

Multi timescale battery modeling: Integrating physics insights to data-driven model

Desai, Tushar; Gallo, Alexander J.; Ferrari, Riccardo M.G.

DOI

[10.1016/j.apenergy.2025.126040](https://doi.org/10.1016/j.apenergy.2025.126040)

Publication date

2025

Document Version

Final published version

Published in

Applied Energy

Citation (APA)

Desai, T., Gallo, A. J., & Ferrari, R. M. G. (2025). Multi timescale battery modeling: Integrating physics insights to data-driven model. *Applied Energy*, 393, Article 126040. <https://doi.org/10.1016/j.apenergy.2025.126040>

Important note

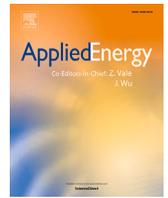
To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



Multi timescale battery modeling: Integrating physics insights to data-driven model

Tushar Desai^{a,*} , Alexander J. Gallo^b , Riccardo M.G. Ferrari^a 

^a Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, Delft, 2628 CD, South Holland, Netherlands

^b Department of Electronics, Information and Bioengineering, Politecnico di Milano, Via Ponzio 34, 20133, Milano, Italy

HIGHLIGHTS

- An encoder–decoder framework is presented to separate battery fast and slow dynamics.
- The model is made interpretable by integrating physics into loss and architecture.
- The latent space maps to quantifiable states without requiring explicit labels.
- The model is validated using sparse intermittent cycles for single/multi-cells.

ARTICLE INFO

Keywords:

Encoder–decoder
Physics informed machine learning
Multi timescale battery modeling
Sparse-unlabeled data

ABSTRACT

Developing accurate models for batteries, capturing ageing effects and nonlinear behaviors, is critical for the development of efficient and effective performance. Due to the inherent difficulties in developing physics-based models, data-driven techniques have been gaining popularity. However, most machine learning methods are black boxes, lacking interpretability and requiring large amounts of labeled data. In this paper, we propose a physics-informed encoder–decoder model that learns from unlabeled data to separate slow-changing battery states, such as state of charge (SOC) and state of health (SOH), from fast transient responses, thereby increasing interpretability compared to conventional methods. By integrating physics-informed loss functions and modified architectures, we map the encoder output to quantifiable battery states, without needing explicit SOC and SOH labels. Our proposed approach is validated on a lithium-ion battery ageing dataset capturing dynamic discharge profiles that aim to mimic electric vehicle driving profiles. The model is trained and validated on sparse intermittent cycles (6 %–7 % of all cycles), accurately estimating SOC and SOH while providing accurate multistep ahead voltage predictions across single and multiple-cell based training scenarios.

1. Introduction

As the electric vehicle (EV) market experiences rapid growth and the demand for stationary energy storage solutions continues to surge, lithium-ion (Li-ion) batteries are emerging as a crucial technology in this industry. These batteries offer a higher energy density and extended cycle life. However, batteries experience degradation over their lifetime, leading to decreased performance. This degradation results from several complex phenomena that occur at the microscale, such as solid electrolyte interphase (SEI) formation, lithium plating, and particle fracture. At the macroscopic level, this degradation manifests as decreasing capacity and increasing resistance.

Given this degradation behavior of batteries, accurate knowledge about their status, such as state of charge (SOC) and state of health (SOH), is important for effective control and optimization. However, in both battery research and practical applications, the ability to monitor batteries remains confined primarily to measuring current, voltage, and temperature. Therefore, relying on mathematical models is essential to extract valuable insights into battery performance, particularly for estimating SOC and SOH. Various approaches are used to estimate a battery's behavior, with physics-based, empirical, and data-driven methods being prominent.

* Corresponding author.

Email addresses: t.k.desai@tudelft.nl (T. Desai), alexanderjulian.gallo@polimi.it (A.J. Gallo), r.ferrari@tudelft.nl (R.M.G. Ferrari).

1.1. Existing modeling approaches

Physics-based techniques focus on Electrochemical (EC) models, which use partial differential equations (PDEs) to offer a deeper understanding of battery behavior. The widely-used pseudo-two-dimensional (P2D) model, based on principles such as porous electrode theory and concentrated solution theory, provides a detailed view of battery dynamics [1,2]. The single particle model (SPM) simplifies the electrode representation, making it a computationally efficient choice for battery modeling [3]. Although offering good performance, these methods require many parameters to be either measured or estimated from data, and involve a high computational cost [4]. Also, it's worth noting that to achieve a comprehensive understanding of battery behavior over its lifetime, it is necessary to integrate degradation models with these core EC models. Battery degradation can result from various mechanisms, either individually or in combination. While significant progress has been made in the development of models to understand individual degradation mechanisms within the framework of EC models, such as solid–electrolyte interface (SEI) formation [5], lithium plating [6,7], and particle fracture [8,9], the intricate interplay between multiple degradation mechanisms remains relatively less accurate [10]. Understanding and modeling multiple mechanisms is even more challenging, exacerbating the parameterization problem for models built upon multiple degradation mechanisms, as emphasized by [11].

Equivalent circuit models (ECM), on the other hand, offer an empirical approach by modeling batteries as a series of resistor–capacitor elements, thus capturing the macroscopic behavior of the battery. This modeling approach strikes a balance between accuracy and complexity, effectively describing battery performance at specific points in its lifetime and under fixed operating conditions. In spite of this, however, ECM lacks a microscopic physics foundation, making it incapable of capturing the nonlinear and time-varying dependencies of parameters on factors like state of charge (SOC), temperature, and current [12]. This limitation greatly constrains ECM's utility and accuracy over extended periods, making it unsuitable for modeling the evolving degradation processes that occur over time [13].

To overcome these limitations, data-driven models based on machine learning (ML) techniques have been proposed due to their ability to automate feature extraction and discover complex patterns within high-dimensional datasets [14]. ML applications in battery modeling are diverse, ranging from estimation of SOC, to health metrics like capacity and internal resistance [15,16]. Although these ML-based methods show promising performance, they often rely on supervised learning and require labeled data, such as SOC, capacity, or resistance at different degradation levels. Obtaining accurate labels is challenging in practical scenarios due to dynamic and varying load conditions [17,18].

ML has also been used to develop health-aware battery models that simulate battery behavior over its lifetime. For instance, Zhao et al. [19] and Hong et al. [20] used recurrent neural networks (RNNs) for multistep-ahead voltage prediction and fault diagnosis. However, the limited interpretability of these models, acting as black boxes, makes it difficult to gain insights into battery states, especially the interplay between slowly varying charge and degradation dynamics and fast voltage dynamics.

1.2. Related work

In recent years, encoder–decoder architecture-based data-driven approaches have gained popularity for identifying nonlinear dynamical systems [21–23]. These models map sequences of past input–output data into lower-dimensional latent spaces representing system states. The flexibility of these models allows for the incorporation of state-space structures, which improves interpretability compared to other data-driven methods.

Encoder–decoder architectures have been applied to various tasks in battery management. For SOC estimation, encoder–decoder models have been used to denoise the signals, thereby increasing estimation

accuracy [24]. Furthermore, advanced variants incorporating attention mechanisms and bidirectional long short-term memory (LSTM) networks have been developed to estimate SOC under complex ambient temperature conditions [25,26]. For SOH estimation, encoder–decoder models have been used to map battery charging curves to SOH values [21]. Encoder–decoder architectures have also been used to predict batteries' remaining useful life (RUL), assuming labeled capacity data are available [27,28]. Despite their widespread usage, a common limitation of these encoder–decoder-based approaches is their reliance on explicit labels for SOC and SOH.

1.3. Proposed approach and contributions

In this paper, to address the challenges of limited interpretability and reliance on labeled data in ML-based battery modeling, we propose a physics-informed encoder–decoder model. Our work integrates known physical principles into an encoder–decoder model to estimate SOC and SOH without explicit labels. By disentangling slowly varying dynamics from fast voltage dynamics within the latent space, our model increases interpretability and enables accurate state estimation from unlabeled data. To the best of the authors' knowledge, this article provides the first attempt at directly leveraging an encoder–decoder model's ability to learn representations that separate slow and fast dynamics in battery systems.

Compared to our previous work [29], we further improve the model's interpretability by constructing a physics-guided architecture. Indeed, while the data-driven encoder–decoder model proposed in [29] could capture slow dynamics and predict multistep-ahead voltage based on the encoder output, it could not offer quantifiable state estimates. This limitation is addressed in this paper by inducing a mapping of the latent representation to measurable battery states. Specifically, by incorporating physics-informed loss functions that capture known physical characteristics of the battery, more interpretable and accurate predictions are achieved.

The key contributions of this work are as follows:

- We propose a physics-informed encoder–decoder framework that separates slow-changing battery states, such as SOC and SOH, from fast voltage dynamics, capturing the batteries' multi-timescale behavior. This separation increases interpretability compared to conventional machine learning methods that often act as black boxes.
- By integrating physics-guided architecture and physics-informed loss functions, the encoder–decoder framework directly maps the latent space to quantifiable battery states without requiring explicit SOC and SOH labels and by just using SOC boundary conditions (BCs). This addresses the common problem of the unavailability of labeled data.
- The model is trained on sparse, intermittent cycle data, showing that the need for SOC BCs can be considerably reduced. It delivers consistent performance across both single-cell and multi-cell-based training scenarios, accurately estimating SOC and SOH while providing reliable voltage predictions multiple steps ahead.

We validate our proposed framework on Stanford's open-source lithium-ion battery ageing dataset, which features dynamic discharge profiles designed to mimic EV driving patterns [30]. While laboratory data cannot fully capture the complexity of real-world EV operations – which include driver-specific behaviors, partial charging/discharging cycles, and varying environmental conditions – this dataset offers several advantages over traditional constant-current discharge tests [31]. The dataset has highly dynamic discharge profiles based on the Urban Dynamometer Driving Schedule (UDDS), incorporating both charging and discharging segments within cycles, simulating regenerative braking typical in EVs. Additionally, the drive profiles follow standardized urban driving schedules, making the dataset more representative of real-world EV load profiles compared to simplified constant current testing protocols [32,33].

1.4. Paper organization

This paper is organized as follows: We discuss the importance and complexities of general battery multi-timescale modeling in Section 2. Data-driven and physics-informed data-driven model constructions are discussed in Sections 3 and 4, respectively. The data used for model validation is described in Section 5. Section 6 covers model training and algorithms. Section 7 presents the results for single and multiple-cell-based training scenarios. Lastly, Section 8 summarizes conclusions and outlines future directions.

2. Multi timescale battery modeling

In the context of lithium-ion batteries, temporal dynamics exhibit distinct scales. Fast timescale phenomena, encompassing intervals on the order of microseconds to a few seconds, are associated with ohmic resistance, charge-transfer processes, and diffusion. On the other hand, the state of charge, representing available energy, experiences timescales typically spanning several hours, contingent upon charge–discharge rates. Conversely, battery degradation is generally a prolonged process unfolding over months and years, often dependent on a cumulative number of charging and discharging cycles.

