section 2 reviews traditional ML-based SOC estimation, highlighting their principles, advantages, and limitations in practical applications. Section 3 examines recent advancements in DL methods, focusing on the roles of MLPs, CNNs, and RNNs in SOC estimation. Finally, Section 4 addresses the challenges of data-driven SOC estimation and suggests future research directions. This review aims to offer a comprehensive technical perspective to researchers, fostering the advancement of intelligent BMS.

SOC ESTIMATION BASED ON TRADITIONAL ML METHODS

Traditional ML methods are crucial for the SOC estimation of LIBs, relying on measured data rather than detailed battery or complex mathematical models. Traditional ML methods include KNNs, decision trees (such as XGBoost), SVMs, ELMs, and GPR, which use manually selected features for model training leading to SOC prediction. Despite challenges in feature selection, computational complexity, and model robustness, these methods offer effective SOC estimation, particularly when computational resources are limited. The definition of SOC of a battery is the percentage of its current available capacity to its rated total capacity, expressed as: $SOC_t = Q_t/Q_{max}$, where Q_t denotes the remaining capacity of the battery at moment *t*. Q_{max} is the total rated energy of the battery, which can be obtained by measurement or estimation[27,28]. Figure 1 illustrates the basic steps or key techniques of the four traditional ML methods described in this section for SOC estimation.

KNNs

KNN is an instance-based ML algorithm commonly applied to classification and regression tasks. It predicts unknown data points by analyzing their proximity to instances in the training dataset. For SOC estimation, KNN identifies the neighboring points by computing distances between the target and training data points. KNN algorithm typically employs metrics such as Euclidean distance or Manhattan distance to measure similarity between data points. Once the distance metric is selected, the KNNs are constructed at the target point. The SOC value is then inferred by averaging the SOC values of these neighbors, either weighted or unweighted. This simple, assumption-free method is intuitive. It does not require assumptions about data distribution and can yield effective results in certain cases.

KNN has demonstrated notable accuracy and adaptability in SOC estimation for LIBs. For instance, Talluri *et al.* used charge-discharge cycle data, including voltage, current, and time, to train and test a KNN model, achieving a 98% accuracy rate with an average absolute error of 0.74%[14]. Similarly, Ghassani *et al.* applied the KNN algorithm to smartphone charging prediction, using periodic SOC data collected to

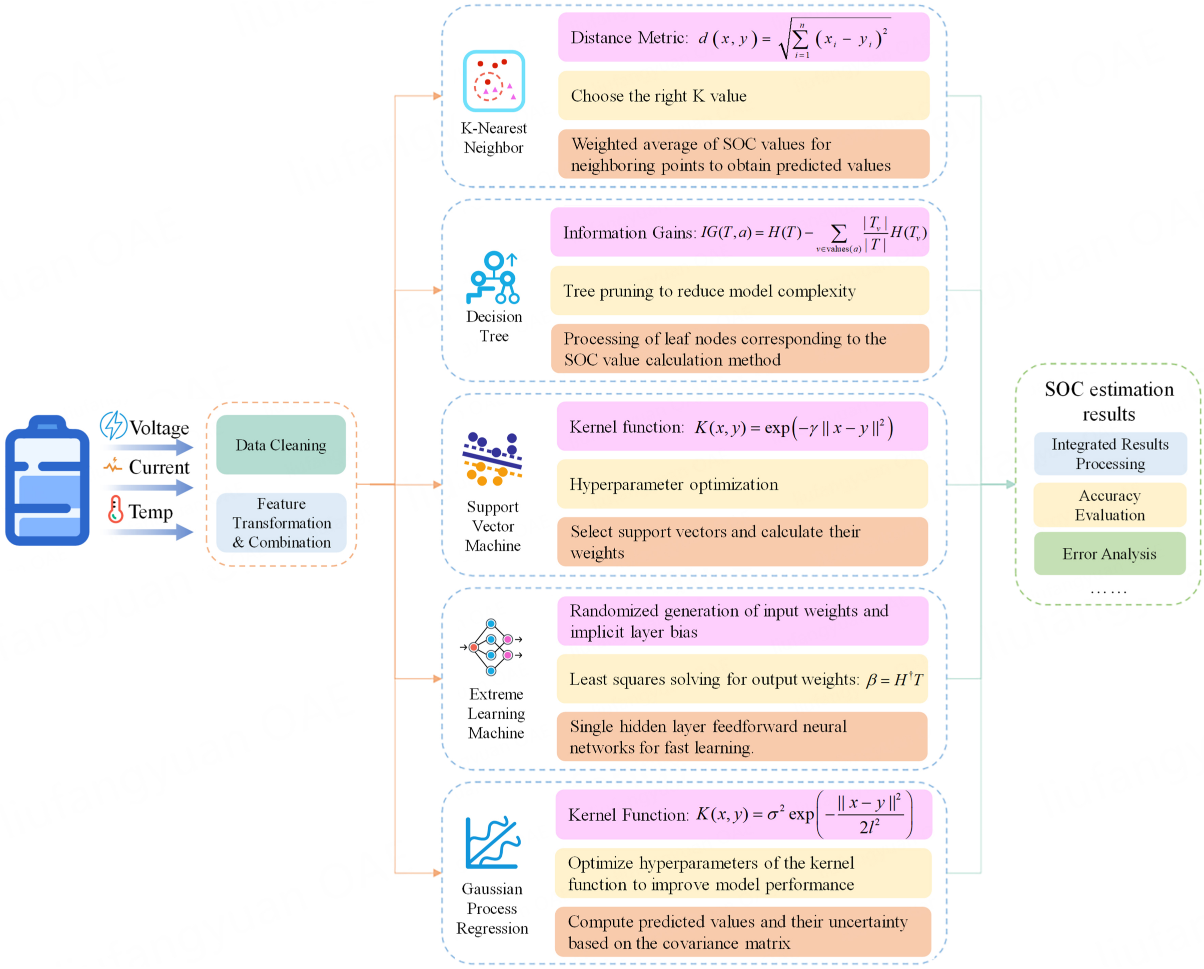
Voltage
Current
Temp
Data Cleaning
Feature Transformation & Combination
K-Nearest Neighbor
Distance Metric: $d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$
Choose the right K value
Weighted average of SOC values for neighboring points to obtain predicted values
Decision Tree
Information Gains: $IG(T,a)=H(T)-\sum_{v\in \mathrm{values}(a)}\frac{|T_v|}{|T|}H(T_v)$
Tree pruning to reduce model complexity
Processing of leaf nodes corresponding to the SOC value calculation method
Support Vector Machine
Kernel function: $K(x,y)=\exp\left(-\gamma\|x-y\|^2\right)$
Hyperparameter optimization
Select support vectors and calculate their weights
Extreme Learning Machine
Randomized generation of input weights and implicit layer bias
Least squares solving for output weights: $\beta=H^{\dagger}T$
Single hidden layer feedforward neural networks for fast learning.
Gaussian Process Regression
Kernel Function: $K(x,y)=\sigma^2\exp\left(-\frac{\|x-y\|^2}{2l^2}\right)$
Optimize hyperparameters of the kernel function to improve model performance
Compute predicted values and their uncertainty based on the covariance matrix
SOC estimation results
Integrated Results Processing
Accuracy Evaluation
Error Analysis
……

Figure 1. Flowchart depicting traditional ML methods for estimating SOC. Input variables (voltage, current, and temperature) undergo data cleaning and feature transformation before being processed by machine learning models: KNN, Decision Trees, SVM, ELM and GPR. The final step involves results processing, accuracy evaluation, and error analysis. SOC: State of charge; KNN: k-nearest neighbor; SVM: support vector machine; ELM: extreme learning machine; GPR: gaussian process regression.

enhance the accuracy of charging time[29]. The result indicates that setting the number of nearest neighbors k = 2 yields highly precise prediction. While simple averaging in SOC estimation overlooks distance information and is sensitive to noise, weighted averaging has been introduced to address these limitations. For example, Hu *et al.* proposed a method for LIB capacity estimation, combining particle swarm optimization (PSO) with KNN regression[30]. This approach defines five key features related to battery capacity - initial charging voltage, constant current charging capacity, constant voltage charging capacity, final charging voltage, and final charging current - capturing the complex relationship between the battery capacity and the KNN model. This method, which is validated with a decade of continuous cycle data, effectively estimates battery capacity across battery's lifespan. By optimizing feature weights through PSO and minimizing cross-validation errors, the model enhances accuracy and mitigates the effects of suboptimal weighting strategies.

The KNN method stands out for its simplicity and adaptability, as it does not rely on assumptions about data distribution, making it well-suited for scenarios with complex patterns. However, it has notable drawbacks, including high computational demands due to its reliance on the entire training predictions and challenges with the "curse of dimensionality," which can impair distance-based calculations in high-dimensional spaces. The effectiveness in SOC estimation depends heavily on the availability of high-quality, representative training data, which can significantly change the prediction accuracy. Moreover, the sensitivity of KNN to noise and outliers necessitates thorough data preprocessing and careful feature selection. Despite these limitations, the KNN method performs well in certain contexts, particularly when the data distribution is relatively uniform.

Decision trees

Decision trees are hierarchical ML algorithms that map features to target variables through recursive data partitioning. During construction, the algorithm selects features to split the data, optimizing subset propriety based on criteria such as information gain or Gini index. To evaluate the feature effectiveness, nodes represent feature-based decisions, branches correspond to feature values, and leaf nodes provide predictions for the target variables. In SOC estimation for LIBs, the algorithms of decision trees are effective for modeling complex, nonlinear relationships and multivariate input-output problems. SOC depends on factors such as voltage, temperature, and charge-discharge rates, which often interact nonlinearly. The hierarchical structure of decision trees efficiently captures these nonlinear relationships and adapts to varying battery operating conditions.

Recent studies have aimed to enhance the accuracy and resilience of SOC estimation. Jiang *et al.* addressed effects of battery aging by introducing two new features - voltage drop rate and temperature rise rate - and employed the XGBoost algorithm to model their relationship with SOC[31]. This method does not require an initial SOC value and can perform reliably across various battery conditions, showing strong potential for practical applications. However, the data for battery operation often contain noise, and directly using such data for training can severely degrade model accuracy. Consequently, combining model-based filtering techniques with ML methods has gained significant attention. To mitigate measurement noise and external disturbances, filtering techniques have been incorporated. For example, Song *et al.* proposed a hybrid method combining the XGBoost and the KF[15]. The XGBoost models the nonlinear relationship between features and SOC during the offline training phase, while the KF refines the XGBoost estimation in real time during online estimation. A more stable and accurate SOC estimation is achieved. This method has demonstrated high accuracy in random walk discharge tests, effectively addressing nonlinear problems and suppressing noise. However, as the KF is designed for linear systems, its application is limited by its inherently dynamic characteristics. Liu *et al.* improved the accuracy and robustness of SOC estimation by integrating extended KF (EKF) with the XGBoost algorithm[32]. The EKF estimated the nonlinear system states of the battery, while the XGBoost algorithm trained and validated the resulting data. Another study utilized the improved tree seed algorithm (TSA) with EKF, optimizing the battery model parameters globally through TSA and combining EKF for dynamic SOC estimation[33]. Simulation and experimental results demonstrate high estimation accuracy and stability. Similarly, Wang *et al.* developed a SOC estimation model for hybrid EV batteries using classification and regression trees (CARTs)[34]. By accounting for energy feedback during regenerative braking, the model achieved high accuracy, with relative errors below 0.035 in simulations and 0.05 in experiments.

Decision trees offer significant advantages in SOC estimation, including simplicity, ease of implementation, and suitability for modeling multivariate and nonlinear relationships. Their adaptability allows SOC estimation under varying battery conditions without requiring an initial SOC value. The accuracy and robustness of SOC estimation can be enhanced by integrating methods such as the KF, EKF, and improved TSAs. However, single Decision trees are prone to overfitting, especially with limited or noisy data, necessitating advanced algorithms such as the XGBoost to enhance the model generalization. The Decision trees model is sensitive to initial parameters, often requiring extensive experimentation and tuning for optimal performance. The model, with high computational complexity and large datasets, can result in long training times, emphasizing the need for algorithmic improvements and parallel processing to boost efficiency.

SVMs

SVMs are statistical learning-based ML models widely applied in classification and regression. They operate by identifying the optimal hyperplane to partition data points into distinct categories. The strong generalization ability of SVMs stems from maximizing the classification margin, relying on support vectors. It encompasses data points closest to the decision boundary, enhancing predictive ability on unseen samples. For nonlinear problems, SVMs employ a kernel function to map data into higher-dimensional space, enabling linear partition in the transformed space even when the original data is nonlinearly separable. Common kernel functions, such as the radial basis function (RBF), polynomial kernel, and sigmoid kernel, effectively model complex data relationships.

The use of SVMs in SOC estimation for LIBs has attracted significant attention. Álvarez Antón *et al.* developed an SVM-based model for SOC estimation using current, voltage, and temperature as inputs[16]. By optimizing hyperparameters with the RBF kernel and cross-validation, the model achieved a determination coefficient (R^2) of 0.98 and a maximum error below 6%. Building on this approach, Álvarez Antón *et al.* further applied SVM to $LiFePO_4$ batteries, achieving an R^2 of 0.97 and demonstrating its suitability for low-cost, microcontroller-based BMS[35]. The result highlights SVM's feasibility across different battery types. Additionally, in the SOC estimation of unmanned aerial vehicle batteries, Wei *et al.* employed support vector regression (SVR) method trained on charge-discharge data, achieving 98.42% accuracy with a mean square error (MSE) of 1.783%[36]. These results confirm the potential of SVMs in high-precision, real-time SOC estimation.

Various optimization strategies have been implemented to improve the performance of SVMs in SOC estimation. Li *et al.* integrated the grey wolf optimization (GWO) algorithm with the least squares SVM (LSSVM) method, leveraging global optimization for enhanced parameter selection[37]. This approach reduced the root MSE from 0.89% to 0.22%, significantly boosting accuracy and robustness. Additionally, ensemble learning methods have been applied to refine the estimation performance of SVMs. Sheng and Xiao combined the LSSVM with fuzzy reasoning and nonlinear correlation measurement, mitigating the influence of low-confidence samples and further improving the accuracy of SOC estimation[38]. Li *et al.* enhanced the accuracy and stability of SOC estimation by combining the SVM with ensemble learning and PSO for optimized parameter selection[39]. In hardware implementation, Stighezza *et al.* developed a field-programmable gate array (FPGA)-based SVM algorithm optimized with the ant colony optimization (ACO) method for real-time SOC estimation[40]. Experimental results demonstrate the hardware model's ability to accurately track SOC changes of the battery, with a maximum error of 3.1%, highlighting the practical feasibility and effectiveness of the SVM algorithm.

Despite its advantages, SVM has limitations. Ipek *et al.* compared SVR with XGBoost, two ML algorithms, concluding that the XGBoost is more efficient for SOC estimation due to faster computation speed and lower error rates[41]. However, the performance of SVR heavily depends on the proper configuration of the kernel function and parameter. While Ipek highlighted the challenges in parameter configuration for SVR, Song *et al.* proposed a method combining SVM with cubature KF (CKF)[42]. This approach leverages the generalization capability of SVM and filtering properties of CKF, improving the accuracy and robustness of SOC estimation. Experimental results show significant error reduction under complex driving conditions.

SVM can be applied independently or integrated with other algorithms to enhance the estimation accuracy. Xie *et al.* introduced a method combining the unscented KF (UKF) with SVM, significantly improving the accuracy by using the UKF outputs as inputs for a second estimation with the SVM[43]. Experimental results demonstrate that this approach reduced the error of SOC estimation to below 1%, outperforming the UKF alone. Similarly, Hu *et al.* developed a method for SOC estimation integrating the barnacle mating optimizer (BMO) and SVM, optimizing the SVM parameters to further enhance accuracy and stability[44]. The use of SVMs in SOC estimation extends beyond theoretical research. Hansen and Wang evaluated an SVM model using dynamic US06 operation data from the U.S. Department of Energy's hybrid EV project, demonstrating its effectiveness in real-world driving conditions[45]. The experimental result reveals that the root MSEs are 5%, 5.76%, and 2.54% in three scenarios and confirm the strong performance of SVM in SOC estimation under complex driving conditions.

The primary advantages of SVMs in SOC estimation are their ability to model nonlinear relationships and strong generalization capacity. By selecting the right kernel function, the SVM can identify the optimal separating hyperplane in high-dimensional feature spaces. It is effective with small datasets and high-dimensional data, maintaining accurate SOC estimation even under complex driving conditions. However, the disadvantages of the SVM cannot be ignored. Its high computational complexity, particularly with large datasets, can result in prolonged training time. The SVM is highly sensitive to parameter selection, with the KF and hyperparameters critically influencing model performance, necessitating optimization through methods such as cross-validation. Additionally, noisy and outlier data can reduce the estimation accuracy. Therefore, practical applications require careful parameter adjustment to fully leverage the advantages of the SVM.

Extreme learning machine

ELM is a novel learning algorithm primarily used for training single-layer feedforward neural networks (SLFN). It accelerates learning by randomly initializing the weights and biases of the hidden layer nodes and keeping them fixed during the training process, which enhances learning speed and efficiency. Traditional neural network (NN) training relies on the time-consuming backpropagation algorithm and is prone to local optima. ELM achieves rapid learning mainly through the following steps. Initially, the hidden layer parameters, such as weights and biases, are randomly assigned, rather than adjusted according to training data. The input data is then mapped into a high-dimensional feature space using the activation function of the hidden layer nodes. Nonlinear activation functions, such as sigmoid, and rectified linear unit (ReLU), are used to map inputs into a high-dimensional feature space, enhancing the representational capacity. Subsequently, the hidden layer output matrix is constructed, and the output layer parameters are determined using the least squares method, eliminating the need of complex iterative training process. The advantages of the ELM lie in its simplicity and efficiency, achieving faster training speed on multiple datasets while maintaining accuracy comparable to traditional DL models across various datasets. In addition, due to its unique random characteristics, the ELM provides robustness with noise in training data, making it well-suited for large-scale datasets. Consequently, it has been widely employed in SOC estimation. By utilizing battery voltage and current as input variables, ELM bypasses complex battery voltage and current as input variables to predict SOC, thus avoiding the complex battery modeling and enabling a direct, data-driven approach to SOC prediction.

