

Enhancing Battery Capacity Estimation Accuracy through the Neural Network Algorithm

Mouncef El Marghichi^{1*}, Abdelilah Hilali², Azeddine Loulijat³

¹ Intelligent Systems Design Laboratory (ISDL), Faculty of Science, Abdelmalek Essaadi University, 93000 Tetouan, Morocco

² Faculty of Sciences, Moulay Ismail University, 11201 Meknes, Morocco

³ Faculty of Sciences and Technology, Hassan first University, 26002 Settat, P.O.B. 577, Morocco

* Corresponding author, e-mail: m.elmarghichi@uae.ac.ma

Received: 14 July 2023, Accepted: 07 March 2024, Published online: 09 May 2024

Abstract

Accurate estimation of battery metrics, such as state of health (SOH), is crucial for effective battery management systems (BMS) due to capacity degradation over time. This paper proposes a methodology to enhance battery capacity estimation accuracy by addressing uncertainties related to state of charge (SOC) estimation and measurement. The methodology employs the Neural Network Algorithm (NNA), an optimization algorithm inspired by artificial neural networks (ANNs). The NNA generates an initial population of pattern solutions and iteratively updates them using a weight matrix, bias operator, and transfer function operator. By combining the advantages of ANNs and optimization techniques, the NNA aims to find an optimal solution considering interdependent variables and incorporating global and local feedbacks. Leveraging the capabilities of the NNA, our objective is to identify the candidate that minimizes a specified cost function, ensuring up-to-date cell capacity through a memory forgetting factor. The algorithm's precision was validated using NASA's Prognostic Data, demonstrating outstanding performance by surpassing two aggressive algorithms in terms of accuracy. In the most severe case scenario, the algorithm achieved a peak error of less than 0.4%. Furthermore, the algorithm consistently demonstrated predictive performance measures that were superior to those of the compared algorithms.

Keywords

Neural Network Algorithm (NNA), battery aging, capacity, lithium-ion, Battery Management Systems

1 Introduction

Lithium batteries are highly valued for their exceptional efficiency, energy density, and extended lifespan, making them an ideal choice for many applications. A well-functioning Battery Management System (BMS) plays a vital role in evaluating the battery's State of Health (SOH), State of Charge (SOC), and Remaining Useful Life (RLU) to ensure the battery's safety and reliability. Real-time and accurate estimation of these parameters is of utmost importance. The accuracy of capacity estimation significantly impacts SOC, SOH, and SOE, as incorrect estimates can lead to critical issues like battery pack failure or inaccurate autonomy estimation. Thus, ensuring efficient capacity estimation is imperative [1].

Various techniques, including coulombic counting, incremental capacity analysis (ICA), differential voltage analysis (DVA), and electrochemical impedance spectroscopy (EIS), can be employed for estimating battery capacity [2–17]. However, implementing these techniques in

online applications presents challenges due to the need for specific charge/discharge profiles to replicate the equilibrium state. Additionally, noise may be introduced during the computation process, particularly when dealing with flat battery curves. Typically, a suitable filter is employed to mitigate this issue, making these techniques more suitable for offline estimation.

To overcome these limitations, it is essential to develop approaches that can effectively utilize sensor data obtained from a battery in operation without disrupting its regular functioning. Leveraging time series data from charge-discharge curves during normal battery operation provides optimal inputs for estimation techniques [18].

We have identified various approaches for estimating battery capacity in online scenarios, which can be classified into two main categories: model-based techniques and data-driven methods. In the model-based category, capacity estimation is accomplished by employing

electrochemical or empirical models. Data-driven techniques, on the other hand, utilize kernel-based methods or neural network (NN) approaches to estimate the capacity.

Model-based techniques involve the utilization of electrochemical or empirical models in conjunction with data to evaluate the battery's capacity. The widely employed models in this context are the electrochemical (EM) model and the equivalent circuit model (ECM). The EM model describes the battery's behavior using partial differential equations [19–21], while the ECM [22–24] represents the battery using electrical components such as capacity, resistance, and voltage source.

Data-based approaches serve as an alternative to model-based techniques, addressing the limitations associated with accurate cell models and the availability of extensive data. These approaches employ machine learning methods to capture the complex relationship between cell capacity and measurements such as voltage and current. Within the data-based category, different methodologies, including kernel methods [25–27] and neural network methods [24, 28, 29], can be utilized to achieve accurate capacity estimation without relying on battery operating theory.

In recent research, a kernel-based approach combined with EMD (empirical mode decomposition) denoiser and MKRVM (multiple kernel relevance vector machine) techniques was employed to estimate battery capacity [25]. The EMD was utilized to preprocess the capacity measurement data and generate noise-free capacity records. The MKRVM model, constructed using the noiseless capacity data, preserved its diversity through a multiple kernel heterogeneous learning method. The sparse weights of the kernel basis functions were obtained through particle swarm optimization.

Another study introduced the use of Gaussian process regression for capacity estimation, relying on voltage measurements during short galvanic operation periods [26]. This approach eliminated the need for interpreting voltage-time records as incremental capacity or differential voltage curves, avoiding noise amplification through differentiation and the requirement for voltage range measurements to cover the peaks of IC/DV curves.

Deep neural networks (DNN), particularly those employing ensemble and transfer learning, have also been employed to assess battery capacity. In a recent study, a DCNN (deep convolutional neural network) was developed using the concepts of ensemble and transfer learning [30]. Transfer learning allowed the transfer of knowledge from a source task to enhance performance in a related but distinct task, while

ensemble-based learning ensured robustness by blending outputs from various learning algorithms. Additionally, a one-dimensional convolutional neural network was utilized in another method, where random segments of charge waveforms were used as inputs for capacity estimation [31]. To enhance the methodology's robustness and accuracy, a linear decreasing particle swarm weighted optimization was applied to optimize the neural network's parameters.

Data-based methods face limitations in terms of the need for abundant and accurate data, as well as the time required for training [1].

Joint estimation techniques have been employed to overcome these limitations and achieve precise capacity estimations. However, these methods heavily depend on the accuracy of the underlying model and may encounter convergence or instability issues due to the interconnected nature of capacity and SOC. Several recent studies have addressed these challenges:

One study [32] proposed a multi-step model fusion algorithm for co-estimating capacity and SOC. The algorithm utilized the covariance and mean of model errors at different levels of degradation to compute weights and establish a fusion model with stable parameters. Additionally, a proportional-integral observer incorporated a forward capacity gain to enhance convergence speed. The resulting fusion method integrated multistep modeling and a proportional-integral-differential observer for SOC and capacitance co-estimation. In another study [33], a second-order equivalent circuit model (ECM) combined with the square root cubature Kalman filter (SRCKF) was employed to estimate SOC, considering capacity degradation and parameter variations. The capacity was determined using a genetic approach. An enhanced adaptive extended Kalman filter (AEKF) was proposed in [34] to achieve co-estimation of SOC and battery capacity. The online open circuit voltage, detected using the forgetting factor recursive least squares (FFRLS), was considered as an observer state for merging SOC and capacity in the filter. Additionally, an OCV-SOC-temperature relationship was developed to improve the AEKF's adaptability to temperature variations. Reference [35] presented a co-estimation method for battery capacity and SOC. A new capacity health indicator, derived from lithium battery charge data, was utilized in combination with a least square support vector machine (LSSVM) for capacity evaluation. The obtained outcomes were then used as inputs for SOC estimation. To address dependency loss in recurrent neural networks (RNN), a moving window technique was employed.

1.1. Paper objective

The main objective of this work is to enhance battery capacity estimation accuracy by addressing measurement noises and errors in SOC prediction. It is crucial to note that our focus is not on proposing a specific SOC estimation algorithm, but rather on introducing an independent methodology for estimating battery capacity that avoids circular dependencies with SOC. Traditional methods like Coulomb counting, which rely on an accurate total capacity estimate for precise SOC estimation, can introduce unstable dependencies when used in conjunction with our approach. Instead, we suggest SOC estimation techniques that solely rely on voltage measurements.

One recommended approach is to use Kalman filter-based methods, particularly sigma-point Kalman filters (SPKF) as mentioned in reference [36]. These methods effectively combine current and voltage data and are less susceptible to errors in capacity estimation, resulting in accurate SOC estimates. Integrating the nominal capacity as a fixed constant within the SPKF further enhances the reliability of SOC estimates.

For capacity estimation, our goal is to employ a recursive approach that directly utilizes estimated SOC and/or measured quantities (voltage/current) while minimizing data volume requirements and avoiding the need for a detailed cell model. Heavy computations are also reduced to minimize memory and computational demands.