Although batteries exhibit dynamics on multiple timescales, for modeling purposes, we combine the SOC dynamics and degradation processes into a single slow timescale, since both evolve much more slowly than the fast electrochemical processes. The concurrent presence of fast and slow dynamics in batteries motivates us to consider a general, two-time-scale nonlinear system of the form:

$$\Sigma : \begin{cases} \dot{\xi}_f &= \psi(\xi_f, \xi_s, u, \Theta), \\ \dot{\xi}_s &= \phi(\xi_f, \xi_s, u, \Theta), \\ y &= \chi(\xi_f, \xi_s, u, \Theta), \end{cases} \quad (1)$$

where $\xi_f \in \mathbb{R}^{n_{\xi_f}}$ is the fast varying state vector and $\xi_s \in \mathbb{R}^{n_{\xi_s}}$ is the slowly varying one associated with SOC and degradation. $u \in \mathbb{R}^{n_u}$ is a controlled input vector and $y \in \mathbb{R}^{n_y}$ is a measured output vector. Finally, $\Theta \in \mathbb{R}^{n_\Theta}$ is a parameter vector associated with battery characteristics. Assume $\|\psi\| \gg \|\phi\|$ to distinguish between fast and slow dynamics.

The separation of timescales allows for more efficient and effective control system design. Fast timescales are often associated with local, rapid responses, while slow timescales govern global or long-term behavior. Control strategies are designed to meet each timescale's unique characteristics and demands. Such multiscale dynamical systems find relevance across various scientific domains, including chemical, biological, and industrial processes. When an accurate description of the system's dynamics is available, these multi-timescale models can be identified and controlled with established techniques such as Mori–Zwanzig [34] or singular perturbation theory [35].

However, the modeling and control complexity escalates when precise equations governing specific dynamics are unavailable. Within many such systems, a superposition of multiple timescales occurs, rendering the clear separation of these timescales a challenge. This phenomenon is particularly relevant to battery systems, notably in the context of degradation; indeed, the effect of slowly evolving degradation dynamics at the micro level and their impact on macro-level battery behavior still remains poorly understood [11]. Consequently, there have been limited efforts in the literature to develop battery models explicitly designed with multi-timescale separation [36].

Modern machine learning methods, especially those leveraging the encoder–decoder approach, attempt to separate multiscale dynamics [37]. This neural network architecture finds common application in sequence-to-sequence learning tasks, exemplified in significant studies such as language translation [38]. Comprising two main components, the encoder processes input data, transforming it into a latent representation that captures essential features. The subsequent step involves the decoder exploiting this representation to generate an output sequence.

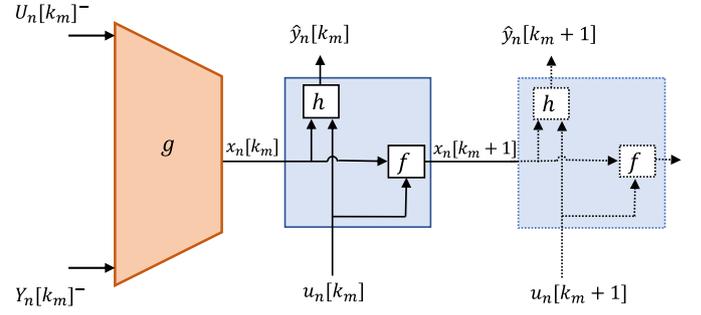


Fig. 1. Encoder–decoder battery model.

The application of the encoder–decoder structure extends beyond language tasks. In molecular dynamics, it has been employed to distinguish slow dynamics from fast ones through step-ahead prediction in the decoder [37]. Similarly, in the study of fluid flow, this architecture has been applied to learning the temporal evolution of complex fluid dynamics [39]. This showcases the broad applicability of the encoder–decoder structure in effectively capturing and separating dynamics in various domains.

In the following sections, we first describe the data-driven encoder–decoder model for separating battery dynamics from previous work. Then, we demonstrate how integrating basic physics insights into the model improves its interpretability.

3. Data-driven approach

We have previously shown that battery modeling can be successfully achieved by building an architecture based on an encoder–decoder structure, as shown in Fig. 1 [29]. In this structure, a multi-step ahead prediction of the battery voltage is included to ensure that stiff latent space generation is avoided [22]. In the remainder of this section, we summarize this general structure. Following this, in Section 4, we provide a specific formulation that integrates physics-based knowledge of the system within the model.

Our proposed encoder–decoder model is structured to process input/output data from full charge–discharge cycles,¹ while generating estimates of the output. Specifically, the encoder processes a “window” of length n_e of current (input) and voltage (output) data, denoted as $U_n[k_m]^-$ and $Y_n[k_m]^-$, respectively, while the decoder generates n_d output estimates, $\hat{Y}_n[k_m]^+$. The subscript $n \in \{1, \dots, N\}$ is used to indicate that the data is associated to one of the N charge–discharge cycles, of length K_n . Furthermore, the notations $[k]^- := [k - n_e, \dots, k - 1]$ and $[k]^+ := [k, \dots, k + n_d - 1]$ denote, respectively, a time window in the past and in the future with respect to the generic time instant k .

The time instants $k_m \in \{1, \dots, K_n\}$ at which the encoder–decoder is evaluated are spaced apart by n_d time-steps, thus avoiding redundant output predictions and reduce the computational burden, i.e., $k_{m+1} = k_m + n_d$, with index $m \in \{1, \dots, M_d\}$, and $M_d := \lfloor (K_n - n_e) / n_d \rfloor$.

Formally, the encoder (Σ_e)–decoder (Σ_d) based battery model can be defined through the following set of difference equations:

$$\Sigma_e : x_n[k_m] = g(U_n[k_m]^-, Y_n[k_m]^-, \theta_g), \quad (2a)$$

$$\Sigma_d : \begin{cases} x_n[k+1] = f(x_n[k], u_n[k], \theta_f), \\ \hat{y}_n[k] = h(x_n[k], u_n[k], \theta_h), \quad k \in [k_m]^+. \end{cases} \quad (2b)$$

The encoder, parameterised by $\theta_g \in \mathbb{R}^{n_{\theta_g}}$, maps the past data samples $U_n[k_m]^-, Y_n[k_m]^-$ to a battery state $x_n[k_m]$ in the model's latent space, using the static mapping function $g : \mathbb{R}^{(n_u+n_y) \times n_e} \rightarrow \mathbb{R}^{n_x}$. For future

¹ Each cycle is defined as the combination of a full charge, followed by a full discharge.

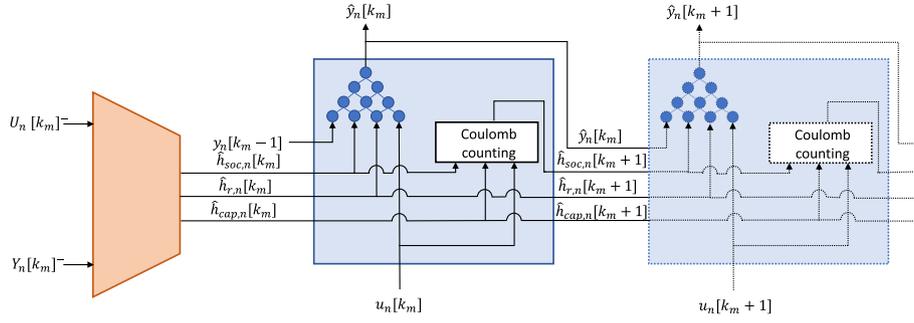


Fig. 2. Encoder–decoder battery model with disentangled latent space and modified recurrent neural network with interpretable state updates.

predictions, the decoder uses $x_n[k_m]$ and “future” samples of the input current $U_n[k_m]^+ \in \mathbb{R}^{n_d \times n_u}$. The state transition function $f : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_x}$ and output function $h : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_y}$ in the decoder, both parameterised respectively by $\theta_f \in \mathbb{R}^{n_{\theta_f}}$ and $\theta_h \in \mathbb{R}^{n_{\theta_h}}$, concur to define the multistep ahead voltage predictions $\hat{Y}_n[k_m]^+ \in \mathbb{R}^{n_d \times n_y}$. Here θ_g , θ_f and θ_h are the tuneable weights-biases vector associated with g , f and h respectively. For simplicity, when discussing the model as a whole, these parameters are collectively referred to as $\theta = [\theta_g^T \theta_f^T \theta_h^T]^T$.

Having provided the architecture of the encoder–decoder model we present in this paper, we now aim to inform the model with physical interpretability. Indeed, the model (2) can already be tuned to accurately predict multistep-ahead voltage, with the latent space $x_n[k_m]$ capturing information about the slowly varying SOC and health states [29]. Still, this latent space output does not quantify individual battery states, as it lacks any inherent mapping to them. Moreover, on the decoder side, the learning-based function f further complicates the interpretability of the decoder dynamics. Solving this issue is the topic of the following sections.

4. Integrating physics into the data-driven approach

The data-driven approach presented in Section 3 demonstrates the potential of encoder–decoder models for battery modeling. However, the lack of physical interpretability in the learned latent space limits our ability to gain insights into the underlying battery dynamics. To address this limitation, we propose an integration of basic physics principles into the encoder–decoder framework. Our approach involves two key modifications:

1. Physics-guided encoder–decoder architecture: We restructure the model’s architecture to incorporate physically meaningful states in the latent space.
2. Physics-informed loss function: We develop a loss function that enforces physical constraints and encourages the model to learn the battery dynamics that align with physical principles.

While the first modification is an assumption about the latent space interpretation, the second one is our key mechanism for aligning the neural network with our desired interpretation. We thus start by redefining the latent state as follows:

$$\hat{x}_n[k_m] = \begin{bmatrix} \hat{x}_{f,n}[k_m] \\ \hat{x}_{s,n}[k_m] \end{bmatrix} \in \mathbb{R}^d, \quad (3)$$

$$\hat{x}_{s,n}[k_m] = [\hat{h}_{\text{soc},n}[k_m] \quad \hat{h}_{\text{cap},n}[k_m] \quad \hat{h}_{\text{r},n}[k_m]]^T,$$

where $\hat{x}_{f,n}[k_m]$ captures fast dynamics, while $\hat{x}_{s,n}[k_m]$ represents slow dynamics-based states. Each state component is further assumed to represent what follows:

- $\hat{h}_{\text{cap},n}[k_m]$ represents the capacity-based SOH factor, calculated from the comparison of the battery’s estimated cycle capacity ($\hat{Q}_{\text{act},n}[k_m]$) to its initial nominal capacity (Q_{nom}), as given by the equation:

$$\hat{h}_{\text{cap},n}[k_m] = \frac{\hat{Q}_{\text{act},n}[k_m]}{Q_{\text{nom}}}. \quad (4)$$

As the capacity factor $\hat{h}_{\text{cap},n}[k_m]$ is slowly varying, we assume it remains constant within a single cycle but may vary across different cycles. This is based on the physical knowledge that significant capacity degradation or recovery, indicative of changes in SOH, does not occur within the relatively short duration of a cycle.

- The SOC, denoted as $\hat{h}_{\text{soc},n}[k_m]$, is defined as the ratio of the remaining charge ($\hat{Q}_{\text{cur},n}[k_m]$) to the current cycle capacity ($\hat{Q}_{\text{act},n}[k_m]$) of the battery, as:

$$\hat{h}_{\text{soc},n}[k_m] = \frac{\hat{Q}_{\text{cur},n}[k_m]}{\hat{Q}_{\text{act},n}[k_m]}. \quad (5)$$

- The state $\hat{h}_{\text{r},n}[k_m]$ captures dynamics not directly linked to SOC or SOH, such as temperature variation or internal resistance that can be useful for voltage prediction but not captured by other states. The state $\hat{h}_{\text{r},n}[k_m]$ is slowly varying and assumed to remain constant for the decoder’s n_d prediction timesteps.
- The voltage from the previous timestep is used as the fast state $\hat{x}_{f,n}[k_m]$, as it captures rapid changes in battery state due to electrochemical processes and polarization effects. For the first timestep in the decoder, this value is the actual measured voltage, while predicted values are used for the subsequent steps as follows:

$$\hat{x}_{f,n}[k_m] = \begin{cases} y_n[k-1], & \text{for } k = k_m \\ \hat{y}_n[k-1], & \text{for } k > k_m \end{cases} \quad (6)$$

In the following subsections, we detail how this physics-informed approach is implemented, starting with the architecture of the encoder–decoder model and the physics-informed loss functions used to align the model with the meaningful battery states described above.