Densmore *et al.* applied an ELM model to predict the SOC and State of Health (SOH) of LIBs using the NASA-AMES dataset, achieving a minimum SOC error of 3.1% and demonstrating the high accuracy of ELM[46]. To further improve the estimation precision, the ELM model has been optimized. Dou *et al.* integrated the gravitational search algorithm (GSA) with an ELM model, significantly enhancing its computational intelligence and robustness and enabling accurate SOC estimation with low error rates across varying temperatures and driving cycles[17]. The GSA with its global search capability optimizes hidden layer neurons and network parameters, overcoming the limitations of traditional ELM with complex data. Jiao *et al.* demonstrated an ELM model enhanced by GSA, achieving high accuracy and stability across various operating conditions[47]. The salp swarm algorithm (SSA) further refines the weights and hidden layer biases of ELM's network, improving generalization and computational precision. The SSA-ELM model achieves an average absolute error of 0.538, highlighting its precision and robustness in SOC estimation. Wang and Yang enhanced regularized ELM (RELM) by integrating the alternating direction method of multipliers with regularization techniques, mitigating overfitting while improving stability and accuracy of SOC prediction[48]. By optimizing output weights, the RELM enhances the generalization of the model, delivering robust performance under complex conditions.

Chin and Gao introduced a mixed generalized maximum correlation criterion (MGMCC) to develop a robust ELM model, which minimizes the effects of non-Gaussian noise and significantly improves the accuracy of SOC estimation[49]. Zhao *et al.* improved the MGMCC-ELM with an enhanced loss function, boosting robustness against noisy data[50]. The adaptive online sequential ELM (AOS-ELM) achieved efficient SOC estimation across varying ambient temperatures by dynamically adjusting model parameters online. The AOS-ELM[51] uses sequential data and limited samples for training, reducing errors and computation time in traditional NN training, and demonstrating its potential in practical applications. For diverse operating conditions, multi-input ELM (MI-ELM) incorporates online model parameter recognition technology with the recursive least squares (RLS) method, providing accurate SOC estimation under varying operating conditions. The MI-ELM[52] demonstrated notable strengths in SOC estimation by effectively utilizing complex nonlinear relationships, enhancing both estimation accuracy and robustness. To address data quality issues, the outlier robust ELM (OR-ELM) utilized the L1 norm loss function and the alternating direction multiplier method (ADMM) to significantly boost noise resistance. The OR-ELM[53] exhibits superior robustness in datasets with outliers, surpassing traditional ELM and RELM models, and improves stability and accuracy in handling noisy data through advanced training methods. The improved PSO-ELM (IPSO-ELM) utilizes nonlinear inertia weights to enhance global optimization, significantly improving the accuracy and stability of SOC estimation. Experimental results confirm its high precision and low error rates, highlighting its practical applications. By leveraging advanced global search optimization techniques, the IPSO-ELM effectively addresses complex conditions, demonstrating robust performance.

While the ELM and its enhanced variants excel in SOC estimation with benefits such as fast computation, simplicity, and adaptability, challenges remain in managing high-noise data and large-scale datasets. Traditional ELM models are sensitive to data quality, often yielding significant estimation errors with noisy data, and may struggle to fully capture complex nonlinear dynamics. Future research could utilize DL and advanced algorithms to enhance the ELM performance in SOC estimation, enabling better handling of complex practical application scenarios. Additionally, exploring new feature variables and data preprocessing techniques can further improve the accuracy and robustness, ultimately enhancing the reliability and performance of the BMS.

GPR

GPR is a Bayesian, nonparametric learning method widely used in regression tasks. Instead of finding a single best-fit function, GPR places a prior distribution over the space of possible functions, capturing uncertainty directly. By conditioning on observed data, it derives a posterior distribution for both mean predictions and predictive variances, enabling robust uncertainty estimation. Similar to SVMs, GPR leverages kernel functions - such as the RBF, Matérn kernel, and others - to characterize relationships between data points in potentially complex, high-dimensional spaces. This kernel-based flexibility allows GPR to adapt to nonlinear patterns while providing principled confidence intervals, making it particularly effective in scenarios with limited data or when precise uncertainty quantification is desired.

Building on this foundational understanding of the flexibility of GPR, researchers have increasingly explored its application to SOC estimation in LIBs, recognizing its capacity to model highly nonlinear system dynamics while directly quantifying prediction uncertainty. Ozcan *et al.* proposed a GPR-based online SOC estimation framework, leveraging different kernel functions - such as the squared exponential, Matérn, rational quadratic, and quasi-periodic kernels - to effectively capture the complex interactions among voltage, current, and temperature[54]. This approach yields high accuracy under both dynamic and constant loading conditions and provides a probabilistic confidence interval for each estimate. To address the often high computational burden of GPR, Ozcan *et al.* employ a sparse formulation that reduces the complexity from O (N^3) to O (NM^2) (N is the number of original samples, M is the number of Inducing points, and $N \gg M$) by introducing a smaller set of inducing points[55]. This makes real-time SOC estimation feasible even with large-scale datasets. Meanwhile, Sahinoglu *et al.* extend GPR by incorporating recurrent structures to account for time-series dependencies, thereby refining the prediction of SOC across successive measurements and achieving remarkably low errors in multi-condition experiments[56].

Beyond these developments, hybrid methods also leverage the nonparametric strength of GPR in tandem with state estimators. Chen *et al.* proposed coupling a UKF with GPR to adaptively adjust for measurement noise, resulting in robust performance across diverse driving cycles and temperatures[57]. Similarly, Lee *et al.* proposed data-driven Gaussian Process Kalman and Particle Filters (GP-UKF and GP-PF) to learn both the prediction and observation models from data, offering significant improvements in accuracy and uncertainty quantification compared to classical filtering schemes[58]. Extending the scope further, Yi *et al.* demonstrated a novel approach that incorporates battery expansion features alongside voltage and current measurements, revealing that GPR can capture subtle mechanical changes associated with SOC variations[59]. This multimodal integration reinforces the versatility of GPR, although the computational demands and sensitivity of the technique to kernel hyperparameters underscore the importance of methodical model optimization. Deng *et al.* focused on the challenges of SOC estimation in LIB packs caused by cell inconsistencies[60]. By employing feature extraction methods such as correlation analysis and principal component analysis (PCA), they optimized the input data for GPR modeling, effectively capturing the nonlinear dynamics of battery packs. Additionally, they validated the autoregressive approach proposed by Sahinoglu *et al.* within their framework, demonstrating its robustness across dynamic cycles, temperatures, and aging states[56]. This work achieved estimation errors below 3.9%, highlighting a practical path for accurate and efficient SOC estimation in complex battery pack systems. These studies collectively highlight the potential of GPR as a powerful, uncertainty-aware tool for LIB SOC estimation - particularly when accurate modeling of nonlinearities and robust real-time performance are paramount.

Regarding its advantages, GPR excels at capturing complex, nonlinear relationships without needing an explicit physical model, and it inherently provides uncertainty quantification through posterior predictive variance. These strengths facilitate high accuracy, flexibility, and robust performance even with limited data or under varying operational conditions. However, GPR also faces challenges. Its computational cost typically scales cubically with the dataset size, although sparse representations can partially alleviate this burden. Moreover, model performance heavily depends on appropriate kernel selection and hyperparameter tuning, necessitating careful calibration or automated optimization methods. Despite these limitations, the growing body of research underscores the strong potential of GPR in delivering reliable SOC estimation and uncertainty analysis for LIBs, supporting both practical deployment and future innovation in BMSs.

Summary

Data-driven SOC estimation for LIBs leverages ML techniques that bypass the need for battery working principles. The evolution of ML technologies has advanced these methods, with traditional algorithms such as KNN, Decision trees, SVM, ELM, and GPR commonly employed. Table 1 summarizes the advantages and drawbacks of these algorithms.

Table 1. Advantages and drawbacks of conventional ML techniques for SOC estimation in LIBs

Method	Advantages	Drawbacks
KNN	Simple and intuitive, strong adaptability for multi-dimensional features	Computationally intensive for large datasets, prone to errors with noisy data and outliers
Decision trees	Simple structure, effectively handle complex nonlinear relationships	Prone to overfitting, high computational complexity and long training times, sensitive to initial parameter selection
SVM	Efficiently manage complex nonlinear relationships and multi-dimensional data, good generalization	Computationally intensive, long training times, sensitive to kernel and hyperparameter selection, requires cross-validation
ELM	Fast learning, streamlined computational process, highly adaptable model structure	Highly sensitive to data quality, prone to noise interference, struggle to capture dynamic variations in complex nonlinear relationships
GPR	Capture complex nonlinear relationships, provide uncertainty quantification, suitable for limited data scenarios	High computational cost, dependent on kernel selection and hyperparameter tuning

SOC: State of charge; LIB: lithium-ion battery; ML: machine learning; KNN: k-nearest neighbor; SVM: support vector machine; ELM: extreme learning machine; GPR: gaussian process regression.

Despite advancements in traditional ML methods for SOC estimation, they still face limitations. These traditional ML methods rely on manual feature design and selection, demanding expertise and extensive experimentation, often using linear or basic nonlinear models with limited expressive power. The computational demands are high, particularly for large datasets, resulting in low efficiency. Additionally, these methods are highly dependent on parameter selection and data preprocessing, with any oversight potentially degrading the model performance. KNN is sensitive to noisy data, and Decision trees are prone to overfitting, while SVM, ELM and GPR depend heavily on kernel functions and hyperparameter choices, necessitating optimization through methods such as cross-validation. Moreover, traditional ML methods struggle with noisy or outlier-laden data, reducing their robustness and estimation accuracy. Future research should focus on refining these algorithms to enhance their efficiency and robustness in handling complex and large datasets. While traditional ML methods have advanced SOC estimation, as discussed above, they still face challenges in feature engineering, model complexity, parameter sensitivity, and data robustness. As a result, more advanced ML techniques are explored to address these issues and improve estimation accuracy. The following sections explore DL models such as MLP, CNN, RNN, and encoder-decoder (ED)-based sequence models, which have demonstrated significant potential across various domains. These models, known for their strong nonlinear fitting capabilities and automated feature extraction, are transforming approaches to estimating complex system states. We will examine their working principles, applications in SOC estimation, and how they address the limitations of traditional methods.

DL-BASED SOC ESTIMATION

DL, a subset of ML, leverages NNs and shows significant potential for estimating the SOC of LIBs. NNs can be trained to extract optimal feature representations from raw data, and demonstrate strong capability in learning complex patterns and relationships. There are two major stages in DL-based SOC estimation. The first stage, excluding attention mechanisms, involves the development of numerous SOC estimators using basic sequence modeling techniques. The second stage introduces attention mechanisms to better capture long-term sequential relationships and replace recursive processes. According to recent publications, research in the first stage appears to have plateaued. In this section, we outline practical strategies for SOC estimation, followed by a concise overview of the theory and application of two basic network types. We then review the theories, applications, and integrations of these techniques, focusing on the progression of sequence modeling technologies without attention mechanisms. The summary assesses its influence on the current thriving attention mechanism-based SOC estimators.

SOC estimation task design

Apart from OCV, all aspects of battery performance, including SOC, are dynamic[61]. As SOC estimation relies on sequences such as current, voltage, and surface temperature, it is inherently a time-series prediction task. In contrast to typical time-series prediction tasks, such as weather forecasting, power consumption prediction, and product sales prediction, SOC estimation faces unique challenges. In conventional prediction scenarios, target parameters are typically observable. In SOC estimation, the SOC of a battery cannot be continuously measured. It is only measured under specific temperature conditions and when the battery reaches its cutoff voltage the SOC can be confirmed as fully charged (100%) or completely discharged (0%)[61]. Therefore, SOC values at individual points in a charge/discharge cycle can only be determined through the proportional relationship between the accumulated charge/discharge and the total battery capacity after completing a full cycle. This characteristic complicates the direct use of real SOC measurements in practice. Consequently, SOC estimation primarily depends on external variables rather than the target variables. As a result, standard time-series prediction models cannot be directly applied to SOC estimation, necessitating the development of tailored modeling standards to address actual conditions.

Table 2 summarizes practical SOC estimation strategies for various dynamic NN (DNN)-based estimators. Here, w indicates the sliding window length of the input sequence. The external variable input $x_i = \{I_i, V_i, T_i$ $,...\}$ at time i may include voltage V_i, current I_i, temperature T_i, and other variables, where NN(•) refers to the NN function, and $\hat{y}_i$ signifies the estimated SOC value at time i. If the variable to be estimated is included in the input, the non-continuous nature of SOC measurement means that only prior estimation values can be used, a method known as recursive prediction. In contrast, non-recursive estimation approaches exclude the variable, such as SOC, from being estimated in the input sequence. Given the nonlinear dynamic characteristics and dependence on external variables, the nonlinear autoregressive model with exogenous inputs (NARX) is commonly applied for continuous recursive estimation of SOC. While single-step estimation predicts the next time step's observation, multi-step estimation requires SOC predictions for multiple time points.

Various estimation strategies are applicable to different application scenarios. Single-step non-recursive estimation is simple to implement and typically delivers high accuracy, making it the most widely used approach for SOC estimation. Multi-step non-recursive estimation, derived by modifying the output layer of a single-step non-estimation network, provides SOC estimations for multiple historical time points, simultaneously offering more comprehensive data for the BMS system; however, it may suffer from lower accuracy. Non-recursive estimation avoids initial errors by not using historical SOC values, but it may result in discontinuities in continuous outputs due to the lack of historical constraints. In contrast, recursive estimation enhances stability, particularly in LIBs with distinct voltage platforms, though it may lead to drift in continuous estimation due to initial and cumulative errors.

Basic components and applications of NN

Multilayer perceptron

MLP is a fundamental component of DL comprising FC Layers and activation functions, as shown in Figure 2. FC layers transform arrays between dimensions via linear transformations, enabling comprehensive feature analysis. Activation functions, including Sigmoid, Tanh, ReLU, and SoftPlus, introduce nonlinearity to the network. By layering multiple "FC-activation function" units, MLPs can model nonlinear relationships between inputs and outputs, establishing their role as a cornerstone of DL

Table 2. Strategy for SOC estimation

Strategy	Formulaic expression
Single-step non-recursive estimation	$\hat{y}_t = \mathrm{NN}\left(x_{t-w}, x_{t-w+1}, \ldots, x_t\right)$
Multi-step non-recursive estimation	$\left\{\hat{y}_{t-w}, \hat{y}_{t-w+1}, \ldots, \hat{y}_t\right\} = \mathrm{NN}\left(x_{t-w}, x_{t-w+1}, \ldots, x_t\right)$
Single-step recursive estimation	$\hat{y}_t = \mathrm{NN}\left(x_{t-w}, x_{t-w+1}, \ldots, x_t; \hat{y}_{t-w}, \hat{y}_{t-w+1}, \ldots, \hat{y}_{t-1}\right)$

SOC: State of charge.

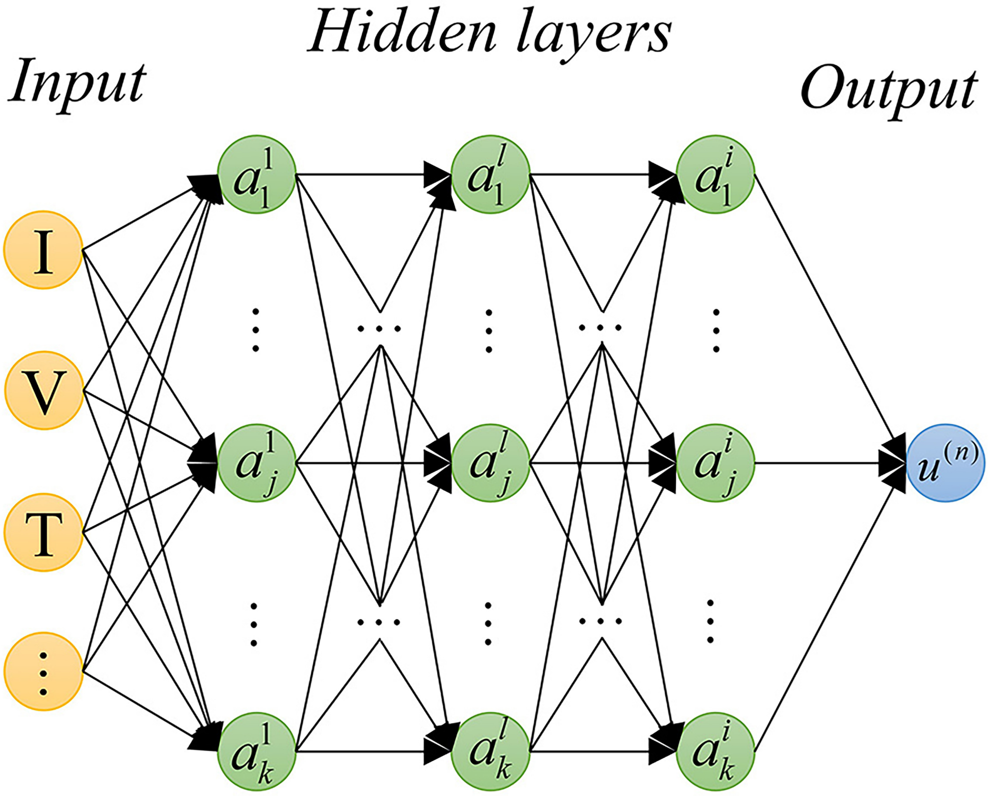

Figure 2. Architecture of a multilayer perceptron.

technology.

