International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(10s) 2026 pp. 1-22
https://doi.org/10.70917/ijcisim-2026-3562
Article

An Improved Salamander-Optimized Dual-Attention-Based Residual Bidirectional LSTM Deep Convolutional Maxout Network for EV Battery SOC and SOH Prediction

Ashish Kumar Singh , Pushparajesh Viswanathan

Department of Electrical and Electronics Engineering , Faculty of Engineering and Technology, Jain Deemed - to - be University, Bengaluru , Karnataka, India.
s.ashish@jainuniversity.ac.in , v.pushparajesh@jainuniversity.ac.in

Abstract : Lithium-ion batteries (LIBs) are widely used in various sectors including electronic devices, electric vehicles (EVs) and energy storage systems due to their high energy density, low self-discharge rate and long cycle life. Despite widespread challenges, the inability of traditional estimation methods to accurately monitor the State of Charge (SOC) and State of Health (SOH) of batteries systems from the complex, non-linear degradation patterns that occur under various operating conditions. To resolve these problems, this study proposes a novel Dual-Attention-based Residual bidirectional long short-term memory Deep Convolutional Maxout Neural Network (DA-Res-Bi LSTM-MaxNet) for precise co-estimation of SOC and SOH in LIBs. Here, the Improved Salamander Optimization Algorithm (ISOA) is used for tuning the hyperparameters and weight parameters, ensuring maximum predictive accuracy. The ISOA is improved using the sine cosine algorithm (ISOA-SCA). The MATLAB/Simulink results show that the proposed values of Root mean square error (RMSE) of 0.131%, Mean absolute error (MAE) of 0.132%, Mean Squared Error (MSE) of 0.085% and Mean Absolute Percentage Error (MAPE) of 13.5%. As a result, these very low error measurements conclude that the proposed ISOA-SCA-optimized DA-ResBiLSTM-MaxNet architecture provides superior accuracy and robustness for simultaneous co-estimation of SOC and SOH under complex, nonlinear battery degradation conditions.
Keywords: Dual-Attention Residual bidirectional LSTM-Based Deep Convolutional Maxout Neural Network, Improved Salamander Optimization Algorithm, State of Charge, State of Health.

1. Introduction

LIBs are used in energy storage and EVs due to high energy density, long lifespan and low self-discharge rates [1]. Accurate SOC assessment is necessary to enhance the operation and reliability of LIBs in EVs [2]. Accurately predicting battery life is essential to improve energy management, increase safety and ensure system reliability, especially the focus is early-stage batteries [3]. For efficient battery recycling, secondary use and timely warnings, necessary to accurately forecast SOH and remaining useful life (RUL) LIBs at the early cycle stage [4]. Amidst rising global temperatures, waves are increasingly being recognized as an excellent solution for reducing greenhouse gas emissions and addresses climate change [5].

LIBs are commonly subject to problems such as temperature, overcharging and over discharging [6]. Battery management systems in consumer electronics (CE), it is essential to reliably predict the SOH of a lithium-ion battery [7]. Accurately detecting the condition of the battery is essential for timely maintenance, replacement and safe operation of EVs [8]. In EV applications, detecting LIB degradation is essential to ensure safety and reliability [9]. The increasing use of EVs, an opportunity is emerging to reuse end-of-life batteries for secondary level (SL) applications such as energy storage systems [10].

To improve quality of the dataset and reliability model, data alignment and training data renormalization strategies were implemented [11]. An energy management system (EMS) integrates forecasting and reinforcement learning (RL) to optimize the control of vehicle-to-grid (V2G) energy transfer, utilizing EV usage patterns and projected energy demand [12]. The equivalent charge and discharge voltage difference intervals are extracted as key aging features, since they accurately reflect the voltage profile variations induced by battery degradation [13]. Advanced battery management system (BMs) strategies are essential to improve accurate prediction of the RUL, SOC and SOH of EV LIBs [14]. The health index (HI) is derived from the very short time interval (less than 0.1 seconds) during which the battery transitions from a discharged state to a resting state [15]. Existing studies face significant challenges due to nonlinear battery behavior, dynamic environmental conditions, poor generalization and the need for an intelligent AI framework. To address these limitations in dynamic situations, this study introduces an improved, attention-based joint co-estimation strategy.

This research aims to resolve the significant challenges in achieving high-precision SOC and SOH estimation under stable operating conditions, thereby ensuring the reliable operation of EV LIB management systems. Conventional systems often suffer from slow convergence speeds, instability and poor transient responses, when it deals with non-linear dynamics of lithium-ion batteries. To overcome these limitations, this research proposes the DA-ResBiLSTM-MaxNet framework, which utilizes dual-attention mechanisms and residual bidirectional LSTM to capture multi-scale features without signal loss. The ISOA-SCA tuned DA-ResBiLSTM-MaxNet facilitates automated hyperparameter and weight tuning, which ensures superior stability, faster convergence and enhanced battery durability across diverse operational landscapes. The main objectives of this paper are;

- To design DA-ResBiLSTM-MaxNet architecture using dual-attention, residual bidirectional LSTM and Maxout units for spatiotemporal battery features extraction without signal loss.
- To perform simultaneous co-estimation of battery SOC and SOH by using a custom multi-objective loss function to capture their internal dependencies.
- To optimize ISOA-SCA for dynamic hyperparameter tuning to aggressively minimize RMSE and MAE prediction errors.

The remaining section of the paper is given in section 2, which explains the related works based on the prediction of SOC and SOH. The proposed methodology based on EV is discussed in Section 3. The results as well as the discussion for evaluating the performance under the conditions are detailed in Section 4. The conclusion and future scope is also discussed in Section 5.

2. Literature survey

Some of the authors discussed related works for predicting EV battery SOC and SOH.

Alwesabi *et al.* [18] published walrus optimization algorithm-Convolutional Neural Network-Bidirectional Long Short-Term Memory-Attention Mechanism (WaOA-CNN-BiLSTM-AM) architecture that combines the benefits of local feature extraction, bidirectional dependency learning and adaptive weight allocation for better SOC modeling. The WaOA technique was used to optimize critical hyperparameters for improved convergence and better robustness of estimation. A multiple-temperature learning approach, along with transfer learning, improved the adaptability of model in unseen temperature scenarios. The architecture has very high computational cost and latency as a result of using a multi-layered hybrid network and metaheuristic optimization techniques.

Kant *et al.* [19] suggested novel bidirectional recurrent neural network (biRNN) with a (LSTM) network captures temporal dependencies from the NASA battery dataset; their linear sequential architecture lacked an advanced multi-scale feature extraction backbone and an automated optimization engine, making it prone to perform bottlenecks and accuracy degradation under highly dynamic operating conditions. The experimental validation in the study was conducted using the LIB dataset from the NASA Prognostics Data Repository. Compared to traditional feed-forward neural networks, RNNs are designed to learn temporal dependencies and perform sequence recognition on the original data.

Sakr *et al.* [20] created an IoT-fog-cloud framework to select and customize ML models for SOH estimation; their approach focused primarily on deployment infrastructure and relied on shallow learning algorithms that lacked advanced deep spatiotemporal feature extraction and automated hyperparameter optimization, limiting its prediction accuracy under dynamic real-world EV operations. Requirements regarding EV battery data and SoH estimation accuracy point to optimal methods for enhancing predictive performance.

Rajasekaran and Venkatanarayanan [21] suggested an adversarial model combining CNNs, LSTM and echo state networks (ESNs) within a generative adversarial network (GAN) architecture to extract spatiotemporal battery features. Their model suffers from inherent GAN training instabilities and lacks a multi-objective metaheuristic optimization engine, resulting in perform bottlenecks and higher baseline estimation errors under dynamic operating profiles.

Naresh and Sriram [22] suggested the homomorphic encryption (HE) enabled privacy-preserving deep neural network (PPDNN) to secure battery data streams. A security-centric framework introduces heavy computational overhead and lacks a hyperparameter optimization engine, resulting in lower real-time estimation accuracy compared to prediction-optimized architectures. Security analysis reveals the effectiveness of this method against various attacks across the stages of machine learning (ML).

Alharbi *et al.* [23] developed a distributed learning model with a federated averaging technique using a 1D CNN to preserve data privacy across multiple clients; their model's simplified sequential architecture lacks an attention mechanism and a global hyperparameter optimization engine, resulting in a loss of fine-grained spatiotemporal degradation features and estimation accuracy. The summarization of existing methods is given in Table 1.

Table 1: Summarization of existing methods

Author name & References	Method Name	Dataset	Key findings	Limitations
Alwesabi *et al.* [18]	WaOA-CNN-BiLSTM-AM	Center for Advanced Life Cycle Engineering (CALCE)	Achieves accurate, multi-temperature SOC estimation by combining local features.	High computational complexity.
Kant *et al.* [19]	biRNN-LSTM	NASA prognostics data repository.	Integrated biRNN-LSTM for SOH prediction.	Accurately identifying SOH requires a high computational cost.
Sakr *et al.* [20]	ML	IoT-Enabled data.	Using IoT-Fog-Cloud enables ML model selection to handle complex battery degradation.	Limited by high computational latency.
Rajasekaran and Venkatanarayanan [21]	GANs	NASA Battery Data Set	Improve battery SOH prediction using a CNN-LSTM generator.	The reliance on large, high-quality, long-term decomposition datasets.

Naresh & V S Sriram [22]	PPDNN	Tenseal package on real battery cell.	Achieve high-accuracy, privacy-preserving inference.	Limited by high computational overhead and latency.
Alharbi *et al.* [23]	CNN, LSTM	NASA battery	1D CNN has the best performance on the NASA Battery dataset.	Limited by low operation and high error rates due to data differences between different clients.