4.1. Encoder

The encoder function g from Eq. (2a) is implemented using a one-dimensional convolutional neural network (1D-CNN) [40]. The 1D-CNN has been shown to effectively extract complex patterns from the longer time-series data of $U[k_m]^-$ and $Y[k_m]^-$, and has a lower computational burden compared to RNNs [14]. To enforce state invariance of $\hat{h}_{\text{cap},n}[k_m]$ as a hard constraint over a cycle, we propose a multistage 1D-CNN as illustrated in Fig. 3. This multistage 1D-CNN processes data in two stages: first, it estimates $\hat{h}_{\text{soc},n}[k_m]$ and $\hat{h}_{\text{r},n}[k_m]$, and extracts intermediate capacity features ($\hat{h}_{\text{cap},n}[k_m]$) from each batch. In the second stage, it aggregates $\hat{h}_{\text{cap},n}[k_m]$ across all batches of a cycle to estimate the overall capacity-based SOH. This two-stage process is summarized in Algorithm 1.

The slowly varying states $\hat{h}_{\text{soc},n}^n[k]$, $\hat{h}_{\text{cap}}^n[k]$ and $\hat{h}_{\text{r},n}^n[k]$ estimated by the encoder are used as initial state values for the decoder model.

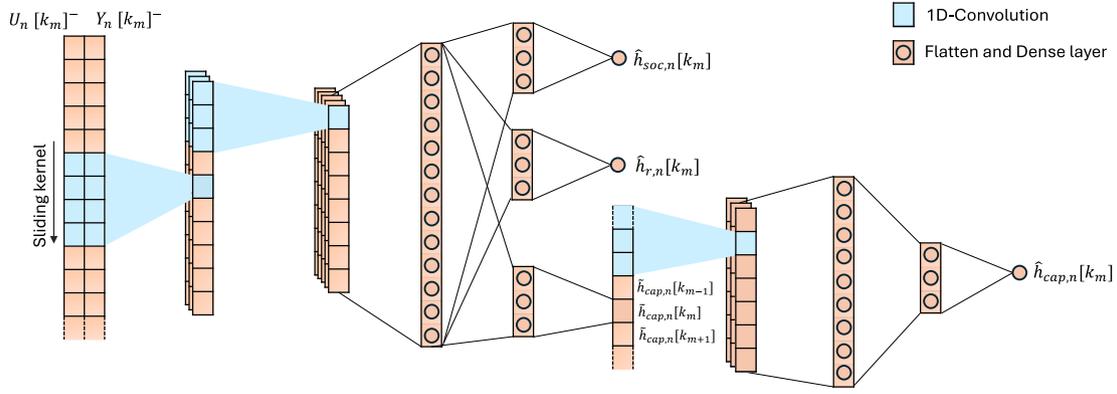


Fig. 3. 1D-CNN based Multistage encoder: first stage convolution with sliding kernel and densely connected layer processes voltage and current data of any batch (m) of a cycle data and estimates SOC and intermediate capacity feature. The second-stage encoder processes these intermediate capacity features from all cycle batches and estimates a single SOH for a cycle.

Algorithm 1: Multistage 1D-CNN encoder.

Input : $X_{n,1:M} = \{U_n[k_m]^-, Y_n[k_m]^-\}_{m=1}^M$ for all batches m of cycle n

Output: Estimated states $\hat{x}_n[k_m]$

Stage 1:

for each batch m in cycle n do

1. Perform multiple 1D convolutions on $X_{n,m}$.
2. Flatten the output.
3. Apply dense layers to estimate states and intermediate capacity feature:

$$[\hat{h}_{soc,n}[k_m], \hat{h}_{cap,n}[k_m], \hat{h}_{r,n}[k_m]] = g_1(X_{n,m}, \theta_{g1}) \quad (7)$$

end

Stage 2:

1. Aggregate $\hat{h}_{cap,n}[k_m]$ from all batches:

$$H_{cap,n} = [\hat{h}_{cap,n}[k_1], \dots, \hat{h}_{cap,n}[k_M]]^\top$$

2. Perform multiple 1D convolutions on $H_{cap,n}$.
3. Apply dense layers to estimate the overall capacity-based SOH for the cycle

$$\hat{h}_{cap,n}[k_m] = g_2(H_{cap,n}, \theta_{g2}) \quad (8)$$

4.2. Decoder

The decoder uses a modified RNN architecture, as illustrated in Fig. 2, to increase the interpretability. The decoder updates the estimated states using the known physics governing these states rather than the learning-based function f in (2b).

- The state $\hat{x}_{f,n}[k_m]$ is obtained from the previous timestep's output, as shown in Eq. (6).
- $\hat{x}_{s,n}[k_m]$ includes health indicator $\hat{h}_{cap,n}[k_m]$ and $\hat{h}_{r,n}[k_m]$, which remain constant across n_d timesteps of decoder as detailed in the Section 4. However, $\hat{h}_{soc,n}[k_m]$ is dynamically updated based on the current input and actual capacity $\hat{Q}_{act,n}[k_m]$, using the Coulomb counting method. This fundamental electrochemical principle of charge accumulation is described by:

$$\hat{h}_{soc,n}[k_m + 1] = \hat{h}_{soc,n}[k_m] + \frac{u_n[k_m] \cdot \Delta t \cdot \eta}{Q_{nom} \cdot \hat{h}_{cap,n}[k_m]}, \quad (9)$$

where Δt is the sampling time, and η is the Coulombic efficiency. The battery's current capacity is calculated based on Q_{nom} and the SOH factor estimated by the encoder, as per Eq. (4).

The nonlinear voltage prediction function h in (2b) is implemented through a time-distributed, shallow,² two-layer dense neural network as shown in Fig. 2 [41]. This network takes $\hat{h}_{soc,n}[k_m]$, $\hat{h}_{r,n}[k_m]$, $\hat{x}_{f,n}[k_m]$ and $u_n[k_m]$ to predict multi-step ahead voltage as follows:

$$\hat{y}_n[k_m] = h(\hat{x}_{f,n}[k_m], \hat{h}_{soc,n}[k_m], \hat{h}_{r,n}[k_m], u_n[k_m], \theta_h), \quad (10)$$

here, h represents a neural network that is trained to learn the relationship between the input, states, and voltage.

4.3. Learning

The learning process of the physics-informed encoder–decoder model involves optimizing multiple objectives to achieve accurate voltage prediction and physically meaningful state estimation. We use an unsupervised learning approach for state estimation, leveraging physical principles and constraints to guide the learning process. The comprehensive loss function consists of several components: the multistep ahead voltage prediction loss (\mathcal{L}_{pred}), which quantifies the accuracy of voltage predictions over multiple timesteps; the SOC dynamics loss ($\mathcal{L}_{dyn,soc}$), which enforces consistency in SOC estimation over successive batches; and the SOC constraint loss ($\mathcal{L}_{con,soc}$) along with the SOH constraint loss ($\mathcal{L}_{con,soh}$), which impose soft constraints to keep the estimated SOC and SOH values within physically realistic bounds.

The overall training process for the physics-informed encoder–decoder model involves solving the following constrained optimization problem numerically:

$$\begin{aligned} \min_{\theta} \quad & \mathcal{L}_{pred} + \lambda_{dyn,soc} \mathcal{L}_{dyn,soc} + \lambda_{con,soc} \mathcal{L}_{con,soc} + \lambda_{con,soh} \mathcal{L}_{con,soh}, \\ \text{s.t.} \quad & \text{constraints of (13)} \end{aligned} \quad (11)$$

where $\lambda_{dyn,soc}$, $\lambda_{con,soc}$, and $\lambda_{con,soh}$ are weighting factors for the respective terms in the cost function. Each of these loss functions and the constraints are detailed as follows:

4.3.1. Multistep ahead voltage prediction loss (\mathcal{L}_{pred})

The multistep ahead voltage prediction loss, based on the mean squared error (MSE), is defined as:

$$\mathcal{L}_{pred} = \frac{1}{N \cdot M \cdot n_d} \sum_{n=1}^N \sum_{m=1}^M \sum_{k=k_m}^{k_{m+1}-1} \|\hat{y}_n[k] - y_n[k]\|^2, \quad (12)$$

² The detailed parameter count of the physics-informed encoder–decoder model is provided in Appendix A.

where, $y_n[k]$ is the true measured voltage from data and $\hat{y}_n[k]$ is the predicted voltage from the model.

4.3.2. SOC dynamics loss ($\mathcal{L}_{\text{dyn},\text{soc}}$)

This dynamic loss leverages the multiple-shooting method to enforce consistency in SOC estimation over successive batches [42,43]:

$$\mathcal{L}_{\text{dyn},\text{soc}} = \frac{1}{N \cdot M} \sum_{n=1}^N \sum_{m=1}^M \left\| \hat{h}_{\text{soc},n}[k_m] - \bar{h}_{\text{soc},n}[k_m] \right\|, \quad (13)$$

where,

$$\bar{h}_{\text{soc},n}[k_m] = \begin{cases} h_{\text{soc},n,\text{begin}} & \text{if } m = 1 \\ \hat{h}_{\text{soc},n}[k_{m-1}] + \sum_{k=k_{m-1}}^{k_m-1} \frac{u_n[k] \cdot \Delta t \cdot \eta}{\hat{Q}_{\text{act},n}[k_m]} & \text{if } 1 < m < M \\ h_{\text{soc},n,\text{end}} & \text{if } m = M. \end{cases} \quad (14)$$

where $\hat{Q}_{\text{act},n}[k_m]$ represents the actual battery capacity derived from the estimated health indicator ($\hat{h}_{\text{cap},n}[k_m]$) using Eq. (4). For intermediate batches ($1 < m < M$), the SOC evolution is calculated by applying Coulomb counting to the SOC estimate from the previous batch ($m - 1$), incorporating the current input ($u_n[k]$), sampling time (Δt), and Coulombic efficiency (η). The boundary conditions $h_{\text{soc},n,\text{begin}}$ and $h_{\text{soc},n,\text{end}}$ define the initial and final SOC values for each cycle, adapting to partial charge/discharge scenarios in practical applications. This physics-informed formulation enables unsupervised learning of both SOC and SOH by incorporating fundamental charge conservation principles into the loss function.

4.3.3. SOC constraint loss ($\mathcal{L}_{\text{con},\text{soc}}$)

This loss ensures that estimated SOC values remain within physically realistic bounds:

$$\mathcal{L}_{\text{con},\text{soc}} = \frac{1}{N \cdot K} \sum_{n=1}^N \sum_{k=1}^K \left(\max(0, h_{\text{soc},\text{min}} - \hat{h}_{\text{soc},n}[k]) + \max(0, \hat{h}_{\text{soc},n}[k] - h_{\text{soc},\text{max}}) \right), \quad (15)$$

where $h_{\text{soc},\text{min}}$ and $h_{\text{soc},\text{max}}$ denote the minimum and maximum limits for SOC, respectively. This formulation penalizes any SOC estimates that fall outside the predefined range, imposing soft constraints on the learned SOC values.

