# An Adaptive Kalman-Guided Soft Sensor Using Feedforward Neural Networks for SOC Estimation in Lithium-Ion Batteries

Mostafa Mahdi Yousef
*Dept. of Electrical Engineering and Computer Science*
*York University*
Toronto, Canada
mostafa6@yorku.ca

Mohammad Shaterabadi
*Dept. of Electrical Engineering and Computer Science*
*York University*
Toronto, Canada
shater94@yorku.ca

Houshang Karimi
*Dept. of Electrical Engineering and Computer Science*
*York University*
Toronto, Canada
hkarimi6@yorku.ca

***Abstract*—Accurate State-of-Charge (SOC) estimation is crucial for ensuring the battery's safe operation and prolonged lifespan, but it remains a challenge due to sensor noise and system nonlinearity. This paper proposes a novel covariance-adaptive hybrid approach that interprets neural network outputs as virtual measurements and dynamically integrates them with the *a priori* state predictions from a Discrete Kalman filter to yield refined *a posteriori* estimates. The introduced adaptive scheme uses exogenous inputs derived from spectral analysis to provide past information to the Feedforward Neural Network (FNN) to perform soft sensing and generate network-aided measurements. The novel processing occurs after this step by adaptive integration with Kalman filter results. The developed hybrid architecture is tested on real experimental battery datasets for LG 18650 HG2, and the results prove its superior performance against existing state-of-the-art algorithms. Experimental results averaged over twenty Monte Carlo trials demonstrated nearly 28% and 39% improvement in the mean absolute error and root mean square error, respectively, compared to the baseline results.**

***Index Terms*—Battery Management System, State of Charge Estimation, Covariance-Adaptive Hybrid Approach, Discrete Kalman Filter, Neural Network Virtual Measurements**

## I. Introduction

Battery technology plays a pivotal role in reducing carbon footprints in the transportation sector [1]. Lithium-ion batteries have been widely adopted in Electric Vehicles (EVs) due to their high energy density, low maintenance, and limited environmental impact [2]. The Battery Management System (BMS) ensures the safe and prolonged operation of the battery [3]. State-of-charge (SOC) is one of the key internal parameters estimated by the BMS and serves as an input to various control functions such as cell balancing and protection [4]. The state of charge cannot be measured directly and must be estimated from observable signals, but SOC estimation during operation still remains a critical research challenge [5]. SOC estimation methods can be divided into three main categories: direct, model-based, and data-driven methods [2]. The Coulomb Counting Method (CCM) and OCV–SOC look-up tables are open-loop direct methods and computationally lightweight, but they suffer from inherent drawbacks. As CCM is an open-loop scheme, it is prone to accumulated errors due to inaccurate initialization and integration errors. Look-up tables are impractical for online applications, as the battery should rest until the terminal voltage reflects the true open circuit voltage [6].

Model-based schemes typically use Kalman Filter (KF) variants [7]–[10] due to their low latency and adaptability during dynamic conditions. However, their performance heavily relies on accurate parameter characterization and fine-tuned Kalman filter parameters (i.e., initial guesses and noise covariance matrices) [11]. The parameters of an electrical equivalent circuit are a function of temperature, c-rate, and SOC [12], making it difficult to capture the nonlinearities of the battery's electrochemical behavior [13]. On the other hand, data-driven estimation strategies excel at mapping a nonlinear relationship between measured signals and SOC and do not need knowledge of the battery's parameters and electrochemical behavior. In practice, lightweight architectures like Feedforward Neural Networks (FNNs) are preferred over complex networks, especially for embedding in resource-constrained BMS [14]. Additionally, more complicated neural networks do not necessarily outperform a light FNN. A study in [15] compared Long-term Short-term Memory (LSTM) neural networks with a lightweight and temporal-aware FNN and demonstrated that FNN, in most cases, outperforms LSTM or achieves the same accuracy with significantly lower computational burden. Although neural networks capture nonlinearities and are an attractive option for real-world implementation, they lack adaptability in changing conditions and may generalize poorly outside the trained domain [14].