In an MLP, the learnable parameters include the transformation coefficient matrix and bias vector within FC layers. Training involves forward propagation to process input arrays and generate output, with supervised learning calculating the deviation from pre-defined labeled values through a certain norm. Backpropagation then distributes this loss, updating parameters via optimization algorithms. After a sufficient number of iterations, the network converges to the training set distribution, freezing parameters for inference where only forward propagation is performed.

As one of the earliest deep NNs applied to SOC estimation, MLP demonstrates its effectiveness. Chemali *et al.* developed an MLP-based estimator with high accuracy across various temperatures[62]. Their method mapped SOC by relating average current and voltage at the current moment to voltage, current, temperature, and data from 50-400 historical moments, using a fixed number of MLP layers and neurons per layer. Hannan *et al.* eliminated historical averages, basing SOC estimation solely on real-time current, voltage, and temperature; they also used a backtracking search algorithm during training to optimize the number of hidden neurons and learning rate in a single-hidden-layer MLP[21]. How *et al.* analyzed how the number of hidden layers in an MLP affects the accuracy of SOC estimation. Their results revealed a U-shaped relationship: too few layers hinder learning capacity, while excessive layers increase overfitting and estimation error for unknown cycles[63]. Jung *et al.* addressed this issue, drawing comparable conclusions[64]. They further incorporated single-step voltage changes as an input variable to account for the instantaneous dynamics of the power battery.

The early MLP studies established foundational strategies for single-step SOC estimation. These efforts also highlight the significant impact of NN hyperparameters on estimation accuracy, directing subsequent research toward hyperparameter optimization.

MLP estimators have become less prevalent in SOC estimation due to their inability to effectively handle temporal data, which is critical for accurate and stable SOC estimations[63]. The MLP structure is inherently unsuited for processing time-series information and faces three severe challenges in this context. MLP is unsuitable for SOC estimation due to its lack of memory structure, preventing it from processing sequential time-series data effectively. It treats all inputs as independent, failing to capture the relationship between past measurements and the current SOC. Additionally, the FC structure of MLP, with its numerous parameters, leads to high computational demands and overfitting risks when handling longer data sequences. Furthermore, its one-dimensional input design struggles with multi-two-dimensional, multi-moment, and multi-variable data typical in SOC estimation, requiring additional designs such as auto-encoding or dimensionality reduction. The FC structure is difficult to use directly for SOC estimation of batteries. These limitations have prompted the adoption of more advanced DNN architectures for SOC estimation, largely replacing MLP.

CNNs

CNNs are a fundamental class of NN models distinguished by their extraction method of information. In contrast to MLPs, which analyze all features through FC layers, CNNs use convolutional layers that focus on local regions (referred to as receptive field) during each convolution operation. By sliding the receptive field across the input, they extract local information and compile it into a feature map, as shown in Figure 3. This process involves globally shared convolutional kernel parameters, significantly reducing the number of learnable parameters and improving computation speed through parallel processing. Convolutional kernels are categorized as 1D-CNN, 2D-CNN, and 3D-CNN, with the first two CNNs being prevalent in SOC estimation. Following the convolution, activation functions and pooling layers extract key features from the feature maps. Pooling layers operate locally, applying rule-based operations such as maximum, minimum, or mean functions without performing linear transformations.

Various CNN architectures are applied in SOC estimation. Bhattacharyya *et al.* employed two 2D convolutions with a stride of 2 to extract features, followed by an MLP for final estimation, achieving a mean absolute percentage error (MAPE) below 0.5% on two datasets with ambient temperatures[22]. This CNN-MLP hybrid model has inspired advancements such as ResNet in computer vision. However, there are debates that 2D convolutions may obscure temporal relationships among variables[65], leading to the preference for 1D-CNNs in later studies to better capture the temporal dynamics of individual variables. Bhattacharjee *et al.* supported this perspective and utilized receptive fields of varying lengths within single-variable channels to capture multi-scale temporal patterns, later merging these features before applying an MLP for the final output[66]. Hannan *et al.* replaced the MLP with a full CNN for the output layer, creating a SOC estimator based on a fully convolutional network (FCN)[67]. Resultantly, it surpasses LSTM, GRU, and

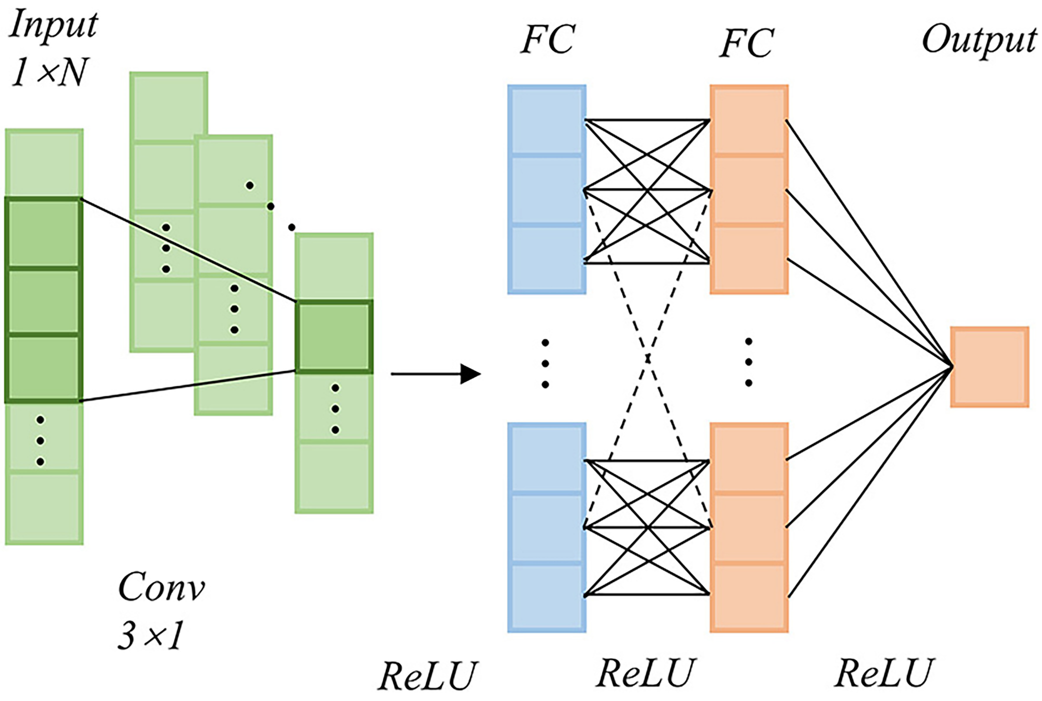

Figure 3. Illustration of CNNs. CNNs: Convolutional neural network.

CNN-MLP methods in estimation accuracy, generalization ability, floating point operations per second (FLOPs), and runtime speed. Li *et al.* addressed the correlation between SOH and SOC by first estimating SOH with a 3D-CNN, and then combining it with a 2D-FCN for SOC estimators[68]. Mohanty *et al.* adopted a continuous wavelet transform-based CNN (CWT-CNN), leveraging CWT to analyze input signals in both time and frequency domains and extract features before CNN-based SOC estimation[69]. This approach transforms low-dimensional variable signals into 2D feature maps, leveraging visual networks for analysis, achieving an MAE below 0.7%.

CNN-based SOC estimators show significant potential but are primarily utilized for data preprocessing in most studies. Table 3 compares various MLP and CNN studies on SOC estimation for LIBs, including references, model structures, evaluation methods, window sizes, benchmark comparisons, battery types, battery operating temperatures, test datasets, and error ranges.

Basic sequence modeling networks and applications

Since SOC estimation is fundamentally a time-series prediction task, early MLP-based strategies led researchers to incorporate fundamental sequence modeling techniques from DL, marking the beginning of deep SOC estimators and sparking initial research. This section provides a summary of basic sequence modeling techniques derived from MLP and CNN and outlines the development of CNN/RNN networks and their combined technologies.

Table 4 summarizes various studies on sequence modeling techniques for SOC estimation of LIBs, detailing references, model names, model structures, evaluation methods, window sizes, benchmark comparisons, battery types, operating temperatures, test profiles, and error ranges. This comparison allows for the evaluation of the effectiveness of various methods and highlights key challenges and advancements in sequence modeling techniques.

Table 3. Comparison of SOC estimation performance between MLP and CNN

Reference	Model name	Model structure/Strategy	Estimation mode	Window size	Compared models	Dataset	Temperature (°C)	TestProfile	Error (%)
[62]	DNN	MLP	NRSS	1		NCR-PF18650[70]		US06 HWFET	MAE < 1
[21]	BPNN-BSA	MLP	NRSS	1	RBFNN-BSA GRNN-BSA ELM-BSA	INR-18650-20R[4]	0 °C 25 °C 45 °C	DST FUDS	MAE < 1.74 RMAE < 0.87 MAPE < 20.09
[22]	CNN	CNN-FC				Samsung 18650	15 °C 25 °C 45 °C		RMSE < 0.19 MAPE < 1.14
[66]	CNN	CNN-FC Transfer Learning	NRSS	500	LSTM GRU	NCR-PF18650[70] LG 18650 HG2[71]	0 °C 10 °C 25 °C	US06 HWFET	MAE = 0.81 (NCR) MAX = 1.78 (NCR) MAE < 1.20 (LG)
[67]	FCN	FCN	NRSS	400	GRU LSTM CNN	NCR-PF18650[70]	25 °C -20 °C to 25 °C	US06 HWFET	MAE = 0.7 MAX = 2.96 MAE = 1.55 (v,t.) MAX = 7.63 (v.t.)
[69]	CWT-CNN	CWT-CNN	NRSS			INR-18650-20R[4]	25 °C	DST US06 FUDS	MAX < 0.76 RMSE < 1.26
[72]	GA-CNN on FPGA	CNN	NRSS	30	CNN				MAE < 3
[73]	CNN	1D-CNN	NRSS		GRU	NCA-PF18650ZY (Private data)		CCCV CC	MAE = 1.62 (32 type) MAE = 4.35 (8 type)
[74]	NARX	MLP	RSS	2	BPNN	NCA (Private data)		0.5C 1C 2C	MAX < 3

SOC: State of charge; MLP: multilayer perceptron; CNN: convolutional neural network; DNN: deep neural network; NRSS: non-recursive single-step; RSS: recursive single step; ELM: extreme learning machine; FC: fully connected layer; BSA: backtracking search algorithm; LSTM: long short-term memory; GRU: gated recurrent unit; FCN: fully convolutional network; CWT: continuous wavelet transform; NARX: nonlinear autoregressive model with exogenous inputs; RSS: recursive single-step; NCA: Nickel-cobalt-aluminum oxide.

Sequence modeling techniques based on RNN and its variants

Following challenges similar to those in the NLP field, various NNs have been developed with memory structures. The most common models include RNN, as illustrated in Figure 4A. The figure depicts MLP-based networks such as simple RNNs (SRNNs)[94], LSTM [Figure 4B][23], and GRU [Figure 4C][24]. This subsection will review these models, their variants, and their applications in SOC estimation.

Table 4. Comparison of SOC estimation methods using sequence modeling approaches

Reference	Model name	Model structure/Strategy	Estimation mode	Window size	Compared models	Dataset	Temperature (°C)	TestProfile	Error (%)
[25]	RNN	SRNN	NRSS	1		16Ah Softclad		CC	MAX < 0.1
[75]	RNN	SRNN	NRSS		BPNN-FA BFNN-FA	NMC NCA (Private data)		SD HPPC	MAE < 1.43 (SD) MAE < 0.42 (HPPC)
[76]	NARX RNN	SRNN	RSS		NARX BPNN-PSO ELM-GWO LSTM UKF	NCR18650PF (Private data) INR-18650-20R[4]	0 °C (INR) 25 °C (INR) 45 °C (INR)	CCCV (NCR) DST (NCR) UDDS (NCR) FUDS (INR)	MAE < 0.33 (NCR) RMSE < 0.49 (NCR) MAE < 0.29 (INR) RMSE < 0.68 (INR)
[77]	RNN	SRNN-NARX	RMS	30	LSTM-Atten	A123-18650[1]	20 °C 25 °C 30 °C 40 °C 50 °C	DST US06 FUDS	MAE < 2.10 RMSE < 2.94
[78]	LSTM	LSTM-NARX	RSS		BPNN-PSO LS-SVM LSTM	18650 (Private data)	20 °C	UDDS DST	MAE < 0.72 RMSE < 0.78 MAPE < 1.28
[79]	LSTM with Attention	LSTM-AM	NRSS		SVM SRNN LSTM SVM-PF SRNN-PF LSTM-PF LSTM-OP BDLSTM	INR-18650-20R[4] NCR-PF18650[70]	0 °C (INR) 25 °C (INR) 45 °C (INR) 0 °C (NCR) 10 °C (NCR) 25 °C (NCR)	US06 DST FUDS US06 HWFET	RMSE < 1.02 (INR) MAE < 0.24 (NCR) +
[80]	PSO-LSTM	LSTM	NRSS		EKF LSTM PSO-LSTM (without noise)	Private data	25 °C	CC UDDS	MAE < 0.43 RMSE < 0.58
[81]	GA-GRU	GRU			RMM LSTM	US18650VTC (Private data)	0 °C 10 °C 20 °C 30 °C 45 °C	DST FUDS US06 BJDST	MAE < 0.28 RMSE < 0.22 MAX < 2.06
[82]	LSTM	LSTM	NRSS	50	SRNN SVR RF	NCR-PF18650[70]	0 °C 10 °C 25 °C	UDDS NN	MAE = 0.63 (Ave.)
[83]	LSTM	LSTM	NRSS	1	UKF	A123-18650		DST FUDS US06	MAE < 2.45
[84]	DBLSTM	SBLSTM	NRSS	100	GRU LSTM Resnet VGG MLP	INR-18650-20R[4]	0 °C 10 °C 25 °C	FUDS US06	RMSE < 0.75
[85]	NAG-Bi-GRU	BiGRU	NRMS		LSTM SVMELMBPNN RBFNN GNN EKF	INR-18650-20R[4] LG 18650-HG2[71]	0 °C 25 °C 45 °C	FUDS (INR) LA92 (LG) UDDS (LG)	RMSE < 1.9 (INR) MAE < 2.2

					UKF PF CKF AWCPF and their improved methods				(INR) RMSE < 1.40 (LG) MAE < 1.13 (LG) MAX < 5 (LG)
[86]	PG-BiGRU	BiGRU	NRSS		LSTM GRU BiGRU PIO-BIGRU	NCR-PF18650[70]	0 °C 10 °C 25 °C	UDDS NN	RMSE < 1.83 MAE < 1.70
[87]	SBLSTM	SBLSTM	NRSS	1	SBSRNN SBGRU	NCR-PF18650[70] INR-18650-20R[4]	0 °C 10 °C 25 °C	US06 HWFET	MAE < 1.2 (NCR) MAX < 6 (NCR) MAE < 3 (INR) MAX < 6 (INR)
[88]	BiLSTM-PANN	BiLSTM	MRSS	30	FNN LSTM GRU BiLSTM LSTM-SA BiLSTM-ED and some with PANN	INR-18650-20R[4]	0 °C 25 °C 45 °C	FUDS DST US06	RMSE < 2.13
[89]	LSTM with stress	LSTM	NRSS	100		pouch-type (Private data)		Train: Test 1:3 1:1 3:1	RMSE < 5 (1:3) RMSE < 4 (3:1)
[90]	OPSLTSM	LSTM	NRSS	1000	TCN RF SVM LSTM GRU CWRNN PCA	NCR-PF18650[70]	0 °C 10 °C 25 °C	US06 HWFET UDDS	RMS < 2.14
[91]	MSFGRU	GRU	NRMS	360	LR LSTM GRU Transformer PatchTST	EV-driven data (Private data)			MAE < 0.61
[92]	LSTM-TL	LSTM	NR		AEKF SVM	NCM data	30 °C	UDDS	MAE < 5.8 RMSE < 3.9
[93]	RNNs-TL	RNNs	NRSS	30	LSTM BiLSTM GRU BiGRU different TL improvements	LG 18650-HG2[71] INR-18650-20R[4] PoliMi (Private data)	-20 °C -10 °C 0 °C (INR) 10 °C 25 °C (INR) 35 °C (PM) 45 °C (INR)	UDDS US06 LA92 DST FUDS BJDST	MAE = 1.5 (INR, TL5) MAE = 2.2 (PM, TL6) MAE = 1.1 (LG)

SOC: State of charge; RNN: recurrent neural network; SRNN: simple recurrent neural network; NRSS: non-recursive single-step; NCA: nickel-cobalt-aluminum oxide; NARX: nonlinear autoregressive model with exogenous inputs; RSS: recursive single-step; PSO: particle swarm optimization; ELM: extreme learning machine; GWO: grey wolf optimizer; LSTM: long short-term memory; UKF: unscented kalman filter; RMSE: root mean square error; SVM: support vector machine; EKF: extended kalman filter; GRU: gated recurrent unit; SVR: support vector regression; MLP: multilayer perceptron; GNN: graph neural network; CKF: cubature kalman filter; BiLSTM: bidirectional long short-term memory networks; ED: encoder-decoder; TCN: temporal convolutional network; PCA: pearson correlation; EV: electric vehicles; FNN: feedforward neural network.