In this particular context, we introduce a fresh approach centered around the Neural Network Algorithm (NNA), which draws inspiration from artificial neural networks (ANNs). The NNA operates by generating an initial set of pattern solutions and subsequently refining them iteratively using a weight matrix, bias operator, and transfer function operator. By harnessing the strengths of ANNs and optimization techniques, the NNA endeavors to discover an optimal solution, taking into account the interconnected variables and incorporating both global and local feedback. Utilizing the capabilities of the NNA, our primary goal is to identify the most suitable candidate that minimizes a pre-defined cost function, while ensuring current cell capacity through the utilization of a memory forgetting factor.

This paper makes significant contributions, which can be summarized as follows:

- A novel framework for recursive online battery capacity estimation based on the Neural Network Algorithm (NNA) is introduced. The evaluation was performed using datasets from NASA that included

tests conducted on Samsung (INR 18650) batteries. The study demonstrates the accuracy and robustness of the NNA method, representing one of the initial attempts to estimate battery capacity using this approach.

- The proposed technique addresses various sources of capacity error, including measurement and estimation noise.
- To evaluate the performance of the proposed approach, a comparison is made with other recursive techniques such as Total Least Squares (TLS), and the Approximate Weighted Total Least Squares (AWTLS).

This research paper is structured as follows: Section 2 offers a thorough formulation of the capacity assessment problem. Section 3 introduces the novel NNA-based framework. Section 4 provides in-depth details on the implementation configuration, analysis, findings, and discussion of the NASA aging battery data. Lastly, Section 5 summarizes the key findings and provides concluding remarks.

2 Problem definition

2.1 Problem definition

The expression for coulomb counting is as follows [36, 37]:

$$\text{SOC}(t_2) = \frac{1}{C} \int_{t_1}^{t_2} \eta i(t) dt + \text{SOC}(t_1). \quad (1)$$

In the provided context, the variables used are as follows: SOC represents the State of Charge, C represents the battery capacity, and $i(t)$ represents the battery current. Furthermore, the symbol η denotes the efficiency factor.

By rearranging the terms of Eq. (1), we can observe a linear relationship expressed as $y = C \cdot x$ in Eq. (2). This linear relationship allows us to estimate the battery capacity C using a standard linear regression approach, assuming we have knowledge of the y and x values. However, a challenge arises due to the presence of noise in both the x -axis (SOC difference) and the y -axis (cumulated current). Consequently, Eq. (2) is modified to $(y - \Delta y) = C(x - \Delta x)$ [36, 37].

$$\int_{t_1}^{t_2} \eta i(t) dt = C \times (\text{SOC}(t_2) - \text{SOC}(t_1)) \quad (2)$$

y x

To address the noise in the y and x axes, researchers in [36, 37] have explored the least squares technique for

predicting battery capacity. They propose four algorithms: AWTLs, WTLS, WLS, and TLS. To derive the formulas for these methods, they adopt the loss function expressed as:

$$F_{\text{loss}} = \sum_{j=1}^N \frac{(y_j - C_h x_j)^2}{\sigma_{y_j}^2 + \sigma_{x_j}^2 (C_h)^2} \quad (3)$$

The estimated battery capacity is represented by C_h , with x_j indicating the estimated SOC during the time interval $[t_1 - t_2]$, and y_j representing the accumulated amp-hours over the same period. The variances on the y and x axes are denoted as $(\sigma_{y_j})^2$ and $(\sigma_{x_j})^2$, respectively.

In the study conducted by the authors [36, 37], they developed four recursive techniques to predict the total battery capacity using Eq. (3). Among these techniques, the AWTLs method was identified as the most accurate, yielding favorable results. The methods were derived with certain considerations, such as applying different weightings to x_j and X_j compared to y_j and Y_j (where Y_j and X_j are points on the line $Y_j = C_h X_j$). Moreover, the methods ensured that the line connecting the data points (Y_j, X_j) and (x_j, y_j) was perpendicular to the line $Y_j = C_h X_j$. Further details can be found in the referenced work [36, 37].

In this study, the NNA algorithm is utilized to minimize Eq. (3) without resorting to any approximations. As an optimizer, the algorithm requires the definition of an objective function. In line with previous work [36, 37], we employ the same loss function Eq. (3). The NNA algorithm aims to minimize this function while considering the following constraints: C_{low} and C_{high} , which represent the minimum and maximum capacity values, respectively.

$$C_{\text{low}} < C_h < C_{\text{high}} \quad (4)$$

The subsequent section provides a detailed description of the NNA algorithm.

3 Neural network algorithm NNA

3.1 Basic idea

Artificial neural networks (ANNs) are computational models that draw inspiration from the structure and functional aspects of biological neural networks. ANNs consist of interconnected computing units, known as artificial neurons, which mimic the behavior of biological nervous systems [38]. The network's overall function is largely determined by the connections between these units.

Based on their connectivity pattern or architecture, ANNs can be categorized into two main types: feedforward

neural networks and recurrent networks [38]. Feedforward networks have a non-looping architecture, meaning they produce a single set of output values for a given input dataset. They are often considered "static" networks. On the other hand, recurrent networks incorporate loops through feedback connections, introducing a time parameter to the model. This makes them "dynamic" neural networks.

Recurrent networks utilize two types of feedback connections: local feedbacks and global feedbacks. Local feedbacks refer to links that transmit the output of a neuron back to itself, while global feedbacks involve links that pass the output of a neuron to other neurons within the same or lower layers of the multilayer network architecture. Fig. 1 illustrates typical architectures for both feedforward and recurrent neural networks. Additional details about ANNs are provided at each relevant step of the suggested optimization method.

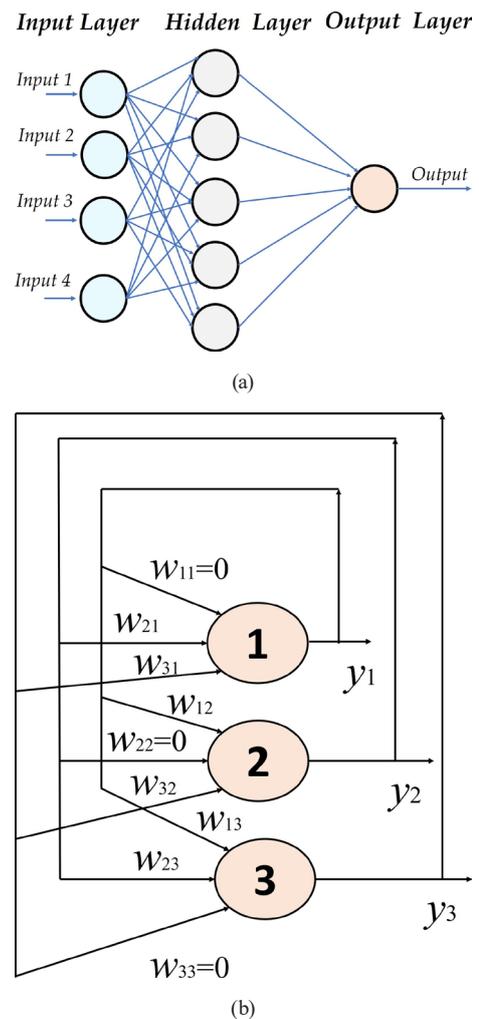


Fig. 1 ANNs with: (a) feed forward neural network, (b) recurrent neural networks

3.2 NNA method

The NNA, similar to other metaheuristic optimization algorithms, begins with an initial population of pattern solutions. While ANNs are commonly used for prediction, in the NNA, the best solution obtained at each iteration is treated as the target data, aiming to minimize the error among the target and predicted pattern solutions. Inspired by ANNs, the NNA seeks to find an optimal solution through iterative updates. NNA is a new optimization algorithm based on the neural network structure [38], incorporating specific mathematical formulations and concepts. Detailed descriptions and processes of the NNA can be found in the subsequent subsections.

3.2.1 Generating initial population

To address an optimization problem, it is often required to represent the decision variables as an array. Before delving into the processes of the Neural Network Algorithm (NNA), it is essential to introduce the key terminology associated with this algorithm. Every agent, which comprises a set of values for each optimization variable, is referred to as a "pattern solution" (e.g., in the Genetic Algorithm, this array is known as a "Chromosome"). For a D -dimensional optimization problem, a pattern solution is represented by a $1 \times D$ array that serves as input data in the NNA. The array is formulated as:

$$\text{Pattern solution} = [x_1, x_2, \dots, x_D]. \quad (5)$$

The population of pattern solutions in the optimization algorithm is analogous to the input data in Artificial Neural Networks (ANNs). To initiate the optimization process, a candidate matrix of pattern solutions with dimensions $N_{\text{pop}} \times D$ is generated. This matrix, denoted as X , is randomly generated within the lower and upper bounds defined by the decision maker for the problem. The matrix X can be expressed as follows, with the rows representing the population size (N_{pop}) and the columns representing the dimension size (D):

Population of
 pattern solutions $X = \begin{bmatrix} x_1^1 & \cdots & x_D^1 \\ \vdots & \ddots & \vdots \\ x_1^{N_{\text{pop}}} & \cdots & x_D^{N_{\text{pop}}} \end{bmatrix}. \quad (6)$

Each decision variable value (x_1, \dots, x_D) can be expressed as a floating-point number (i.e., a real value) or can be defined within a set of discrete variables. The cost (or fitness) of a pattern solution is determined by evaluating the

cost function (or fitness function), denoted as CF , at the corresponding pattern solution. The cost function is calculated as follows:

$$CF_i = f(x_1^i, x_2^i, \dots, x_D^i), \quad i = 1, 2, \dots, N_{\text{pop}}. \quad (7)$$

In the objective function, denoted as f , the notations with a vector sign represent vector values (arrays), while the remaining notations and parameters are considered scalar values. After computing the cost function (fitness function) for all pattern solutions, the best pattern solution is determined as the target solution which is defined as the candidate solution with the minimum objective function value.