4.3.4. SOH constraint loss ($\mathcal{L}_{\text{con},\text{soh}}$)

Similarly, the SOH constraint loss keeps the estimated SOH within realistic bounds:

$$\mathcal{L}_{\text{con},\text{soh}} = \frac{1}{N \cdot M} \sum_{n=1}^N \sum_{m=1}^M \left(\max(0, h_{\text{cap},\text{min}} - \hat{h}_{\text{cap},n}[k_m]) + \max(0, \hat{h}_{\text{cap},n}[k_m] - h_{\text{cap},\text{max}}) \right), \quad (16)$$

where $h_{\text{cap},\text{min}}$ and $h_{\text{cap},\text{max}}$ represent the minimum and maximum values for SOH, respectively.

5. Data

We use Stanford's open-source lithium-ion battery ageing datasets to validate the proposed approach [30]. This dataset includes discharge cycles that aim to mimic typical electric vehicle usage per the UDDS

Table 1

Summary of charge C-rates for different cells.

Charge C-rate	3C	1C	C/2	C/4
Cells	W3, W10, G1	W9, V5	W5, W8	W4, W7, V4

cycle. While this dataset represents an improvement over traditional constant-current cycling tests by incorporating dynamic load profiles, it is important to acknowledge its limitations in fully representing real-world EV operations. Laboratory data inherently cannot capture all complexities of field conditions, where batteries experience driver-specific usage patterns, partial cycling, varying environmental conditions, extended rest periods leading to calendar ageing, and combinations thereof [31].

The dataset comprises data from INR21700-M50T NMC battery cells, each with a Q_{nom} of 4.85 Ah. Throughout 23 months, 10 of these cells underwent a testing protocol that combined a constant current-constant voltage (CCCV) charging, spanning charging rates³ from C/4 to 3C, as detailed in Table 1. All procedures were conducted at a constant temperature of 23 °C. For a further detailed description of the experimental setup, we refer interested readers to [30].

The experiment adopts a detailed charging and discharging protocol, depicted in Fig. 4. This protocol is structured into a six-step cycle. The cycle initiates with a CC charging phase at a particular C-rate mentioned in Table 1 (Step 1). This phase transitions to CV charging once the battery's voltage reaches 4 V (Step 2), persisting until the charging current is reduced to below 50 mA. In the second stage, Steps 3 and 4 are designed to extend the battery's voltage to 4.2 V, equating to a 100 % SOC through CC charging, followed by another CV phase. The subsequent step (Step 5) involves reducing the SOC from 100 % to 80 % through a constant current discharge at C/4. The cycle terminates with Step 6, where a sequence of UDDS cycles is used to deplete the battery's charge from 80 % SOC down to 20 %, completing the one cycle.

To establish a reliable baseline for battery health, a series of reference performance tests (RPTs) were conducted at regular intervals every 25–30 cycles. These tests included capacity measurements, hybrid pulse power characterization (HPPC), and electrochemical impedance spectroscopy (EIS). The intermittent actual capacity ($Q_{\text{act},n}$) of the cell was assessed by discharging at a rate of C/20 from a state of full charge. Over time, $Q_{\text{act},n}$ of the cells declines due to ageing, as depicted in Fig. 6. The capacity-based health indicator at these intermittent cycles is calculated using:

$$h_{\text{cap},n} = \frac{Q_{\text{act},n}}{Q_{\text{nom}}} \quad (17)$$

$$= \frac{\int_{\text{charged}}^{\text{discharged}} I_{C/20}(t) dt}{Q_{\text{nom}} \cdot 3600}.$$

The UDDS discharge profiles from six battery cells are selected for validation. The remaining cells are excluded due to inconsistencies observed during RPTs, leading to the discontinuation of their ageing campaign. UDDS profiles were chosen for their dynamic nature, which simulates urban driving conditions. As illustrated in Fig. 5, these profiles include both charging and discharging phases, simulating the typical regeneration braking scenarios of EVs that aim to represent more realistic driving scenarios. Data are collected at a frequency of 10 Hz.

In the dataset, while the direct SOC indicator is absent, signals for charge capacity (Q_{dis}) and discharge capacity (Q_{dis}) are included, derived by integrating current during charging and discharging phases,

³ C-rate indicates the charge or discharge rate relative to the battery's capacity. For example, a 1C rate charges or discharges the battery in 1 h, while 2C does so in 30 minutes.

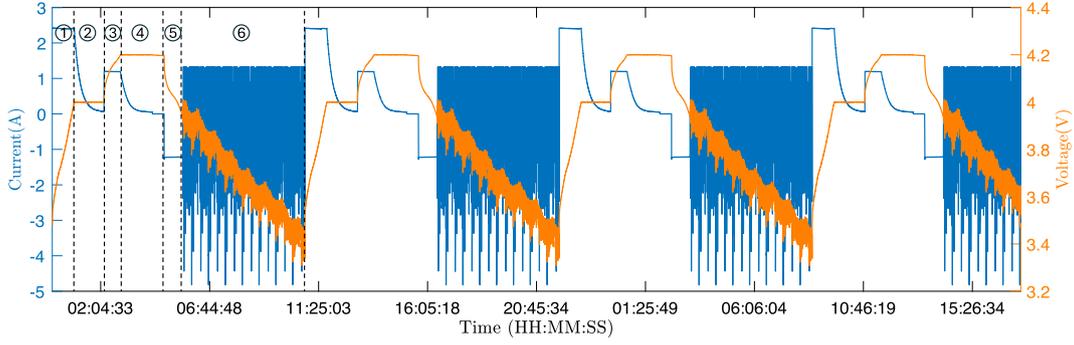


Fig. 4. Battery ageing campaign with repeated charging and discharging steps: (1) Initial CC charge to 4 V, (2) Transition to CV charging until the current cut-off of 50 mA, (3) CC charging to reach 4.2 V, (4) CV phase to sustain 4.2 V, (5) CC discharge to 80 % SOC, and (6) Multiple UDDS cycles down to 20 % SOC.

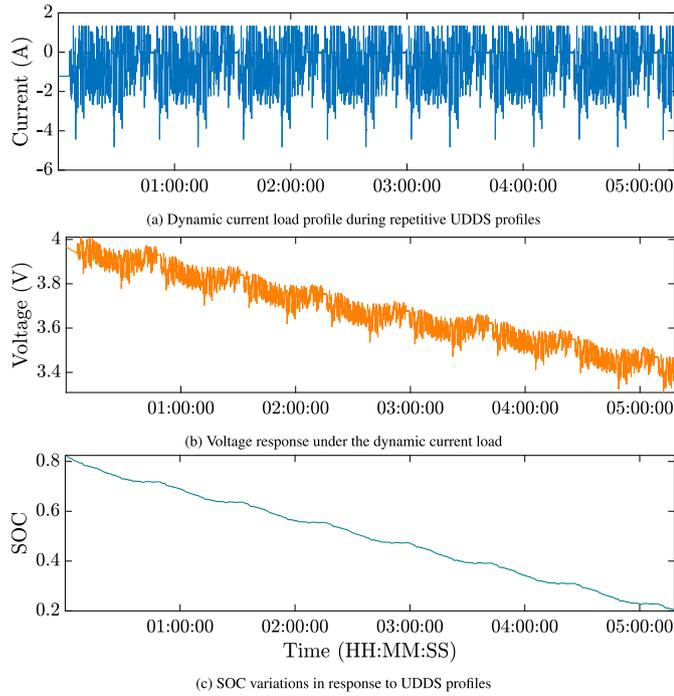


Fig. 5. Variations in current, voltage, and SOC under repetitive UDDS profiles (Step 6), with SOC varying from 80 % to 20 %.

respectively. At the start of Step 5, both Q_{ch} and Q_{dis} are reset. The cell is then discharged at a $C/4$ rate until Q_{dis} reaches 0.2 times the actual capacity $Q_{act,n}$, marking time t_b when the SOC is 80 %. In Step 6, discharge continues until $(Q_{dis}(t_c) - Q_{ch}(t_c)) \geq 0.8 \times Q_{act,n}$, marking time t_c , corresponding to an SOC of 20 %. Depth of discharge (DOD) is then used to calculate SOC for UDDS discharge profiles, providing the ground truth:

$$\begin{aligned} h_{soc} &= 1 - DOD \\ &= 1 - \frac{Q_{dis} - Q_{ch}}{Q_{act,n}} \end{aligned} \quad (18)$$

6. Training and deployment

The overall methodology for training and deploying the model is illustrated in Fig. 7. During training, the model requires voltage, current, and cycle-specific boundary conditions (BCs) for the SOC. Specifically, the model can use any cycle segment for training provided that the SOC BCs are known, as shown in Equation (13), which is feasible in many

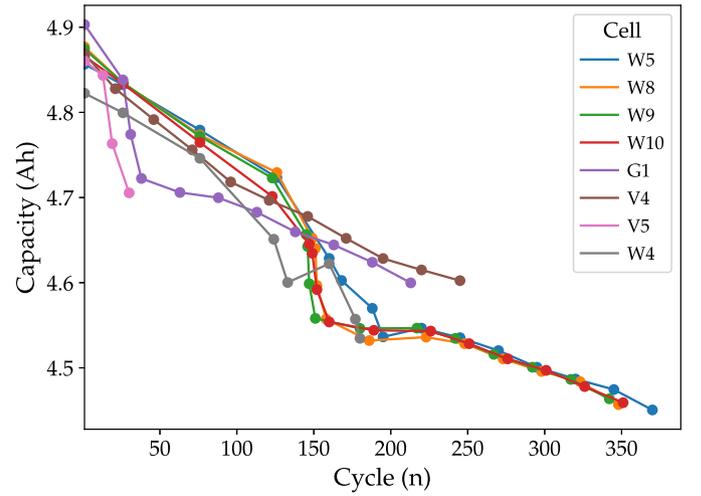


Fig. 6. Periodic assessments of actual battery capacity ($Q_{act,n}$) through capacity test over multiple cycles [30].

scenarios. However, there is a possibility that these SOC BCs are not scaled in accordance with the battery's current degradation level, as described by Eq. (18), a scenario that could affect the model's ability to learn the SOH. In our work, we use cycle segments where these SOC BCs are accurately known as detailed in Section 5. However, we recognize that obtaining SOC BCs for all cycles is challenging in practice. To address this issue, we use sparse intermittent cycles where SOC BCs are more likely to be accurately known. The SOC BCs that are scaled appropriately can potentially be obtained during fully charged or fully discharged conditions or through SOC-OCV mapping after an appropriate rest period. Although these intermittent cycles represent only about 6 % to 7 % of the total cycles used for training, this approach allows the model to learn the SOC and SOH dynamics without requiring SOC BCs for every cycle.

The voltage and current data from intermittent cycles undergo a data preprocessing step: each cycle's voltage and current data are normalized, downsampled to 1 Hz to reduce computational load, and segmented into training and validation. Specifically, 80 % of the cycles are used for training and 20 % for validation. Some other random cycles are chosen for testing the trained model, as illustrated in Fig. 8. The test cycles undergo the same preprocessing as the training and validation cycles. As mentioned in Section 3, to avoid overlap in multistep-ahead voltage predictions, each cycle data is partitioned into m non-overlapping batches, and each batch spans a length of $n_e + n_d$ timesteps, with a lag of n_d steps between successive batches.