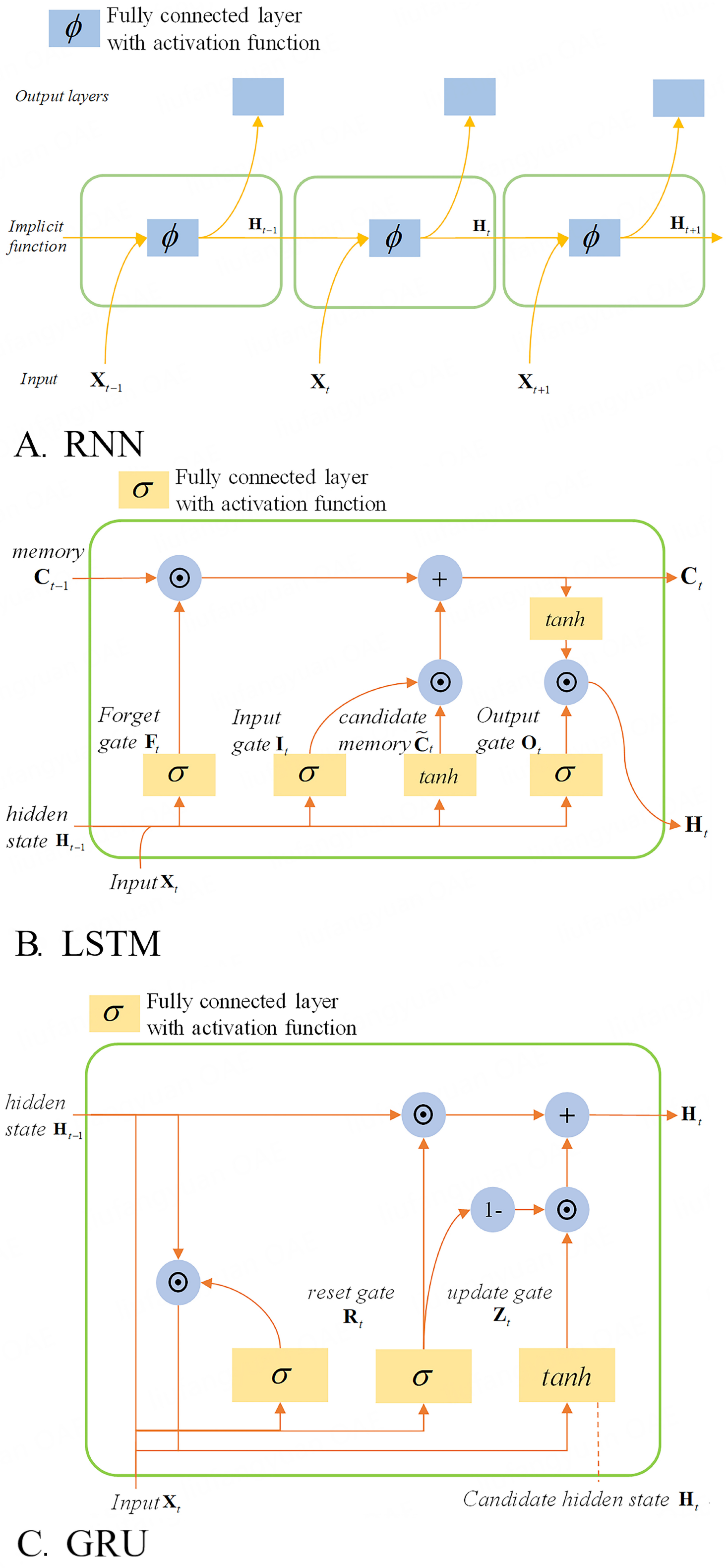
ϕ Fully connected layer with activation function
Output layers
Implicit function
ϕ
$\mathbf{H}_{t-1}$
$\mathbf{H}_{t}$
$\mathbf{H}_{t+1}$
Input
$\mathbf{X}_{t-1}$
$\mathbf{X}_{t}$
$\mathbf{X}_{t+1}$
A. RNN
σ Fully connected layer with activation function
memory $\mathbf{C}_{t-1}$
$\odot$
+
$\mathbf{C}_{t}$
tanh
Forget gate $\mathbf{F}_{t}$
Input gate $\mathbf{I}_{t}$
candidate memory $\tilde{\mathbf{C}}_{t}$
Output gate $\mathbf{O}_{t}$
σ
tanh
hidden state $\mathbf{H}_{t-1}$
$\mathbf{H}_{t}$
Input $\mathbf{X}_{t}$
B. LSTM
σ Fully connected layer with activation function
hidden state $\mathbf{H}_{t-1}$
$\odot$
+
$\mathbf{H}_{t}$
1-
reset gate $\mathbf{R}_{t}$
update gate $\mathbf{Z}_{t}$
σ
tanh
Input $\mathbf{X}_{t}$
Candidate hidden state $\mathbf{H}_{t}$
C. GRU

Figure 4. Structure of RNN and its variants: (A) RNN; (B) LSTM; (C) GRU. RNN: Recurrent neural network; LSTM: long short-term memory; GRU: gated recurrent unit.

SRNNs are designed to process sequential input by sharing the same set of parameters across time steps and iteratively updating the hidden states, enabling memory functionality. In each RNN unit, the input array, combining the historical hidden state h_{t-1} and input vector x_t, undergoes linear transformation and nonlinear activation based on the Tanh function, producing the hidden state for the current time step h_t. The h_t dimension, determined by the preset hidden layer size, may not align with the required output dimension, necessitating an MLP for decoding the hidden state to generate the final network output.

As the most basic RNN, the effectiveness of SRNN in SOC estimation has been extensively confirmed. Liu *et al.* developed a single-step, non-recursive SOC estimator using SRNN[25]. Lipu *et al.* applied this method to $LiNi_{1-x-y}Co_xMn_yO_2$ (NCM) and $LiNi_{1-x-y}Co_xAl_yO_2$ (NCA) batteries and proposed a hyperparameter optimization technique using the firefly algorithm[75]. After the hyperparameter optimization, SRNN performed MLP in predictive accuracy on two datasets. The inherent recursive nature of RNNs makes them ideal for recursive SOC estimation, inspiring numerous early studies in this area. Liu *et al.* developed a NARX-based SOC estimator using SRNN and used the value of impedance as an input variable[74]. Wang *et al.* employed a sliding window approach to incorporate multiple historical measurements, leveraging the memory of RNN to capture time-varying patterns[76]. They improved this method with a dynamic window size that adapts to data fluctuations, reducing the maximum RMSE to 0.34%, outperforming the fixed window approach. Sadykov *et al.* showed that SRNN estimators based on recursive strategies had half the error rate of non-recursive models when handling untrained driving dynamics[77]. While the dynamic window length in the model of Wang *et al.* is unspecified, judging from the lengths of comparison models of 225 and 450, it likely exceeds 100[76]. Considering that the total length of the training samples is only 3,000, with each estimation using about 1/30 of the total data processed per estimation, its feasibility could be limited in certain contexts.

SRNNs face a major limitation in retaining short-term important information. While the hidden state h_t generated by SRNN at time step t should reflect earlier input changes, deep or long sequences of SRNN experience issues due to activation functions such as tanh reaching saturation zone. In this state, their derivative approach zero, causing gradients to diminish during backpropagation. As the network deepens, weights in the earlier layers update minimally and are overshadowed by new information. Conversely, when the derivative of the activation function exceeds 1, particularly in the non-saturated region of the activation function, weight updates can be excessively fast. In long sequences, large gradients can escalate significantly through multiple layers, leading to the exploding gradient problem. Both issues hinder the effective training and application of SRNNs.

To overcome the limitations of SRNNs, LSTM and GRU are introduced, which incorporate gated mechanisms, regulating information flow within NNs through “gates” that control data transmission. An LSTM unit features three gates and, as opposed to the computations in SRNN, those in an LSTM unit retain a hidden state h_t with short-term memory and add a context vector or cell state c_t for long-term memory. Upon receiving the input for a given time step, the unit updates the long-term memory c_t using the Input gate and Forget gate. Specifically, the Input gate computes an input weight vector based on the current input x_t and the previous hidden state h_{t-1}, determining the relevance of the input data to be stored in the cell state c_t. The Forget gate generates a forgetting weight vector, determining through a dot product, while holding information in the cell state c_t should be discarded or forgotten. After updating the long-term memory c_t, the Output gate retrieves the short-term memory h_t needed for output. It computes a third weight vector using the current input x_t and the previous hidden state h_{t-1}, and then multiplies the output weight by the long-term memory c_t to generate the final output h_t.

The solution to vanishing and exploding gradient issues with gated mechanisms is simple: minimize the number of times the gradient passes through nonlinear layers and regulate their derivatives of nonlinear layers within a suitable range. In LSTM, the cell state is directly transmitted to subsequent layers without alteration, enabling identity mapping. Consequently, this mechanism facilitates smoother information flow in deep networks, addressing the vanishing and exploding gradient problems. Identity mapping preserves long-term memories even with long sequences, while the gated mechanism filters irrelevant information and retains valuable data, effectively addressing memory coverage issues. The gated mechanism controls information flow through weights calculated by the sigmoid activation function, which adjusts gate openness between 0 and 1. This regulation helps keep gradients within a manageable range. With a maximum derivative of 0.25, the sigmoid function limits the scale of weight updates, preventing excessively large changes in weights even when gradients are large.

LSTM introduces significantly more learnable parameters than SRNN, complicating training. To address this issue, GRU, a simplified variant of LSTM and a form of RNN, maintains the use of gated mechanisms to tackle gradient problems and long-term memory limitations while simplifying the structure of LSTM to include only two gates. In the Update Gate, the weights are computed based on the current input x_t and the prior hidden state h_{t-1}, determining the retention or replacement of the previous state. Similarly, the Reset gate calculates weights to control the influence of the prior hidden state on the current candidate state, generating short-term memory through the generation of the candidate hidden state $\tilde{h}_t$. Finally, this short-term memory is integrated into the long-term memory h_t.

In GRU networks, the long-term memory h_t, akin to the cell state c_t in LSTM, is managed differently. In contrast to LSTM, which outputs short-term information while keeping the cell state hidden, GRU outputs all information across periods. This simplification arises from two assumptions. First, GRU believes that historical long-term memories can be retained and updated without distinguishing short-term memories. Secondly, when LSTM updates the cell state, the input increment and the legacy of the historical cell state are in a zero-sum relationship, so there is no need to calculate two sets of weights separately. As a result, GRU combines these gates into a single update gate, reducing parameter complexity and accelerating convergence. With fewer parameters, faster convergence, and improved efficiency, GRU has become a widely used technique in SOC estimation tasks.

LSTM and GRU, two prominent RNN variants, are commonly used in deep SOC estimators. Abbas *et al.* designed LSTM-based SOC estimators using the NARX model of hyperparameters such as input delay, feedback delay, and hidden layer size[95]. They proposed a Bayesian regularization optimization approach, which outperformed the "trainlm" training function in accuracy. Wei *et al.* improved the error reduction to within 0.78%[78]. Chemali *et al.* and Li *et al.* developed single-step non-recursive SOC estimators using LSTM, demonstrating excellent accuracy of LSTM in SOC estimation compared to SRNN[79,96]. They also found that increasing the scale of the LSTM hidden layer enhanced estimation accuracy. Li *et al.*, Jiao *et al.* and Duan *et al.* achieved comparable results using GRU[97-99]. Key LSTM parameters, such as hidden layer size, learning rate, and iteration count, were crucial to these outcomes. Ren *et al.* developed a PSO-based optimization algorithm to align LIB data characteristics with the network topology[80]. They also introduced random noise in the input layer during the training to enhance the network's resistance to interference. The resulting PSO-LSTM network outperformed others, achieving a 0.43% reduction in MSE for isothermal estimation. Chen *et al.* applied genetic algorithms to optimize GRU hyperparameters, achieving an impressive RMSE of 0.07%[81]. Ma *et al.* employed a sliding window technique to develop an LSTM-based SOC estimator[82]. In contrast to the work by Wang *et al.*, Ma *et al.* significantly reduced the window length, with the optimal length set to just 50 seconds, resulting in a slight decrease in estimation accuracy but still within an acceptable range[76,82].

Certain variations are also commonly used in SOC estimation, primarily stack-RNNs and bidirectional RNNs (BiRNNs). Stack-RNNs compute hidden layer states for sequential inputs repeatedly, using the output from the upper layer in subsequent iterations. BiRNNs, on the other hand, process both forward and reverse sequences of inputs simultaneously, combining the resulting hidden states using a sigmoid function. The Stack-BiLSTM merges both variations, with the bidirectional layer enabling the model to capture temporal features from both directions, while the stacked layer distributes parameters to prevent excessively large single hidden layer dimensions, enhancing the nonlinear processing of input data[87].

Yang *et al.* first applied Stack-LSTM to SOC estimation, developing a single-step continuous estimator that outperformed UKF in accuracy[83]. However, deviations appeared post-tests at 60% and 80% SOC states after the 250 s, and significant errors arose when the voltage mode fluctuated. Hannan *et al.* introduced a Stack-GRU-based estimator, achieving an RMSE error metric of 0.65%, surpassing Stack-LSTM[84]. Zhang *et al.* validated BiLSTM as a viable SOC estimator, though its RMSE remained below 2.5%[85]. Yang *et al.* optimized BiLSTM parameters with Bayesian optimization algorithms[100]. Chen *et al.* enhanced BiGRU hyperparameters using an improved pigeon-inspired genetic algorithm, yielding an RMSE estimation error of under 1%[86]. Bian *et al.* demonstrated the efficacy of Stack-BRNNs, with Stack-BiLSTM achieving estimation accuracies of 0.46% and 0.73% in isothermal and temperature-varying discharge tests, outperforming single-mechanism LSTM networks[87]. However, Ma *et al.* found that stacking did not improve the accuracy or stability of LSTM-based NARX estimators[82].

In addition to expanding the number of LSTM or GRU layers in a single unit, ensemble learning can estimate the same set of inputs using multiple RNN units in parallel, combining their outputs for a final SOC estimation value. Manoharan *et al.* employed this approach, with the BiLSTM-based SOC estimator, and achieved an RMSE under 0.74% in three dynamic discharge tests at 45 °C[88].

LSTM and GRU-based estimators outperform RNN methods. With optimized hyperparameters, their performance is similar; however, the GRU structure is simpler, requiring fewer training samples, and facilitates long-term memory updates via the Forget and Input gates. Therefore, there is no clear advantage between LSTM and GRU, as their performance differences are often minimal. Switching networks or adding bidirectional structures yields less improvement than hyperparameter tuning and stacking, with final outcomes dependent on the specific dataset and training methods.

Further exploration of sequence modeling strategies

SOC estimators, whether based on physical principles, models, or data-driven approaches, require specific parameters for different battery types with varying physicochemical compositions. However, as lithiation characteristics of batteries evolve with time or usage conditions, these parameters may lose effectiveness, causing errors in SOC estimation. Data-driven DL estimation algorithms, particularly NNs, heavily rely on the quality and quantity of training data, significantly increasing costs for parameter calibration compared to model-based methods. While RNNs are commonly used for sequence modeling, challenges remain in developing effective data-driven SOC estimators. Since changes in the physicochemical properties of a battery during charging and discharging introduce noise interference, gathering real SOC data required for training DL learning models for SOC is both costly and time-consuming. To overcome these issues, various

RNN-based strategies, akin to MLPs, have been explored for SOC estimation. Figure 5 illustrates improvements in sequence modeling strategies.

To address the first issue, incorporating new input variables has been suggested beyond the typical three input variables of current, voltage, and battery temperature to enhance deterministic measurement accuracy. Jiang *et al.* explored the relationship between battery stress and SOC, replacing the temperature variable with battery stress as input for Stack-LSTM, resulting in a mechanics-based SOC estimation approach[89]. While the improvement in accuracy was minimal, this method achieved comparable performance to longer input windows with shorter ones. Also, during dynamic charging and discharging processes, stress changes more gradually than other variables, capturing the overall effect of dynamic processes and reducing transient noise, thereby improving the robustness of estimators against battery measurements.

Due to the high cost of measuring new variables such as stress, most studies opt to derive additional input variables from existing ones to enhance reliability through comprehensive data. Some works focused on mechanistic approaches. For example, Chen *et al.* introduced the voltage mean of window data as an additional EI-LSTM (extended input LSTM) input, significantly improving the estimation accuracy during high-temperature dynamic discharge[101]. This concept aligns with Chemali *et al.*, both using the mean to represent the charging and discharging behaviors over time[62]. Yu *et al.* expanded the input to the DL model by extracting physical variables from raw measurements using a simplified electrochemical model[102]. The results showed that OCV and reaction polarization resistance improved the LSTM-based SOC estimation. Compared to the original input model, the extended input reduced all three error metrics by over 50%, though this improvement remains relatively modest when compared to other studies.

A common approach to address this issue is employing ML methods to extract features from the original input data. For example, Wu *et al.* applied random forest methods to reduce the dimensionality of battery aging and thermoelectric characteristics before feeding the data into LSTM and GRU networks for SOC estimation[103]. Jayaraman *et al.* utilized PCA for data reduction prior to inputting it into the LSTM[90]. Liu *et al.* identified six key variables strongly correlated with SOC through Pearson correlation analysis on real-world driving data, then adaptively combined these features using dynamic weights before inputting them into the GRU network[91]. This approach effectively predicts the SOC of EV batteries within 1-5 min, achieving an RMSE error lower than that of the Transformer and the PatchTST network - the top performers in standard time-series tasks. This concept has since been expanded, leading to numerous studies that use DL methods for feature extraction prior to inputting data into LSTM. Liu *et al.* introduced an LSTM-based estimator that activates a specific sub-network for the current SOC stage, based on the previous stage's output, while other sub-networks remain dormant[104]. This strategy focuses on the relevant stage characteristics, minimizing input interference. These approaches, categorized as temporal representation[105,106], divide the entire framework into two stages: "temporal representation" and "temporal prediction." Later, as illustrated in Figure 6, these methods are superseded by autoencoders (AEs), CNNs, and other advanced techniques, establishing the third major class of SOC estimators based on the ED architecture.