The Neural Network Algorithm (NNA) shares similarities with ANNs, where N_{pop} input data is associated with D dimension(s) and a single target data or response (refer to Fig. 1 (a)). Once the target solution (X^{Target}) is established among the other pattern solutions, the corresponding weight (W^{Target}), which belongs to the population of weights (weight matrix), needs to be selected.

3.2.2 Weight matrix

In Artificial Neural Networks (ANNs), the artificial neurons or processing units can have multiple input paths analogous to dendrites. These units combine the weighted values of these input paths through a simple summation, resulting in an internal activity level for the unit [38]. The output path of a unit may be connected to the input path of other units through connection weights, representing the synaptic strength of biological neural connections. Each connection is assigned a weight (w) (Fig. 1 (b)), which modifies or weights the signals on the input lines before they are summed [38].

In ANNs, initial weights are typically assigned random numbers, and as the iteration progresses, these weights are updated based on the calculated network error. In the case of the Neural Network Algorithm (NNA), initial weights are defined according to Eq. (8):

$$W(t) = [W_1, \dots, W_{N_{\text{pop}}}] = \begin{bmatrix} w_1^1 & \cdots & w_1^{N_{\text{pop}}} \\ \vdots & \ddots & \vdots \\ w_{N_{\text{pop}}}^1 & \cdots & w_{N_{\text{pop}}}^{N_{\text{pop}}} \end{bmatrix} \quad (8)$$

$$= \begin{bmatrix} w_{11} & \cdots & w_{N_{\text{pop}}1} \\ \vdots & \ddots & \vdots \\ w_{1N_{\text{pop}}} & \cdots & w_{N_{\text{pop}}N_{\text{pop}}} \end{bmatrix}.$$

The weight matrix W , with dimensions $N_{\text{pop}} \times N_{\text{pop}}$, is responsible for generating random numbers uniformly

distributed between zero and one during iterations, denoted by t . The first subscript of the weight indicates its association with a specific pattern solution (e.g., $w_{2 \times}$ corresponds to the second pattern solution), while the second subscript is shared among the other pattern solutions (e.g., w_{23} is shared with the third pattern solution). Each pattern solution has its corresponding weight value, which contributes to generating a new candidate solution.

However, there is a constraint imposed on the weight values. This constraint ensures that the sum of weights for a particular pattern solution does not exceed one. Mathematically, this constraint can be defined as follows:

$$\sum_{j=1}^{N_{pop}} w_{ij}(t) = 1, \quad i = 1, 2, \dots, N_{pop}, \quad (9)$$

$$w_{ij} \in U(0,1), \quad i, j = 1, 2, \dots, N_{pop}. \quad (10)$$

The weight values in the Neural Network Algorithm (NNA) are generated as uniformly distributed random numbers between zero and one, as shown in Eq. (10). However, there is a constraint that the sum of weight values for each pattern solution should not exceed one, as expressed in Eq. (9). This constraint is necessary to control the bias of movement and the generation of new pattern solutions or individuals. Without this constraint, the weight values tend to grow disproportionately in a specific direction, leading the algorithm to get trapped in a local optimum point. The constraint plays a similar role to the pheromone parameter in ant colony optimization (ACO), where excessive pheromones on a route attract more ants. By enforcing this constraint, the NNA ensures that its agents have controlled movement with a mild bias, ranging from zero to one.

Once the weight matrix (W) is formed, new pattern solutions (X_{New}) are calculated using an equation inspired by the weight summation technique employed in ANNs.

$$X_j^{New}(t+1) = \sum_{i=1}^{N_{pop}} w_{ij}(t) \times X_i(t), \quad j = 1, 2, \dots, N_{pop} \quad (11)$$

$$X_i(t+1) = X_i^{New}(t+1) + X_i, \quad i = 1, 2, \dots, N_{pop} \quad (12)$$

In the NNA, where t represents the iteration index, the new pattern solution is updated for iteration $t + 1$ using Eq. (11) and Eq. (12). To illustrate this process, consider an example with six pattern solutions (equivalent to six neurons or a population size of 6). The update of the first new pattern solution can be computed as follows:

$$\begin{aligned} X_1^{New}(t+1) &= w_{11}(t)X_1(t) + w_{21}(t)X_2(t) \\ &+ w_{31}(t)X_3(t) + w_{41}(t)X_4(t) \\ &+ w_{51}(t)X_5(t) + w_{61}(t)X_6(t). \end{aligned} \quad (13)$$

Additionally, for enhanced elucidation, Fig. 2 illustrates the process by which the NNA generates its fresh set of pattern solutions for D dimension(s). Once the new pattern solutions have been derived from the existing population of patterns, the weight matrix should be adjusted to account for the optimal weight value known as the "target weight". The ensuing equations propose a method for updating the weight matrix.

$$\begin{aligned} W_i^{Updated}(t+1) &= W_i(t) + 2 \times rand \\ &\times (W^{Target}(t) - W_i(t)), \quad i = 1, 2, \dots, N_{pop} \end{aligned} \quad (14)$$

During the optimization process, it is crucial to ensure that the weight matrix consistently adheres to the constraints Eq. (9) and Eq. (10).

3.2.3 Bias operator

The dynamics of the neural networks model are significantly influenced by the bias current. This current is intricately linked to a surrounding condition, such as noise, in order to ensure that the output of each neuron aligns with the surrounding condition [38]. In the NNA, the bias operator introduces noise by modifying a certain percentage of the pattern solutions in the new population of pattern solutions ($X_i^{New}(t+1)$) and the updated weight matrix ($W_i^{New}(t+1)$). Essentially, the bias operator in the NNA functions as a means of exploring the search space, similar to the mutation operator in the GA.

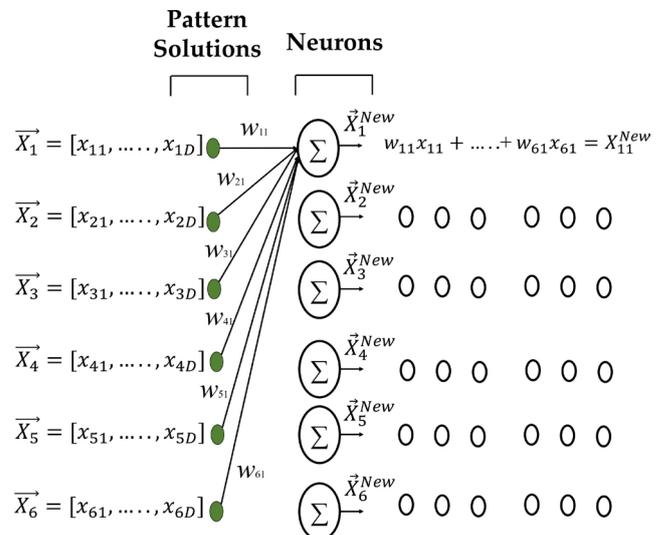


Fig. 2 Schematic view of generating new pattern solutions

In essence, the bias operator prevents premature convergence of the algorithm, particularly in the early iterations, by modifying several individuals within the population. It introduces noise to the new pattern solutions (Eq. (11)) and the updated weight matrix (Eq. (14)). To achieve this, the provided Pseudo code in Algorithm 1 has been employed for the new pattern solutions and updated weight matrix.

Algorithm 1 displays the lower bound (LB) and upper bound (UB) of a given problem. It also showcases the modification factor (β), which determines the percentage of pattern solutions that need to be altered. Initially, β is set to 1, indicating a 100% chance of modifying all individuals in the population. However, its value is adaptively reduced at each iteration using a specific reduction formulation, as suggested below:

$$\beta(t+1) = \beta(t) \times 0.99, \quad t = 1, 2, \dots, \text{Max}_{\text{iteration}}, \quad (15)$$

$$\beta(t+1) = 1 - \frac{t}{\max_{\text{iteration}}}, \quad t = 1, 2, \dots, \text{Max}_{\text{iteration}}. \quad (16)$$

To achieve this objective, either Eq. (15) or Eq. (16), or any other reduction equation, can be employed. The bias operator is adaptively decreased to facilitate the algorithm in searching for an optimal solution closer to the target solution. Additionally, this reduction in β helps prevent drastic alterations in the pattern solutions during the final iterations.