Multiple studies have explored hybrid approaches with different implementation techniques [16]–[18]. In [16], current, temperature, estimated SOC, and recursive exponential filters of current (referred to as polarization states) are fed to a feedforward neural network to estimate terminal voltage. Then, the terminal voltage inferred from the neural network is used in an extended Kalman filter (EKF) as the model-based voltage prediction, eliminating the need for parameter characterization. The inclusion of polarization states enables temporal dependencies to be captured but increases the input dimensionality, which may raise computational complexity. In [17], a radial basis function neural network (RBFNN) is trained offline using previous voltage, current, and estimated SOC to predict the next terminal voltage. This neural network serves as the nonlinear observation model within an adaptive extended Kalman filter framework. SOC is updated using Coulomb counting and refined using voltage error feedback, enabling real-time correction without explicit battery modeling or parameter extraction. The inclusion of past voltage as a network input provides memory, reducing the need for additional filtering structures or multi-scale inputs. In [18], there is no need for parameter characterization, as SOC is directly estimated by a feedforward neural network based on current, voltage, temperature, and moving averages. This estimate is treated as a virtual measurement and corrected using a linear Kalman filter that also incorporates Coulomb counting. While this approach simplifies implementation and improves accuracy over standalone FNN, it lacks adaptive mechanisms to adjust model uncertainty or track system dynamics in real-time.

Unlike conventional hybrid estimators, this paper introduces a hybrid technique that significantly enhances adaptability by exploiting the synergy of neural networks and Kalman filtering. In this architecture, the need for parameter characterization is eliminated, thanks to the neural networks. The suggested hybrid scheme fills the adaptability weakness, and the closed-loop fusion ensures a high level of adaptability and performance in real-world BMS applications, all without using complex, computationally heavy algorithms. Therefore, the key contributions of this work include:

- **Using the neural network as a soft sensor:** FNN is used as an inferential sensor to adaptively adjust the output of the CCM, removing accumulated errors. This fusion eliminates the need for parameter estimation and improves the adaptability of purely data-driven approaches while ensuring implementation simplicity.
- **Lightweight and hardware-efficient architecture:** Designed with computational simplicity, making it ideal for real-time deployment in embedded battery management systems with no need for recurrent neural networks.
- **Robust performance across conditions:** Achieves significant error reductions compared to strong baseline models, validated across multiple temperatures and realistic driving cycles to ensure robust generalization and practical applicability.

This paper is organized as follows: Section II describes the proposed methodology, which is divided into three main steps. Section III introduces the training dataset and initializes the estimator, followed by a presentation of the experimental results. Finally, Section IV summarizes the findings and discusses the future directions of this work.

## II. Proposed Methodology

The proposed methodology has three major stages. The first one involves incorporating additional inputs to the FNN to provide temporal dependencies, which are not inherently captured, unlike recurrent neural networks. The next crucial step is leveraging a powerful feedforward neural network to accurately map the nonlinear relationship between feature vectors and SOC as the target label. Finally, the trained neural network's outputs will be treated as synthetic measurements followed by an adaptive correction step, ensuring the model's robustness and accuracy. These steps are detailed in the following subsections.

### *A. Neural Network-Aided Observations*

This section explains the first two stages of the methodology mentioned earlier. Firstly, exogenous inputs should be created to address FNN's lack of inherent memory cells. Spectral analysis of the voltage and current signals revealed that the energy concentration occurs in low frequencies, less than 5 mHz, proving the effectiveness of low-pass filters in capturing temporal dependencies. Therefore, due to its efficacy, the Butterworth filter is specifically designed for pre-processing the raw data [15]. In Equation 1, the filter design determines $b_i$ and $a_j$ coefficients, where $M$ and $N$ denote the filter order.

$$\mathbf{U}_{\text{filt}}(k) = \sum_{i=0}^{M} b_i\, \mathbf{U}(k-i) - \sum_{j=1}^{N} a_j\, \mathbf{U}_{\text{filt}}(k-j) \tag{1}$$

where

$$\mathbf{U}(k) = \begin{bmatrix} V_k & I_k \end{bmatrix},$$
$$\mathbf{U}_{\text{filt}}(k) = \begin{bmatrix} V_{\text{LTD}} & V_{\text{STD}} & I_{\text{LTD}} & I_{\text{STD}} \end{bmatrix}.$$

After capturing the long-term and short-term dependencies, network-aided observations are provided by a feedforward neural network shown in Figure 1.