Some studies demonstrated that LSTM effectively manages noise without requiring additional input variables. For example, Almaita *et al.* utilized LSTM to estimate the SOC of utility-scale LIB storage systems, achieving an MSE below 0.62%[107]. Despite fewer epochs, the LSTM outperformed an MLP model, reducing the MSE by a factor of 11 and exhibiting superior capability to capture battery dynamics. Almaita *et al.* demonstrated that the inherent noise resistance of LSTM remains effective even when transitioning from

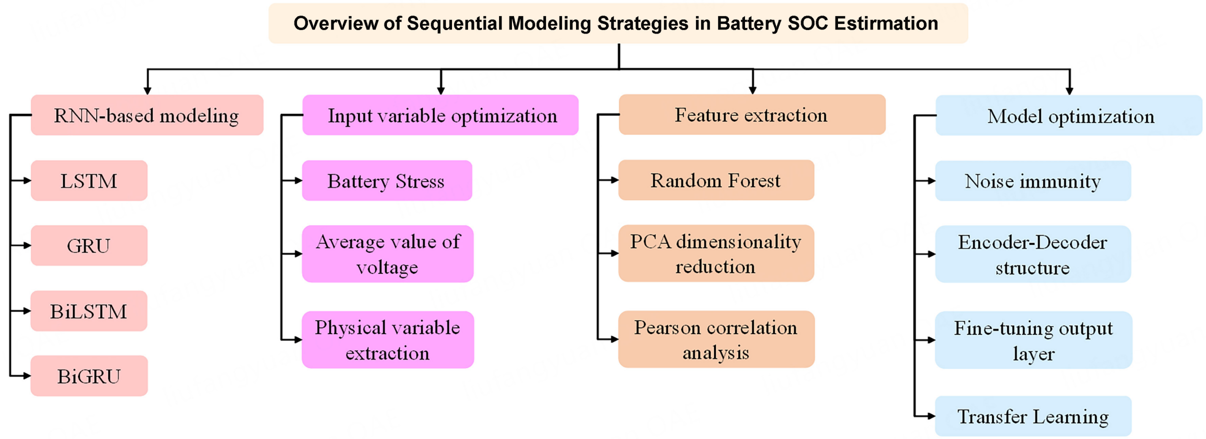
Overview of Sequential Modeling Strategies in Battery SOC Estirmation
RNN-based modeling
LSTM
GRU
BiLSTM
BiGRU
Input variable optimization
Battery Stress
Average value of voltage
Physical variable extraction
Feature extraction
Random Forest
PCA dimensionality reduction
Pearson correlation analysis
Model optimization
Noise immunity
Encoder-Decoder structure
Fine-tuning output layer
Transfer Learning

Figure 5. Sequential approaches for estimating SOC of batteries. Key strategies include RNN-based modeling (LSTM, GRU, BiLSTM, BiGRU), input variable optimization (battery stress, average voltage, physical variables), feature extraction (random forest, PCA), and model optimization (noise immunity, Encoder-Decoder structures, transfer learning). SOC: State of charge; LSTM: long short-term memory; GRU: gated recurrent unit; BiLSTM: bidirectional long short-term memory networks; PCA: pearson correlation; RNN: recurrent neural network.

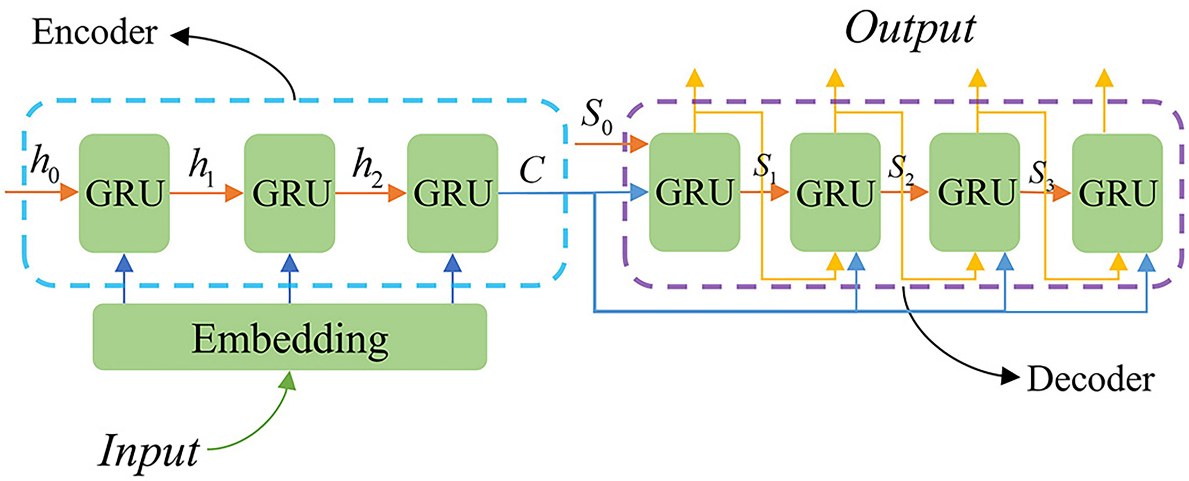

Figure 6. Structure of ED based on a GRU-GRU framework. GRU: Gated recurrent unit; ED: encoder-decoder.

single cells to complex battery packs, validating its reliability as a SOC estimator[107].

Addressing the second issue, transfer learning, initially adopted in image recognition to mitigate annotation costs, has been successfully applied to SOC estimation. Liu *et al.* employed transfer learning to fine-tune LSTM output layer parameters using minimal data from the NCM battery, achieving accurate SOC estimations[92]. This method reduced the required data volume by 70%, significantly enhancing the practicality of deep estimators. Wang *et al.* utilized transfer learning for GRU-based estimators, cutting the training time for target batteries by nearly 4 min[108]. Eleftheriadis *et al.* analyzed the impact of eight different transfer learning techniques on four different methods for SOC estimation, such as LSTM, GRU, BiLSTM, and BiGRU[93]. Comparative results with trained models optimized via Bayesian hyperparameter tuning demonstrated that hybrid architectures and adaptive hidden state techniques are always one of the most effective technologies in both accuracy and computation time. Che *et al.* proposed a multi-task learning approach for joint prediction of SOC, state of energy (SOE), and future temperature, emphasizing the role of transfer learning[109]. Inspired by Caruana, they employed shared layers to extract hidden features of different battery states and task-specific layers to achieve accurate multi-state monitoring[110]. By incorporating a transfer learning strategy to retrain specific NN layers, the method effectively adapts to varying discharge profiles, environmental temperatures, and aging conditions, significantly enhancing the model's generalization across diverse application scenarios.

In conclusion, RNN-based networks offer a simple structure with significant effectiveness, particularly in SOC estimation. LSTM and GRU networks, with their inherent noise resistance and strong sequence modeling capabilities, have proven to be effective for a range of research applications. As a result, they remain widely applied in SOC estimation. However, despite extensive research, the limitations of RNNs have constrained further improvements in the accuracy of SOC estimation. Additionally, the emergence of ED structures in other fields has highlighted the benefits of combining various networks, leading to more robust performance under complex load conditions and disturbances than standalone RNNs do.

Sequence modeling techniques based on CNN

Similar to MLPs, CNNs lack inherent historical memory but can sequentially process data through the sliding of the receptive field, offering limited temporal functionality. On this basis, temporal convolutional networks (TCNs) extend CNNs by introducing causal convolution and residual links, creating efficient time-series prediction models. As opposed to RNNs derived from MLPs, TCNs provide memory capabilities with lower complexity costs and avoid vanishing or exploding gradients issues. Guo and Ma found that TCNs outperformed FCNN, LSTM, and GRU in robustly estimating SOC under noisy battery conditions[111].

CNNs offer deployment advantages due to their efficient convolutional mechanisms, with most learnable parameters concentrated in the convolutional kernels. In contrast to the gate structures in SRNN and its variants, CNNs require fewer parameters, enabling easier device-side deployment and hardware acceleration. For example, Guo and Li implemented a trained FCNN on FPGAs[72]. Mazzi *et al.* compared CNNs and GRUs, deploying a more accurate and less complex 1D-CNN on an STM32 embedded chip[73]. Using the STM32Cube AI framework, the 1D-CNN quantification model achieved an inference speed of 0.971 ms. For complex TCNs, Pau *et al.* demonstrated that, although TCNs initially required the most MACCs and had the longest inference time on two microcontroller unit (MCUs) without hardware acceleration, STM32MP257F-EV1 optimization significantly improved the performance[112]. The accelerated parallel TCN achieved the lowest MSE and fastest inference time among tested networks, making it the most efficient solution for SOC estimation in EV batteries.

ED-based sequence modeling techniques

RNNs and CNNs are commonly employed for sequence modeling in SOC estimation, but their fixed hidden state size often exceeds the required estimation steps. To address this issue, a FC layer processes the final hidden state vector to generate outputs, forming an ED framework. This NNs-MLP structure enables flexible estimation lengths by adjusting the decoding of the output size of MLP. The ED structure effectively reads and generates sequences of arbitrary length, which is more challenging for traditional networks.

Viewed through the lens of information compression and restoration, stacks of multiple MLPs and CNN layers can be seen as simplified versions of the ED structure, despite differing design philosophies. The Encoder handles information compression, extracting relevant features from input data, while the Decoder reconstructs or predicts output using these features. ED-based estimators can be classified into CNN-CNN, RNN-RNN, and CNN/AE-RNN categories, depending on the networks used in the Encoder and Decoder.

CNN-CNN and RNN-RNN

The design philosophy of CNN-CNN and RNN-RNN ED structures is similar: both use MLPs and CNNs to capture long-term dependencies and sequential patterns through specialized designs, with Encoding and Decoding performed by the same structure. This approach offers the advantage of structural consistency, simplifying the model design and training. It facilitates hyperparameter optimization algorithms and embedded deployment, as the encoder and decoder can share similar configurations.

Significant research has focused on SOC estimation using RNN-ED structures. Cui *et al.* applied an LSTM-based ED model to learn the nonlinear relationship between SOC and measurable current and voltage[113]. Bian *et al.* employed BiLSTM as a recursive unit to capture temporal dependencies of sequence data in both the Encoder and Decoder[114], enhancing and improving estimation accuracy, as depicted in Figure 7A. Among various recurrent units, such as SRNN, GRU, and LSTM, the bidirectional structure typically yields the highest estimation accuracy, effectively capturing battery dynamics and delivering precise SOC sequence estimation across diverse ambient temperature conditions. Research shows that the ideal input sequence length improves the model's learning capacity and prediction precision, while the model's depth, determined by the number of hidden layers, also influences the final output quality. This finding was corroborated by Terala *et al.*[115]. Ma *et al.* enhanced the LSTM unit, introducing the PLSTM, which integrates state information and processes it together[116]. Using this approach, the sequence-to-sequence mapping with process information (SSMPI) model was developed, which leverages empirical process data between input sequence steps, beyond just the relationship between multi-step measurable parameters and SOC. It also incorporated a two-stage pre-training approach to boost the feature extraction capabilities of the model, enabling it to tackle challenges from diverse load configurations and fluctuating sampling frequencies. The first phase used an unsupervised AE pre-training technique, enhancing the model's ability to learn sequence-to-sequence mappings and improve generalization. The second phase involved supervised training for parameter fine-tuning. Experimental results suggested that, even under complex unknown dynamic load profiles and varying sampling frequency rates, the SSMPI model achieved precise SOC estimation with certain robustness.

Research on SOC estimators using CNN-ED is limited. The work by Fan *et al.* was the only one to apply the U-Net architecture, originally designed for image segmentation to SOC estimation, as demonstrated in Figure 7B[117]. The convolutional layer structure of this method accommodates variable-length input data and produces SOC estimation of corresponding lengths. By incorporating symmetric padding convolution layers to minimize boundary effects and using total variation loss functions to improve smoothness and reduce maximum errors, the study tackled the inaccuracies of the initial estimation of traditional SOC estimators and enhanced both accuracy and stability across varying input lengths through the CNN structure.

As discussed above, the use of RNN-ED in SOC estimation significantly exceeds that of CNN-ED, primarily for two main reasons. First, RNNs were built on their early dominance in early sequence modeling. They are adept at handling temporal dependencies in sequence data, addressing challenges faced by early models with long sequences, which made them the dominant approach until the late 2010s. With the rise of DL platforms such as TensorFlow and PyTorch, RNNs remained prevalent, particularly in domains where they had already proven effective. Consequently, in the early 2020s, RNNs continued to be widely used in sequence modeling tasks. The slow adoption of new ML technologies in SOC estimation ensures that once a technique is established and researchers are proficient with it, its usage remains widespread for an extended

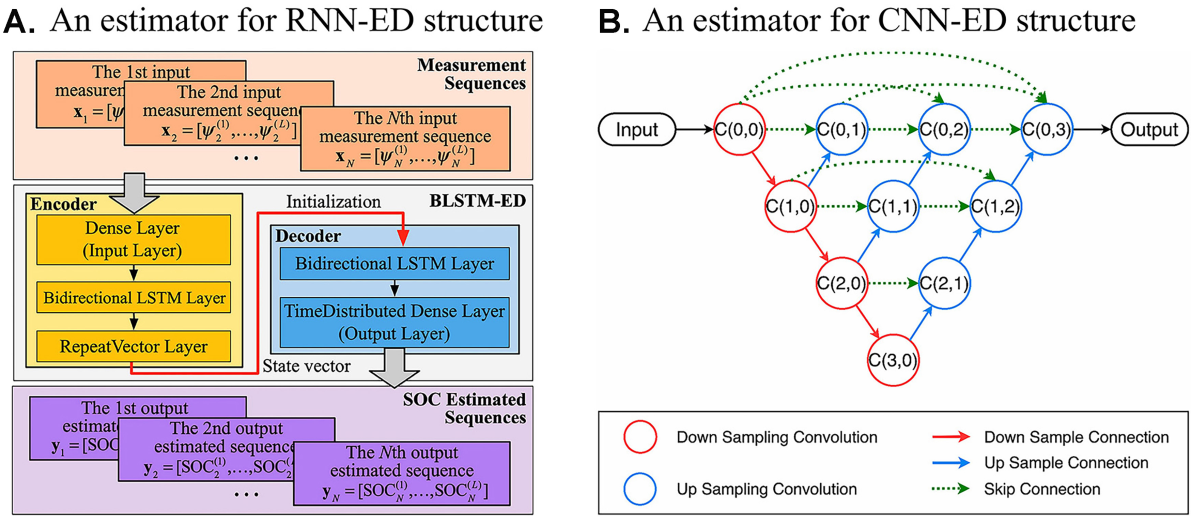
A. An estimator for RNN-ED structure
Measurement Sequences
The 1st input measurement sequence $\mathbf{x}_1 = [\psi$
The 2nd input measurement sequence $\mathbf{x}_2 = [\psi_2^{(1)},\ldots,\psi_2^{(L)}]$
The Nth input measurement sequence $\mathbf{x}_N = [\psi_N^{(1)},\ldots,\psi_N^{(L)}]$
. . .
Encoder
Dense Layer (Input Layer)
Bidirectional LSTM Layer
RepeatVector Layer
Initialization
BLSTM-ED
Decoder
Bidirectional LSTM Layer
TimeDistributed Dense Layer (Output Layer)
State vector
SOC Estimated Sequences
The 1st output estimated sequence $\mathbf{y}_1 = [\mathrm{SOC}$
The 2nd output estimated sequence $\mathbf{y}_2 = [\mathrm{SOC}_2^{(1)},\ldots,\mathrm{SOC}_2^{(L)}]$
The Nth output estimated sequence $\mathbf{y}_N = [\mathrm{SOC}_N^{(1)},\ldots,\mathrm{SOC}_N^{(L)}]$
. . .
B. An estimator for CNN-ED structure
Input
C(0,0)
C(0,1)
C(0,2)
C(0,3)
Output
C(1,0)
C(1,1)
C(1,2)
C(2,0)
C(2,1)
C(3,0)
Down Sampling Convolution
Up Sampling Convolution
Down Sample Connection
Up Sample Connection
Skip Connection

Figure 7. Structures of RNN-ED (A) and CNN-ED (B) SOC estimators[114,117]. Elsevier Copyright 2024. RNN: Recurrent neural network; CNN: convolutional neural network; ED: encoder-decoder; SOC: state of charge.

period.

The second reason is that CNN-based sequence data processing technologies emerged later, and CNN-based ED networks were primarily focused on the image field, making their application to SOC estimation more complex and costly. Among CNN-EDs models such as U-Net, DeepLab v3+, and Mask R-CNN, only U-Net has been used for SOC estimation[117]. While CNNs are very effective in handling fixed-length data and can process sequences through techniques such as one-dimensional convolution, they are generally less suited than RNNs for capturing long-term dependencies in sequence data. In contrast to CNN-ED, RNN-ED models are more widely employed in tasks such as SOC estimation, which require time series analysis. In SOC estimators based on RNN-ED, SRNN networks are no longer employed, as LSTM and GRU models, which incorporate gating mechanisms, have proven to overcome the defects of SRNN since 2019.

CNN/AE-RNN

In addition to the standard RNN-ED and CNN-ED architectures, the CNN/AE-RNN model is another widely used ED structure. Three factors drive the use of this asymmetric model in SOC estimation. First, ML methods for feature extraction from raw input data are highly effective[90,91,103], with some methods even surpassing[91] Transformer models and top implementations such as PatchTST in time series prediction. This "temporal representation-temporal prediction" two-stage approach is effectively carried forward in ED through the asymmetric structure. Using AEs, CNNs, and other NNs for temporal representation simplifies training and deployment within a unified framework and environment, enhancing practical network applications. While RNN-based estimators and their variants meet current accuracy requirements, they lack sufficient internal filtering capabilities, struggling to handle local disturbances during dynamic battery loading. This limitation leads to significant peaks in estimation errors, with MAX values substantially larger than MAE and RMSE. Additionally, although CNNs excel at capturing local features, RNNs, particularly those using gating mechanisms such as LSTM or GRU, are better suited for modeling long-distance dependencies. The CNN/AE-RNN structure leverages the strengths of both networks, with CNN/AE extracting local time-varying or spatial features and RNN capturing long-term dependencies. Alternatively, specialized CNNs such as TCNs can replace RNNs, forming CNN-TCN estimators (which can actually be classified as CNN-ED). However, this approach has not yet been explored in SOC estimation.