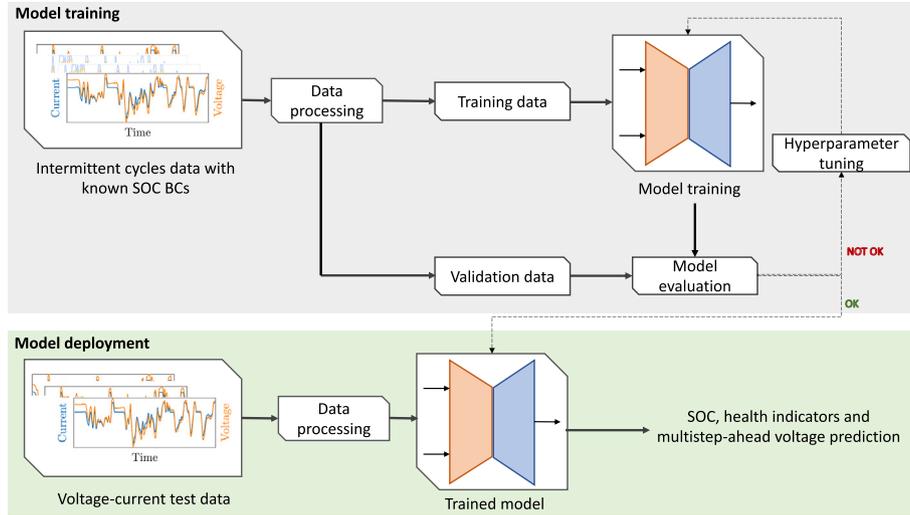


Fig. 7. Schematic representation of encoder-decoder based battery model training and deployment framework.

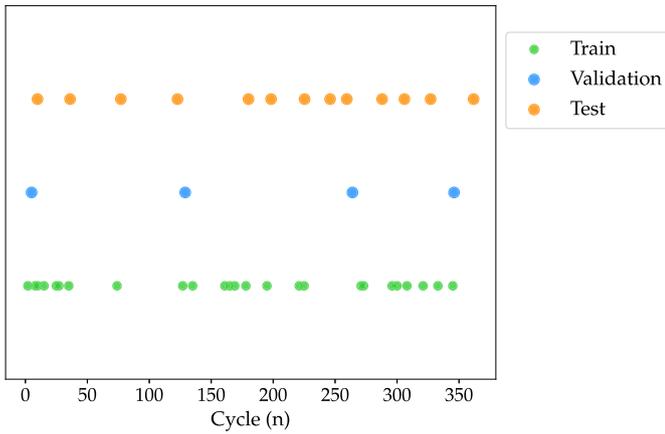


Fig. 8. For cell W5, randomly chosen intermittent cycles for training, validation, and testing of the encoder-decoder battery model.

6.1. Model training

The data preprocessing is followed by model training, summarized in Algorithm 2. Model weights and biases (θ) are initialized according to the efficient Glorot scheme [44,45]. During the forward pass, the total loss, \mathcal{L}_{total} , is computed based on the predicted voltage and the encoder's latent space output as described in Section 4.3.

In the backward pass, gradients are calculated based on the total loss function \mathcal{L}_{total} , employing the backpropagation through time (BPTT) with the modified RNN framework of the decoder [46]. These gradients are subsequently propagated back through the latent space to the encoder, facilitating a comprehensive encoder-decoder learning as outlined by [47]. To optimize the adjustment of model parameters in response to these gradients, the Adam optimizer, known for its adaptive learning rate capabilities, is used [48]. To prevent overfitting, early stopping is implemented, which monitors validation loss and halts the training process when there is no significant improvement.

To ensure physical plausibility and numerical stability, additional constraints are applied: for the SOC, we enforce $0 \leq \hat{h}_{soc,n}[k_m] \leq 1$, and for the SOH, $0.1 \leq \hat{h}_{cap,n}[k_m] \leq 1.05$. The lower bound for SOH is set to prevent numerical issues in the Coulomb counting Eq. (9). In contrast, the upper bound accommodates scenarios where a new battery's capacity may exceed $Q_{nom} = 4.85$ Ah, as shown in Fig. 6. The

above-described constraints are derived from typical battery behavior, observed under controlled conditions. However, the actual vehicle operation may introduce factors such as temperature fluctuations, varying loads, and ageing that affect battery response. Thus, these fixed bounds defined above should be considered initial guidelines that must be tuned based on specific system data and real-world observations.

During training, not only the primary prediction loss \mathcal{L}_{pred} is monitored, but also the soft-constraint penalties $\mathcal{L}_{con,soc}$ and $\mathcal{L}_{con,soh}$ defined in Eqs. (15) and (16), respectively. Achieving convergence of these losses during training, meaning they are driven toward approximately zero, is crucial. If either penalty fails to decline toward zero, it indicates persistent SOC or SOH estimates outside the permissible range. Such violations propagate through the recurrent decoder, since \hat{h}_{soc} and \hat{h}_{cap} directly influence future voltage predictions, and degrade multi-step forecast accuracy. Therefore, we require

$$\mathcal{L}_{con,soc} \rightarrow 0, \quad \mathcal{L}_{con,soh} \rightarrow 0,$$

before accepting a training model. If, in either training or online deployment, the constraint losses plateau above a small threshold (e.g., $\epsilon = 10^{-3}$), we need an adaptive recalibration:

- Adjust or shift $[h_{soc,min}, h_{soc,max}]$ based on in-field OCV measurements after rest, to accommodate systematic SOC drift.
- Adjust $[h_{cap,min}, h_{cap,max}]$ in proportion to observed SOH deviations.

A coupled adjustment is essential because the two states are algebraically linked through the definitions in (4) and (5). This dynamic adjustment ensures that our soft bounds remain representative of the actual operating condition, thereby preserving both the physical plausibility of \hat{h}_{soc} and \hat{h}_{cap} and the accuracy of the downstream voltage predictions.

6.2. Evaluation criteria

The model's performance is assessed using the following performance metrics: root mean square error (RMSE), mean absolute percentage error (MAPE), and maximum absolute error (Max):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}, \quad (19)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad (20)$$

$$Max = \max_i |y_i - \hat{y}_i| \quad (21)$$

Algorithm 2: Training of the physics-informed encoder–decoder battery model.**Input :** model, training-validation cycles data**Output:** Trained model with optimized parameters θ **for each epoch do****Training:** $total\ training\ loss \leftarrow 0$ **for cycle (n) \in training cycles do** $X_{n,1:M} = \{U_n[k_m]^-, Y_n[k_m]^-\}_{m=1}^M, \{U_n[k_m]^+\}_{m=1}^M, \{Y_n[k_m]^+\}_{m=1}^M, h_{soc, begin}, h_{soc, end} \leftarrow cycle$ **Encoder:****for $m \in [1, \dots, M]$ do** $[\hat{h}_{soc,n}[k_m], \hat{h}_{cap,n}[k_m], \hat{h}_{r,n}[k_m]] \leftarrow g_1(X_{n,m}, \theta_{g1})$ (7)**end** $\hat{h}_{cap,n}[k_m] \leftarrow g_2([\hat{h}_{cap,n}[k_1], \dots, \hat{h}_{cap,n}[k_M]]^\top, \theta_{g2})$ (8)**Decoder:****for $m \in [1, \dots, M]$ do****for $k \in [k_m, \dots, k_m + n_d - 1]$ do** $\hat{h}_{soc,n}[k+1] \leftarrow \hat{h}_{soc,n}[k] + \frac{u_n[k] \cdot \Delta t \cdot \eta}{Q_{nom} \cdot \hat{h}_{cap,n}[k]}$ (9) $\hat{h}_{cap,n}[k+1] \leftarrow \hat{h}_{cap,n}[k]$ $\hat{h}_{r,n}[k+1] \leftarrow \hat{h}_{r,n}[k]$ $\hat{y}_n[k] = h(\hat{x}_{f,n}[k], \hat{x}_{s,n}[k], u_n[k], \theta_h)$ (10)**end****end****Loss Computation:** $\mathcal{L}_{pred} \leftarrow \mathbb{E} \left[\left\| \hat{Y}_n[k_m]^+ - Y_n[k_m]^+ \right\|^2 \right]$ (12) $\mathcal{L}_{dyn,soc} \leftarrow \mathbb{E} \left[\left\| \hat{h}_{soc,n}[k_m] - \begin{cases} h_{soc,begin} & \text{if } m = 1 \\ \hat{h}_{soc,n}[k_{m-1}] + \sum_{k=k_{m-1}}^{k_m-1} \frac{u_n[k] \cdot \Delta t \cdot \eta}{Q_{act,n}[k_m]} & \text{if } 1 < m < M \\ h_{soc,end} & \text{if } m = M \end{cases} \right\|^2 \right]$ (13) $\mathcal{L}_{con,soc} \leftarrow \mathbb{E} \left[\max(0, h_{soc,min} - \hat{h}_{soc,n}[k_m]) + \max(0, \hat{h}_{soc,n}[k_m] - h_{soc,max}) \right]$ (15) $\mathcal{L}_{con,soh} \leftarrow \mathbb{E} \left[\max(0, h_{cap,min} - \hat{h}_{cap,n}[k_m]) + \max(0, \hat{h}_{cap,n}[k_m] - h_{cap,max}) \right]$ (16)**Optimization:** $model.loss \leftarrow \mathcal{L}_{pred} + \lambda_{dyn,soc} \mathcal{L}_{dyn,soc} + \lambda_{con,soc} \mathcal{L}_{con,soc} + \lambda_{con,soh} \mathcal{L}_{con,soh}$ (11) $model.loss.backward()$ $model.optimizer.step()$ $total\ training\ loss \leftarrow total\ training\ loss + model.loss$ **end****Validation:** $total\ validation\ loss \leftarrow 0$ **for cycle \in validation cycles do**

Apply the trained model to validation cycle data to compute validation loss.

 $total\ validation\ loss \leftarrow total\ validation\ loss + validation\ loss$ **end**

Compute average training and validation loss.

if average validation loss < average training loss then

Save model checkpoint.

end**end**

where N represents the number of samples, y_i denotes the true value of the i th sample, and \hat{y}_i indicates the predicted value.

6.3. Deployment scenario

In deployment scenarios, data quality needs to be strictly monitored. Real-world vehicle data are often incomplete or noisy, underscoring the need for a robust data pipeline to clean and preprocess signals [18,31]. In our work, we assume that the cycle data selected for training are complete and free of missing values, which allows us to isolate model performance from data quality issues.

With this assumption in place, the model's deployment can be categorized into two distinct strategies: offline and online training. Offline training adopts a more conventional approach, wherein the model is

developed using a dataset derived from multiple cells. This methodology uses intermittent cycle data from multiple cells for training and validation in order to capture cell-to-cell variation. Importantly, by using only current and voltage measurements obtained during UDDS discharge cycles, the model remains independent of specific charging profile features, such as the multi-stage CCCV protocol used in the Stanford dataset. This complete dependence on current and voltage data eliminates the necessity for historical charging information, thereby mitigating the risk of data leakage and improving the model's capacity to generalize across different cells [49]. The model is trained on data from multiple cells and tested on a holdout cell to demonstrate this.