2.1. Research gaps

The existing approaches, such as WaOA-CNN-BiLSTM-AM, suffer from high computational complexity and significant processing delay, which makes their implementation difficult for real-time battery management system applications [16]. The integration of the biRNN-LSTM network is effective in the case of temporal dependency, but it tends to be unstable due to vanishing gradients and requires high memory capacity for long sequences [17]. Hence, the application of the proposed ML within the IoT-fog-cloud architecture in real-time applications may create latency and synchronization problems between different architectural levels [18]. The conventional GAN framework suffers from training difficulty and may experience the problem of mode collapse or instability between the generator and discriminator [19]. The hybrid PPDNN network for privacy preservation, the problem of high computational cost caused by encryption operations may affect its practicality for online battery management [20]. Finally, the federated learning approach faces communication bottlenecks and potential performance degradation if the data distributed across local clients is non-independent and identically distributed (non-IID) [21].

3. Proposed methodology

This work presents a robust method for estimating SoC and SoH of the battery system. The proposed system handles non-linear battery dynamics under high changing operating conditions. The architecture of proposed methodology shown in Figure 1.

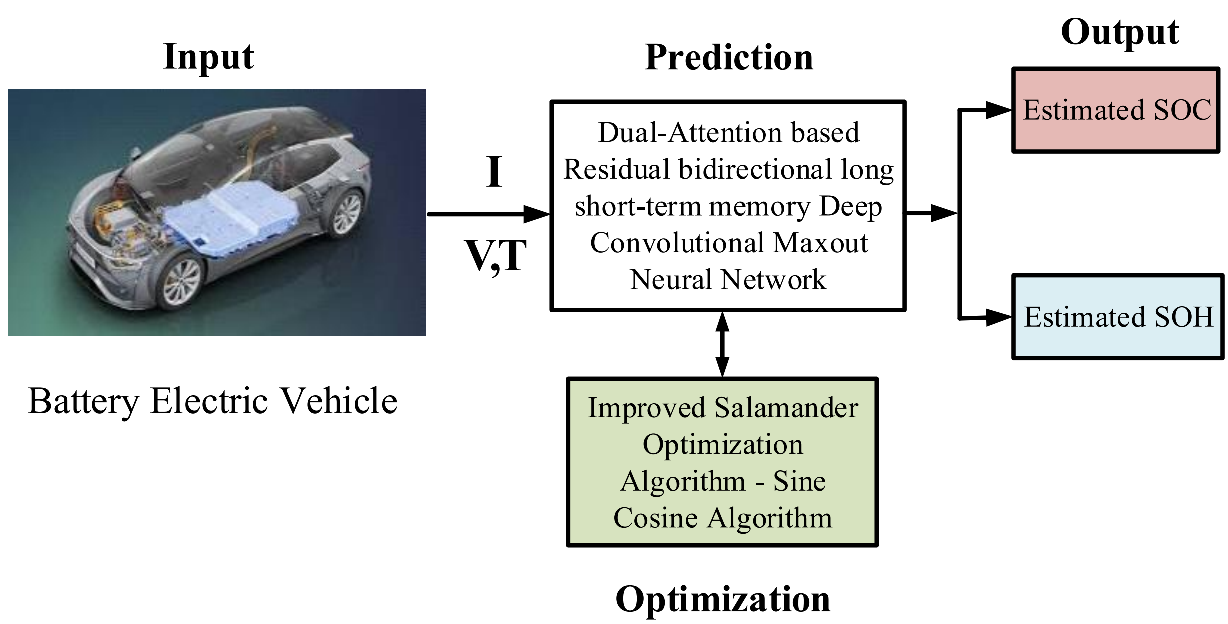

Figure 1. Structure of the proposed methodology

This work starts by obtaining vehicle telemetry data from a BEV, including operational input parameters such as voltage (V), current (I) and temperature (T). These sequential input streams are fed directly into the DA-ResBiLSTM-MaxNet to extract robust temporal and spatial feature representations. To minimize prediction errors and maximize model accuracy, the structural hyperparameters and weights of the network are dynamically tuned by the ISOA-SCA. Finally, the optimized neural network processes the extracted high-level features to execute simultaneous joint estimation, yielding highly accurate, parallel outputs for both battery SOC and SOH.

3.1. Modelling of State of Charge

The SOC is a parameter that ensure reliable and efficient operation of EV BMS. Because intrinsic chemical and electrical properties of a battery are highly complex and non-linear, it is impossible to directly measure SOC. Instead, it must be calculated using indirect methods [24]. This work estimates SOC using battery current integration and deep learning-based nonlinear feature extraction to improve tracking accuracy during dynamic charging and discharging conditions. Mathematically, the Coulomb counting method is calculated as follows:

$$SOC = SOC_0 - \frac{1}{c_{CAP}} \int_0^{\tau} \eta I_{BATT} d\tau \tag{1}$$

$$SOC = \frac{\eta I_{BATT}}{c_{CAP}} \tag{2}$$

Where, η is the coulombic coefficient associated with the process of charging and discharging, I_{BATT} represent maximum capacity of electric current storage in the battery, SOC_0 is the initial SOC, c_{CAP} represent total capacity of battery and τ represent time variable.

3.2. *Modelling of State of Health*

The SOH indicates battery aging and degradation level based on capacity reduction and internal resistance variation over time. This degradation leads to a decrease in energy due to high ohmic and polarization resistance. While traditional methods rely on static relationships, the proposed model continuously monitors battery degradation characteristics to accurately estimate battery health during long-term EV operation [25]. The critical indicators of battery health is expressed in equation (3).

$$\left(SOH\right)_T = \frac{c_i}{c_n} \tag{3}$$

Where, $\left(SOH\right)_T$ represent state of health at time, c_i represent present effective capacity, and c_n shows the nominal capacity of the battery at time T. Moreover, equation (4) shows how the internal resistance of a battery increases over time, compared to its initial value at the beginning of life.

$$\left(SOH\right)_T = \frac{r_{0,T}}{r_{0,0}} \tag{4}$$

Here, $r_{0,T}$ represent current ohmic resistance and $r_{0,0}$ represent initial ohmic resistance. The ML algorithms predict both capacity, resistance of a battery. Finally SOH of the battery can be determined using Equation (3) for capacity or Equation (4) for resistance. For modern advanced batteries, establishing this accurate SOH value is essential for accurate detection and monitoring of the battery's real-time operational status.

3.3. Prediction of Dual-Attention-based Residual Bidirectional LSTM Deep Convolutional Maxout Neural Network

For the purpose of estimating the battery states of SOC and SOH with accuracy under dynamic operation scenarios, this study proposes the DA-ResBiLSTM-MaxNet model that provides an exhaustive pipeline based on deep learning feature extraction, bidirectional LSTM modeling, dual attention-based selection and piecewise linear mapping. The DA-ResBiLSTM-MaxNet model is shown in Figure 2.

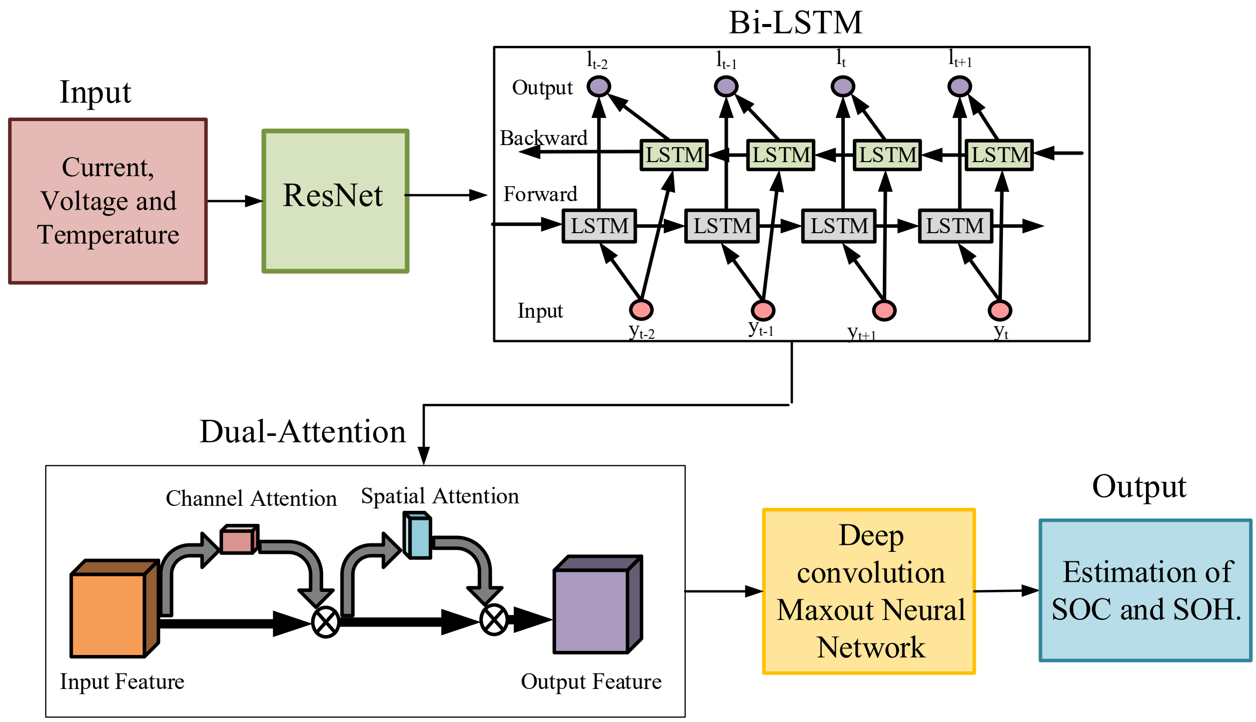

Figure 2: Structure of proposed DA-ResBiLSTM-MaxNet

The process starts from the Input block, providing the sequential input of battery I, V and T values, which are directly fed into the ResNet block. The ResNet model isolates and identifies relevant local features, like short-term voltage changes or spikes, whereas identity skip connections ensure that gradient problems would not arise even during deep layering. As a result, those relevant local features would then be transmitted to the Bi-LSTM model, where long-term time patterns or processes of battery deterioration will be found via parallel analysis of the hidden layers in two directions, forward and backward. After establishing local and temporal features of the batteries, the resulting sequence will enter the Dual-Attention block that applies channel and spatial-temporal attentions to identify the most important local and temporal features for further analysis. Finally, the selected local and temporal features are provided to the Maxout-based neural network architecture, consisting of interleaved convolution, dropout, fully connected and maxout blocks for the execution of piecewise linear mapping.