Bhattacharjee *et al*. simplified and expanded the use of CNN in the Encoder of CNN/AE-RNN by replacing varying receptive fields with fixed-length fields to independently extract time-varying information across different channels[66]. For extracting time-varying features from a single channel, both 1D-CNN and 2D-CNN can be utilized. For example, Ardeshiri *et al.* converted battery signals into multi-channel images, applied 2D-CNN for feature extraction and then employed stacked bidirectional LSTM networks (SBiLSTM) to capture temporal sequence dependencies, achieving precise SOC estimation under varying ambient temperatures[118]. Bian *et al.* introduced a multi-channel CNN (MCNN) utilizing 1D-CNNs for multi-scale feature extraction and fusion, followed by BiLSTM-based bidirectional time learning in the decoder stage, as demonstrated in Figure 8[65]. This approach effectively captures local invariant characteristics and time dependencies in battery data, demonstrating superior SOC estimation performance across multiple public LIB datasets under both stable and fluctuating ambient temperatures. Shen *et al.* developed the source-free temperature transfer network (SFTTN) framework for passive SOC estimation, leveraging pre-trained active models and target data through a 2D-CNN-BiLSTM network to overcome challenges of distribution differences and data scarcity in cross-domain scenarios[119]. Extensive experiments confirmed that SFTTN outperforms or matches other methods in both fixed-temperature and cross-domain SOC estimation under varying ambient temperatures.

The use of AEs as encoders, similar to CNN, is aimed at feature extraction and denoising. Chen *et al.* applied a denoising AE (DAE) in the encoder to extract key data features, reduce noise, and expand the measurement dimensionality of battery data[120]. They then used the processed data to train a GRU-RNN, improving the accuracy and robustness of SOC estimation. The DAE enhanced the ability of GRU to capture the nonlinear relationship between the data sequence and SOC. Savargaonkar *et al.* proposed a SOC estimator based on uncorrelated sparse AE with LSTM (USAL)[121]. This method leverages sparse AE and LSTM's strengths, with the AE in USAL generating effective SOC encoding while reducing multicollinearity to identify significant long and short-term features, aiding LSTM in efficiently modeling time-series data.

ED-based SOC estimators enhance estimation accuracy and robustness, but their network components face limitations due to spatial and temporal constraints. In the ED structure, the encoder compresses the entire input sequence into a fixed-length context vector, which must carry sufficient information for the decoder to generate output. However, for long sequences, a single vector may not encapsulate all necessary details, creating an information bottleneck. While LSTMs can form long-term memories, they may struggle to capture distant dependencies. Similarly, CNNs, which use a sliding window approach, are restricted by the local receptive field and mainly capture local information, whereas global information is crucial in SOC estimation. Additionally, RNNs face challenges with network interpretability. These issues can be mitigated by incorporating an important structure.

Summary

Table 5 summarizes various studies on SOC estimation of LIBs using ED structures, with detailed references, model names, model architectures, evaluation methods, window sizes, benchmark comparisons, battery types, battery operating temperatures, test profiles, and error ranges for each approach. This provides a comprehensive framework for assessing the performance of different methods and provides key insights into the challenges of ED modeling and potential future development.

This review has traced the evolution of SOC estimators, initially without attention mechanisms. Network structures have advanced from MLP to CNN/RNN, and then later to ED networks combining both CNN and RNN, resulting in increased complexity, more learnable parameters, and improved overall accuracy. Early attention-based models augment existing networks to address memory decay, while later research

Table 5. Comparison of SOC estimation methods using ED structures

Reference	Model name	Model structure/Strategy	Estimation mode	Window size	Compared models	Dataset	Temperature (°C)	TestProfile	Error (%)
[113]	ED	LSTM-LSTM-FC	NRSS	50	LSTM GRU	INR-18650-20R[4]	0 °C 25 °C 45 °C	FUDS DST US06	MAE < 0.69 RMSE < 0.86
[114]	BLSTM-ED	BLSTM-BLSTM	NRMS	40	SRNN-ED GRU-ED LSTM-ED	NCR-PF18650[70]	-20 °C -10 °C 0 °C 10 °C 25 °C	US06 HWFET	MAE < 2.87 MAX < 6.39
[116]	SSMPL	PLSTM-PLSTM	NRMS	10	LSTM	satellite battery (Private data) INR-18650-20R[4]	0 °C 25 °C 45 °C	FUDS US06 BJDST	MAE < 1.45
[117]	U-Net	U-Net (CNN-ED)	NRMS		EKF RNN	NCR-PF18650[70]	-20 °C -10 °C 0 °C 10 °C 25 °C 10 °C -25 °C -20 °C -10 °C	US06	MAE < 1.1 RMSE < 1.4 MAX < 7.4 MAE < 1.2 (v.t.) RMSE < 1.5 (v.t.) MAX < 5.0 (v.t.)
[118]	CNN-SBLSTM	CNN-SBLSTM	NRSS	20		NCR-PF18650[70] Lishen LP2770102AC (Private data)	0 °C 25 °C	US06 HWFET	MAE < 0.54 MAX < 3.39
[65]	MCNN-BLSTM	CNN-BLSTM	NRMS	128	MCNN-BSRNN MCNN-BGRU MCNN-LSTM MLP GRU LSTM LSTM-ED SVM CNN-GRU CNN-LSTM	NCR-PF18650[70] INR-18650-20R[4] A123-18650[1]	-20 °C -10 °C 0 °C 10 °C 25 °C 10 °C -25 °C -20 °C -10 °C	LA92 (NCR) UDDS (NCR) BJDST (INR) US06 (INR) FUDS (INR) DST (A123) US06 (A123) FUDS (A123)	MAE < 1.01 (NCR) RMSE < 1.61 (NCR) MAX < 5.27 (NCR) MAE = 0.98 (NCR, v.t.) MAE < 1.08 (INR) RMSE < 1.39 (INR) MAX < 3.04 (INR) MAE < 0.82 (A123) RMSE < 1.36 (A123) MAX < 2.53 (A123)
[119]	SFTTN	CNN-LSTM-FC With TL	NRSS	256	GRU LSTM AUKF	NCR-PF18650[70] LG 18650-	-20 °C -10 °C 0 °C	Other temp to 25 °C Constant	MAE = 4.17 (temp

					EIS-GPR	HG2[71] A123-18650[1]	10 °C 25 °C 30 °C 40 °C 50 °C	temp to vary NCR to LG NCR to A123	c. to c., Ave.) MAE = 6.11 (temp c. to v., Ave.) MAE = 2.14 (NCR to LG, Ave.) MAE = 2.19 (NCR to A123, Ave.)
[120]	DAE-GRU	DAE-GRU	NR	50	SRNN GRU	NCM (Private data)	25 °C	UDDS HWFET NEDC	MAE < 3.36 RMSE < 4.19
[121]	USAL	SAE-LSTM	NRSS		DNN LSTM AE-LSTM GPR RGPR AR-RGPR SLSTM	NASA Ames Labs data[122] APR18650M1A[123] Diao stress acc-aging[124]	0 °C 25 °C 45 °C	CCCV US06 FUDS DST BJDST	MAE < 0/50 RMSE < 0.99

SOC: State of charge; ED: encoder-decoder; LSTM: long short-term memory; FC: fully connected layer; NRSS: non-recursive single-step; GRU: gated recurrent unit; NRMS: non-recursive multi-step; SRNN: simple recurrent neural network; CNN: convolutional neural network; EKF: extended kalman filter; RNN: recurrent neural network; MCNN: multi-column convolutional neural network; MLP: multilayer perceptron; SVM: support vector machine; GPR: gaussian process regression.

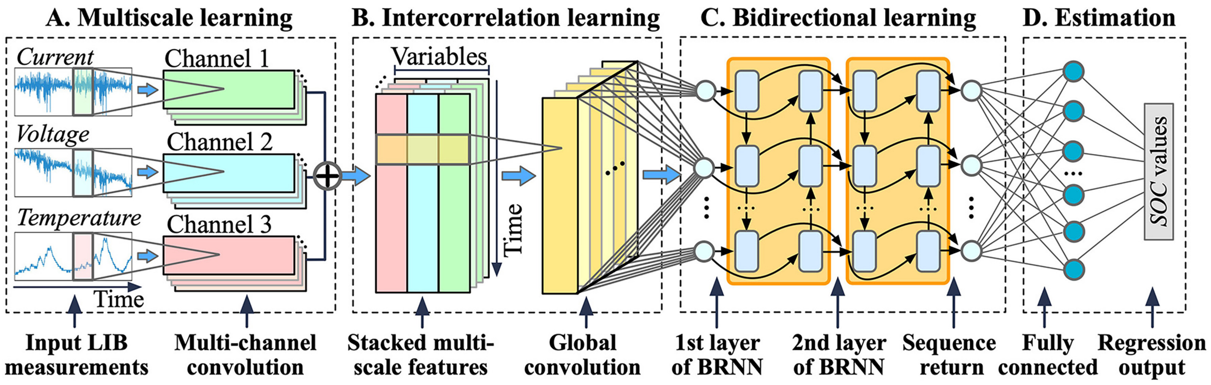

Figure 8. Proposed MCNN-BRNN architecture for SOC estimation[65].(A) Multiscale learning; (B) Intercorrelation learning; (C) Bidirectional learning; (D) Estimation. Elsevier Copyright 2024. MCNN: Multi-column convolutional neural network; BRNNs: bidirectional recurrent neural network; SOC: state of charge; LIB: lithium-ion battery.

introduces Transformer models with attention layers for SOC estimation, achieving strong performance in time series tasks. Despite this obstruction, many studies still integrate these models with Transformers to enhance SOC estimation, a trend likely to persist.

Early SOC estimation research focused on MLP and RNN, establishing three basic modeling approaches for practical SOC estimation tasks. Later studies incorporated true value inputs from standard time series predictions, but these models lack practical relevance and are difficult to compare with existing studies. Therefore, when building deep SOC estimators, it is essential to critically assess the three modeling strategies mentioned in the "SOC estimation task design" section, as they remain fundamental in the short term.

The initial phase of research encompassed a wide range of training methods, including hyperparameter optimization through advanced intelligent optimization algorithms, rapid multi-battery and multi-scenario model generation based on transfer learning, and the "unsupervised pre-training" along with "supervised fine-tuning"[116] approach. These strategies not only enhance modeling accuracy and stability but also accelerate the creation of deep estimators for new estimation, reducing the need for extensive data collection and lengthy training periods. Despite these advancements, practical deployment of SOC estimators still faces significant challenges, particularly when applied across varying battery chemistries and pack configurations, such as lithium-ion, nickel-metal hydride (NiMH), and lead-acid batteries, which exhibit unique OCV-SOC characteristics. For instance, LIBs typically show nonlinear OCV-SOC relationships with distinct plateau regions, while NiMH and lead-acid batteries may have more linear profiles. These differences are further complicated by environmental factors, including temperature fluctuations and aging effects, which alter the OCV-SOC curve over time and under different operating conditions[125].

In battery packs, the issue of cell inconsistencies becomes particularly prominent. Factors such as manufacturing differences, uneven aging, and varying usage conditions lead to disparities in individual cell SOCs, which propagate estimation errors at the pack level. Some methodologies, including fuzzy adaptive federated filtering[125] and GPR[60], have been developed to address these challenges. These approaches leverage multi-source data fusion and adaptive estimation frameworks to improve accuracy and fault tolerance in complex systems. However, achieving universally robust SOC estimation frameworks that effectively address these cross-chemistry and cross-configuration challenges remains an open research question.

SUMMARY AND OUTLOOK

This work primarily reviews the research progress in SOC estimation of LIBs, focusing on both traditional ML techniques and DL-based methods. In traditional ML, we reviewed the application and limitations of KNN, Decision trees, SVM, ELM, and GPR for SOC estimation. While effective in certain contexts, these methods require extensive feature engineering and are sensitive to parameter choices, ultimately becoming computationally intensive with large datasets. Their performance is often hindered by battery aging and nonlinear dynamics. In contrast, DL technology and methods, particularly sequence models such as RNN, LSTM, and GRU, have shown significant promise in advancing SOC estimation. These networks autonomously extract features and identify complex patterns and relationships in the data, outperforming traditional methods in accuracy and reliability. Additionally, ED-based networks enhance SOC estimation by integrating various network architectures to capture both local and global information. Future research directions for ML methods in SOC estimation are outlined in Figure 9.

Distributed training

The methods for SOC estimation, as reviewed in this work, rely on traditional centralized learning approaches, which require substantial amounts of data from battery operation, typically gathered from controlled laboratory environments. However, real-world battery usage scenarios are much more complex, with dynamic, multi-dimensional changes. To improve model generalization and adaptability, one must collect data from batteries operating under diverse conditions for training and prediction. Although this approach enhances model accuracy, it introduces challenges such as high data transmission volumes, elevated transmission costs, and data privacy concerns. Frequent data transmission can also cause interference and distortion, compromising the accuracy of SOC estimation. To mitigate these issues,

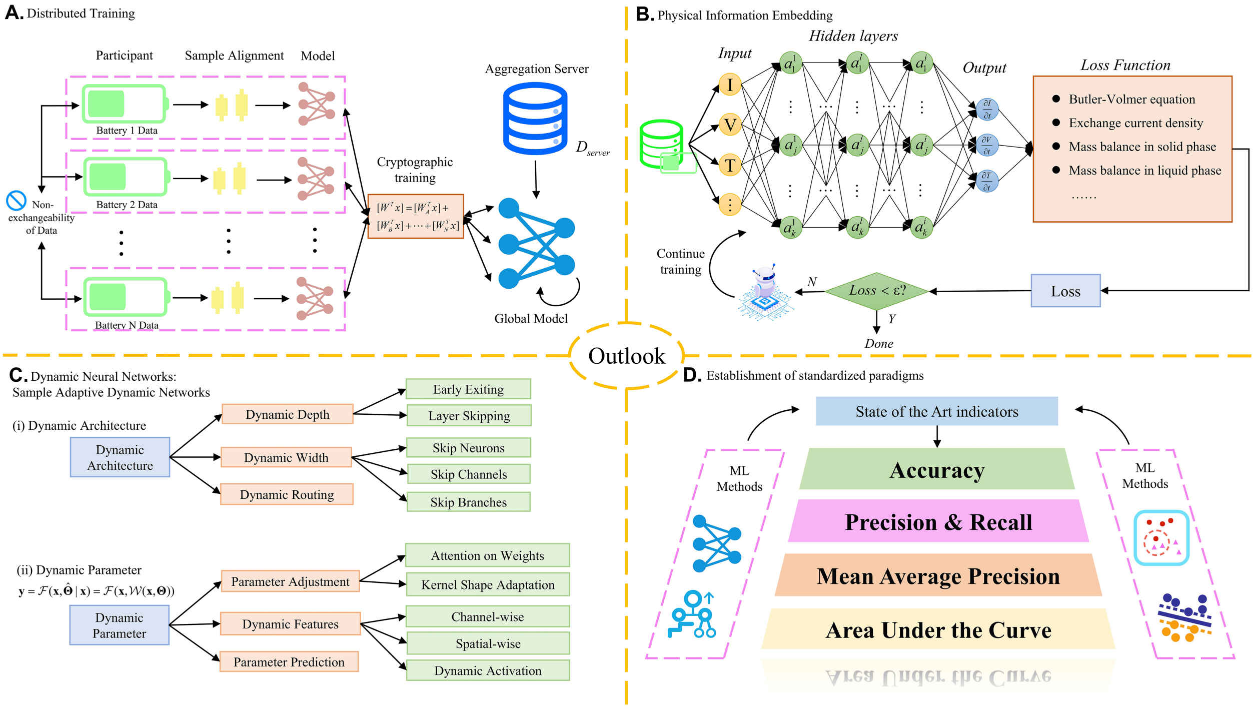

Figure 9. Potential research directions for SOC estimation of LIBs. (A) Distributed training; (B) Embedding of physical information; (C) Dynamic neural network integration; (D) Development of standard paradigms. SOC: State of charge; LIB: lithium-ion battery; ML: machine learning.

distributed training presents a promising solution. In this method, only model parameters are exchanged and updated on the server side, eliminating the need to transmit extensive raw data. This method reduces transmission volume and minimizes interference risks. With advancements in edge computing, distributed training enhances local computing power, enabling model training at the source of data generation. Federated learning (FL), a distributed training approach, allows multiple participants to approach models while preserving data privacy and security. In this framework, one can locally train models and only share updates, minimizing data transfer and enhancing security. Additionally, FL accommodates the heterogeneous nature of battery data, improving the model's adaptability across various batteries and application scenarios. This method offers considerable promise for the research of SOC estimation and has been recently used to enhance traditional NNs for battery SOH estimation. *Lv et al.* combined artificial NNs (ANNs) with FL to predict LIB SOH in real-world scenarios[126]. Their local ANN model outperformed the widely used LSTM model, proving more suitable for EVs while minimizing the need for extensive data uploads. Wong *et al.* employed an LSTM-based ED structure for LIB SOH prediction, demonstrating that their decentralized model performed similarly to centralized approaches[127]. The success of FL highlights its potential for SOC estimation, offering innovative approaches to improve accuracy while preserving data privacy and minimizing transmission. By integrating FL with NNs such as ANNs and LSTM, precise battery status predictions can be achieved without extensive data transfer. These research advancements suggest that future SOC estimation may increasingly rely on distributed learning methods to enhance performance while reducing reliance on data transmission.