However, applying the model to entirely different charging profiles may introduce new degradation patterns that are not captured in our

current training setup. While the model can adapt to varying charging behaviors using voltage–current data, its performance could be affected if the underlying degradation mechanisms differ significantly. This limitation underscores the importance of ensuring that the training dataset contains a diverse range of charging profiles to improve the model’s robustness.

In contrast, the emergence of cloud-based battery management systems introduces the flexibility of not only allowing for offline training, but also online model training and deployment [50,51]. The model, therefore, benefits from a dynamic, iterative training and deployment process in the online training framework. Similar to offline training, we focus on using current and voltage measurements, primarily from highly dynamic discharge cycles. As with offline deployment, encountering different charging profiles in online operations may introduce new degradation patterns. However, the continuous learning capability of the online model helps address this issue by allowing it to adapt to new data reflecting the actual usage patterns of the battery. The model is updated in real-time as new data becomes available, particularly cycle data with known SOC BCs. However, as discussed earlier, obtaining accurate SOC BCs for all cycles is challenging in practice, and this issue is particularly critical in online learning scenarios. To address this, we rely on sparse intermittent cycles where SOC BCs are accurately known. This continuous adaptive learning allows the model to improve with each new cycle without requiring SOC BCs for every cycle. The efficacy of this approach is assessed through training on intermittent cycle data from individual cells, with detailed results presented in Section 7.

7. Results and discussion

The performance of the proposed model is evaluated for multi-step ahead voltage prediction, state of charge estimation, and health indicators estimation ($h_{cap,n}[k_m]$, and $h_{r,n}[k_m]$). Evaluation metrics include RMSE, MAPE, and Max, as defined in Section 6.2. The model is trained using 6%–7% of the available cycles, selected as sparse intermittent cycles. For the online scenario, the model is trained independently on six individual cells, with data partitioned into training, validation, and test sets as shown in Fig. 8. In the offline scenario, the model is trained on sparse training cycles from five cells and tested over 10% of the cycles from a holdout cell (W8) across different ageing levels.

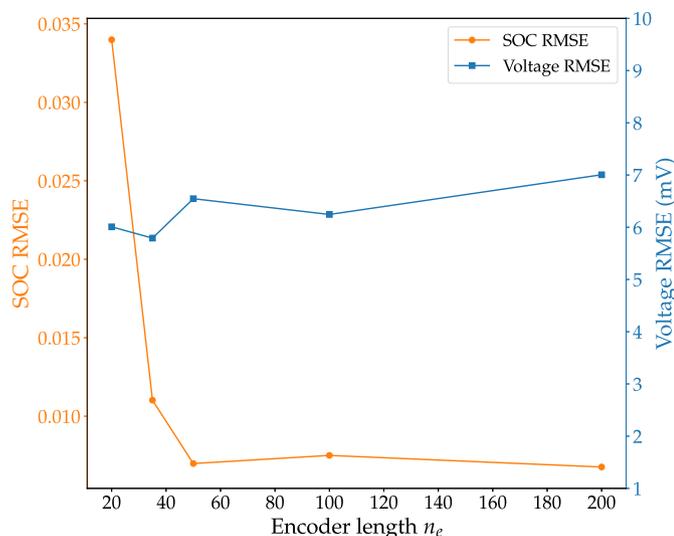


Fig. 9. Variation of the encoder sequence length n_e , and its impact on SOC estimation and voltage prediction in RMSE values.

Table 2

Training epoch time for different decoder sequence lengths (n_d) for single-cell data.

n_d	Time per epoch (s)
50	0.57
100	1.00
200	2.01
300	2.89
400	3.83

Hyperparameter tuning primarily focuses on the learning rate, weight factor λ for loss functions in (11), and sequence lengths for the encoder (n_e) and decoder (n_d). After extensive experiments, $\lambda_{\text{dyn, soc}}$ was set to 1000 for SOC-related dynamic loss, with other λ values set to 1. A learning rate of $5 \cdot 10^{-4}$ balances convergence speed with stability. As shown in Fig. 9, on test cycles of a single cell, varying the encoder length n_e from 20 to 200 indicated a sharp decline in SOC RMSE until $n_e = 50$, after which it plateaued, while voltage RMSE remained steady at 6–7 mV. The encoder length 50 shows substantially lower SOC estimation error, whereas lengths above 50 provided only marginal accuracy gains at the expense of increased computational cost. Consequently, $n_e = 50$ was selected for all subsequent experiments.

Through the experimentation process, it was found that the decoder sequence length n_d is an important hyperparameter, impacting not only the quality of the multistep-ahead voltage prediction but also the accuracy of the estimators of SOC and other health indicators. The influence of n_d on model performance is extensively evaluated in the remaining section, with a detailed comparison of evaluation metrics.

Computational burden. All experiments were implemented in TensorFlow and run on a cluster equipped with an AMD EPYC 7402 CPU (24 cores at 2.80 GHz), 200 GB RAM, and an NVIDIA Tesla V100S-PCIE-32GB GPU. Decoder-length tests, as shown in Table 2 indicate per-epoch⁴ training time scales linearly from 0.57 s at $n_d = 50$ to 3.83 s at $n_d = 400$. This is because the modified RNN decoder must perform backpropagation through time over n_d steps [46], as discussed in Section 6.1. Each additional step adds another layer to the computational graph and incurs extra matrix multiplies and gradient computations. Over 2000–3000 epochs, total training time thus ranges from roughly 1–7 h, depending on sequence lengths and dataset size.

7.1. Multi-step ahead voltage prediction

The performance metrics for multistep-ahead voltage prediction across test cycles for each individual cell with varying n_d are presented in Fig. 10. The results show that $n_d = 50$ gives the lowest RMSE, MAPE, and Max. As n_d increases, prediction errors also increase due to the accumulation of errors propagated through each prediction step, where the one-step voltage prediction serves as the input (x_f) for the subsequent step, as defined in Eq. (10). Despite this, RMSE varies only from approximately 6–8 mV, across n_d values from 50 to 400. The voltage predictions and errors, for $n_d = 50$ and $n_d = 400$, is presented in Fig. 11. This result is based on a test cycle of cell W5, with 92% SOH and a SOC ranging from 80% to 20%. The results indicate the model’s ability to predict voltage across these different n_d accurately.

In the multiple cell-based training-validation and testing on holdout cell, the performance metrics with varying n_d are shown in Table 3. The results are comparable to those of individual cell-based training. However, the errors are lower in the multiple cell-based training. Specifically, RMSE ranges from approximately 5–7.5 mV as n_d varies from 50 to 400, compared to higher RMSE values of 6–8 mV in individual

⁴ An epoch is one complete pass through the entire training dataset.

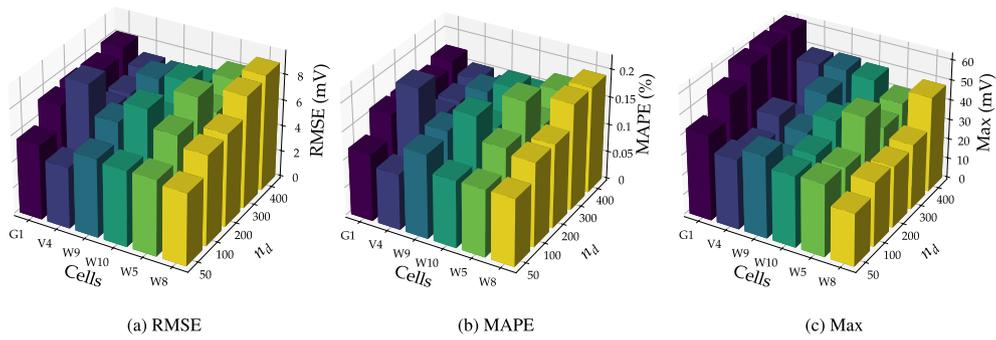


Fig. 10. Individual cell-based training: performance metrics for multi-step ahead voltage prediction, evaluated across the test cycles of each cell with varying n_d .

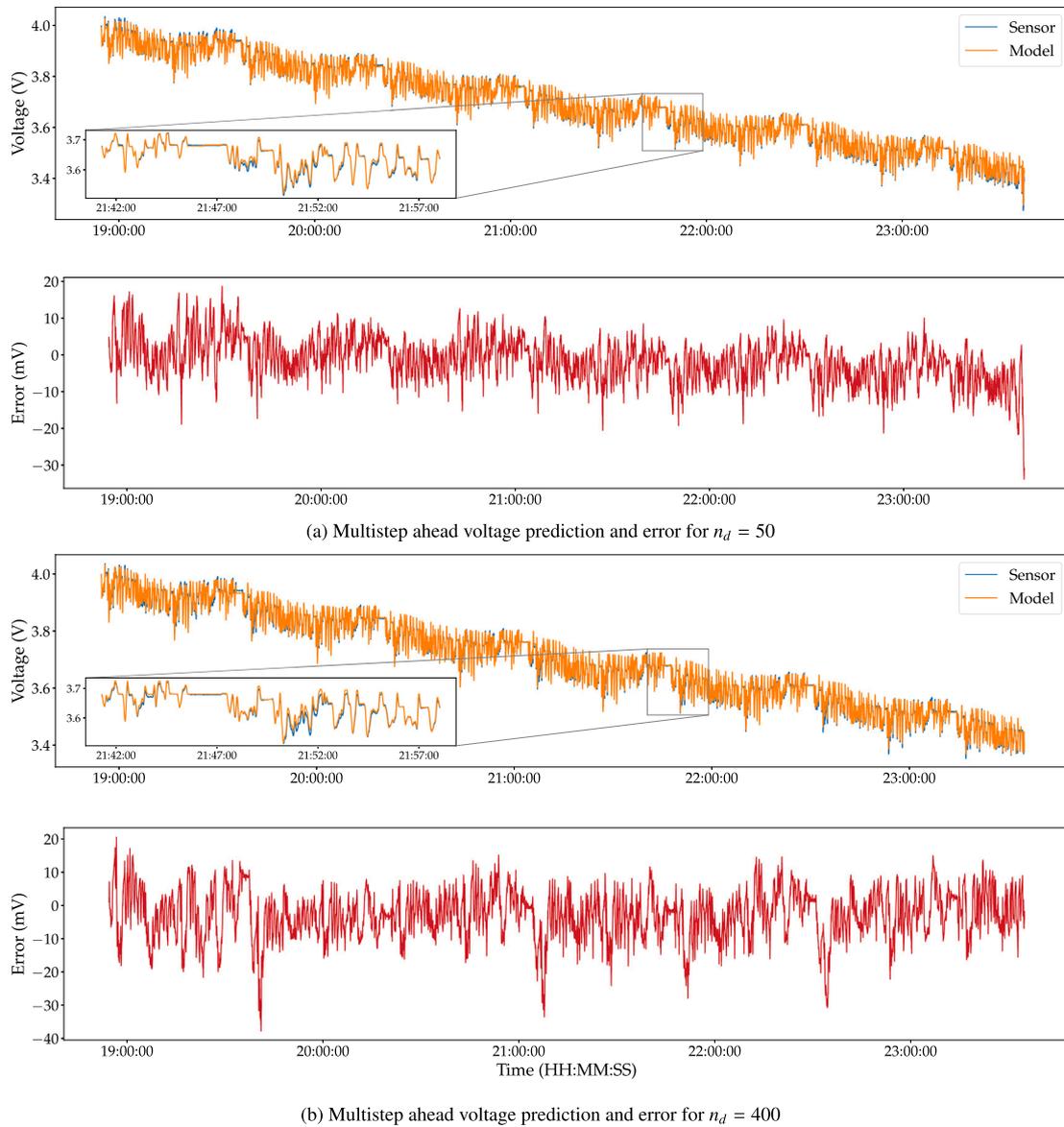


Fig. 11. Individual cell-based training: a comparative analysis of multi-step ahead voltage prediction for a test cycle (92 % SOH, with SOC varying from 80 % to 20 %) of cell W5, with n_d at 50 and 400 steps.

cell-based training. MAPE follows a similar pattern. This demonstrates the model's improved performance and generalization when trained on a diverse set of cycles from multiple cells.