3.2.4 Transfer function operator

In the NNA, the transfer function operator is responsible for updating and improving the new pattern solutions by moving them closer to the target solution. The transfer function operator, defined by Eq. (17), utilizes a constant value of two to search both before and after the target solution. This allows for exploration on both sides of the target solution. The collaboration between the bias and transfer function operators in the NNA is detailed in Algorithm 2, where the

Algorithm 1 Pseudocode for the generating the new pattern solutions

```

For i=1 to Npop
  If rand ≤ β
    % -- Bias for New Pattern Solution---%
    Nb= Round (D × β)
    For j=1 to Nb
      XInput (i,Integer rand [0,D])=LB+(UB-LB) × rand.
    End For
    % -- Bias for Updated Weight Matrix---%
    Nwb= Round (Npop × β)
    For j=1 to Nwb
      XInput (j,Integer rand [0, Npop])=U(0,1).
    End For
  End If
End For

```

Algorithm 2 Pseudocode for Bias and TF combination

```

For i=1 to Npop
  If rand ≤ β
    % -- Bias Operator---%
    Bias Operator (refer to subsection 3.2.3)
  Else (rand > β)
    % -- Transfer Function (TF) Operator---%
    Apply Eq. (17)
  End If
End For

```

bias operator is more influential in generating new pattern solutions during early iterations, while the transfer function operator plays a significant role in later iterations.

$$\begin{aligned}
 X_i^*(t+1) &= \text{TF}(X_i(t+1)) = X_i(t+1) \\
 &+ 2 \times \text{rand} \times X^{\text{Target}}(t) \\
 &- X_i(t+1), \quad i = 1, 2, \dots, N_{\text{pop}}
 \end{aligned} \quad (17)$$

Utilizing Eq. (13), the new pattern solution, $X_1(t+1)$ is transferred from its current position to an updated position, $X_i^*(t+1)$. The collaboration between the bias and transfer function (TF) operators in the NNA is elaborated in detail in Algorithm 2.

Algorithm 2 highlights that during early iterations, the bias operator has a greater likelihood of generating new pattern solutions and discovering unvisited patterns, along with new weight values. However, as the iteration number increases, this probability diminishes, and the TF operator assumes a more significant role in the NNA, particularly during the final iterations.

3.2.5 Steps of the NNA

The NNA is inspired by ANNs, incorporating their advantages. Fig. 3 provides an overview of the NNA's procedures, while Fig. 4 illustrates its schematic representation. The NNA employs weight matrix, bias, and transfer

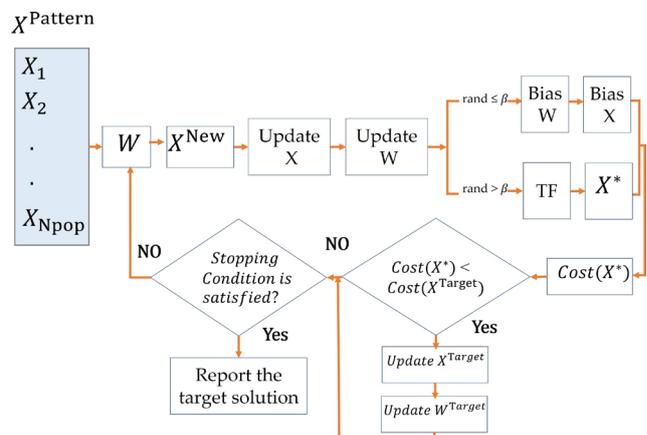


Fig. 3 NNA process

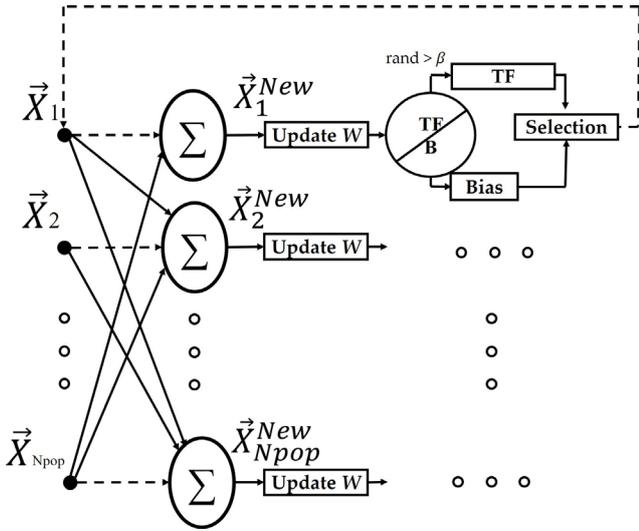


Fig. 4 Schematic depiction of NNA performance

function operators denoted as "W", "Bias", and "TF" respectively. It features both self-feedback and feedback to other neurons, depicted by dashed and solid lines in Fig. 4. The weight matrix is modified iteratively during the 4-optimization process. With interdependent current and next values of design variables, the NNA is categorized as a dynamic optimization model.

$$X_i(t + \Delta t) = f(X_i(t), P(t)), \quad i = 1, 2, \dots, N_{pop} \quad (18)$$

Eq. (18) represents the general trend of the NNA as a dynamic optimization model, where $X_i(t + \Delta t)$ and $X_i(t)$ denote the next and current locations of the i -th pattern solution, respectively. The NNA utilizes concepts and strategies from ANNs, making it an associated memory-based algorithm that incorporates global and local feedbacks. The summarized steps of the proposed method are as follow.

In the proposed algorithm, the first step involves determining the population size and maximum number of iterations. Subsequently, an initial population of pattern solutions is generated randomly within a specified range. The cost of these initial solutions is then calculated. Following this, a weight matrix is generated randomly between zero and one, adhering to the given constraint. A target solution and its corresponding weight are set as the minimum value for minimization problems. New pattern solutions are generated and updated using specific equations. The weight matrix is also updated based on the applied constraints. The algorithm checks the bias condition, performing the bias operator if the condition is met, or applying the transfer function operator if the condition is not met. The objective function value is calculated for all updated pattern solutions, and the target solution and

its weight are updated accordingly. The value of a variable is updated using a reduction formulation. The algorithm checks a predefined stopping condition, and if satisfied, the process ends. Otherwise, it returns to Step 6 and repeats until the stopping criterion is met.

3.2.6 Proposed algorithm

Fig. 5 illustrates the proposed algorithm to estimate battery capacity with two main phases: data pre-processing and estimation.

In the first stage, data pre-processing focuses on eliminating unreliable measurements and preparing the data for further analysis. Corrupted, incomplete, or overlapping data points are removed, and the remaining data is evenly spaced. SOC estimation, although not the primary focus of this study, is performed using an independent approach that avoids circular dependencies with SOC. Specifically, SOC estimation techniques based solely on voltage measurements, such as sigma-point Kalman filters (SPKF), are utilized. SPKF effectively integrates current and voltage data to minimize capacity estimation errors and achieve accurate SOC estimation. Other approaches for SOC prediction can also be explored (refer to [39, 40] for additional information). The result of this stage is the generation of the y and x vectors.

The second stage involves the utilization of the NNA algorithm to estimate the total battery capacity. The NNA algorithm is configured with appropriate parameters. For each

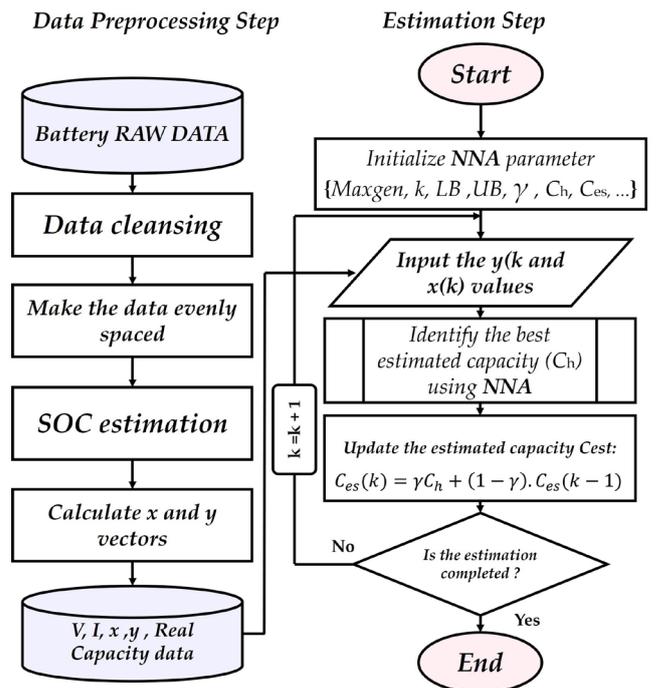


Fig. 5 The proposed framework to estimate battery capacity

new x (Δ SOC) and y value, the NNA algorithm minimizes the loss function Eq. (3) to identify the optimal candidate (C_h). The estimated capacity (C_{es}) is computed by incorporating a forgetting factor (γ) Eq. (19), typically ranging from 0.9 to 1 [41–44], which facilitates rapid convergence of the NNA algorithm towards the true capacity value.

$$C_{es}(k) = \gamma C_h + (1 - \gamma) \times C_{es}(k - 1) \quad (19)$$

4 Results and discussion

In Section 4, we conduct a comparative analysis between the Neural Network Algorithm (NNA) and two least squares approaches, namely Total Least Squares (TLS) and Adaptive Weighted Total Least Squares (AWTLS), as discussed in references [36, 37]. The specific parameters employed in the NNA are outlined in Table 1, encompassing the boundaries for battery parameters such as the lower bound capacity (C_{low}) and the higher bound capacity (C_{high}), along with the forgetting factor γ .