In the second stage, forward propagation occurs during training, where inputs pass through the network, and corresponding outputs are produced. Equation (2) describes the forward propagation in an FNN for SOC estimation [4]. The activation functions used in this study, defined in Equations (3), (4), and (5), were selected based on [19]. After each batch, backpropagation is performed to compute gradients and update weights in order to minimize the loss function defined in Equation (6), where $S\hat{O}C(k)$ is the estimated SOC and $SOC(k)$ is the ground truth value. In this study, the neural network is a soft sensor, and $S\hat{O}C(k)$ acts as an artificial measurement, not the final prediction.

$$\text{SOC}_i = f_{\text{FNN},i}\left(\sum_j W_{j,i} O_j + \theta_{j,i}\right) \tag{2}$$

$$y = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

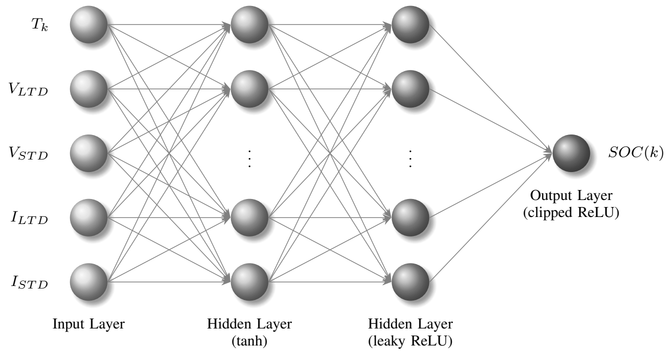


Fig. 1: Feedforward neural network for SOC estimation.

$$y = \text{LeakyReLU}(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0 \end{cases} \tag{4}$$

$$y = \text{ClippedReLU}(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases} \tag{5}$$

$$J = \frac{1}{K} \sum_k \left( SOC(k) - S\hat{O}C(k) \right)^2 \tag{6}$$

### *B. Adaptive Model-Based Correction*

This section describes the final stage of the proposed hybrid approach, which enhances estimator robustness. Specifically, the Coulomb counting method is used to construct a physical state-space model. However, due to its sensitivity to initial conditions and accumulation of integration errors, CCM alone is insufficient. To mitigate this, a discrete Kalman filter is employed to correct these errors recursively. Nevertheless, both process and measurement models are subject to uncertainties caused by sensor noise, environmental variations, battery aging, and capacity fading. The continuous-time Coulomb Counting method expresses SOC as:

$$\text{SOC}(t) = \text{SOC}(t_0) - \frac{\mu}{C_\text{n}} \int_{t_0}^{t} I(\tau)\, \mathrm{d}\tau \tag{7}$$

Differentiating (7) yields the corresponding differential equation:

$$\frac{\mathrm{d}\,\text{SOC}(t)}{\mathrm{d}t} = -\frac{\mu}{C_\text{n}} I(t) \tag{8}$$

The forward Euler method is commonly used to discretize such first-order systems:

$$x_{k+1} = x_k + \Delta t \cdot f(x_k, t_k) \tag{9}$$

Applying (9) to the SOC differential equation (8), the discrete-time representation is derived as follows:

$$\text{SOC}[k] \approx \text{SOC}[k-1] - \frac{\mu\,\Delta t}{C_\text{n}} \cdot I[k-1] \tag{10}$$

As mentioned earlier, the CCM limitations necessitate the use of adaptive strategies capable of capturing dynamic behaviors in real time under varying operating conditions. To improve robustness under realistic noise conditions, the feedforward neural network is trained using multiple noisy versions of the original sensor recordings. These versions are generated by introducing various combinations of offset and gain errors from sensors into the measurements. While this practice enhances the network's tolerance to sensor imperfections, it still only partially accounts for hidden or unmodeled dynamics. Consequently, neither CCM nor FNN alone can ensure robust SOC estimation. To address this limitation, the proposed hybrid estimator adaptively updates the measurement noise covariance using real-time innovation analysis. The resulting Kalman-based hybrid estimation equations are summarized as follows:

*1) Prediction Step:* The SOC state and covariance are predicted using the discrete-time model:

$$\hat{x}_{k|k-1} = \hat{x}_{k-1|k-1} - \frac{\mu\,\Delta t}{C_\text{n}} I_{k-1} \tag{11a}$$

$$P_{k|k-1} = P_{k-1|k-1} + Q \tag{11b}$$

*2) Measurement Step:* The measurement $y_k$ is generated by the trained neural network:

$$y_k = f_{\text{FNN}}(\boldsymbol{U}_{\text{filt},k}; \boldsymbol{\theta}) \tag{12}$$