Overall, both individual and multiple cell-based training scenarios suggest that shorter n_d results in lower errors. While RMSE and MAPE values increase with longer n_d , the increases are moderate, with

Table 3

Multiple cell-based training: performance metrics for multi-step ahead voltage prediction, trained and validated on sparse cycles from multiple cells and tested on the cycles of the holdout cell with varying n_d . (Note: Bold values indicate the lowest error in each column.)

n_d	Train			Validation			Test		
	RMSE (mV)	MAPE	Max (mV)	RMSE (mV)	MAPE	Max (mV)	RMSE (mV)	MAPE	Max (mV)
50	5.3	0.11 %	81.2	5.3	0.11 %	40.5	5.1	0.11 %	35.0
100	5.8	0.12 %	77.0	5.9	0.12 %	38.8	6.4	0.14 %	40.6
200	7.3	0.15 %	94.0	7.4	0.16 %	53.1	7.4	0.16 %	39.1
300	6.7	0.14 %	96.4	6.7	0.14 %	53.3	7.0	0.15 %	48.6
400	7.0	0.14 %	90.3	6.9	0.15 %	48.4	7.5	0.16 %	49.5

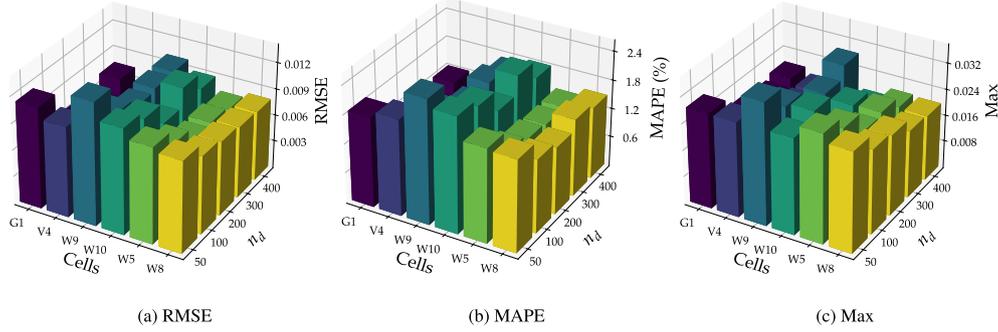


Fig. 12. Individual cell-based training: performance metrics for SOC estimation, evaluated across the test cycles of each cell with varying n_d .

MAPE ranging from 0.11 % to 0.16 %. Considering a sampling frequency of 1 Hz, the results show that the decoder can accurately predict ageing-aware voltage 50–400 seconds ahead. These results confirm the model's ability to maintain accuracy over longer prediction horizons using a shallow and modified RNN decoder model, which offers better interpretability compared to standard RNN networks.

7.2. State of charge

The performance metrics for SOC estimation across various test cycles for each cell, with different n_d , are presented in Fig. 12. The results indicate that SOC estimation accuracy improves with increasing n_d , as evidenced by lower RMSE, MAPE, and Maximum error values. On average, the MAPE error for each cell varies from 2 % to 1.5 % as n_d increases from 50 to 400. This trend is further illustrated in Fig. 13, which compares SOC estimation and ground truth for one of the test cycle (92 % SOH) of cell W5 for the range of n_d values.

The SOC estimation results for the multiple cell-based training, validation, and testing on a holdout cell are shown in Table 4. The results are comparable to or better than those from the single cell-based training, with MAPE values decreasing from 1.98 % to 1.65 % as n_d increases from 50 to 400 in the test cycles. This improvement is due to the inclusion of many cycles from various degradation levels across multiple cells in the training set.

The improvement in SOC estimation with increasing n_d is due to the learning process based on the multiple shooting method, as described by the dynamic loss function in Eq. (13). This method uses only the cycle's SOC BCs to learn each batch's SOC. For every batch, the SOC estimate $h_{soc,n}[k_m]$ is treated as a free variable during optimization, constrained by the previous batch estimate ($h_{soc,n}[k_{m-1}]$) and the Coulomb counting-based update for the lag between consecutive batches. Consequently, as n_d decreases, the number of batches (M) within each cycle increases, potentially increasing the overall error due to the more frequent estimation required by the dynamic loss function. However, the error remains relatively low even for $n_d = 50$, with RMSE around 0.0107 and MAPE approximately 198 % in the test set, as shown in Table 4. These results demonstrate that the proposed model can estimate the SOC across different ageing levels using only SOC BCs of sparse intermittent cycles.

7.3. Health indicator: h_{cap}

The capacity-based health indicator $h_{cap,n}[k_m]$, also referred to as SOH, is estimated using the multiple shooting-based SOC dynamic loss function described in Eq. (13). During the optimization process, $h_{cap,n}[k_m]$ is considered as a free variable in addition to $h_{soc,n}[k_m]$. Given the absence of explicit known dynamics for h_{cap} , its accuracy is affected by the accuracy of SOC estimation. As discussed in Section 7.2, lower values of n_d lead to higher SOC estimation errors, subsequently affecting the accuracy of SOH estimation. The maximum error for some of the test cycles goes beyond 5 % for $n_d < 200$ and hence not included here.

Performance metrics for test cycles with higher values of n_d in individual cell-based training are shown in Fig. 15. The estimated SOH for all train, validation and test cycles is compared with the ground truth obtained through RPTs in Fig. 14. In the case of multiple cell-based training, the performance metrics are shown in Table 5. The results are consistent with those from the individual cell-based training with maximum errors of 0.0224 and 0.0244 for $n_d = 300$ and $n_d = 400$, respectively. For each test cycle of the holdout cell, the estimated and ground truth SOH are plotted in Fig. 16. These results demonstrate that the proposed model can learn SOH using SOC BCs alone, without explicit SOH labels, especially for higher n_d values where SOC estimation error is lower.

7.4. Health indicator: h_r

The other health indicator $h_{r,n}[k_m]$, as previously detailed in the Section 4, remains constant across the decoder's prediction horizon (n_d). This indicator is unconstrained due to the absence of explicit physics-based dynamics and is updated using the gradient derived from the multi-step ahead voltage prediction loss. As illustrated in Fig. 17, $h_{r,n}[k_m]$ varies with SOC within a cycle, and it also varies with ageing, as shown for two different test cycles at different ageing levels.

The mean value of h_r across the SOC range of 80 % to 20 % is plotted against the cycle number (n) in Fig. 18. Fig. 18a shows the mean h_r values for training, validation, and test cycles for cell W5, while Fig. 18b depicts the mean h_r for the holdout cell W8's cycles. These plots indicate an increase in the mean h_r value with cycle number, suggesting the ageing state captured by h_r .

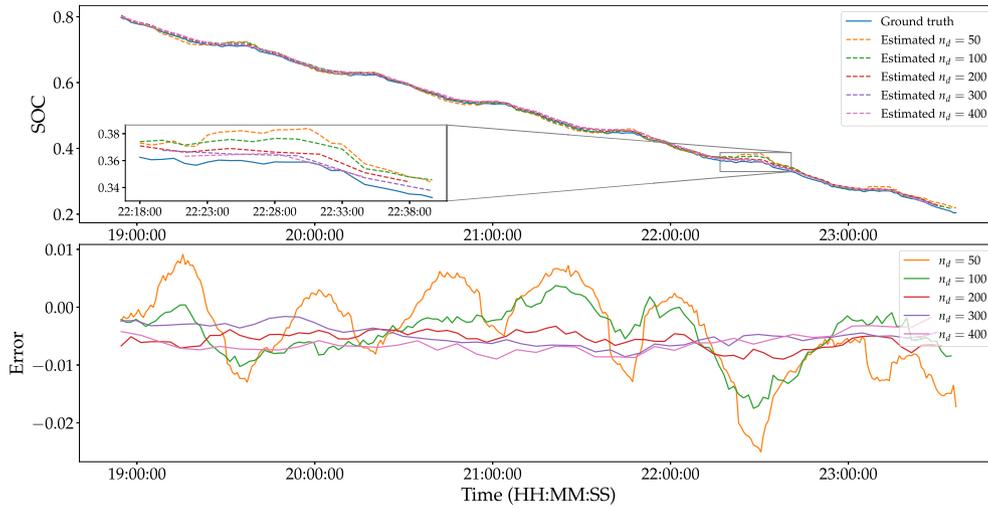
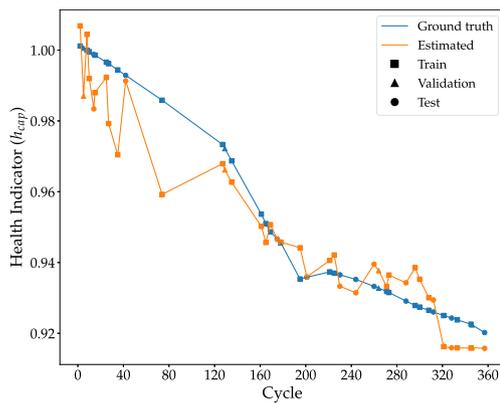


Fig. 13. Individual cell-based training: SOC estimation and error for cell W5 during a test cycle (92 % SOH) with n_d values ranging from 50 to 400 steps.

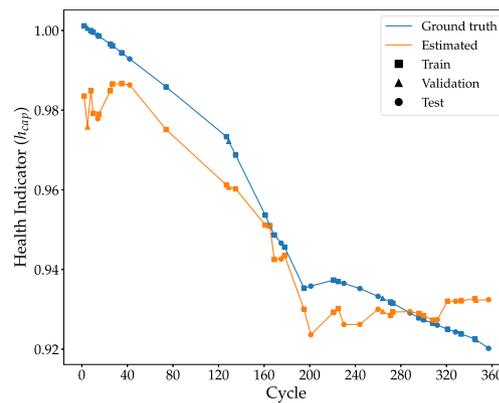
Table 4

Multiple cell-based training: performance metrics for SOC estimation, trained and validated on sparse cycles from multiple cells and tested on the cycles of the holdout cell with varying n_d . (Note: Bold values indicate the lowest error in each column.)

n_d	Train			Validation			Test		
	RMSE	MAPE	Max	RMSE	MAPE	Max	RMSE	MAPE	Max
50	0.0106	1.95 %	0.0375	0.0114	2.15 %	0.0318	0.0107	1.98 %	0.0283
100	0.0092	1.80 %	0.0261	0.0099	1.99 %	0.0263	0.0095	1.86 %	0.0270
200	0.0086	1.72 %	0.0273	0.0092	1.87 %	0.0266	0.0086	1.75 %	0.0200
300	0.0090	1.78 %	0.0237	0.0098	1.99 %	0.0228	0.0093	1.90 %	0.0313
400	0.0077	1.53 %	0.0219	0.0084	1.73 %	0.0204	0.0084	1.65 %	0.0237



(a) h_{cap} estimation for $n_d = 300$



(b) h_{cap} estimation for $n_d = 400$

Fig. 14. Individual cell-based training: SOH estimation for train, validation and test cycles of cell W5, with $n_d = 300$ and $n_d = 400$ steps.