Physical information embedding

A key issue in the research literature on SOC estimation of LIBs is that existing models often neglect the internal physical and chemical mechanisms of batteries. Although data-driven approaches, such as DL, have advanced in modeling the complex nonlinear relationships of battery state, they generally lack an intuitive grasp of the underlying physical processes. This limitation reduces model generalization and leads to significant degradation in performance, especially when the battery conditions change due to aging or varying operational environments. Moreover, these models usually require extensive labeled data for training, which may be impractical and costly. Recently, the rise of AI for Science has highlighted physical information embedding as a promising approach. By incorporating the physical and electrochemical properties of batteries into data-driven models, this method enhances both accuracy and robustness. It not only improves the understanding of battery behavior but also aids in data-limited scenarios by relying on fundamental physical principles rather than solely on data. For example, embedding electrochemical parameters or state equations[128] into the models offers prior knowledge, enabling more precise predictions when data is scarce. Incorporating physical information enhances model adaptability, enabling it to address challenges such as battery aging and varying operating conditions more effectively. Battery aging, characterized by the changes in capacity, internal resistance, and other key parameters impacted by complex physical and chemical changes, can be better predicted by embedding the aging mechanisms into the model, resulting in a more accurate SOC estimation. Additionally, batteries exhibit varying behaviors under different operating conditions, including various temperatures, charge/discharge rates, and load conditions. Embedding physical information enables the model to account for the variations, improving its reliability and performance across diverse operating conditions, reducing reliance on extensive training data, and thereby lowering experimental costs and data collection requirements. This approach is particularly beneficial for advancing new battery technologies and optimizing existing battery systems. For example, Nascimento *et al.* introduced a hybrid physical-NN method for LIBs, integrating known discharge physics, such as the Nernst and Butler-Volmer equations, with flexible ML to enhance uncertainty quantification and improve consistency with experimental data[129]. Future research should focus on integrating physical information with data-driven models to enhance SOC estimation across varying battery states and operating conditions, including developing advanced algorithms for the seamless incorporation of physical information and data-designing architectures that accurately reflect the physical and chemical characteristics of batteries. Additionally, utilizing physical information to improve model interpretability can make SOC estimation more transparent and reliable, fostering trust among engineers and researchers to understand the model's predictions. These advancements will enable the creation of precise and robust models for SOC estimation, providing essential support and designing BMS.

Integration of DNN

Recent research on battery SOC estimation has heavily utilized NN methods. However, most studies concentrated on refining model architecture or specific network design, often neglecting a deeper theoretical explanation of their operational mechanisms. This "building block" approach creates challenges. Without theoretical guidance, determining the optimal network configuration is difficult, leading researchers to rely on empirical adjustments, such as increasing network layers or neurons, to improve data fitting. This empirical approach often introduces challenges, such as deeper networks complicating training and increasing the risk of overfitting. Therefore, achieving both computational efficiency and model accuracy in network design requires careful exploration of optimal parameter configuration. However, most studies fail to provide these configurations or justify the choice of specific parameters, leaving future research without a clear direction. DNNs[130] offer a potential solution, advancing the research of SOC estimation through their abilities to adaptively adjust their structure and parameters based on input data during the inference phase, providing greater flexibility than traditional static networks. In contrast, static networks maintain fixed configurations post-training, which may be suboptimal in practical applications due to limited representational ability and inflexible resource allocation. DNNs, however, allocate computational resources efficiently by activating specific modules as needed. For example, in SOC estimation, dynamic networks can adapt to changes in battery usage conditions, optimizing both inference and energy efficiency while maintaining accuracy. Accurate SOC estimation necessitates a model with robust representational capacity to predict charge/discharge and lifespan effectively. The architecture design of static networks is constrained by fixed parameters, limiting their ability to adapt to complex inputs. In contrast, DNNs enhance expressive ability by dynamically adjusting the network depth or activating certain specific modules as needed. One of the advantages of DNNs in SOC estimation is their adaptability to varying conditions. In contrast to static networks, which need frequent parameter adjustments, DNNs can flexibly adjust their structure during inference, ensuring an optimal balance between accuracy and efficiency across diverse environments. DNNs are compatible with existing NN technologies and can seamlessly incorporate various advanced DL technologies. Current SOC estimation methods using CNN and RNN have shown practical success, but their static structures restrict model optimization for specific tasks. Integrating dynamic techniques could potentially improve the performance of these models.

Establishment of standard paradigms

Finally, a key issue in the current ML methods, particularly in SOC estimation of batteries, is the limited comparability of reported results. This discrepancy arises because different research groups use proprietary datasets for model training and testing, with varying dataset quality and non-standardized evaluation methods. The lack of standardized datasets and evaluation methods makes it difficult to compare the advantages and disadvantages of different approaches, hindering the in-depth research of the academic community. Therefore, establishing a unified evaluation framework for SOC estimation is crucial, offering a consistent benchmark for fair performance comparison across methods using the same datasets and metrics. This not only facilitates the identification of optimal technologies but also ensures fair and transparent model comparisons. An effective evaluation framework should encompass key metrics such as accuracy, stability, and efficiency. Accuracy measures how closely the SOC estimation values of the model align with true values under varying conditions; stability assesses the robustness of the model to data fluctuations and environmental changes; and efficiency evaluates the processing speed and resource usage of the model. Evaluating these indicators across various environmental conditions, such as temperature variations, enables a thorough assessment of model performance. In addition, this evaluation standard should allow for the identification of the strengths and weaknesses of different methods, thereby facilitating the selection of the most appropriate model. The field of SOC estimation can benefit from the best practices of other domains. For instance, in image recognition and natural language processing, fully established state-of-the-art (SOTA) models serve as benchmarks, reflecting the latest advancements and offering a clear target for new research to catch up.

DECLARATIONS

Authors' contributions

Conceived and designed the article: Li, Y.; Yang, F.
Collected and analyzed data: Wu, Y.; Bai, D.
Developed programmers and performed data visualization: Wu, Y.; Bai, D.; Zhang, K.
Drafted manuscript: Wu, Y.
Reviewed and edited the manuscript: Zhang, K.; Li, Y.; Yang, F.
Supervised the project and provided financial support: Li, Y.; Yang, F.; Zhang, K.
All authors reviewed the results and approved the final version of the manuscript.

Availability of data and materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Financial support and sponsorship

Li, Y. is grateful for the support from the Natural Science Foundation of Shanghai under grant (No. 24ZR1425400). Zhang, K. is grateful for the support from the National Natural Science Foundation of China under grant (No. 12372173) and the Natural Science Foundation of Shanghai under grant (No. 23ZR1468600).

Conflicts of interest

Li, Y. and Yang, F. are Guest Editors of the special issue "*Machine Learning for Materials Development and State Prediction in Lithium-ion Batteries*" but were not involved in any steps of editorial processing, notably including reviewer selection, manuscript handling, or decision-making. The other authors declare that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Xing, Y.; He, W.; Pecht, M.; Tsui, K. L. State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures. *Appl. Energy*. **2014**, *113*, 106-15. DOI
2. Ng, K. S.; Moo, C.; Chen, Y.; Hsieh, Y. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Appl. Energy*. **2009**, *86*, 1506-11. DOI
3. He, H.; Xiong, R.; Fan, J. Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *Energies* **2011**, *4*, 582-98. DOI
4. Zheng, F.; Xing, Y.; Jiang, J.; Sun, B.; Kim, J.; Pecht, M. Influence of different open circuit voltage tests on state of charge online estimation for lithium-ion batteries. *Appl. Energy*. **2016**, *183*, 513-25. DOI
5. He, L.; Guo, D. An improved coulomb counting approach based on numerical iteration for SOC estimation with real-time error correction ability. *IEEE. Access*. **2019**, *7*, 74274-82. DOI
6. Liu, P.; Xu, R.; Liu, Y.; Lin, F.; Zhao, K. Computational modeling of heterogeneity of stress, charge, and cyclic damage in composite electrodes of Li-ion batteries. *J. Electrochem. Soc.* **2020**, *167*, 040527. DOI
7. Xu, R.; Yang, Y.; Yin, F.; et al. Heterogeneous damage in Li-ion batteries: experimental analysis and theoretical modeling. *J. Mech. Phys. Solids*. **2019**, *129*, 160-83. DOI
8. Ng, M.; Zhao, J.; Yan, Q.; Conduit, G. J.; Seh, Z. W. Predicting the state of charge and health of batteries using data-driven machine learning. *Nat. Mach. Intell.* **2020**, *2*, 161-70. DOI
9. Hossain, L. M.; Hannan, M.; Hussain, A.; et al. Data-driven state of charge estimation of lithium-ion batteries: algorithms, implementation factors, limitations and future trends. *J. Clean. Prod.* **2020**, *277*, 124110. DOI
10. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE. Trans. Inf. Theory*. **1967**, *13*, 21-7. DOI
11. Salzberg, S. L. C4.5: programs for machine learning by j. ross QuinLan. Morgan Kaufmann Publishers, Inc., 1993. *Mach. Learn.* **1994**, *16*, 235-40. DOI
12. Hearst, M.; Dumais, S.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE. Intell. Syst. Appl.* **1998**, *13*, 18-28. DOI
13. Huang, G.; Zhu, Q.; Siew, C. Extreme learning machine: theory and applications. *Neurocomputing* **2006**, *70*, 489-501. DOI
14. Talluri, T.; Chung, H. T.; Shin, K. Study of battery state-of-charge estimation with kNN machine learning method. *IEIESPC*. **2021**, *10*, 496-504. DOI
15. Song, S.; Zhang, X.; Gao, D.; et al. A hierarchical state of charge estimation method for lithium-ion batteries via xgboost and kalman filter. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Toronto, Canada, October 11 - 14, 2020; IEEE Press; pp 2317-22. DOI
16. Antón JC, García Nieto PJ, Blanco Viejo C, Vilán Vilán JA. Support vector machines used to estimate the battery state of charge. *IEEE. Trans. Power. Electron.* **2013**, *28*, 5919-26. DOI
17. Dou, J.; Ma, H.; Zhang, Y.; et al. Extreme learning machine model for state-of-charge estimation of lithium-ion battery using salp swarm algorithm. *J. Energy. Storage*. **2022**, *52*, 104996. DOI

18. Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533-6. DOI
19. Lecun, Y.; Boser, B.; Denker, J. S.; et al. Backpropagation applied to handwritten zip code recognition. *Neural. Comput.* **1989**, *1*, 541-51. DOI
20. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U. S. A.* **1982**, *79*, 2554-8. DOI PubMed PMC
21. Hannan, M. A.; Lipu, M. S. H.; Hussain, A.; Saad, M. H.; Ayob, A. Neural network approach for estimating state of charge of lithium-ion battery using backtracking search algorithm. *IEEE. Access.* **2018**, *6*, 10069-79. DOI
22. Bhattacharyya, H.S.; Yadav, A.; Choudhury, A.B.; Chanda, C.K. Convolution neural network-based SOC estimation of Li-ion battery in EV applications. In *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*, Dublin, Ireland, August 30-September 01, 2023; IEEE, 2023; pp 587-92. DOI
23. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural. Comput.* **1997**, *9*, 1735-80. DOI PubMed
24. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* 2014, arXiv:1412.3555. DOI
25. Liu, Z.; Jia, W.; Liu, T.; Chang, Y.; Li, J. State of charge estimation of lithium-ion battery based on recurrent neural network. In *2020 Asia Energy and Electrical Engineering Symposium (AEEES)*, IEEE, 2020; pp 742-6. DOI
26. Cheng, M.; Lee, Y.; Liu, M.; Sun, C. State-of-charge estimation with aging effect and correction for lithium□ion battery. *IET. Electr. Syst. Transp.* **2015**, *5*, 70-6. DOI
27. Tian, J.; Xiong, R.; Shen, W. A review on state of health estimation for lithium ion batteries in photovoltaic systems. *eTransportation* **2019**, *2*, 100028. DOI
28. Tian, J.; Chen, C.; Shen, W.; Sun, F.; Xiong, R. Deep learning framework for lithium-ion battery state of charge estimation: recent advances and future perspectives. *Energy. Storage. Mater.* **2023**, *61*, 102883. DOI
29. Ghassani, F.; Abdurohman, M.; Putrada, A.G. Prediction of smarthphone charging using K-nearest neighbor machine learning. In *2018 Third Int Conf Inform Comput ICIC*, Palembang, Indonesia, October 17-18, 2018; IEEE, 2018; pp. 1-4. DOI
30. Hu, C.; Jain, G.; Zhang, P.; Schmidt, C.; Gomadam, P.; Gorka, T. Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery. *Appl. Energy.* **2014**, *129*, 49-55. DOI
31. Jiang, F.; Yang, J.; Cheng, Y.; et al. An aging-aware soc estimation method for lithium-ion batteries using xgboost algorithm. In *2019 IEEE Int Conf Progn Health Manag ICPHM*, San Francisco, USA, June 17-20, 2019; IEEE, 2019; pp. 1-8. DOI
32. Liu, X.; Li, K.; Wu, J.; He, Y.; Liu, X. An extended Kalman filter based data-driven method for state of charge estimation of Li-ion batteries. *J. Energy. Storage.* **2021**, *40*, 102655. DOI
33. Chen, W.; Cai, M.; Tan, X.; Wei, B. Parameter identification and state-of-charge estimation for Li-ion batteries using an improved tree seed algorithm. *IEICE. Trans. Inf. &. Syst.* **2019**, *E102.D*, 1489-97. DOI
34. Wang, Q.; Luo, Y.; Han, X. Research on estimation model of the battery state of charge in a hybrid electric vehicle based on the classification and regression tree. *Math. Comput. Modell. Dyn. Syst.* **2019**, *25*, 376-96. DOI
35. Antón J, García Nieto P, de Cos Juez F, Sánchez Lasheras F, González Vega M, Roqueñí Gutiérrez M. Battery state-of-charge estimator using the SVM technique. *Appl. Math. Modell.* **2013**, *37*, 6244-53. DOI
36. Wei, K.; Wu, J.; Ma, W.; Li, H. State of charge prediction for UAVs based on support vector machine. *J. Eng.* **2019**, *2019*, 9133-6. DOI
37. Li, J.; Ye, M.; Meng, W.; Xu, X.; Jiao, S. A novel state of charge approach of lithium ion battery using least squares support vector machine. *IEEE. Access.* **2020**, *8*, 195398-410. DOI
38. Sheng, H.; Xiao, J. Electric vehicle state of charge estimation: nonlinear correlation and fuzzy support vector machine. *J. Power. Sources.* **2015**, *281*, 131-7. DOI
39. Li, R.; Xu, S.; Li, S.; et al. State of charge prediction algorithm of lithium-ion battery based on PSO-SVR cross validation. *IEEE. Access.* **2020**, *8*, 10234-42. DOI
40. Stighezza, M.; Bianchi, V.; De, M. I. FPGA implementation of an ant colony optimization based SVM algorithm for state of charge estimation in Li-ion batteries. *Energies* **2021**, *14*, 7064. DOI
41. Ipek, E.; Eren, M.K.; Yilmaz, M. State-of-charge estimation of li-ion battery cell using support vector regression and gradient boosting techniques. In *2019 Int Aegean Conf Electr Mach Power Electron ACEMP 2019 Int Conf Optim Electr Electron Equip OPTIM*, Istanbul, Turkey, August 27-29, 2019; IEEE, 2019; pp 604-9. DOI
42. Song, Q.; Wang, S.; Xu, W.; Shao, Y.; Fernandez, C. A novel joint support vector machine - cubature kalman filtering method for adaptive state of charge prediction of lithium-ion batteries. *Int. J. Electrochem. Sci.* **2021**, *16*, 210823. DOI
43. Xie, F.; Wang, S.; Xie, Y.; Fernandezb, C.; Li, X.; Zou, C. A novel battery state of charge estimation based on the joint unscented kalman filter and support vector machine algorithms. *Int. J. Electrochem. Sci.* **2020**, *15*, 7935-53. DOI
44. Hu, J.; Hu, J.; Lin, H.; et al. State-of-charge estimation for battery management system using optimized support vector machine for regression. *J. Power. Sources.* **2014**, *269*, 682-93. DOI
45. Hansen, T.; Wang, C. Support vector based battery state of charge estimator. *J. Power. Sources.* **2005**, *141*, 351-8. DOI
46. Densmore, A.; Hanif, M. Modeling the condition of lithium ion batteries using the extreme learning machine. In *2016 IEEE PES PowerAfrica, Livingstone, Zambia*, June 28-July 03, 2016; IEEE, 2016; pp. 184-8. DOI
47. Jiao, M.; Wang, D.; Yang, Y.; Liu, F. More intelligent and robust estimation of battery state-of-charge with an improved regularized extreme learning machine. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104407. DOI