*3) Adaptive Measurement Noise Covariance:* To reflect dynamic uncertainty levels, the measurement noise covariance $R_k$ is updated adaptively based on the operational condition [20]:

*During charging* $(I_k > 0)$:

$$R_k(\alpha_R) = (1-\alpha_R) R_{k-1} + \alpha_R \left( y_k - \hat{x}_{k|k-1} \right)^2 \tag{13}$$

*During discharging or rest* $(I_k \leq 0)$:

$$R_k(\alpha_R) = R_{\text{discharge}} \tag{14}$$

*4) Kalman Gain Computation:* The Kalman gain $K_k$ is calculated as:

$$K_k = P_{k|k-1} H_k^\top \left( H_k P_{k|k-1} H_k^\top + R_k(\alpha_R) \right)^{-1} \tag{15}$$

*5) Correction Step:* The state estimate and covariance are corrected based on measurement innovation:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left( y_k - H_k \hat{x}_{k|k-1} \right) \tag{16a}$$

$$P_{k|k} = (I - K_k H_k)\, P_{k|k-1} \tag{16b}$$

Figure 2 illustrates the closed-loop workflow: temporal-aware inputs are first fed into the FNN, which predicts a synthetic measurement. Meanwhile, the process model uses the measured current to compute *a priori* estimates. The discrepancy between the FNN output and the CCM prediction (the innovation) is then used for adaptive correction. The final SOC estimate is forwarded to the battery management system for downstream tasks such as energy management or charger control in electric vehicles. Table I summarizes the key symbols used in this section and describes their definition concisely.

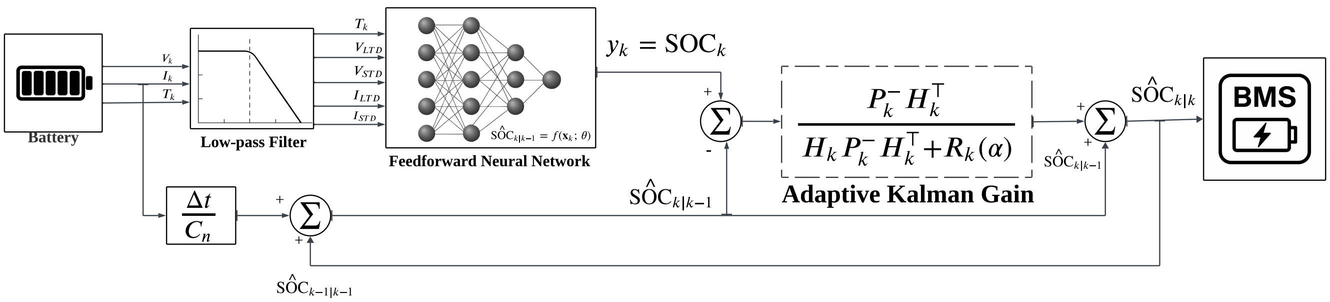


Fig. 2: Hybrid SOC estimation architecture combining a feedforward neural network (FNN) for a priori prediction with an adaptive Kalman filter for residual-based correction.