3.3.1. ResNet

The Input block contains sequential time-series data measurements corresponding to the battery's I, V and temperature T of the battery. The first step involves feature extraction, where the primary features extracted correspond to the voltage drop, current change and thermal variation, respectively. Following the Input block, the ResNet module utilizes Residual learning blocks to extract local nonlinear battery features such as voltage fluctuations and thermal variations. During deep learning, skip connections are used to avoid vanishing gradient problems and preserve important feature information [26]. Figure 3 shows the structure of ResNet.

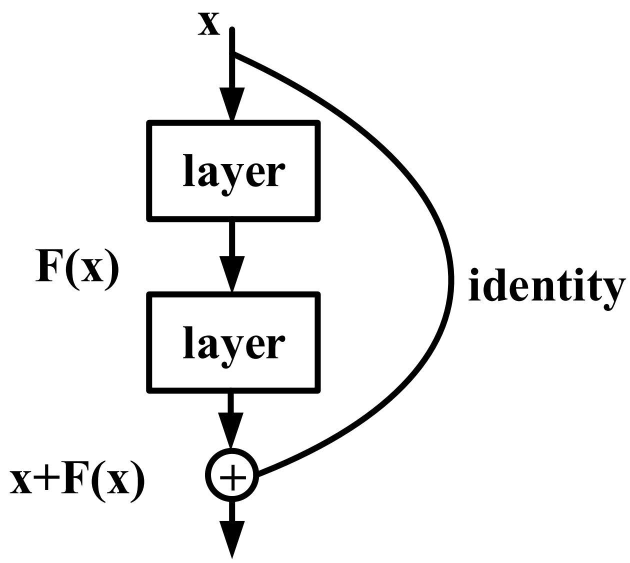

Figure 3: Structure of ResNet

The residual building block with a skip or identity connection, which is the basic architecture employed to train deep neural network architectures such as ResNet. In the above residual building block, the input vector x branches into two streams: the first stream is the primary stream, while the other stream is the shortcut stream. In the first stream, the vector x passes through consecutive weights layers compute residual transformation, which is referred to as $F(x)$. The shortcut stream transports the same x vector without making any alterations whatsoever, passing through no layers at all. The above two streams meet together through an element-wise addition operator to produce the output of the layer $F(x)+x$. By providing a clear highway for the slopes to flow, this structure helps resolve the issue of vanishing slopes.

$$H(x)=F(x)+x \tag{5}$$

It is the basic mathematical expression of a residual mapping, which is usually referred to as a skip connection or an identity shortcut in a ResNet. Where, $H(x)$ represent final output of the residual block, x represents the input and $F(x)$ represent the learned residual mapping. The block is a set of high-dimensional spatial feature maps that encode localized transient variations and short-term battery behavior as direct and immediate output. Such feature maps can be passed to the successive temporal and attention layers in order to generate the exact network outputs.

3.3.2. Bidirectional LSTM

The extracted feature maps from the ResNet blocks are directly given to the BiLSTM module. The BiLSTM module analyzes sequences in both directions to understand long-term temporal dependencies. This effectively understand complex battery degradation mechanisms and dynamic charging operations. The BiLSTM layer consists of two separate hidden layers, where one hidden layer analyzes the temporal sequence in chronological order, whereas the other layer analyzes the temporal information from the future to the past order [27]. The architecture of BiLSTM model is given in Figure 4.

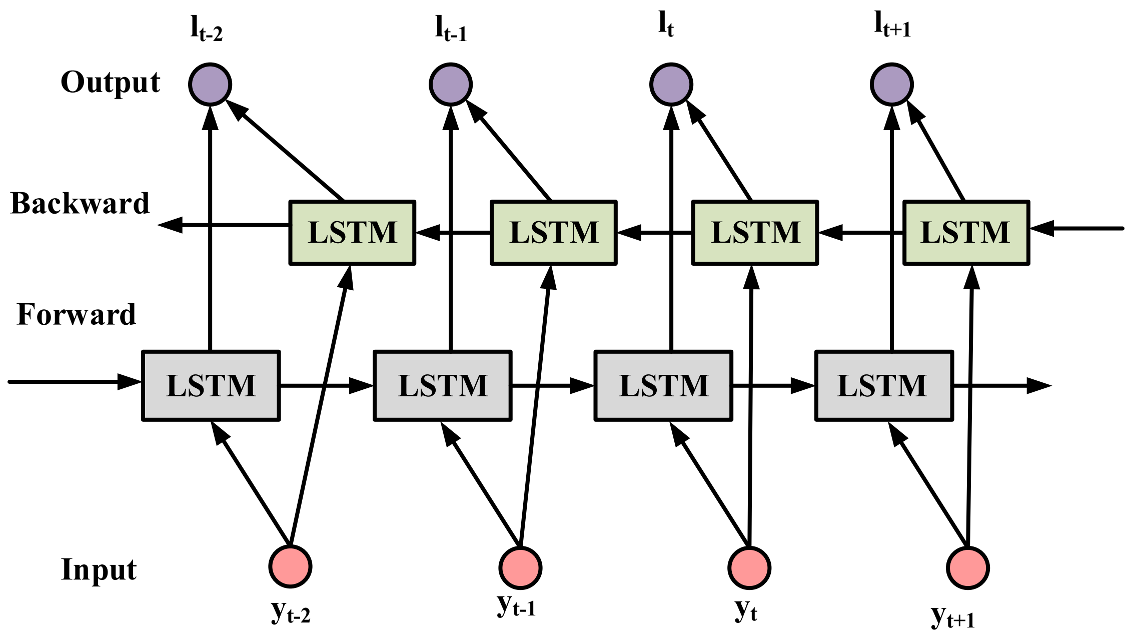

Figure 4: Architecture of bidirectional LSTM

The BiLSTM network that aims at obtaining complete information from time-series data. The stream of input values y_t is divided and simultaneously processed through two separate but parallel streams, in forward direction and backward direction. Forward LSTM stream captures the dependencies of the data from the past to the future direction, whereas the Backward LSTM stream captures succeeding events from the future to the past direction. The hidden states of the parallel streams are then combined to generate a single output value l_t. The mathematical equations of the Bi-LSTM network model are described below:

The mathematical calculation of the hidden state vector in the forward hidden layer, which stores accumulated historical context up to the current time step (t), is given in Equation (6)

$$\vec{l}_t = LSTM\left(\vec{l}_{t-1}, y_t\right) \tag{6}$$

The mathematical expression for the hidden state vector in the backward hidden layer, capturing future dependencies independently from the previous one by processing the input in the reverse order, is shown in equation (7).

$$\bar{l}_t = LSTM\left(\vec{l}_{t+1}, y_t\right) \tag{7}$$

The process of merging both of the forward and backward hidden layers into one complete temporal feature vector y_t reflecting the entire behavior of the battery is given in equation (8).

$$l_t = \left[\vec{l}_t, \bar{l}_t\right] \tag{8}$$

Where, l_t represent output layer at time t, $\vec{l}_t$ represent forward layer at time t, $\bar{l}_t$ represents backward layer at time t and y_t represents the input at time t. The output of this layer is a concatenated state vector l_t, by integrating both directions into one cohesive vector for the entire temporal sequence and represents the trends of battery degradation. Figure 4 shows the structure of a bidirectional LSTM.

3.3.3. Dual-Attention

Dual-Attention block, the primary input comes directly from the output of the previous Bi LSTM layers through l_t. With deep learning models that monitor the dynamics of battery operations, regular sequential models tend to weigh each individual feature and time period equally. This equal weighting often results in loss of state information due to operational noise. The architecture of dual-attention mechanism is shown in Figure 5.

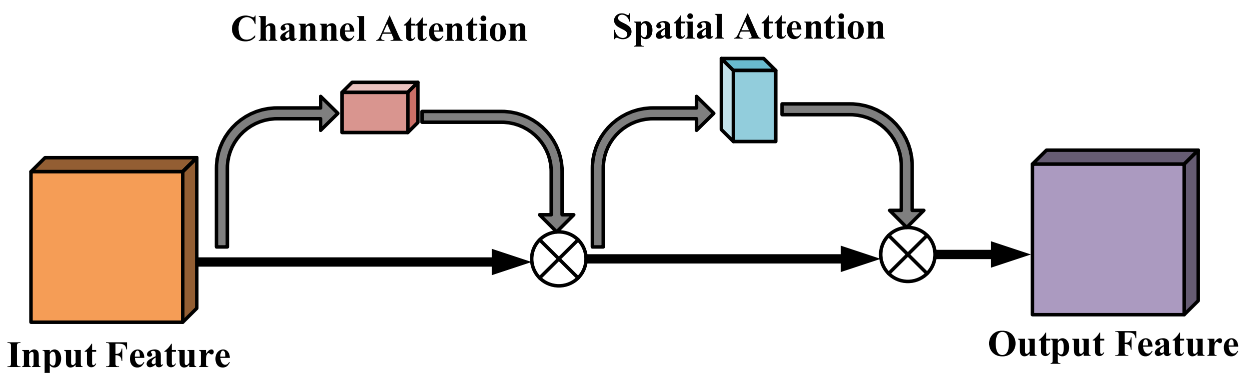

Figure 5: Structure of Dual-attention mechanism

To overcome this problem, the dual-attention mechanism uses channel and spatial-temporal attention to identify important battery features and critical degradation regions. It improves feature prioritization and suppresses irrelevant noisy information during prediction [28].