48. Wang, Z.; Yang, D. State-of-charge estimation of lithium iron phosphate battery using extreme learning machine. *2015. 6th. Int. Conf. Power. Electron. Syst. Appl. PESA*. IEEE 2015; pp 1-5. DOI
49. Chin, C.; Gao, Z. State-of-charge estimation of battery pack under varying ambient temperature using an adaptive sequential extreme learning machine. *Energies* **2018**, *11*, 711. DOI
50. Zhao, X.; Qian, X.; Xuan, D.; Jung, S. State of charge estimation of lithium-ion battery based on multi-input extreme learning machine using online model parameter identification. *J. Energy. Storage.* **2022**, *56*, 105796. DOI
51. Zhang, B.; Ren, G. Li-ion battery state of charge prediction for electric vehicles based on improved regularized extreme learning machine. *WEVJ.* **2023**, *14*, 202. DOI
52. Zhang, C.; Wang, S.; Yu, C.; Xie, Y.; Fernandez, C. Novel improved particle swarm optimization-extreme learning machine algorithm for state of charge estimation of lithium-ion batteries. *Ind. Eng. Chem. Res.* **2022**, *61*, 17209-17. DOI
53. Du, J.; Liu, Z.; Wang, Y. State of charge estimation for Li-ion battery based on model from extreme learning machine. *Control. Eng. Pract.* **2014**, *26*, 11-9. DOI
54. Ozcan, G.; Pajovic, M.; Sahinoglu, Z.; Wang, Y.; Orlik, P.V.; Wada, T. Online state of charge estimation for lithium-ion batteries using Gaussian process regression. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, Italy, October 23-26, 2016; IEEE, 2016; pp 998-1003. DOI
55. Ozcan, G.; Pajovic, M.; Sahinoglu, Z.; Wang, Y.; Orlik, P.V.; Wada, T. Online battery state-of-charge estimation based on sparse gaussian process regression. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*, Boston, USA, July 17-21 2016; IEEE, 2016; pp 1-5. DOI
56. Sahinoglu, G. O.; Pajovic, M.; Sahinoglu, Z.; Wang, Y.; Orlik, P. V.; Wada, T. Battery state-of-charge estimation based on regular/recurrent gaussian process regression. *IEEE. Trans. Ind. Electron.* **2018**, *65*, 4311-21. DOI
57. Chen, X.; Chen, X.; Chen, X. A novel framework for lithium-ion battery state of charge estimation based on Kalman filter Gaussian process regression. *Int. J. Energy. Res.* **2021**, *45*, 13238-49. DOI
58. Lee, K.; Lee, W.; Kim, K. K. Battery state-of-charge estimation using data-driven Gaussian process Kalman filters. *J. Energy. Storage.* **2023**, *72*, 108392. DOI
59. Yi, Y.; Xia, C.; Shi, L.; et al. Lithium-ion battery expansion mechanism and Gaussian process regression based state of charge estimation with expansion characteristics. *Energy* **2024**, *292*, 130541. DOI
60. Deng, Z.; Hu, X.; Lin, X.; Che, Y.; Xu, L.; Guo, W. Data-driven state of charge estimation for lithium-ion battery packs based on Gaussian process regression. *Energy* **2020**, *205*, 118000. DOI
61. Plett, G. L. *Battery management systems: battery modeling*. Artech, 2015. https://ieeexplore.ieee.org/document/9100168 (accessed 2025-02-22).
62. Chemali, E.; Kollmeyer, P. J.; Preindl, M.; Emadi, A. State-of-charge estimation of Li-ion batteries using deep neural networks: a machine learning approach. *J. Power. Sources.* **2018**, *400*, 242-55. DOI
63. How, D. N. T.; Hannan, M. A.; Lipu, M. S. H.; Sahari, K. S. M.; Ker, P. J.; Muttaqi, K. M. State-of-charge estimation of Li-ion battery in electric vehicles: a deep neural network approach. *IEEE. Trans. Ind. Appl.* **2020**, *56*, 5565-74. DOI
64. Jung, G.E; Baek, J.; Liu, J.; et al. The precision SOC estimation method of LiB for EV applications using ANN. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, Republic of, October 20-22, 2021; IEEE, 2021; pp 1052-4. DOI
65. Bian, C.; Yang, S.; Liu, J.; Zio, E. Robust state-of-charge estimation of Li-ion batteries based on multichannel convolutional and bidirectional recurrent neural networks. *Appl. Soft. Comput.* **2022**, *116*, 108401. DOI
66. Bhattacharjee, A.; Verma, A.; Mishra, S.; Saha, T. K. Estimating state of charge for xEV batteries using 1D convolutional neural networks and transfer learning. *IEEE. Trans. Veh. Technol.* **2021**, *70*, 3123-35. DOI
67. Hannan, M. A.; How, D. N. T.; Lipu, M. S. H.; et al. SOC estimation of Li-ion batteries with learning rate-optimized deep fully convolutional network. *IEEE. Trans. Power. Electron.* **2021**, *36*, 7349-53. DOI
68. Li, J.; Jiang, Z.; Jiang, Y.; Song, W.; Gu, J. The state of charge estimation of lithium-Ion battery based on battery capacity. *J. Electrochem. Soc.* **2022**, *169*, 120539. DOI
69. Mohanty, P. K.; Jena, P.; Padhy, N. P. Continuous wavelet transform based CNN model for EV battery state of charge estimation. In *2023. IEEE. 3rd. International. Conference. on. Smart. Technologies. for. Power,. Energy. and. Control. (STPEC), 2023*; 1-5. DOI
70. Kollmeyer, P. Panasonic 18650PF Li-ion battery data. *Mendeley. Data.* **2018**, *3*, 2018. DOI
71. Kollmeyer, P.; Vidal, C.; Naguib, M.; Skells, M. LG 18650HG2 Li-ion battery data and example deep neural network xEV SOC estimator script. *Mendeley. Data.* **2020**, *3*, 2020. DOI
72. Guo, W.; Li, J. A battery SOC prediction method based on GA-CNN network and its implementation on FPGA. *Computer. engineering. and. technology.Singapore*, Springer, 2019; pp 78-90. DOI
73. Mazzi, Y.; Ben, S. H.; Gaga, A.; Errahimi, F. State of charge estimation of an electric vehicle's battery using tiny neural network embedded on small microcontroller units. *Intl. J. of. Energy. Research.* **2022**, *46*, 8102-19. DOI
74. Liu, V.T.; Sun, Y.K.; Lu, H.Y.; Wang, S.K. State of charge estimation for lithium-ion battery using recurrent neural network. In *2018 IEEE International Conference on Advanced Manufacturing (ICAM)*, Yunlin, Taiwan, November 16-18, IEEE, 2018; pp 376-9. DOI
75. Lipu, M.S.H.; Hannan, M.A.; Hussain, A.; Saad, M.H.M.; Ayob, A.; Muttaqi, K.M. Lithium-ion battery state of charge estimation method using optimized deep recurrent neural network algorithm. In *2019 IEEE Industry Applications Society Annual Meeting*, Baltimore, USA, September 29-October 03, 2019; IEEE, 2019; pp 1-9. DOI

76. Wang, Q.; Gu, H.; Ye, M.; Wei, M.; Xu, X. State of charge estimation for lithium-ion battery based on NARX recurrent neural network and moving window method. *IEEE. Access.* **2021**, *9*, 83364-75. DOI
77. Sadykov, M.; Haines, S.; Walker, G.; Holmes, D. W. Feed-forward state of charge estimation of $LiFePO_4$ batteries using time-series machine learning prediction with autoregressive models. *J. Energy. Storage.* **2024**, *100*, 113516. DOI
78. Wei, M.; Ye, M.; Li, J. B.; Wang, Q.; Xu, X. State of charge estimation of lithium-ion batteries using LSTM and NARX neural networks. *IEEE. Access.* **2020**, *8*, 189236-45. DOI
79. Chemali, E.; Kollmeyer, P. J.; Preindl, M.; Ahmed, R.; Emadi, A. Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries. *IEEE. Trans. Ind. Electron.* **2018**, *65*, 6730-9. DOI
80. Ren, X.; Liu, S.; Yu, X.; Dong, X. A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM. *Energy* **2021**, *234*, 121236. DOI
81. Chen, J.; Lu, C.; Chen, C.; Cheng, H.; Xuan, D. An improved gated recurrent unit neural network for state-of-charge estimation of lithium-ion battery. *Appl. Sci.* **2022**, *12*, 2305. DOI
82. Ma, L.; Hu, C.; Cheng, F. State of charge and state of energy estimation for lithium-ion batteries based on a long short-term memory neural network. *J. Energy. Storage.* **2021**, *37*, 102440. DOI
83. Yang, F.; Song, X.; Xu, F.; Tsui, K. State-of-charge estimation of lithium-ion batteries via long short-term memory network. *IEEE. Access.* **2019**, *7*, 53792-9. DOI
84. Hannan, M. A.; How, D. N. T.; Mansor, M. B.; Hossain, L. M. S.; Ker, P.; Muttaqi, K. State-of-charge estimation of Li-ion battery using gated recurrent unit with one-cycle learning rate policy. *IEEE. Trans. on. Ind. Applicat.* **2021**, *57*, 2964-71. DOI
85. Zhang, Z.; Dong, Z.; Lin, H.; et al. An improved bidirectional gated recurrent unit method for accurate state-of-charge estimation. *IEEE. Access.* **2021**, *9*, 11252-63. DOI
86. Chen, L.; Song, Y.; Lopes, A. M.; Bao, X.; Zhang, Z.; Lin, Y. Joint estimation of state of charge and state of energy of lithium-ion batteries based on optimized bidirectional gated recurrent neural network. *IEEE. Trans. Transp. Electrific.* **2024**, *10*, 1605-16. DOI
87. Bian, C.; He, H.; Yang, S. Stacked bidirectional long short-term memory networks for state-of-charge estimation of lithium-ion batteries. *Energy* **2020**, *191*, 116538. DOI
88. Manoharan, A.; Sooriamoorthy, D.; Begam, K.; Aparow, V. R. Electric vehicle battery pack state of charge estimation using parallel artificial neural networks. *J. Energy. Storage.* **2023**, *72*, 108333. DOI
89. Jiang, B.; Tao, S.; Wang, X.; Zhu, J.; Wei, X.; Dai, H. Mechanics-based state of charge estimation for lithium-ion pouch battery using deep learning technique. *Energy* **2023**, *278*, 127890. DOI
90. Jayaraman, R.; Thottungal, R. Accurate state of charge prediction for lithium-ion batteries in electric vehicles using deep learning and dimensionality reduction. *Electr. Eng.* **2024**, *106*, 2175-95. DOI
91. Liu, H.; Liang, F.; Hu, T.; Hong, J.; Ma, H. Multi-scale fusion model based on gated recurrent unit for enhancing prediction accuracy of state-of-charge in battery energy storage systems. *JJ. Mod. Power. Syst. Clean. Energy.* **2024**, *12*, 405-14. DOI
92. Liu, Y.; Shu, X.; Yu, H.; et al. State of charge prediction framework for lithium-ion batteries incorporating long short-term memory network and transfer learning. *J. Energy. Storage.* **2021**, *37*, 102494. DOI
93. Eleftheriadis, P.; Giazitzis, S.; Leva, S.; Ogliari, E. Transfer learning techniques for the lithium-ion battery state of charge estimation. *IEEE. Access.* **2024**, *12*, 993-1004. DOI
94. Lipton, Z. C.; Berkowitz, J.; Elkan, C. A critical review of recurrent neural networks for sequence learning. arXiv 2015, arXiv:1506. 00019. DOI
95. Abbas, G.; Nawaz, M.; Kamran, F. Performance comparison of NARX & RNN-LSTM neural networks for LiFePO4 battery state of charge estimation. In *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Islamabad, Pakistan, January 08-12, 2019; IEEE, 2019; pp 463-8. DOI
96. Li, C.; Xiao, F.; Fan, Y.; Yang, G.; Zhang, W. A recurrent neural network with long short-term memory for state of charge estimation of lithium-ion batteries. In *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, May 24-26, 2019; IEEE, 2019; pp 1712-6. DOI
97. Li, C.; Xiao, F.; Fan, Y. An approach to state of charge estimation of lithium-ion batteries based on recurrent neural networks with gated recurrent unit. *Energies* **2019**, *12*, 1592. DOI
98. Jiao, M.; Wang, D.; Qiu, J. A GRU-RNN based momentum optimized algorithm for SOC estimation. *J. Power. Sources.* **2020**, *459*, 228051. DOI
99. Duan, W.; Song, C.; Peng, S.; Xiao, F.; Shao, Y.; Song, S. An improved gated recurrent unit network model for state-of-charge estimation of lithium-ion battery. *Energies* **2020**, *13*, 6366. DOI
100. Yang, B.; Wang, Y.; Zhan, Y. Lithium battery state-of-charge estimation based on a bayesian optimization bidirectional long short-term memory neural network. *Energies* **2022**, *15*, 4670. DOI
101. Chen, J.; Zhang, Y.; Wu, J.; Cheng, W.; Zhu, Q. SOC estimation for lithium-ion battery using the LSTM-RNN with extended input and constrained output. *Energy* **2023**, *262*, 125375. DOI
102. Yu, H.; Zhang, L.; Wang, W.; et al. State of charge estimation method by using a simplified electrochemical model in deep learning framework for lithium-ion batteries. *Energy* **2023**, *278*, 127846. DOI
103. Wu, X.; Li, M.; Du, J.; Hu, F. SOC prediction method based on battery pack aging and consistency deviation of thermoelectric characteristics. *Energy. Rep.* **2022**, *8*, 2262-72. DOI
104. Liu, D.; Wang, S.; Fan, Y.; Fernandez, C.; Blaabjerg, F. An optimized multi-segment long short-term memory network strategy for

power lithium-ion battery state of charge estimation adaptive wide temperatures. *Energy* **2024**, *304*, 132048. DOI

105. Fawaz H, Forestier G, Weber J, Idoumghar L, Muller P. Deep learning for time series classification: a review. *Data. Min. Knowl. Disc.* **2019**, *33*, 917-63. DOI
106. Kazemi, S. M.; Goel, R.; Eghbali, S.; et al. Time2Vec: learning a vector representation of time. *arXiv1907, arXiv*: 1907.05321. DOI
107. Almaita, E.; Alshkoor, S.; Abdelsalam, E.; Almomani, F. State of charge estimation for a group of lithium-ion batteries using long short-term memory neural network. *J. Energy. Storagee.* **2022**, *52*, 104761. DOI
108. Wang, Y.; Chen, Z.; Zhang, W. Lithium-ion battery state-of-charge estimation for small target sample sets using the improved GRU-based transfer learning. *Energy* **2022**, *244*, 123178. DOI
109. Che, Y.; Zheng, Y.; Wu, Y.; et al. Battery states monitoring for electric vehicles based on transferred multi-task learning. *IEEE. Trans. Veh. Technol.* **2023**, *72*, 10037-47. DOI
110. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41-75. DOI
111. Guo, S.; Ma, L. A comparative study of different deep learning algorithms for lithium-ion batteries on state-of-charge estimation. *Energy* **2023**, *263*, 125872. DOI
112. Pau, D. P.; Aniballi, A. Tiny machine learning battery state-of-charge estimation hardware accelerated. *Appl. Sci.* **2024**, *14*, 6240. DOI
113. Cui, S.; Yong, X.; Kim, S.; Hong, S.; Joe, I. An LSTM-based encoder-decoder model for state-of-charge estimation of lithium-ion batteries. *Intelligent. Algorithms. in. Software. Engineering.Silhavy, R., Eds.; Cham*, Springer International Publishing, 2020; pp 178-88. DOI
114. Bian, C.; He, H.; Yang, S.; Huang, T. State-of-charge sequence estimation of lithium-ion battery based on bidirectional long short-term memory encoder-decoder architecture. *J. Power. Sources.* **2020**, *449*, 227558. DOI
115. Terala, P. K.; Ogundana, A. S.; Foo, S. Y.; Amarasinghe, M. Y.; Zang, H. State of charge estimation of lithium-ion batteries using stacked encoder-decoder Bi-directional LSTM for EV and HEV applications. *Micromachines* **2022**, *13*, 1397. DOI PubMed PMC
116. Ma, L.; Wang, Z.; Yang, F.; et al. Robust state of charge estimation based on a sequence-to-sequence mapping model with process information. *J. Power. Sources.* **2020**, *474*, 228691. DOI
117. Fan, X.; Zhang, W.; Zhang, C.; Chen, A.; An, F. SOC estimation of Li-ion battery using convolutional neural network with U-Net architecture. *Energy* **2022**, *256*, 124612. DOI
118. Ardeshiri, R.R.; Ma, C. State of charge estimation of lithium-ion battery using deep convolutional stacked bidirectional LSTM. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, Kyoto, Japan, June 20-23, 2021; IEEE, 2021; pp 1-6. DOI
119. Shen, L.; Li, J.; Zuo, L.; Zhu, L.; Shen, H. T. Source-free cross-domain state of charge estimation of lithium-ion batteries at different ambient temperatures. *IEEE. Trans. Power. Electron.* **2023**, *38*, 6851-62. DOI
120. Chen, J.; Feng, X.; Jiang, L.; Zhu, Q. State of charge estimation of lithium-ion battery using denoising autoencoder and gated recurrent unit recurrent neural network. *Energy* **2021**, *227*, 120451. DOI
121. Savargaonkar, M.; Oyewole, I.; Chehade, A.; Hussein, A. A. Uncorrelated sparse autoencoder with long short-term memory for state-of-charge estimations in lithium-ion battery cells. *IEEE. Trans. Automat. Sci. Eng.* **2024**, *21*, 15-26. DOI
122. Bakas, S.; Akbari, H.; Sotiras, A.; et al. Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Sci. Data.* **2017**, *4*, 170117. DOI PubMed PMC
123. Severson, K. A.; Attia, P. M.; Jin, N.; et al. Data-driven prediction of battery cycle life before capacity degradation. *Nat. Energy.* **2019**, *4*, 383-91. DOI
124. Diao, W.; Saxena, S.; Pecht, M. Accelerated cycle life testing and capacity degradation modeling of $LiCoO_2$-graphite cells. *J. Power. Sources.* **2019**, *435*, 226830. DOI
125. Hu, L.; Hu, X.; Che, Y.; Feng, F.; Lin, X.; Zhang, Z. Reliable state of charge estimation of battery packs using fuzzy adaptive federated filtering. *Appl. Energy.* **2020**, *262*, 114569. DOI
126. Lv, X.; Cheng, Y.; Ma, S.; Jiang, H. State of health estimation method based on real data of electric vehicles using federated learning. *Int. J. Electrochem. Sci.* **2024**, *19*, 100591. DOI
127. Wong, K. L.; Tse, R.; Tang, S.; Pau, G. Decentralized deep-learning approach for lithium-ion batteries state of health forecasting using federated learning. *IEEE. Trans. Transp. Electrific.* **2024**, *10*, 8199-212. DOI
128. Li, C.; Cui, N.; Wang, C.; Zhang, C. Reduced-order electrochemical model for lithium-ion battery with domain decomposition and polynomial approximation methods. *Energy* **2021**, *221*, 119662. DOI
129. Nascimento, R. G.; Corbetta, M.; Kulkarni, C. S.; Viana, F. A. Hybrid physics-informed neural networks for lithium-ion battery modeling and prognosis. *J. Power. Sources.* **2021**, *513*, 230526. DOI
130. Han, Y.; Huang, G.; Song, S.; Yang, L.; Wang, H.; Wang, Y. Dynamic neural networks: a survey. *IEEE. Trans. Pattern. Anal. Mach. Intell.* **2022**, *44*, 7436-56. DOI