| Symbol | Description |
|---|---|
| $V_{\text{LTD}}$ | Voltage representing long-term dependencies |
| $I_{\text{LTD}}$ | Current representing long-term dependencies |
| $V_{\text{STD}}$ | Voltage representing short-term dependencies |
| $I_{\text{STD}}$ | Current representing short-term dependencies |
| $\boldsymbol{U}(k)$ | Raw input vector: $[V_k,\ I_k]$ |
| $\boldsymbol{U}_{\text{filt}}(k)$ | Filtered input vector: $[V_{\text{LTD}},\ V_{\text{STD}},\ I_{\text{LTD}},\ I_{\text{STD}}]$ |
| $\hat{x}_{k\|k-1}$ | Prior SOC estimate at time $k$ |
| $\hat{x}_{k\|k}$ | Corrected SOC estimate at time $k$ |
| $y_k$ | Measured SOC (FNN output) |
| $I_k$ | Battery current (positive for discharge) |
| $C_{\text{n}}$ | Nominal battery capacity [Ah] |
| $\mu$ | Coulombic efficiency (Assumption: $\mu \approx 1$) |
| $\Delta t$ | Sampling interval |
| $P_{k\|k-1}$ | Prior error covariance |
| $P_{k\|k}$ | Corrected error covariance |
| $Q$ | Process-noise covariance |
| $R_k(\alpha_R)$ | Adaptive measurement-noise covariance |
| $\alpha_R$ | Adaptation rate for $R_k$ |
| $K_k$ | Kalman gain |
| $H_k$ | Measurement model matrix ($H_k = 1$) |
| $J$ | MSE loss function for training |
| $O_j$ | Output of neuron $j$ from previous layer |
| $W_{j,k}$ | Weight connecting neuron $j$ to neuron $k$ |
| $\theta_{j,k}$ | Bias term for neuron $k$ |
| $f_{\text{FNN}}(\cdot)$ | FNN Activation function |
| $\boldsymbol{\theta}$ | Trainable weights and biases of FNN |

TABLE I: Summary of key symbols and their definitions.

## III. Experimental Results

### A. Training Dataset

The training dataset in this study is a publicly available dataset for an LG 18650GH2 lithium-ion battery cell [19]. To generate the dataset, multiple isothermal experiments are conducted on a brand-new battery cell based on standard drive cycles to mimic real-world scenarios. Table II summarizes important information about the battery cell and data generation process. Figure 3 demonstrates how the data is divided into training, validation, and test portions to facilitate a generalized training and bias-variance trade-off.

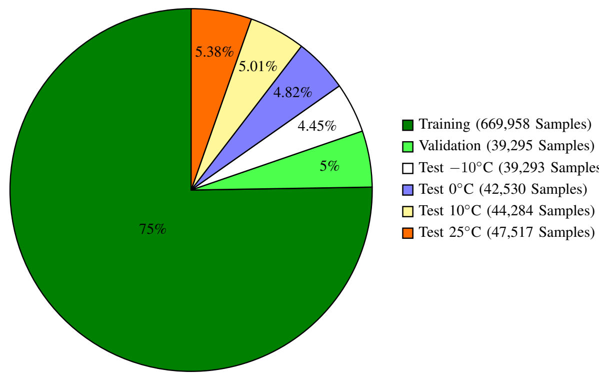


Fig. 3: Sample distribution across training, validation, and test sets at four ambient temperatures ($-10$°C to 25°C), enabling comprehensive evaluation of thermal robustness.

$$\text{SOC} = \frac{\text{Supplied or Consumed AH}}{\text{Nominal AH}} \tag{17}$$

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{18}$$

| Item | Details |
|---|---|
| Battery Type | LG 18650HG2 (3 Ah, 3.6 V nominal) |
| Temperature Range | $-10$°C to 25°C (isothermal tests) |
| Measured Signals | Voltage, current, temperature, ampere-hour |
| SOC Label | Computed via Coulomb counting (Eq. 17) |
| Sampling Rate | Uniformly resampled to 1 Hz |
| Training Drive Cycles | UDDS, HWFET, LA92, US06, Mixed 1–8 |
| Discharge Depth | Up to $\sim$95% of nominal capacity |
| Charging Method | 1C rate, CC-CV |
| Noise Augmentation | 13 variants with gain/offset distortions |
| Normalization | Min–max scaling (Eq. 18) |
| Test Drive Cycles | 3 discharges: UDDS, LA92, US06 |

TABLE II: Summary of training dataset.

### B. Initialization and Hyperparameter Tuning

The feedforward neural network architecture comprises two hidden layers, each with 55 neurons. The hidden layers use

Tanh and Leaky ReLU activation functions ($\alpha = 0.3$), while the output layer employs a Clipped ReLU activation capped at 1.0 to constrain predictions. The model is trained using the Adam optimization algorithm with an initial learning rate of 0.01. Training is conducted over 5500 epochs using full-batch updates without data shuffling. A fixed validation set corresponding to the $-10\,^\circ\mathrm{C}$ condition is used to monitor performance, and MSE is adopted as the loss function. The estimator is initialized as follows: $\alpha_R = 0.3$, $Q = 10^{-8}$, $P_0 = 10^{-2}$, $R_{\text{discharge}} = 10^{-2}$, $R_{\text{charge}} = 10^{-8}$.