3.3.3.1. Channel Attention

The Channel Attention Module takes the temporal features obtained from the Bi-LSTM module as its input. The sequential networks that assign equal weight to each feature can lead to loss of critical diagnostic information due to operational noise. To address this issue, the channel attention mechanism gives adaptive importance weights to different feature channels. The network focus on highly informative degradation-related signals. Due to the changing importance of each channel depending on the current battery load state, an attention weight matrix for such channels is calculated specifically by this equation.

$$M_s(F) = \sigma\left(f^{7\times7}\left(\left[AvgPool(F); MaxPool(F)\right]\right)\right) \tag{9}$$

$$M_s(F) = \sigma\left(f^{7\times7}\left(\left[F^s_{avg}; F^s_{\max}\right]\right)\right) \tag{10}$$

Where, $MaxPool(F)$ represent max-pooling, $AvgPool(F)$ represents average-pooling, σ denotes sigmoid function, $W_0 \in R^{\frac{C}{r\times C}}$ and $W_1 \in R^{C\times\frac{C}{r}}$. The MLP weights, W_0 and W_1 are shared between the two inputs and the rectified linear unit (ReLU) follows the first weight layer W_0 to manage processing bias. The final channel-filtered feature tensor, where the calculated weights have been applied to amplify critical battery signals and suppress noise.

3.3.3.2. Spatial Attention

The channel attention module are inputs to the Spatial Attention Module. While the channel attention process identifies sensor features that need to be emphasized in a battery monitoring application, the spatial attention process identifies important temporal regions during critical battery degradation. It improves the network to track transient battery behavior under varying load conditions. The both max-pooling and average-pooling processes being concurrently on input data on channel dimension. Max pooling identifies extreme values such as high voltage, while average pooling captures the temporal patterns in the input data. These feature maps generated by the two processes are then combined to generate a multi-channel feature map, which is further processed through a convolution filter to yield a single-channel feature map. Lastly, spatial weights are calculated by running the one-channel data through a sigmoid activation function. Multiplying these weights the input feature map helps filter out any constant values that may affect the identification of critical battery conditions.

$$M_s(F) = \sigma\left(f^{7\times7}\left(\left[AvgPool(F); MaxPool(F)\right]\right)\right) \quad (11)$$

$$M_s(F) \quad = \sigma\left(f^{7\times7}\left(\left[F^s_{avg}; F^s_{\max}\right]\right)\right) \quad (12)$$

Here, σ refers to sigmoid function, while $f^{7\times7}$ represent convolution function with a filter size of 7×7. Performing two layers of filtering, the model successfully isolates not only the most relevant sensors but also the exact time points for critical degradation of the battery.

3.3.4. Deep Convolution Maxout Neural Network

The noise-tolerant feature maps that are produced by the Dual-Attention module are fed directly into the Deep Convolutional Maxout Neural Network (DCMNN). The DCMNN applies advanced nonlinear feature mapping using multiple deep convolutional layers that enable abstract features from the weighted sensory time series data. Where gradient vanishing occurs, this module utilizes the Maxout activation function. The Maxout mechanism improves learning flexibility, avoids dead neuron problems and enhances nonlinear battery behavior modeling. Finally, the feedforward propagation process within this network architecture involves an embedding layer, dropout, convolutional blocks, max-pooling layers and a dense linear layer [29]. Figure 6 shows the structure of the deep convolution maxout neural network [29].

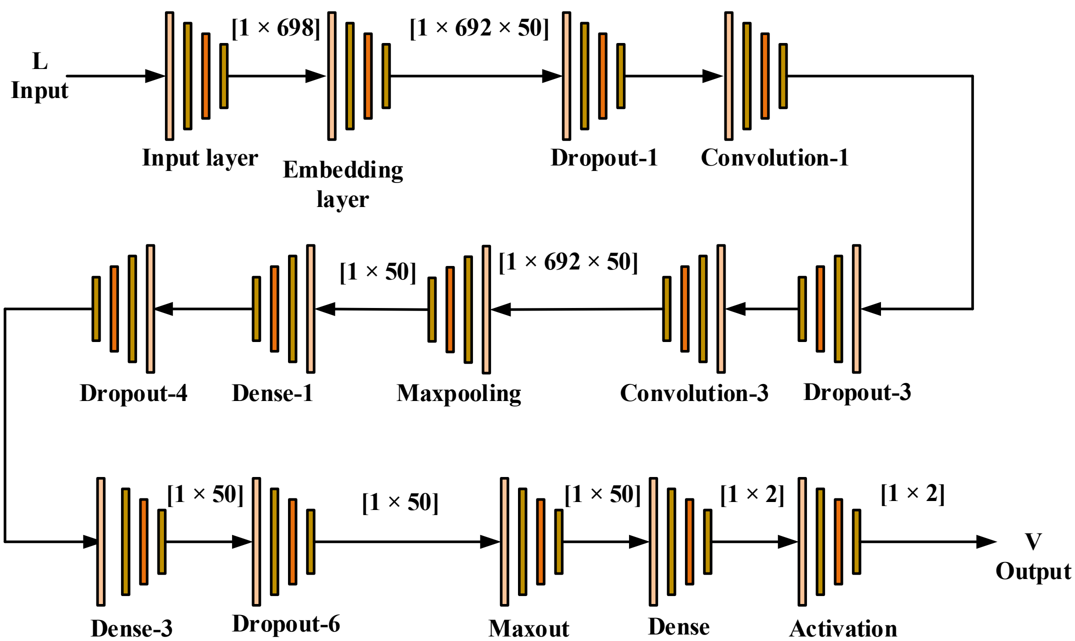

Figure 6: Structure of Deep Convolution Maxout Neural Network

The raw data vector is first fed to the Input layer in a structural feature size of $[1\times698]$. Subsequently, the data vector is projected onto the Embedding layer for extraction of context representation, converting the tensor into a multi-channel feature size of $[1\times692\times50]$. The flow of data further proceeds through a regularization layer composed of Dropout-1 and Convolution-1 to eventually enter a secondary layer block of Dropout-3 and Convolution-3. The output tensor of this stage will retain its previous size of $[1\times692\times50]$. To reduce spatial redundancy and emphasize the significant peak variations, the Maxpooling layer downsamples the sequence data size to $[1\times50]$. The resultant feature map undergoes feature mapping across two dense mapping stages with the inclusion of Dense-1 and Dropout-4 to form a first-layer block. The secondary feature mapping stage comprises Dense-3 and Dropout-6, keeping the dense feature size constant at $[1\times50]$. This vector becomes the input directly fed to the Maxout array of unit functions. Maxout unit has mathematical behavior expressed as follows:

$$S(L)=\max_{Z\in[1,n]}\left(W_{ij}+b_{ij}\right) \tag{13}$$

Where, $S(L)$ represents the output of the maxout unit feature map, L represent vector preceding layer, W represent weight and b represent bias factor are the trainable weight matrices across k pooled feature channel. The network outputs refined and smoothed final predictions for the battery's SOC and SOH that are highly resilient to vanishing gradients. It leads to high resistance against the vanishing gradient problem due to the output produced by the neural network.

Pooling through competitive activation channels, the maxout function outputs an optimized feature size of $[1\times 50]$, which is funneled through a final dense layer to scale the matrix to a dimension of $[1\times 2]$. Finally, the activation layer extract regression state estimation of output vector V with a size of $[1\times 2]$ represents the simultaneous output for SOC and SOH. DCMNN improves feature selection and gradient flow by approximating convex functions, thereby enabling the modeling of complex, nonlinear, and fast battery fluctuations. The DCMNN adapts itself to any behavior exhibited by the batteries through its adaptive activation layer, thus ensuring that there is stability during the learning process and accurate estimation of SOC and SOH.

3.4. Improved Salamander Optimization Algorithm - Sine Cosine Algorithm

The ISOA-SCA algorithm optimizes the hyperparameters of DA-ResBiLSTM-MaxNet, where the conventional ISOA is enhanced using SCA. The configuration for network in order to achieve higher accuracy in co-estimation through the minimization of an objective function based on MAE and RMSE statistical errors. The integration of cyclical mathematical operations of the SCA into the basic ISOA-SCA methodology makes the algorithm stronger by improving its global and local search capability in such a way that the deep learning model does not get trapped in local minima. Therefore, due to this optimization technique, the solution space, including thousands or even millions of combinations of parameters, is thoroughly analyzed to choose the best possible weight and bias matrices for DA-ResBiLSTM-MaxNet [30].

The algorithm starts with the initialization of candidate solutions, where each solution vector is considered a set of network structural parameters, mathematical weights and biases [31].

$$\vec{z}_d^{\,i}(t+1)=\begin{cases}\vec{z}_d^{\,i}(t)+\vec{a}\cdot\sin(r_3)\cdot\left|r_4\cdot\vec{z}^{\,*}(t)-\vec{z}_d^{\,i}(t)\right| & r_5<0.5\\ \vec{z}_d^{\,i}(t)+\vec{a}\cdot\cos(r_3)\cdot\left|r_4\cdot\vec{z}^{\,*}(t)-\vec{z}_d^{\,i}(t)\right| & r_5\geq 0.5\end{cases} \tag{14}$$

Where, $\vec{z}_d^{\,i}$ and $\vec{z}_d^{\,i}(t+1)$ represent the current and updated values of the d^{th} hyperparameter dimension for the i^{th} candidate solution, the index d sequentially tunes the learning rate $(d=1)$, $(d=2)$ represent the hidden layer size, $(d=3)$ represent the dropout rate, $(d\geq 4)$ represent the network weight and bias coefficients, $\vec{z}^{\,*}(t)$ represent towards the best configuration, the dynamic parameter $\vec{a}$ along with random variables r_3 r_4 and r_5, regulate the direction and step size of these adjustments using cyclical sine and cosine operations.