The parameter "population size" (N_p) governs the number of individual solutions or eagles within each generation. A larger population size can promote exploration, but it may also lead to higher computational costs. The "Max_Iteration" parameter defines the maximum number of iterations or generations that the algorithm will execute before termination. γ denotes the forgetting factor employed in the algorithm.

4.1 NASA Battery aging dataset

For performance testing, the NASA PCoE dataset was used [45]. Intensified life tests were conducted on the SAMSUNG INR 18,650 lithium-ion cells. Four batteries (No.18, No.7, No.6, and No.5) with a 2 Ah capacity underwent charging, discharging, and impedance measurement. Charging involved CC (Constant Current) and CV (Constant Voltage) modes, while discharging was performed at specific voltage levels. The tests ended when the batteries lost 30% of their initial capacity. Fig. 6 illustrates the capacity deterioration.

Table 1 NNA algorithm parameters, battery bounds and the forgetting factor

Battery	Population number (N_p)	Number of iterations <i>Max_Iteration</i>	γ	C (Ah)	
				C_{low}	C_{high}
#B05	30	2	0.99	0	2.5
#B06	30	2	0.99	0	2.5
#B07	30	2	0.99	0	2.5
#B18	30	2	0.99	0	2.5

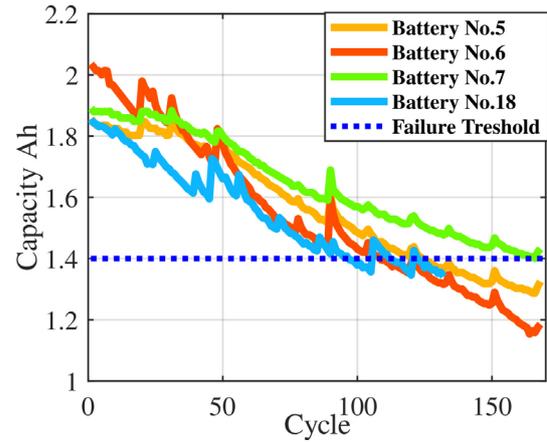


Fig. 6 Trajectories degeneration of battery capacity

4.2 Discussion

We have employed both the least squares and NNA techniques to predict the capacity of previously reported batteries. The results are presented in Fig. 7, which displays the estimated capacity obtained from both algorithms, along with the true capacity. To emphasize the differences between the methods, we have included zoomed-in sections. Notably, the NNA approach exhibits slightly better accuracy compared to the least squares method (TLS, and AWTLs).

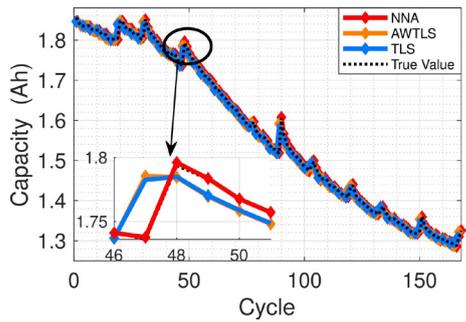
Fig. 8 presents the APE (absolute percentage error) values Eq. (20), demonstrating that NNA outperforms the least squares approaches in terms of error. NNA achieved a maximum error of only 0.39% (B.07), while the AWTLs algorithms had an APE that exceeds 0.9% in most cases. All methods yielded a low mean error that did not exceed 1%.

$$APE(\%) = 100 \times \left| \frac{C_{es}(j) - C_{tr}(j)}{C_{tr}(j)} \right| \quad (20)$$

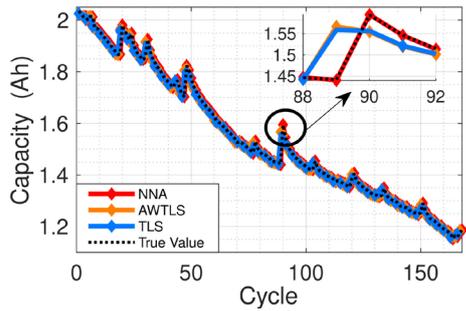
Table 2 summarizes the max and mean error values. By examining the values and understanding their implications, we can understand why NNA is superior to AWTLs and TLS.

Examining Battery #B05, the NNA algorithm demonstrates impressive accuracy with a maximum error of 0.279391% and a mean error of 0.083373%. In contrast, both AWTLs and TLS algorithms exhibit higher errors, with AWTLs recording a maximum error of 0.899295% and a mean error of 0.554360%, and TLS showing a maximum error of 1.171136% and a mean error of 0.554362%. These results clearly indicate that the NNA algorithm provides superior estimation accuracy compared to the other methods.

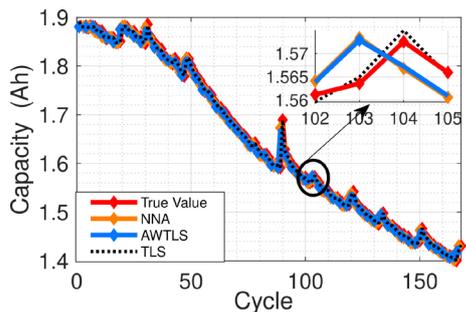
Moving on to Battery #B06, we observe a similar trend. The NNA algorithm achieves a maximum error of



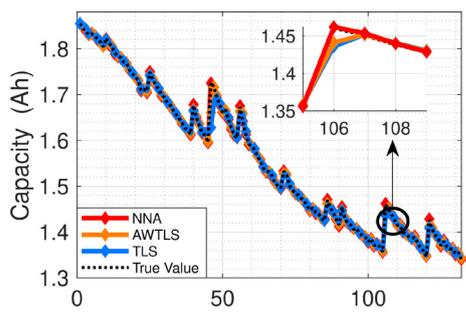
(a)



(b)



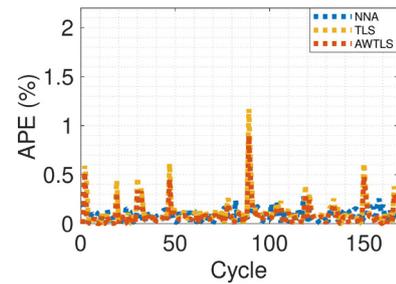
(c)



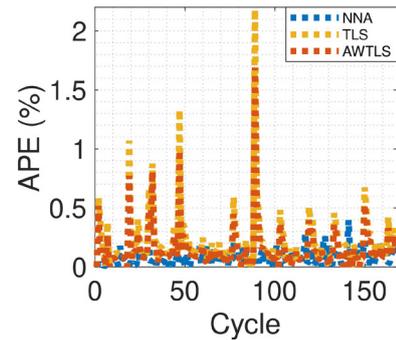
(d)

Fig. 7 The estimated Capacity: (a) B05, (b) B06, (c) B07, (d) B18

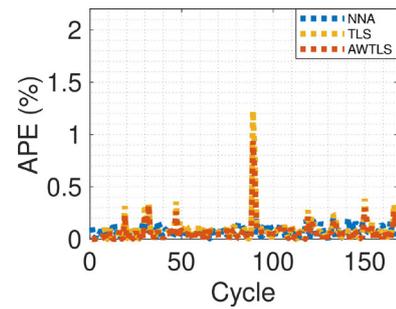
0.396103% and a mean error of 0.084259%, showcasing its precision in estimating battery capacity. Conversely, AWTLS and TLS algorithms display higher errors, with AWTLS recording a maximum error of 1.689592% and a mean error of 0.938652%, while TLS exhibits a maximum error of 2.182708% and a mean error of 0.938669%.