### C. Results

Due to the stochastic noise profile and random weights initialization, the Kalman filter and neural network could have slightly different results at every run. Therefore, twenty experiments were conducted with different seeds rather than relying on a single run result to perform a fair evaluation, and the overall performance is reported using average Root Mean Squared Error (RMSE, Eq. 19) and average Mean Absolute Error (MAE, Eq. 20).

$$\mathrm{RMSE}_{\text{avg}} = \frac{1}{N}\sum_{n=1}^{N}\sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\mathrm{SOC}_k - \hat{\mathrm{SOC}}_k^{(n)}\right)^2} \quad (19)$$

$$\mathrm{MAE}_{\text{avg}} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\left|\mathrm{SOC}_k - \hat{\mathrm{SOC}}_k^{(n)}\right| \quad (20)$$

Figures 4 through 8 collectively demonstrate the superiority of the proposed hybrid SOC estimation framework over the baseline FNN model. Fig. 4 shows the performance of the baseline neural network against the proposed algorithm with the test data gathered for $-10\,^\circ\mathrm{C}$ and examines them in three different driving cycles. The UDDS aims to mimic typical urban driving conditions, while LA92 represents a more dynamic and modern driving cycle designed to better reflect real-world urban and suburban driving conditions. The US06 challenges the estimator more, reflecting aggressive urban and suburban driving, including rapid lane changes, passing and merging onto highways, hard accelerations, frequent throttle changes, and high-speed cruising. The highlighted region in Fig. 4 illustrates a case where the FNN estimator diverges under transient nonlinear conditions, which is effectively corrected by the Kalman filter-based hybrid approach.

Figure 5 further supports this observation by showing reduced error amplitude and variability, highlighting the hybrid method's robustness to high-frequency current and voltage fluctuations even under the challenging US06 cycle. Despite this challenging condition, the proposed hybrid approach decreases the error dramatically and sometimes is around zero, which proves its efficacy even in aggressive driving conditions. As shown in Figure 6, the proposed method closely tracks the ground-truth SOC, exhibiting reduced drift and enhanced consistency during both the recovery and dynamic phases of the LA92 cycle.

Quantitative comparisons in Figure 7 reveal that the hybrid estimator achieves up to 39% lower RMSE, particularly at low

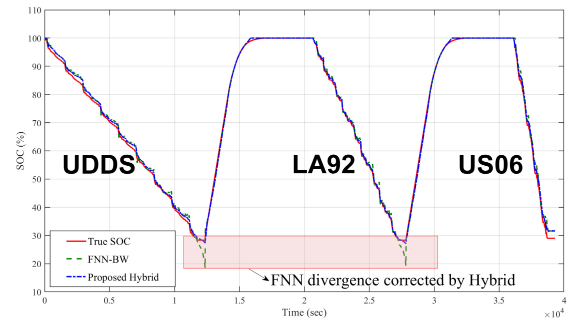


Fig. 4: SOC estimation results on a composite drive cycle at $-10\,^\circ\mathrm{C}$ using the proposed hybrid method and baseline temporal-aware FNN.

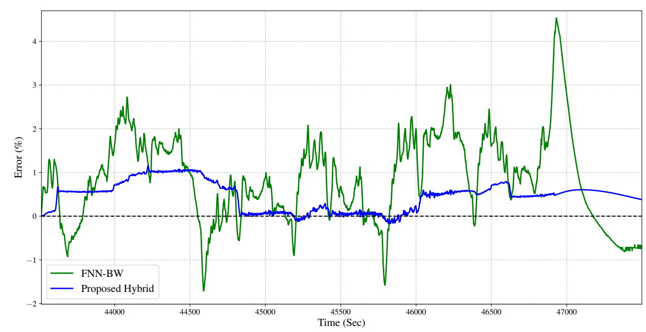


Fig. 5: SOC estimation error on the US06 cycle at $25\,^\circ\mathrm{C}$.