These parameters evolve during iterations to minimize a function depending on errors. The initial population matrix and its boundaries can be stated as follows:

$$Z=\begin{bmatrix} z_{1,1} & \cdots & z_{1,j} & \cdots & z_{1,\mathrm{M}}\\ \vdots & \ddots & \vdots & \iddots & \vdots\\ z_{i,1} & \cdots & z_{i,j} & \cdots & z_{i,m}\\ \vdots & \iddots & \vdots & \ddots & \vdots\\ z_{\mathrm{N},1} & \cdots & z_{\mathrm{N},j} & \cdots & z_{\mathrm{N},m}\end{bmatrix}_{\mathrm{N}\times m} \tag{15}$$

$$Z_i : z_{i,j}=lb_j+r\cdot\left(ub_j-lb_j\right) \tag{16}$$

Where, Z is the population matrix, N represent number of individuals in population, m represent number of decision variables which encompass the core network hyperparameters, weights and biases, Xi is the spatial position vector of the i^{th} individual, $z_{i,j}$ is its specific parameter coordinate along the j^{th} dimension, r represent uniformily distributed random number in the interval [0,1], lb_j and ub_j are lower and upper bounds of the j^{th} target variable. The ISOA-SCA designed to find the lowest possible error point across the network's joint state estimation outputs.

$$f = \min(RMSE)(MAE) \tag{17}$$

The fitness is designed to minimize both the RMSE and MAE prediction errors simultaneously.

$$F_i = \alpha \cdot \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(P_j - A_j\right)^2} + \beta \frac{1}{m}\sum_{j=1}^{m}\left|z_j - \hat{z}_j\right| \tag{18}$$

Where $\left(P_j - A_j\right)$ represents the standard deviation of residuals over N samples, F_i represents the learning rate, z_j represent average linear distance between target values and predictions $\hat{z}_j$ over m samples, α and represent the weight coefficient assigned to balance the impact of the RMSE and MAE errors, which are evaluated by comparing the network's predicted SOC and SOH outputs.

3.4.1: Phase 1: Regeneration (Exploration phase)

The exploration phase mimics salamander regeneration behavior to explore new search regions and avoid local minima. When an individual spatial position representing a specific configuration of network weights, biases or hyperparameters is deemed optimal, it undergoes a regenerative shift. This process alters the coordinate poorly performing individuals to explore unmapped spaces and escape local optima. The position recalculation mechanism is modeled as follows:

$$Z_{ij}^{P1} = z_{ij} + \left(r + \frac{T-t}{T}\right) \cdot \left(Z_{best,j} - I \cdot z_{i,j}\right) \tag{19}$$

Where Z_{ij}^{P1} represents the updated parameter coordinate of the i^{th} individual in the j^{th} dimension during the exploration phase and $Z_{best,j}$ is the best-known optimal configuration of the learning rate, batch size, hidden neurons, dropout rate, and optimizer parameters found up to iteration count t. T is the epoch, I represent a step multiplier randomly selected as either 1 or 2. If the newly calculated weight and bias configurations yield a lower value in the objective function, the individual's position vector updates via a greedy selection rule:

$$Z_i = \begin{cases} Z_i^{P1}, & F_i^{P1} \le F_i \\ Z_i, & else \end{cases} \tag{20}$$

Where, F_i^{P1} represent values in the new position.

3.4.2: Phase 2: Adaptive movement (Exploitation phase)

The second operational phase mimics the salamander's local search and fine-tuning strategies. This phase focuses on local search capabilities, executing fine-tuned adjustments around the promising weight and bias regions identified during the exploration stage. The new local solution coordinates are computed using small variations around the current position:

$$Z_{ij}^{P2} = z_{ij} + (1-2r) \cdot I \cdot \frac{ub_j - lb_j}{t} \tag{21}$$

Where, Z_{ij}^{P2} is the new position of the i^{th} individual in the exploitation phase, the refined local move yields a lower error value, and the position updates successfully:

$$Z_i = \begin{cases} Z_i^{P2}, & F_i^{P2} \le F_i \\ Z_i, & else \end{cases} \tag{22}$$

Through this two-phase iterative cycle, the ISOA-SCA converges toward a global minimum. The optimized hyperparameters include learning rate, hidden layer size, dropout rate and network weight coefficients are applied directly to the internal network layers. The integration of the cyclical trigonometric operators of the SCA into the ISOA, the ISOA-SCA, is able to avoid the pitfalls of premature convergence exhibited by the base algorithm by providing the benefits of enhanced global search and faster local search for obtaining the optimal network weights. This optimization ensures highly accurate, simultaneous tracking for both SOC and SOH outputs. The pseudo-code of ISOA-SCA is given in algorithm 1.

Algorithm 1: Pseudo-code of ISOA-SCA
1. Start 2. Initialize the population matrix Z (having N number of individuals and m decision variables) using Equation (15) 3. Specify lb_j and ub_j values for all parameter domain ranges 4. Calculate fitness/objective function value $F(X_i)$ for each individual using Equation (18) 5. Locate the best candidate position Z_{best} at present 6. Fix iteration counter $t = 1$ 7. While ($t <= T_{\max}$) do 8. Adjust the balancing coefficient r_1 9. Produce random control parameters and search directions using equation (14) 10. For each individual, do 11. Exploration Phase: Regeneration Approach 12. Update the position of the weight/bias through equation (19). 13. Apply the greedy selection/update condition in Equation (20) and evaluate fitness using Equation (18). 14. Exploitation Phase: Adaptive Movement Method 15. Update the position of the weight/bias using equation (21). 16. If improvements in MAE/RMSE errors occur, update the result using equation (22). 17. End for 18. Determine the optimal architecture Z_{best} obtained so far. 19. Increment the iteration counter $t = t + 1$. 20. End while 21. Output the optimal weights, biases and hyperparameters obtained. End ISOA-SCA Process

The proposed methodology has major contributions to battery state detection using sophisticated deep learning models in combination with an innovative hybrid meta-heuristic optimization strategy. First, the methodology leverages the use of a Dual Attention-based module consisting of two stages to extract sensor-based feature information and identify critical points of degradation, complemented by residual bi-directional LSTM structures, which allow detecting complex bi-directional trends without vanishing gradient issues. Besides that, DCMNN applies dynamic and piecewise linear activation functions, making it possible to deal with non-linear electrochemistry phenomena and voltage surges. To achieve maximum accuracy, the network parameters, such as structural hyperparameters, weights and biases, are tuned through hybrid ISOA-SCA with the addition of SCA. Consequently, it guarantees escape from local optima and rapid convergence. The proposed method delivers joint and accurate estimation of SOC and SOH simultaneously.

4. Results and discussion

Simulation of proposed system is performed via MATLAB/Simulink in order to depict the behavior of the system in different operational settings and different driving cycles. The effectiveness of joint optimization of the DA-ResBiLSTM-MaxNet model is validated and verified by comparing it with several benchmark systems through various statistical parameters, including RMSE, MAE, MSE and MAPE. The integrated approach consists of simulations and comparisons across various metrics demonstrates that proposed model outperforms conventional forecasting models in achieving high accuracy, faster convergence and noise robustness. The configuration details of the simulated battery cell and the benchmark network system are presented in Table 2 below.

Table 2. System parameters

Component	Technical Details
Device Name	CPU164
Full device name	CPU164.smg.local
Processor	Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz 3.60 GHz
Installed RAM	16.0 GB
Storage	932 GB HDD ST1000DM014-2UB10D
Graphics Card	Intel(R) HD Graphics 4600 (113 MB)
System Type	64-bit operating system, x64-based processor

4.1. Performance evaluation

4.1.1.RMSE

This statistical indicator emphasizes the magnitude of the differences in the calculation by squaring their values, followed by taking the square root in order to match the units of the errors with those of the original data. It reduces the error, that is closer to the possibile values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{23}$$

Where, n represent total number, y_i represent actual, $\hat{y}$ represent forecasted value and $\left(y_i - \hat{y}_i\right)^2$ squared error.

4.1.2.MSE

By squaring the deviations, the MSE places greater emphasis on large deviations than small ones since their effect is more pronounced when squared. However, square units of this make it difficult to interpret, hence the use of the RMSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y - \hat{y}_i\right)^2 \tag{24}$$

4.1.3.MAE

This is determined by calculating the average of absolute differences between the estimated and real values. To find the average difference, divide the sum of differences by number of data points.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{25}$$

Here, $\left|y_i - \hat{y}_i\right|$ is absolute value. The MAE is different from MSE because the latter gives more weight to larger differences than the former. The MAE is thus an appropriate measure when the objective is not to give more weight to large differences.

4.1.4.MAPE

The MAPE is calculated based on mean of the absolute differences expressed as percentages of the forecasted and actual values. To obtain the average relative error of the forecast, the sum of all absolute percentage differences must be observations.

$$MAPE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y - \hat{y}\right)^2} \tag{26}$$

Here, y represents actual battery capacity, whereas $\hat{y}$ represents the forecasted one.