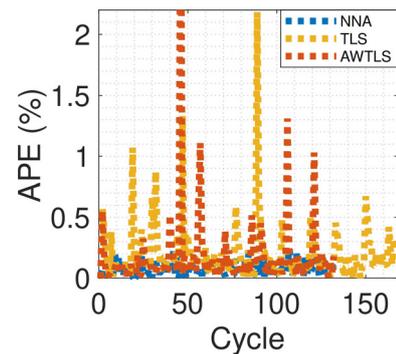
(a)



(b)



(c)



(d)

Fig. 8 The APE (%): (a) B05, (b) B06, (c) B07, (d) B18

Once again, the NNA algorithm outperforms its counterparts in terms of accuracy.

Battery #B07 further supports the superior performance of the NNA algorithm. It achieves a maximum error of 0.216433% and a mean error of 0.073802%, indicating its ability to estimate battery capacity with exceptional

Table 2 Absolute max and mean error for the algorithms in percent

Battery	Methods	Max error (%)	Mean error (%)
#B05	NNA	0.279391	0.083373
	AWTLS	0.899295	0.554360
	TLS	1.171136	0.554362
#B06	NNA	0.396103	0.084259
	AWTLS	1.689592	0.938652
	TLS	2.182708	0.938669
#B07	NNA	0.216433	0.073802
	AWTLS	0.938234	0.450062
	TLS	1.224801	0.450061
#B18	NNA	0.247501	0.080718
	AWTLS	3.822576	0.080718
	TLS	2.182708	0.196232

precision. AWTLS and TLS algorithms, on the other hand, demonstrate higher errors with AWTLS recording a maximum error of 0.938234% and a mean error of 0.450062%, and TLS showing a maximum error of 1.224801% and a mean error of 0.450061%. Once again, the NNA algorithm proves to be the most accurate in estimating battery capacity.

Lastly, analyzing Battery #B18, we find that the NNA algorithm maintains its consistent performance with a maximum error of 0.247501% and a mean error of 0.080718%. In contrast, AWTLS exhibits a significantly higher maximum error of 3.822576% and the same mean error of 0.080718%, highlighting its lower accuracy in estimating the capacity of this particular battery. TLS, with a maximum error of 2.182708% and a mean error of 0.196232%, also shows less precision compared to the NNA algorithm.

In conclusion, the comparison of the algorithms' performance in estimating battery capacity demonstrates that the NNA algorithm consistently outperforms both AWTLS and TLS. It achieves lower maximum and mean errors across all batteries, indicating higher accuracy and precision. These results highlight the NNA algorithm's effectiveness in estimating battery capacity and its potential for reliable applications in the field.

To assess the efficacy of the algorithms and determine their predictive performance, we analyze three key variables: mean squared error (MSE), mean absolute percentage error (MAPE), and root mean square error (RMSE). These metrics allow us to evaluate the accuracy and reliability of the algorithms' predictions. The three metrics are described as:

$$MSE = \frac{1}{p} \sum_{i=1}^p (C_{es}(i) - C_{tr}(i))^2, \quad (21)$$

$$MAE = \frac{100}{\pi} \sum_{i=1}^p \left| \frac{C_{es}(j) - C_{tr}(j)}{C_{tr}(j)} \right|, \quad (22)$$

$$RMSE = \sqrt{\frac{1}{p} \sum_{i=1}^p (C_{es}(i) - C_{tr}(i))^2}, \quad (23)$$

where p is the number of points, C_{tr} is the measured capacity, and C_{es} is the estimated value.

Comparing the numerical results from Table 3, the performance of different algorithms for battery capacity estimation can be assessed. Analyzing the values of RMSE, MAPE, and MSE allows us to evaluate the accuracy and reliability of each method.

In terms of RMSE, NNA consistently outperforms AWTLS and TLS across all batteries. For instance, NNA achieves RMSE values of 1.5693 mAh (Battery #B05), 1.5293 mAh (Battery #B06), 1.4423 mAh (Battery #B07), and 1.5166 mAh (Battery #B18). In contrast, AWTLS and TLS exhibit higher RMSE values, such as 2.2288 mAh and 2.9231 mAh (Battery #B05) for AWTLS and TLS, respectively. The consistently lower RMSE values attained by NNA indicate that its predictions are closer to the true capacity values compared to the other algorithms (Fig. 9).

When considering MAPE, NNA again demonstrates superior performance compared to AWTLS and TLS for most batteries. For example, NNA achieves MAPE values of 0.0834 (Battery #B05), 0.0843 (Battery #B06), 0.0738 (Battery #B07), and 0.0807 (Battery #B18). In contrast, AWTLS and TLS yield higher MAPE values, such as 0.0846 and 0.1129 (Battery #B05) for AWTLS and TLS, respectively. The lower MAPE values obtained by NNA indicate that its predictions have a smaller average

Table 3 Predictive performance indicators

Battery	Methods	RMSE (mAh)	MAPE	Mean squared error (MSE)
#B05	NNA	1.5693	0.0834	2.4628e-06
	AWTLS	2.2288	0.0846	1.8142e-04
	TLS	2.9231	0.1129	1.8143e-04
#B06	NNA	1.5293	0.0843	2.3389e-06
	AWTLS	3.9284	0.1473	5.7219e-04
	TLS	5.1434	0.1962	5.7218e-04
#B07	NNA	1.4423	0.0738	2.0803e-06
	AWTLS	1.9552	0.0649	1.5878e-04
	TLS	2.5534	0.0860	1.5879e-04
#B18	NNA	1.5166	0.0807	2.3000e-06
	AWTLS	6.8082	0.1726	2.3000e-06
	TLS	10.3380	0.2546	0.0102

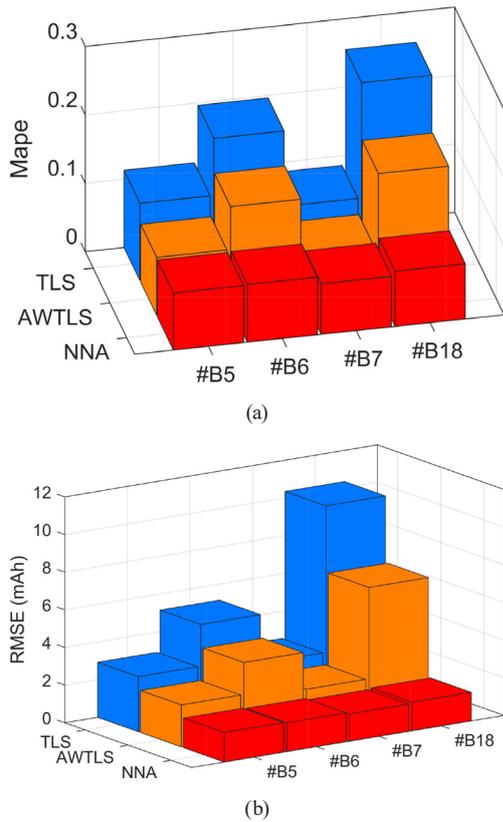


Fig. 9 The predictive performance: (a) MAPE, (b) RMSE

percentage error, signifying its improved accuracy in estimating battery capacity (Fig. 9).

Analyzing the MSE values further supports the superior performance of NNA. With the exception of Battery #B18, where it is tied with AWTLs, NNA achieves the lowest MSE values for all batteries. For instance, NNA yields MSE values of 2.4628×10^{-6} (Battery #B05), 2.3389×10^{-6} (Battery #B06), and 2.0803×10^{-6} (Battery #B07). Conversely, AWTLs and TLS produce higher MSE values, such as 1.8142×10^{-4} (Battery #B05) for AWTLs and 1.8143×10^{-4} (Battery #B05) for TLS. The lower MSE values obtained by NNA indicate its ability to minimize the squared differences between predicted and true values, resulting in improved predictive performance.

Fig. 10 showcases boxplots illustrating the absolute error in battery capacity for each algorithm across all tests. The red horizontal line represents the mean error for each approach. The box height, known as the Interquartile Range (IQR), depicts the data's variability and spread, while outliers lie beyond the whiskers. These boxplots offer valuable insights into the performance and accuracy of the methods.