temperatures where estimation is typically more challenging. Similarly, Figure 8 shows that the hybrid approach reduces MAE by up to 28%, confirming the benefit of adaptive covariance tuning and integrating FNN predictions within the Kalman filtering framework. Increased internal resistance and diffusion delays could cause over/under-estimated SOC estimations in lower temperatures. Thus, SOC estimation in lower temperatures is a demanding task. As Fig. 8 reveals for the $-10\,^\circ\mathrm{C}$ test dataset results, the proposed approach

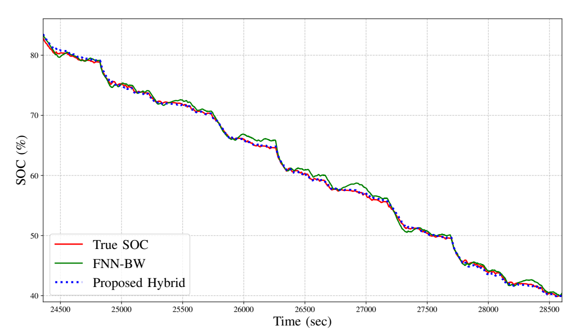


Fig. 6: SOC estimation comparison under the LA92 cycle at $0\,^\circ\mathrm{C}$.

decreases the mean absolute error by 30.93%, and RMSE, which penalizes larger errors more, has dramatically improved by 46.5%, validating the efficacy of the suggested approach in challenging weather conditions and highlighting the adaptability of the hybrid estimator.

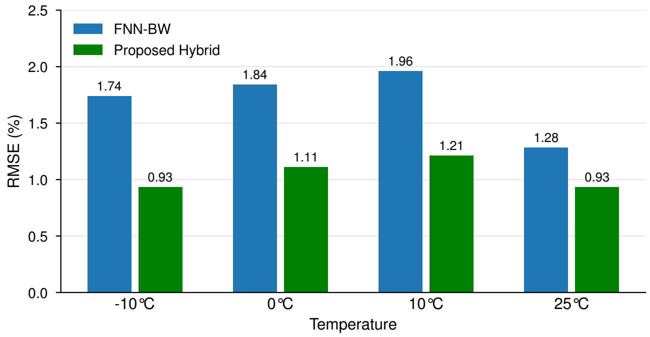


Fig. 7: Root Mean Square Error (RMSE) comparison across four temperatures, averaged over 20 Monte Carlo trials.

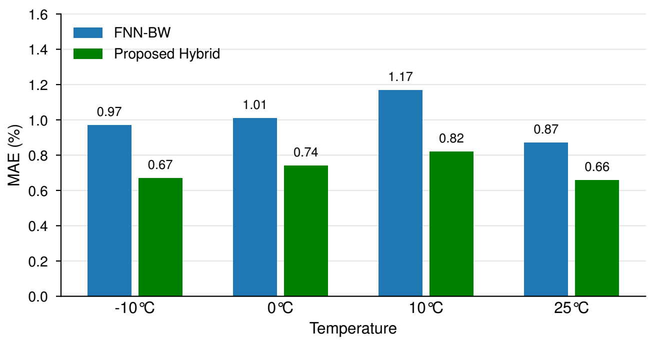


Fig. 8: Mean Absolute Error (MAE) comparison across all tested temperatures, averaged over 20 Monte Carlo trials.

## IV. Conclusion

This paper presents a hybrid framework for robust state-of-charge estimation in lithium-ion batteries that infuses model-based corrections into neural network inferences. A temporal-aware neural network generates synthetic network-aided observation, and an adaptive discrete Kalman filter refines these measurements based on the model-based predictions. Across all tested temperatures, the proposed hybrid architecture cuts the average MAE from 1.01% to 0.72% ($\approx$29% relative reduction) and lowers the average RMSE from 1.71% to 1.00% ($\approx$42% relative reduction). These results demonstrate the method's robustness and superior accuracy under varying thermal conditions, making it well-suited for real-world BMS applications. Future work will comprehensively compare the proposed approach and recurrent neural networks like Long-term Short-term Memory and their computational efficiency in embedded battery management systems through online implementation.

## References


[1] S. Jorkesh, R. Ahmed, S. Habibi, R. Hosseininejad, and S. Xu, "Battery state of charge and state of health estimation using a new hybrid deep neural network approach," *IEEE Access*, vol. 13, pp. 12 566–12 580, 2025.

[2] Y. Chen, Y. Liu, Y. He, Z. Lyu, Y. Cai, and S. Zhang, "Soc estimation of lithium-ion battery pack based on discharge stage division and fusion modeling," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–14, 2025.