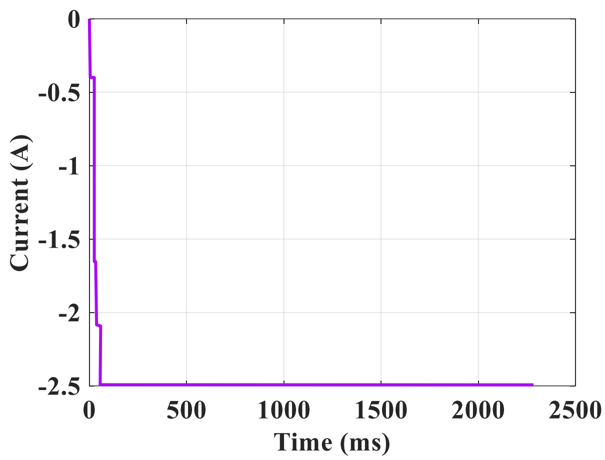

Figure 7. Analysis of current

Figure 7 shows the analysis of the current time graph, where current in Amperes (A) while Time in milliseconds (ms). At initial instant t=0 is characterized by an abrupt drop in the current signal that momentarily stops at -0.4 A and -2.1 A before falling to the lowest point. After about 60 ms from the start, the current signal reaches a stable level of -2.5 A and then remains constant for the rest of the period up to about 2300 ms.

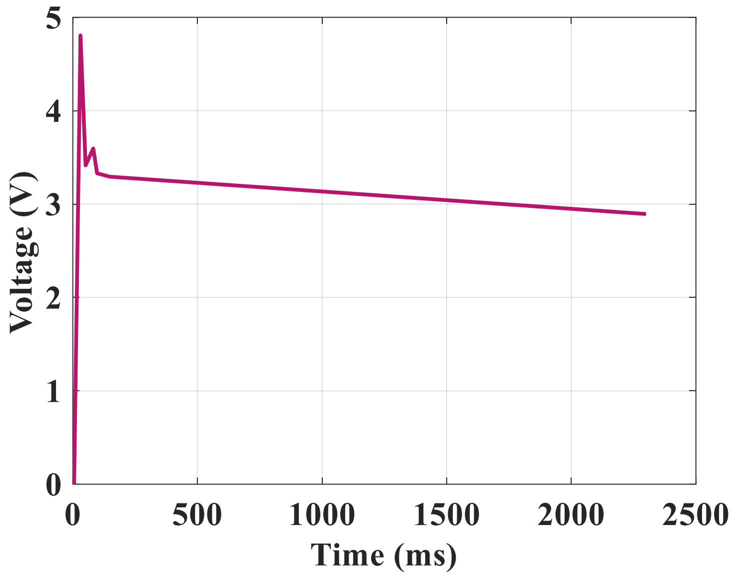

Figure 8. Analysis of voltage

Figure 8 shows the behavior of voltage over time, it starts at 0 V and there is a sudden rise in voltage to 4.8 V within 50 ms. After this rapid rise in voltage, it suddenly falls and shows another small peak of 3.6 V before stabilizing at 3.3 V. This steady state lasts until the end of the experiment period, i.e., about 2300 ms, during which time the voltage declines linearly from 3.3 V to 2.9 V.

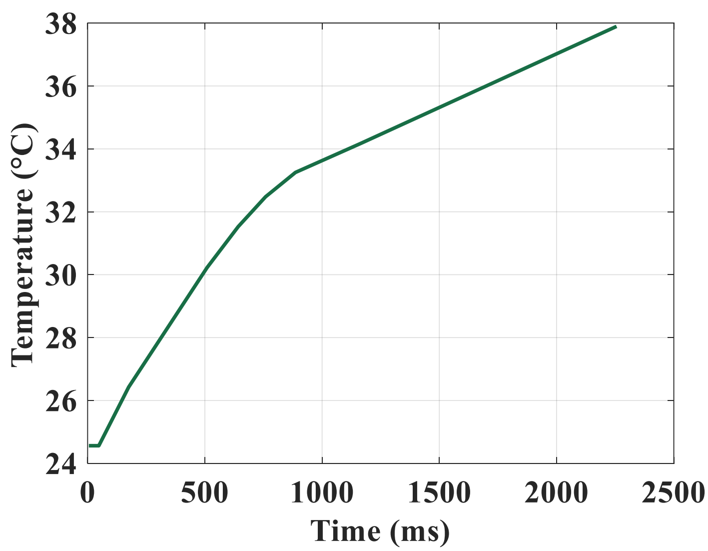

Figure 9. Analysis of temperature

Figure 9 shows the analysis of temperature over time. The Temperature is measured in °C, where the temperature holds steady at 24.5°C up to 60 ms. Afterward, a sudden and curved increase starts instantly. It attains a value of 33.3°C after 900 ms. The rate of temperature increase subsequently decreases. The temperature increases linearly thereafter. It finally peaks at 38°C at 2250 ms.

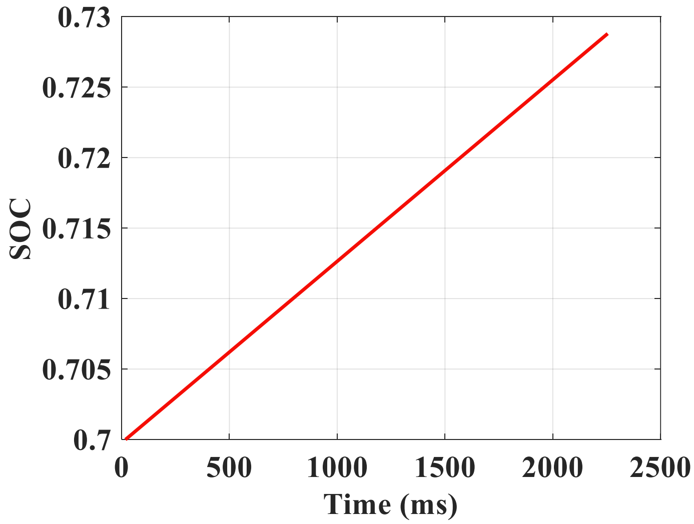

Figure 10. Analysis of SOC

Figure 10 shows the analysis of SOC behavior over time. The vertical y-axis measures the SOC ratio as a decimal number. The horizontal x-axis measures time in ms. At t = 0 ms, the SOC ratio is measured at 0.70. There is a perfectly linear increase throughout the process. It depicts a constant charging rate. At t = 2250 ms, the SOC ratio attains 0.729.

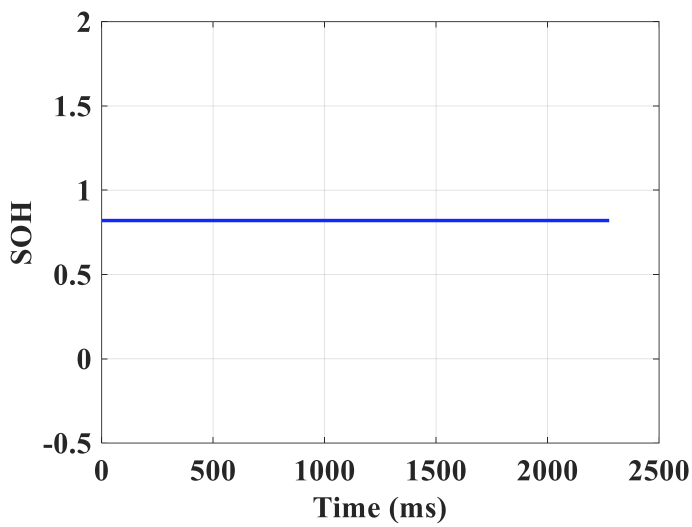

Figure 11. Analysis of SOH

Figure 11 shows the analysis of the SOH characteristic over time. The SOH remains constant during the whole experiment with a steady value of around 0.82. It indicates that there has been no change in the health of the battery over time.

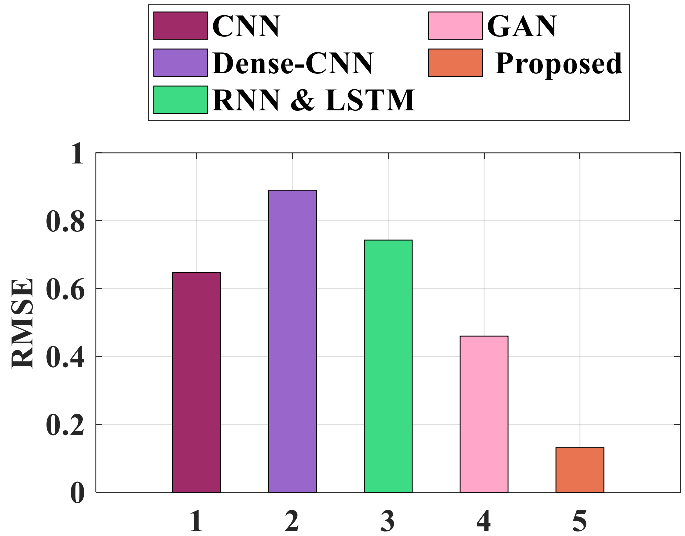

Figure 12. Analysis of RMSE

The performance analysis of RMSE in five different machine learning models is presented in Figure 12. The RMSE with a smaller value reflecting better results. The five models tested, which are CNN with an RMSE of 0.647, Dense-CNN with an RMSE value of 0.89, RNN & LSTM with an RMSE of 0.743 and GAN with RMSE of 0.46. Proposed model has better results with an RMSE value of 0.131.