Remarkably, the NNA algorithm exhibits the lowest spread, with a significantly smaller IQR compared to other

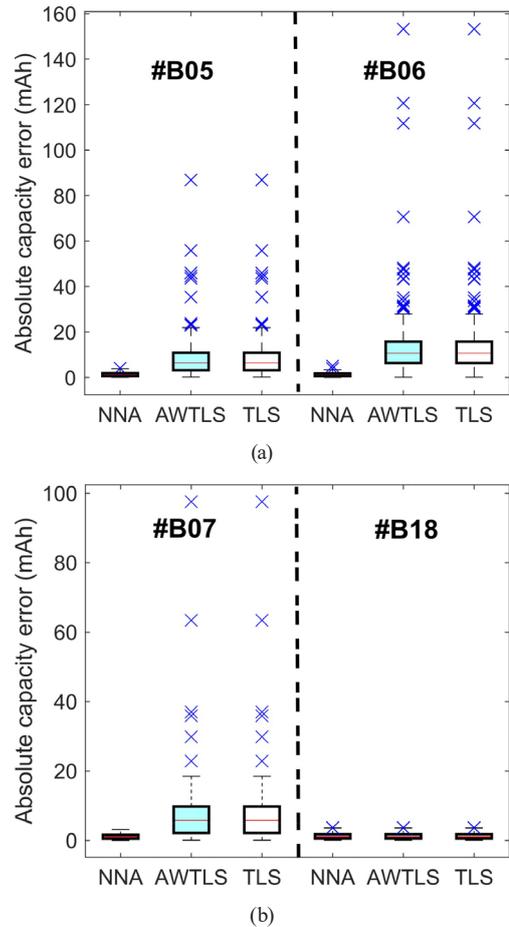


Fig. 10 Boxplots: (a) B05 and B06, (a) B07 and B18

methods. Conversely, the compared techniques demonstrate wider spreads, characterized by larger IQRs and more outliers, indicating lower accuracy relative to the NNA algorithm.

Furthermore, the absence of outliers in all tests for all algorithms confirms their high accuracy in estimating battery capacity. The NNA algorithm, specifically, exhibits a relatively small difference between the 25th and 75th percentiles, as supported by Tables 2 and 3, highlighting its precision and further substantiating its capability for accurate update.

In summary, based on the numerical comparisons, NNA outperforms AWTLs and TLS in terms of RMSE, MAPE, and MSE. The consistently lower values obtained by NNA highlight its superior accuracy and reliability in estimating battery capacity. By leveraging neural networks to capture complex relationships, NNA surpasses the traditional statistical approaches employed by AWTLs and TLS, providing more accurate predictions and enhanced performance in battery capacity estimation.

5 Conclusion

In conclusion, this manuscript presents a novel technique aimed at enhancing the accuracy of battery capacity estimation by minimizing uncertainties associated with SOC estimation and measurement. The proposed methodology utilizes the Neural Network Algorithm (NNA), which draws inspiration from artificial neural networks (ANNs) and employs an optimization approach. Through the iterative updating of pattern solutions using a weight matrix, bias operator, and transfer function operator, the NNA effectively considers interdependent variables and incorporates global and local feedbacks, thus seeking an optimal solution. By leveraging the advantages of ANNs and optimization techniques, the NNA facilitates the identification of a candidate that minimizes a designated cost function, ensuring the up-to-date cell capacity through a memory forgetting factor.

References

- [1] Mouncef, E. M., Mostafa, B. "Battery total capacity estimation based on the sunflower algorithm", *Journal of Energy Storage*, 48, 103900, 2022.
<https://doi.org/10.1016/J.EST.2021.103900>
- [2] Dubarry, M., Svoboda, V., Hwu, R., Liaw, B. Y. "Incremental Capacity Analysis and Close-to-Equilibrium OCV Measurements to Quantify Capacity Fade in Commercial Rechargeable Lithium Batteries", *Electrochemical and Solid-State Letters*, 9(10), A454, 2006.
<https://doi.org/10.1149/1.2221767>
- [3] Anseán, D., García, V. M., González, M., Blanco-Viejo, C., Viera, J. C., Fernández Pulido, Y., Sánchez, L. "Lithium-Ion Battery Degradation Indicators Via Incremental Capacity Analysis", *IEEE Transactions on Industry Applications*, 55(3), pp. 2992–3002, 2019.
<https://doi.org/10.1109/TIA.2019.2891213>
- [4] Pastor-Fernández, C., Uddin, K., Chouchelamane, G. H., Widanage, W. D., Marco, J. "A Comparison between Electrochemical Impedance Spectroscopy and Incremental Capacity-Differential Voltage as Li-ion Diagnostic Techniques to Identify and Quantify the Effects of Degradation Modes within Battery Management Systems", *Journal of Power Sources*, 360, pp. 301–318, 2017.
<https://doi.org/10.1016/J.JPOWSOUR.2017.03.042>
- [5] Li, Y., Abdel-Monem, M., Gopalakrishnan, R., Berecibar, M., Nanini-Maury, E., Omar, N., van den Bossche, P., Van Mierlo, J. "A quick on-line state of health estimation method for Li-ion battery with incremental capacity curves processed by Gaussian filter", *Journal of Power Sources*, 373, pp. 40–53, 2018.
<https://doi.org/10.1016/J.JPOWSOUR.2017.10.092>
- [6] Tian, J., Xiong, R., Yu, Q. "Fractional-Order Model-Based Incremental Capacity Analysis for Degradation State Recognition of Lithium-Ion Batteries", *IEEE Transactions on Industrial Electronics*, 66(2), pp. 1576–1584, 2019.
<https://doi.org/10.1109/TIE.2018.2798606>
- [7] Ma, Z., Yang, R., Wang, Z. "A novel data-model fusion state-of-health estimation approach for lithium-ion batteries", *Applied Energy*, 237, pp. 836–847, 2019.
<https://doi.org/10.1016/J.APENERGY.2018.12.071>
- [8] Bloom, I., Christophersen, J. P., Abraham, D. P., Gering, K. L. "Differential voltage analyses of high-power lithium-ion cells: 3. Another anode phenomenon", *Journal of Power Sources*, 157(1), pp. 537–542, 2006.
<https://doi.org/10.1016/J.JPOWSOUR.2005.07.054>
- [9] Honkura, K., Takahashi, K., Horiba, T. "Capacity-fading prediction of lithium-ion batteries based on discharge curves analysis", *Journal of Power Sources*, 196(23), pp. 10141–10147, 2011.
<https://doi.org/10.1016/J.JPOWSOUR.2011.08.020>
- [10] Xiong, R., Li, L., Tian, J. "Towards a smarter battery management system: A critical review on battery state of health monitoring methods", *Journal of Power Sources*, 405, pp. 18–29, 2018.
<https://doi.org/10.1016/J.JPOWSOUR.2018.10.019>
- [11] Honkura, K., Honbo, H., Koishikawa, Y., Horiba, T. "State Analysis of Lithium-Ion Batteries Using Discharge Curves", *ECS Transactions*, 13(19), 61, 2008.
<https://doi.org/10.1149/1.3018750>
- [12] Bloom, I., Jansen, A. N., Abraham, D. P., Knuth, J., Jones, S. A., Battaglia, V. S., Henriksen, G. L. "Differential voltage analyses of high-power, lithium-ion cells: 1. Technique and application", *Journal of Power Sources*, 139(1–2), pp. 295–303, 2005.
<https://doi.org/10.1016/J.JPOWSOUR.2004.07.021>
- [13] Bloom, I., Christophersen, J., Gering, K. "Differential voltage analyses of high-power lithium-ion cells: 2. Applications", *Journal of Power Sources*, 139(1–2), pp. 304–313, 2005.
<https://doi.org/10.1016/J.JPOWSOUR.2004.07.022>