[3] A. M. Andwari, A. Pesiridis, S. Rajoo, R. Martinez-Botas, and V. Esfahanian, "A review of battery electric vehicle technology and readiness levels," *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 414–430, 2017.

[4] D. N. T. How, M. A. Hannan, M. S. H. Lipu, and P. J. Ker, "State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review," *IEEE Access*, vol. 7, pp. 136 116–136 136, 2019.

[5] P. Xu, C. Wang, J. Ye, and T. Ouyang, "State-of-charge estimation and health prognosis for lithium-ion batteries based on temperature-compensated bi-lstm network and integrated attention mechanism," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 6, pp. 5586–5596, 2024.

[6] Y. Wang, J. Tian, Z. Sun, L. Wang, R. Xu, M. Li, and Z. Chen, "A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems," *Renewable and Sustainable Energy Reviews*, vol. 131, p. 110015, 2020.

[7] R. P. Priya and R. Sakile, "State of charge estimation of lithium-ion battery based on extended kalman filter and unscented kalman filter techniques," *Energy Storage*, vol. 5, no. 3, p. e408, 2023.

[8] H. B. Sassi, F. Errahimi, and E.-S. Najia, "State of charge estimation by multi-innovation unscented kalman filter for vehicular applications," *Journal of Energy Storage*, vol. 32, p. 101978, 2020.

[9] J. Yun, Y. Choi, J. Lee, S. Choi, and C. Shin, "State-of-charge estimation method for lithium-ion batteries using extended kalman filter with adaptive battery parameters," *IEEE Access*, vol. 11, pp. 90 901–90 915, 2023.

[10] I. M. Monirul, L. Qiu, and R. Ruby, "Accurate soc estimation of ternary lithium-ion batteries by hppc test-based extended kalman filter," *Journal of Energy Storage*, vol. 92, p. 112304, 2024.

[11] P. Shrivastava, T. K. Soon, M. Y. I. B. Idris, and S. Mekhilef, "Overview of model-based online state-of-charge estimation using kalman filter family for lithium-ion batteries," *Renewable and Sustainable Energy Reviews*, vol. 113, p. 109233, 2019.

[12] N. Andrenacci, F. Vellucci, and V. Sglavo, "The battery life estimation of a battery under different stress conditions," *Batteries*, vol. 7, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2313-0105/7/4/88

[13] Z. Cui, L. Wang, Q. Li, and K. Wang, "A comprehensive review on the state of charge estimation for lithium-ion battery based on neural network," *Energy Storage Materials*, vol. 3, no. 2, pp. 123–139, 2021.

[14] P. Dini and D. Paolini, "Exploiting artificial neural networks for the state of charge estimation in ev/hv battery systems: A review," *Batteries*, vol. 11, no. 3, p. 107, 2025.

[15] C. J. G. Vidal, "State-of-charge estimation of lithium-ion batteries for electrified powertrains using neural networks," Ph.D. dissertation, McMaster University, Hamilton, Ontario, Canada, 2020. [Online]. Available: http://hdl.handle.net/11375/25851

[16] C. Chen, R. Xiong, R. Yang, W. Shen, and F. Sun, "State-of-charge estimation of lithium-ion battery using an improved neural network model and extended kalman filter," *Journal of Cleaner Production*, vol. 234, pp. 1153–1164, 2019.

[17] M. Charkhgard and M. Farrokhi, "State-of-charge estimation for lithium-ion batteries using neural networks and ekf," *IEEE transactions on industrial electronics*, vol. 57, no. 12, pp. 4178–4187, 2010.

[18] T. Hao, J. Ding, and T. Tu, "A hybrid kalman filter for soc estimation of lithium-ion batteries," in *2022 41st Chinese Control Conference (CCC)*. IEEE, 2022, pp. 5222–5227.

[19] P. Kollmeyer, C. Vidal, M. Naguib, and M. Skells, "Lg 18650hg2 li-ion battery data and example deep neural network xev soc estimator script," Mendeley Data, V3, 2020, available: https://doi.org/10.17632/cp3473x7xv.3.

[20] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation," in *2017 IEEE power & energy society general meeting*. IEEE, 2017, pp. 1–5.