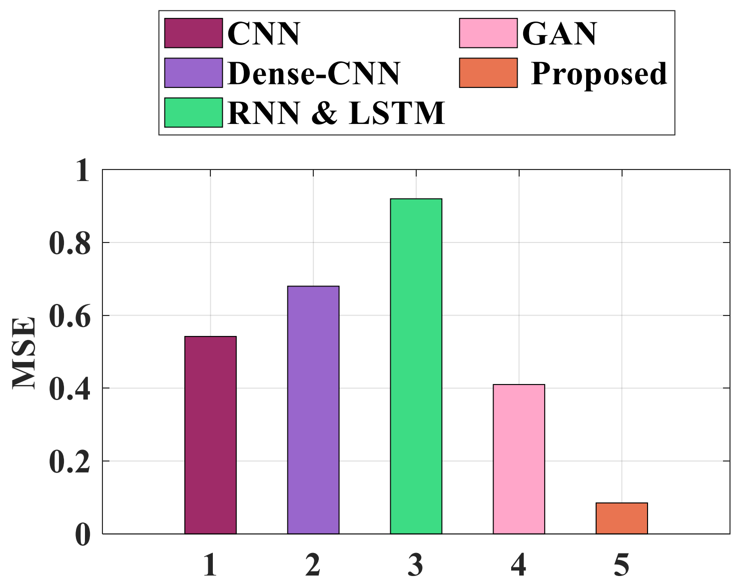

Figure 13. Comparative performance of MSE between the proposed method and existing techniques

The analysis of the MSE performance for each of the five models used is depicted in Figure 13. The MSE with lower values indicates better performance of the model. The x-axis denotes the various machine learning models. For instance, the MSE values for CNN, Dense-CNN, RNN & LSTM and GAN are 0.54, 0.68, 0.92 and 0.41, respectively. The Proposed model has the least error rate of about 0.085.

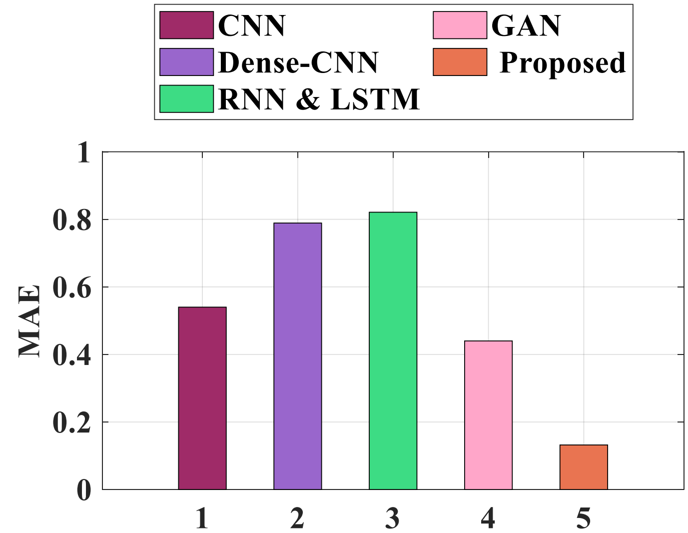

Figure 14. Analysis of MAE

Figure 14 shows the analysis of MAE performance across five distinct machine learning models. The y-axis represents magnitude of the MAE, whereby smaller the MAE, the greater its accuracy in predicting. The different machine learning models according to the color-coded legend: CNN has an MAE of 0.54, Dense-CNN has an MAE of 0.789, RNN and LSTM have the largest MAE of 0.821, while GAN has an MAE of 0.44. The comparative performance of Proposed MAE has smallest error of 0.132.

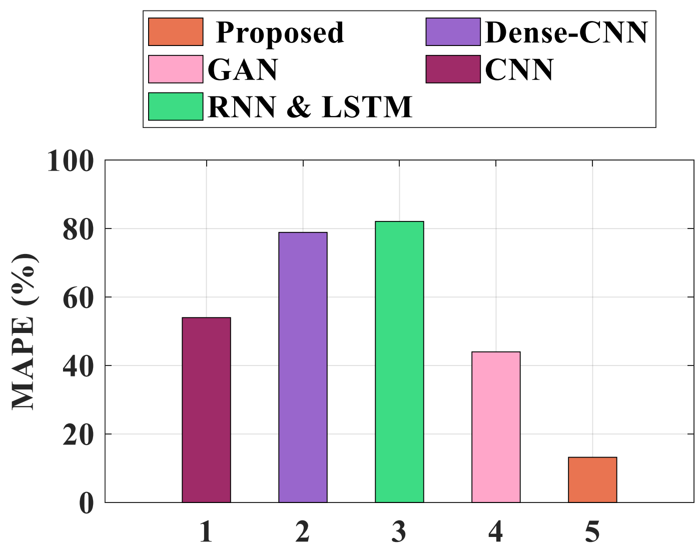

Figure 15. Analysis of MAPE

Figure 15 shows the comparison between five different Deep Learning algorithms concerning their prediction accuracy based on the MAPE. The traditional sequential models and the CNN demonstrate high levels of inaccuracy, with the RNN & LSTM models having the maximum inaccuracy of 82%, while the other two, Dense-CNN and traditional CNN, respectively, show 79% and 54%. Although GAN has lower margins of percentage errors compared to the three models at 44%, it still falls short behind the proposed model has higher accuracy rate of 13.5%.

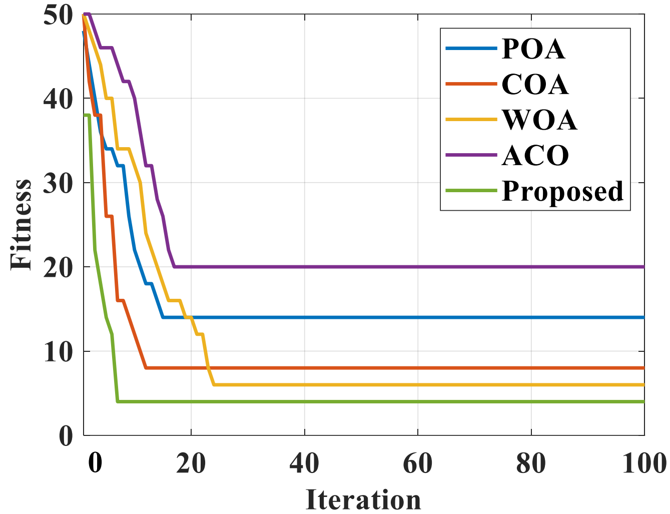

Figure 16. Analysis of the convergence plot

Figure 16 shows the analysis of the convergence plot optimization performance of five different algorithms over 100 iterations. The vertical y-axis represents the fitness value, which means the lower the fitness value, the better it is and represents an optimal solution. The horizontal x-axis is the number of iterations ranging from 0 to 100. The approaches considered begin with high values of fitness, which approach 50. The ACO approach converges by 17 iterations to provide the highest final value of fitness as 20. The POA algorithm converges near 15 iterations to reach the fitness value of 14. COA algorithm decreases the initial iterations and reaches minimum fitness value of 8 after iteration 11. The WOA algorithm converges slowly in comparison but provides the lowest fitness value of 6 after 24 iterations. The proposed algorithm approach converges very quickly in the beginning and reaches the fitness value of approximately 4 after 5 iterations. Therefore, the performance of the proposed algorithm shows that it converges fast and provides optimum results. Table 3 shows the performance comparison.

Table 3. Performance comparison

Methods	RMSE	MAE	MSE	MAPE (%)
CNN	0.647	0.54	0.542	54
Dense-CNN	0.89	0.789	0.68	79
RNN & LSTM	0.743	0.821	0.92	82
GAN	0.46	0.44	0.41	44
Proposed	**0.131**	**0.132**	**0.085**	**13.5**

The predictive accuracy for the five deep learning techniques used, assessed based on four well-known standard errors. From the quantitative analysis performed, it is evident that the proposed method scores excellent accuracy with the least error in terms of RMSE of 0.131, MAE of 0.132, MSE of 0.085 and MAPE of 13.5%. However, the classical networks experience poor performance with the RNN & LSTM networks demonstrating maximum inaccuracy, indicated by a MAPE of 82% and an MSE of 0.92. It is followed by the Dense-CNN network with a MAPE of 79% and the CNN network with a MAPE of 54%. Though the GAN network has improved its tracking performance, as seen in the reduction of its RMSE to 0.46 and MAPE to 44%, it does not perform comparably with the proposed method.

4.3. Discussion

The primary results of these studies show that the development of EV battery state estimation techniques demonstrates an emergence from the initial use of ensemble machine learning toward the implementation of hybrid deep learning sequential networks. At same time, ensemble learning can successfully capture non-linear relationships between data inputs; its inability to account for temporal relationships led to the employment of recurrent neural networks such as bi-directional RNN-LSTMs. For accurate extraction purposes, hybrid spatial-temporal architectures involving combinations of CNN, LSTM and GRU blocks are being used for tracking battery state dynamics. Yet, serious architectural limitations exist, since classical deep learning structures are vulnerable to the effects of data leakage and dead neurons caused by hard activation functions, such as the ReLU activation. At the same time, although modern systems support privacy-preserving and federated computing approaches, these structures are restricted by the requirement of setting manually optimized hyperparameters that get stuck in local optima during training. However, traditional deep learning architectures are prone to data leakage and dead neurons caused by hard processing dependencies, whereas modern CNN are constrained by manually optimized overfitting parameters, which easily get trapped in local optimizations. To overcome these limitations, the proposed DA-ResBiLSTM-MaxNet–ISOA-SCA framework provides highly accurate, real-time battery level monitoring with minimal error by eliminating the dead neuron bottleneck and avoiding local training optimizations. Table 4 shows the comparison of error metrics.