- [14] Bloom, I., Walker, L. K., Basco, J. K., Abraham, D. P., Christophersen, J. P., Ho, C. D. "Differential voltage analyses of high-power lithium-ion cells. 4. Cells containing NMC", *Journal of Power Sources*, 195(3), pp. 877–882, 2010.
<https://doi.org/10.1016/J.JPOWSOUR.2009.08.019>
- [15] Andre, D., Meiler, M., Steiner, K., Wimmer, C., Soczka-Guth, T., Sauer, D. U. "Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. I. Experimental investigation", *Journal of Power Sources*, 196(12), pp. 5334–5341, 2011.
<https://doi.org/10.1016/J.JPOWSOUR.2010.12.102>
- [16] Abe, Y., Hori, N., Kumagai, S. "Electrochemical Impedance Spectroscopy on the Performance Degradation of LiFePO₄/Graphite Lithium-Ion Battery Due to Charge-Discharge Cycling under Different C-Rates", *Energies*, 12(23), 4507, 2019.
<https://doi.org/10.3390/EN12234507>
- [17] Teliz, E., Zinola, C. F., Díaz, V. "Identification and quantification of ageing mechanisms in Li-ion batteries by Electrochemical impedance spectroscopy.", *Electrochimica Acta*, 426, 140801, 2022.
<https://doi.org/10.1016/J.ELECTACTA.2022.140801>
- [18] Li, W., Sengupta, N., Dechent, P., Howey, D., Annaswamy, A., Sauer, D. U. "Online capacity estimation of lithium-ion batteries with deep long short-term memory networks", *Journal of Power Sources*, 482, 228863, 2021.
<https://doi.org/10.1016/J.JPOWSOUR.2020.228863>
- [19] Li, W., Cao, D., Jöst, D., Ringbeck, F., Kuipers, M., Frie, F., Sauer, D. U. "Parameter sensitivity analysis of electrochemical model-based battery management systems for lithium-ion batteries", *Applied Energy*, 269, 115104, 2020.
<https://doi.org/10.1016/J.APENERGY.2020.115104>
- [20] Bartlett, A., Marcicki, J., Onori, S., Rizzoni, G., Yang, X. G., Miller, T. "Electrochemical Model-Based State of Charge and Capacity Estimation for a Composite Electrode Lithium-Ion Battery", *IEEE Transactions on Control Systems Technology*, 24(2), pp. 384–399, 2016.
<https://doi.org/10.1109/TCST.2015.2446947>
- [21] Li, W., Fan, Y., Ringbeck, F., Jöst, D., Han, X., Ouyang, M., Sauer, D. U. "Electrochemical model-based state estimation for lithium-ion batteries with adaptive unscented Kalman filter", *Journal of Power Sources*, 476, 228534, 2020.
<https://doi.org/10.1016/J.JPOWSOUR.2020.228534>
- [22] He, H., Xiong, R., Guo, H. "Online estimation of model parameters and state-of-charge of LiFePO₄ batteries in electric vehicles", *Applied Energy*, 89(1), pp. 413–420, 2012.
<https://doi.org/10.1016/J.APENERGY.2011.08.005>
- [23] Zou, Y., Hu, X., Ma, H., Li, S. E. "Combined State of Charge and State of Health estimation over lithium-ion battery cell cycle lifespan for electric vehicles", *Journal of Power Sources*, 273, pp. 793–803, 2015.
<https://doi.org/10.1016/J.JPOWSOUR.2014.09.146>
- [24] Li, W., Rentemeister, M., Badedo, J., Jöst, D., Schulte, D., Sauer, D. U. "Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation", *Journal of Energy Storage*, 30, 101557, 2020.
<https://doi.org/10.1016/J.EST.2020.101557>
- [25] Wu, J., Wei, Z., Li, W., Wang, Y., Li, Y., Sauer, D. U. "Battery Thermal- and Health-Constrained Energy Management for Hybrid Electric Bus Based on Soft Actor-Critic DRL Algorithm", *IEEE Transactions on Industrial Informatics*, 17(6), pp. 3751–3761, 2021.
<https://doi.org/10.1109/TII.2020.3014599>
- [26] Richardson, R. R., Birkel, C. R., Osborne, M. A., Howey, D. A. "Gaussian Process Regression for *In Situ* Capacity Estimation of Lithium-Ion Batteries", *IEEE Transactions on Industrial Informatics*, 15(1), pp. 127–138, 2019.
<https://doi.org/10.1109/TII.2018.2794997>
- [27] Hu, C., Jain, G., Schmidt, C., Strief, C., Sullivan, M. "Online estimation of lithium-ion battery capacity using sparse Bayesian learning", *Journal of Power Sources*, 289, pp. 105–113, 2015.
<https://doi.org/10.1016/J.JPOWSOUR.2015.04.166>
- [28] Shen, S., Sadoughi, M., Chen, X., Hong, M., Hu, C. "A deep learning method for online capacity estimation of lithium-ion batteries", *Journal of Energy Storage*, 25, 100817, 2019.
<https://doi.org/10.1016/J.EST.2019.100817>
- [29] You, G., Park, S., Oh, D. "Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach", *Applied Energy*, 176, pp. 92–103, 2016.
<https://doi.org/10.1016/J.APENERGY.2016.05.051>
- [30] Shen, S., Sadoughi, M., Li, M., Wang, Z., Hu, C. "Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries", *Applied Energy*, 260, 114296, 2020.
<https://doi.org/10.1016/J.APENERGY.2019.114296>
- [31] Qian, C., Xu, B., Chang, L., Sun, B., Feng, Q., Yang, D., Ren, Y., Wang, Z. "Convolutional neural network based capacity estimation using random segments of the charging curves for lithium-ion batteries", *Energy*, 227, 120333, 2021.
<https://doi.org/10.1016/J.ENERGY.2021.120333>
- [32] Xiong, R., Wang, J., Shen, W., Tian, J., Mu, H. "Co-Estimation of State of Charge and Capacity for Lithium-Ion Batteries with Multi-Stage Model Fusion Method", *Engineering*, 7(10), pp. 1469–1482, 2021.
<https://doi.org/10.1016/J.ENG.2020.10.022>
- [33] Chen, Z., Xiao, J., Shu, X., Shen, S., Shen, J., Liu, Y. "Model-Based Adaptive Joint Estimation of the State of Charge and Capacity for Lithium-Ion Batteries in Their Entire Lifespan", *Energies*, 13(6), 1410, 2020.
<https://doi.org/10.3390/EN13061410>
- [34] Nian, P., Shuzhi, Z., Xiongwen, Z. "Co-estimation for capacity and state of charge for lithium-ion batteries using improved adaptive extended Kalman filter", *Journal of Energy Storage*, 40, 102559, 2021.
<https://doi.org/10.1016/J.EST.2021.102559>
- [35] Wang, Q., Ye, M., Wei, M., Lian, G., Wu, C. "Co-estimation of state of charge and capacity for lithium-ion battery based on recurrent neural network and support vector machine", *Energy Reports*, 7, pp. 7323–7332, 2021.
<https://doi.org/10.1016/j.egyr.2021.10.095>

- [36] Plett, G. L. "Recursive approximate weighted total least squares estimation of battery cell total capacity", *Journal of Power Sources*, 196(4), pp. 2319–2331, 2011.
<https://doi.org/10.1016/j.jpowsour.2010.09.048>
- [37] Plett, G. "Battery management systems, Volume II: Equivalent-circuit methods", Artech, 2015. ISBN 9781630810283 [online] Available at: <http://ieeexplore.ieee.org/document/9100098> [Accessed: 06 March 2024]
- [38] Sadollah, A., Sayyaadi, H., Yadav, A. "A dynamic metaheuristic optimization model inspired by biological nervous systems: Neural network algorithm", *Applied Soft Computing*, 71, pp. 747–782, 2018.
<https://doi.org/10.1016/j.asoc.2018.07.039>
- [39] Elmarghichi, M., Bouzi, M., Ettalabi, N., Derri, M. "Comparison of state of charge estimation algorithms for lithium battery", In: *Proceedings of the 2nd International Conference on Electronic Engineering and Renewable Energy Systems*, Saidia, Morocco, 2020, pp. 293–300. ISBN 978-981-15-6258-7
https://doi.org/10.1007/978-981-15-6259-4_30
- [40] Shrivastava, P., Soon, T. K., Idris, M. Y. I. B., Mekhilef, S. "Overview of model-based online state-of-charge estimation using Kalman filter family for lithium-ion batteries", *Renewable and Sustainable Energy Reviews*, 113, 109233, 2019.
<https://doi.org/10.1016/j.rser.2019.06.040>
- [41] Elmarghichi, M., Bouzi, M., Ettalabi, N. "Robust Parameter Estimation of an Electric Vehicle Lithium-Ion Battery Using Adaptive Forgetting Factor Recursive Least Squares", *International Journal of Intelligent Engineering & Systems*, 13(5), pp. 74–84, 2020.
<https://doi.org/10.22266/ijies2020.1031.08>
- [42] Mouncef, E., Mostafa, B., Naoufl, E. "Online Parameter Estimation of an Electric Vehicle Lithium-Ion Battery Using AFFRLS", In: *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, Kenitra, Morocco, 2020, pp. 1–6. ISBN 978-1-7281-6922-4
<https://doi.org/10.1109/ICECOCS50124.2020.9314577>
- [43] El Marghichi, M., Loulijat, A., El Hantati, I. "Variable Recursive Least Square Algorithm for Lithium-ion Battery Equivalent Circuit Model Parameters Identification", *Periodica Polytechnica Electrical Engineering and Computer Science*, 67(3), pp. 239–248, 2023.
<https://doi.org/10.3311/PPee.21339>
- [44] Mouncef, E., Mostafa, B., Naoufl, E. "Online Parameter Estimation of a Lithium-Ion Battery based on Sunflower Optimization Algorithm", In: *2020 2nd Global Power, Energy and Communication Conference (GPECOM)*, Izmir, Turkey, 2020, pp. 53–58. ISBN 978-1-7281-6265-2
<https://doi.org/10.1109/GPECOM49333.2020.9247936>
- [45] NASA "Li-ion Battery Aging Datasets", [online] Available at: <https://data.nasa.gov/dataset/Li-ion-Battery-Aging-Datasets/uj5r-zjdb/data> [Accessed: 01 January 2024]