Table 4. Comparison of error metrics

Author / Ref.	**Method Used**	**RMSE (%)**	**MAE (%)**
Alwesabi et al. [18]	WaOA-CNN-BiLSTM-AM	0.0203	0.0163
Kant et al. [19]	biRNN-LSTM	1.8	1.6
Sakr et al. [20]	ML	0.0458	-
Rajasekaran and Venkatanarayanan [21]	GANs	0.0113	0.0075
Naresh and Sriram [22]	PPDNN	0.094	0.081
Alharbi *et al.* [23]	CNN, LSTM	0.666	0.626
Proposed method	**DA-ResBiLSTM-MaxNet-ISOA-SCA**	**0.131**	**0.132**

The prediction precision for seven techniques in terms of RMSE and MAE. It is evident from the quantitative analysis of this table that the proposed DA-ResBiLSTM-MaxNet–ISOA-SCA framework has been able to provide reliable results in terms of accurate and stable tracking. The values of RMSE and MAE are 0.131% and 0.132%, respectively, which show that there is a significant enhancement in terms of performance compared to conventional sequential structures. The CNN-LSTM architecture has an RMSE of 0.666% and an MAE of 0.626%, while the biRNN-LSTM has higher tracking errors in the form of an RMSE value of 1.8% and an MAE of 1.6%. Although specialized approaches such as GANs and WaOA-CNN-BiLSTM-AM have lower errors, the framework under consideration performs better than conventional models such as PPDNN, which gives 0.094% RMSE and 0.081% MAE, while ML provides 0.0458% RMSE.

5. Conclusion

This work presents how non-linearity, temporal dependencies and hyperparameter sensitivities were managed in SOC and SOH prediction in the EV's batteries through the developed groundbreaking DA-ResBiLSTM-MaxNet-ISOA-SCA approach. The developed deep network architecture ensured the elimination of the dead neurons bottleneck problem by integrating a residual bidirectional LSTM architecture with a combination of dual attention mechanism and max-out activation layers, thus ensuring that the necessary feature gradients were retained throughout all cycling processes. Additionally, the employment of the novel ISOA-SCA ensured that there was an optimization of network structural weights and biases, thus avoiding local minima problems and ensuring faster convergence of the training process. Through empirical validation with current state-of-the-art methods, it became clear that the hybrid optimizer and network architecture were able to deliver tracking with minimum errors an RMSE of 0.131 %, MAE of 0.132 %, MSE of 0.085%, and MAPE of 13.5 %. Overall, the proposed method is an accurate simultaneous tracking engine for SOC and SOH. Future research will focus on validating this hybrid architecture in various battery chemistries and real-world driving dynamics to assess its scalability.

Reference

1. Liao, W., Chen, Z., Li, P., Tan, J., Li, H., & Zhao, C. (2025). Enhanced battery health monitoring in electric vehicles: A novel hybrid HBA-HGBR model. *Journal of Energy Storage*, *110*, 115316.
2. Haraz, A., Abualsaud, K., & Massoud, A. M. (2025). Ensemble learning for precise state-of-charge estimation in electric vehicles' lithium-ion batteries considering uncertainty. *IEEE Access*.
3. Xing, X., Yan, T., & Xia, M. (2025). Early prediction of battery life using an interpretable health indicator with evolutionary computing. *Reliability Engineering & System Safety*, *260*, 110980.
4. Huang, Y., Yu, H., Lai, Y., Zhu, L., Huang, C., & Dai, H. (2025). State of Health Estimation and Remaining Useful Life Prediction of Lithium-ion Batteries Based on Optimized Echo State Network with Early Data Feature. *IEEE Transactions on Instrumentation and Measurement*.
5. Wang, L., Yang, T., & Hu, B. (2025). A battery state of health Estimation method for real-world electric vehicles based on physics-informed neural networks. *IEEE Sensors Journal*.
6. Kant, A., Kumar, M., & Sihag, S. (2025). Enhanced battery management system for Li-ion batteries using a hybrid machine learning model for accurate state of health prediction. *International Journal of Ambient Energy*, *46*(1), 2503276.
7. Huang, R., Chen, Y., & Liu, C. (2025). Revealing state-feature dependencies in dynamic degradation: an exofeature-aware transformer for battery state-of-health prediction. *IEEE Transactions on Consumer Electronics*.
8. Jing, H., Hu, J., Ou, S. S., Lv, Z., Lyu, R., & Zhao, J. (2025). Scalable and generalizable deep learning for battery state of health estimation in on-road electric vehicles. *Journal of Energy Chemistry*.
9. Hemavathi, S. (2025). Lithium-ion battery state of health Estimation using intelligent methods. *Franklin Open*, *10*, 100237.
10. Al-Alawi, M. K., Jaddoa, A., Cugley, J., & Hassanin, H. (2025). Enhanced state of charge estimation through Cluster-Based Learning Model: Impact study on degradation and profitability of second-life electric vehicle batteries. *Journal of Energy Storage*, *117*, 116071.
11. Liu, J., He, L., Zhang, Q., Xie, Y., & Li, G. (2025). Real-world cross-battery state of charge prediction in electric vehicles with machine learning: Data quality analysis, data repair and training data reconstruction. *Energy*, 138322.
12. Bhat, A. N., Singh, M., & Kasana, S. S. (2026). Deep reinforcement learning for vehicle-to-grid energy management with reward decomposition and battery health preservation. *Sustainable Energy, Grids and Networks*, 102184.
13. Vedhanayaki, S., & Indragandhi, V. (2025). Bayesian optimized support vector regression with a Gaussian kernel for accurate prediction of the state of health of lithium-ion batteries used for electric vehicle applications. *Global Energy Interconnection*.
14. Pratheeba, C., & Sukumar, P. (2025). Hybrid HGRN-SCSO technique for enhanced prediction of remaining useful life in EV batteries. *Electrical Engineering*, *107*(4), 5311-5323.
15. Poh, W. Q. T., Xu, Y., Liu, W., & Tan, R. T. P. (2025). Momentary informatics-based data-driven estimation of lithium-ion battery health under dynamic discharging currents. *Journal of Power Sources*, *629*, 236041.
16. Jasphin, E.T.J., Joice, C.S. Weed Classification and Crop Health Monitoring in Microclimatic Conditions Using Thermal Image Analysis and Deep Learning Algorithms. J Plant Growth Regul 44, 2247–2263 (2025). https://doi.org/10.1007/s00344-024-11542-1
17. A. Venkataramana, K. S. Kumar, N. Suganthi and R. Rajeswari, "Prediction of Brinjal Plant Disease Using Support Vector Machine and Convolutional Neural Network Algorithm Based on Deep Learning," in Journal of Mobile Multimedia, vol. 18, no. 3, pp. 771-788, May 2022, doi: 10.13052/jmm1550-4646.18315.
18. Alwesabi, K., Farh, H. M. H., Ibrahim, A. W., Al-Shamma'a, A. A., Al-Wagih, K. S. M., Al-Ashmoery, Y., & Shafik, M. B. (2026). Hybrid deep learning framework for accurate state-of-charge estimation of electric vehicle batteries under multi-temperature conditions using transfer learning and WaOA optimization. Energy Reports, 15, 109282.
19. Kant, A., Kumar, M., & Sihag, S. (2025). Prediction of electric vehicle battery state of health estimation using a hybrid deep learning mechanism. *International Journal of Green Energy*, *22*(10), 2065-2078.
20. Sakr, H. A., Eladl, A. A., & El-Afifi, M. I. (2025). Leveraging IoT-enabled machine learning techniques to enhance electric vehicle battery state-of-health prediction. *Journal of Energy Storage*, *120*, 116409.
21. Rajasekaran, E., & Venkatanarayanan, S. (2025). State-of-Health (SoH) prediction for electric vehicle battery systems using GAN-based models with triple attention mechanisms. *Journal of Energy Storage*, *134*, 118143.
22. Naresh, V. S., & Sriram, V. S. (2025). PPDNN-SoH: privacy-preserving deep neural network for sustainable electric vehicle battery state of health prediction. *Neural Computing and Applications*, *37*(32), 27445-27473.
23. Alharbi, T., Umair, M., & Alharbi, A. (2025). Lithium-ion battery state of health degradation prediction using deep learning approaches. *IEEE Access*, *13*, 13464-13481.
24. Pandiaraj, P., & Natarajan, S. K. V. (2025). Optimized deep convolutional neural networks for SOC and SOH estimation in electric vehicles using the orangutan algorithm. *Journal of Energy Storage*, *134*, 118140.

25. Jafari, S., & Byun, Y. C. (2022). Prediction of the battery state using the digital twin framework based on the battery management system. *IEEE Access*, *10*, 124685-124696.
26. Khan, F., Yu, X., Yuan, Z., & Rehman, A. U. (2023). ECG classification using a 1-D convolutional deep residual neural network. *Plos one*, *18*(4), e0284791.
27. He, Y., Wen, C., & Xu, W. (2025). Residual life prediction of SA-CNN-BiLSTM aero-engine based on a multichannel hybrid network. *Applied Sciences*, *15*(2), 966.
28. He, J., Yang, H., Zhou, S., Chen, J., & Chen, M. (2022). A dual-attention-mechanism multi-channel convolutional LSTM for short-term wind speed prediction. *Atmosphere*, *14*(1), 71.
29. Peta, J., & Koppu, S. (2022). An IoT-based framework and ensemble-optimized deep maxout network model for breast cancer classification. Electronics, 11(24), 4137.
30. Hamadneh, T., Batiha, B., Al-Refai, O., Ibraheem, I. K., Smerat, A., Werner, F., ... & Eguchi12, K. (2025). Salamander optimization algorithm: a new bio-inspired approach for solving optimization problems. *Int. J. Intell. Eng. Syst.*, *18*(7), 550-561.
31. Cao, Y., Wang, W., & He, Y. (2025). Predicting Glossiness of Heat-Treated Wood Using the Back Propagation Neural Network Optimized by the Improved Whale Optimization Algorithm. *Forests*, *16*(5), 716.