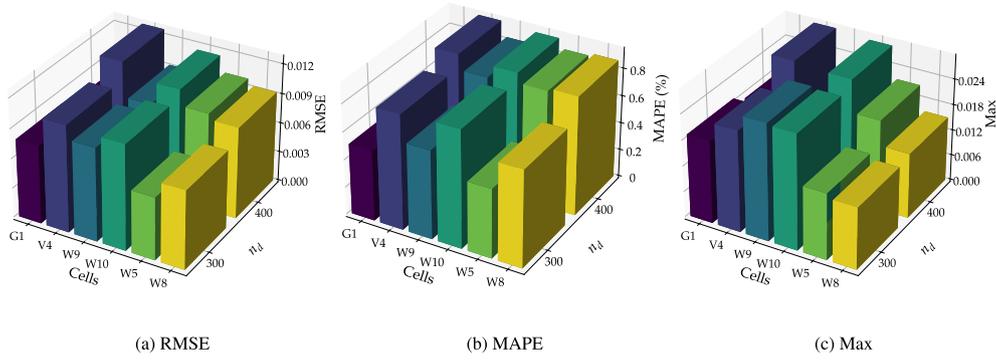
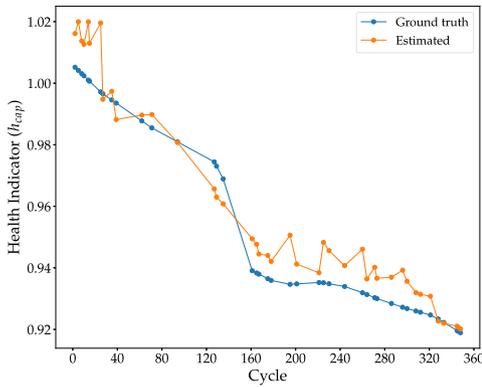


Fig. 15. Individual cell-based training: performance metrics for SOH estimation, evaluated across the test cycles of each cell with varying n_d .

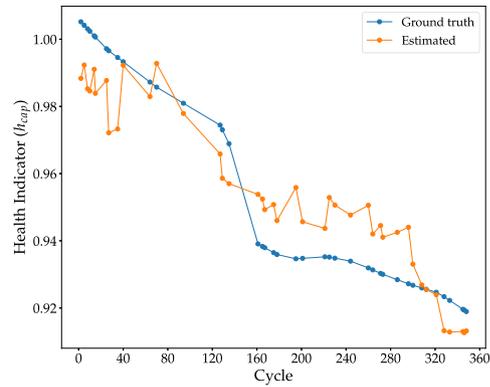
Table 5

Multiple cell-based training: performance metrics for SOH estimation, trained and validated on sparse cycles from multiple cells and tested on the cycles of the holdout cell with varying n_d .

n_d	Train			Validation			Test		
	RMSE	MAPE	Max	RMSE	MAPE	Max	RMSE	MAPE	Max
300	0.0094	0.71 %	0.0339	0.0098	0.80 %	0.0211	0.0118	1.01 %	0.0224
400	0.0151	1.35 %	0.0365	0.0154	1.38 %	0.0354	0.0146	1.35 %	0.0244



(a) h_{cap} estimation for $n_d = 300$



(b) h_{cap} estimation for $n_d = 400$

Fig. 16. Multiple cell-based training: SOH estimation for test cycles of holdout cell W8, with $n_d = 300$ and $n_d = 400$ steps.

7.5. Comparison with existing battery models

The proposed physics-informed encoder–decoder framework offers an effective balance between modeling accuracy and computational efficiency compared to traditional battery modeling approaches, based on Electrochemical (EC) models, or Equivalent Circuit Models (ECMs). Unlike EC models, such as the widely used Doyle-Fuller-Newman (DFN) model, which require solving complex coupled nonlinear partial differential equations (PDEs) and identifying over 20 parameters through cell tear-down procedures or carefully designed experiments [52,53], our method is chemistry agnostic. It leverages common physical relationships to capture battery behavior without extensive parameter identification or elaborate degradation modeling [54].

Similarly, ECMs are known for their runtime computational efficiency. However, they face challenges in capturing the nonlinear and time-varying dependencies of parameters on states like SOC and ageing. ECMs require online adaptive parameter estimation schemes and continuous recalibration of SOC–OCV relationships as capacity fades to maintain model accuracy. Additionally, the nonlinear filters used with ECMs for state estimation, such as extended Kalman filters or

particle filters, necessitate careful tuning of covariance matrices and noise parameters, leading to practical implementation complexities and sensitivity to initialization errors [55,56].

While most of the existing data-driven methods have focused on estimating specific battery states independently, such as SOC estimation [16,57], degradation estimation [58], or voltage prediction [19,20]. These methods require extensive labeled data for model training and validation. For instance, SOC estimation models need reference SOC values across the operating range, while degradation monitoring approaches rely on periodic capacity characterization tests. In contrast, the proposed approach simultaneously estimates battery states and predicts multi-step voltage using only voltage–current measurements and sparse SOC boundary conditions (6 %–7 % of total cycles).

We conducted a comparative evaluation of the proposed physics-informed encoder–decoder approach against the conventional EC and ECM models using the same dataset [30]. All experiments were performed on a single test cycle of cell W8, which comprised 18,835 s of UDDS discharge data. In the remainder of this section we detail the results of this comparison.

Table 6
Comparison of EC, ECM, and proposed approaches: voltage and SOC estimation errors (RMSE) and computational time. (Note: Bold values indicate the lowest error and time in the respective column.)

Approach	Voltage RMSE (mV)	SOC RMSE	Computational time
EC [52]	13.1–15	1.62×10^{-4} – 3.2×10^{-4}	3146 s
ECM [59]	16.6–18.9	1.0×10^{-3} – 1.56×10^{-2}	9.7 s
Physics-informed encoder–decoder	5–7.5	1.3×10^{-3} – 1.07×10^{-2}	0.09 s

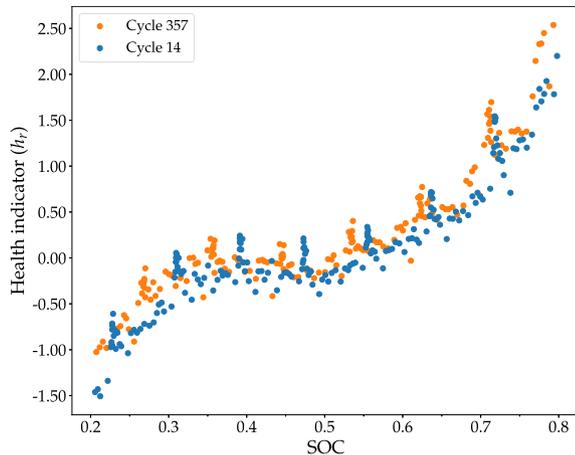


Fig. 17. Variation of the health indicator h_r with the SOC (20 %–80 %) for two different test cycles of cell W8, comparing an early cycle and a later cycle.

7.5.1. Accuracy comparison

Regarding accuracy of voltage estimation, as captured by the RMSE, our approach achieves substantially lower values compared to EC and ECM models, as shown in Table 6. In addition, our model accurately predicts multi-step ahead voltage over a range of 50–400 s. In terms of SOC estimation, while the EC model achieves very low errors, it relies on a detailed and computationally intensive parameter identification process. The ECM, although computationally efficient, shows higher SOC errors compared to our method. Notably, our physics-informed approach strikes a favorable balance by delivering competitive SOC estimation with chemistry-agnostic physics, thereby demonstrating its practical advantage.

7.5.2. Computational efficiency comparison

Computational efficiency is as critical as accuracy for practical battery model deployment. All inference computations were conducted on a benchmark platform using a Quad-Core Intel Core i7 processor running at 2.8 GHz, ensuring fair performance comparisons. In our experiments, the DFN-based EC model was implemented via the COBRAPRO framework using the SUNDIALS IDA solver and CasADi's automatic differentiation [60–62]. As summarized in Table 6, the DFN model required approximately 3146 s (over 52 min) due to the computational overhead of discretizing the governing PDEs into a differential algebraic system. Although ECMs reduce computational demand, their reliance on continuous online adaptive parameter estimation results in inference times of around 9.7 s. In contrast, the proposed physics-informed encoder–decoder framework delivers extremely fast inference times of approximately 0.09 s, although one-off model training takes some hours, as described at the beginning of Section 7.

In summary, the experimental results confirm that the proposed physics-informed encoder–decoder framework outperforms traditional EC and ECM models by achieving lower voltage estimation errors and maintaining competitive SOC estimation, all while significantly reducing inference times. This comprehensive performance improvement, evident from the results in Table 6, indicates that our approach offers a practical advantage in terms of both accuracy and efficiency.

7.6. Challenges in real-world battery conditions and future directions

While our experiments using the Stanford laboratory dataset show that the proposed physics-informed encoder–decoder model accurately predicts multi-step ahead voltage, estimates SOC and infers battery health indicators (SOH), it is important to recognize that laboratory conditions cannot fully replicate the diverse environments experienced in real-world electric vehicle (EV) operations. In practice, battery performance is influenced by several factors that are inherently variable and less controllable than lab settings. Here, we discuss some of the critical challenges and outline potential future directions to ensure the adaptability of the proposed model under actual operating conditions.

Diverse user behaviors. In controlled laboratory tests, cells are cycled under standardized protocols (e.g., UDDS discharge profiles) that aim to mimic average driving conditions. However, in real-world scenarios, driving behavior varies widely among users. Aggressive acceleration, frequent braking, and inconsistent driving patterns lead to different load profiles and transient responses. Although our model uses a high dynamic discharge profile, the variability introduced by diverse driver behavior could affect the accuracy of SOC and SOH estimation and voltage prediction. Future work should incorporate driver-specific usage patterns through additional sensor data or tailored training datasets with synthetic drive cycles or real-world data to improve the model's adaptability to a broader range of driving styles [63].

Varying ambient conditions. Laboratory experiments are typically conducted at constant or mildly fluctuating temperatures, yet actual EVs operate under widely varying ambient conditions. Seasonal and geographic temperature variations influence battery electrochemical dynamics, degradation rates, and thermal behavior. Such ambient variations can alter both the magnitude and evolution of the measured voltage signals. A promising future direction is integrating temperature signals as inputs to the model.

Inconsistent charging patterns. Charging protocols in lab experiments are precise and consistent, using fixed charging profiles. Conversely, EV charging in the field is often irregular; drivers engage in partial charging, experience fluctuating charging rates, or use fast charging methods that differ significantly from standardized protocols. These inconsistencies can lead to varying degradation rates not observed in the training dataset and also make it difficult to determine the SOC boundary conditions. Hence, future studies could benefit from incorporating a wider variety of charging scenarios. This might include hybrid training approaches combining controlled laboratory data with field-collected charge profiles or simulating diverse charging behaviors during model validation.

Linking real-world challenges to the proposed model. The physics-informed encoder–decoder framework proposed in this work is designed with interpretability and computational efficiency in mind. However, adaptation to real-world challenges is essential for practical deployment. Future model iterations could further improve its generalizability by expanding the training set to include diverse cycling and charging patterns, either through field data acquisition or realistic simulation. Incorporating additional sensors (e.g., temperature or driver behavior metrics) into the model's input space might also help capture thermal influences.

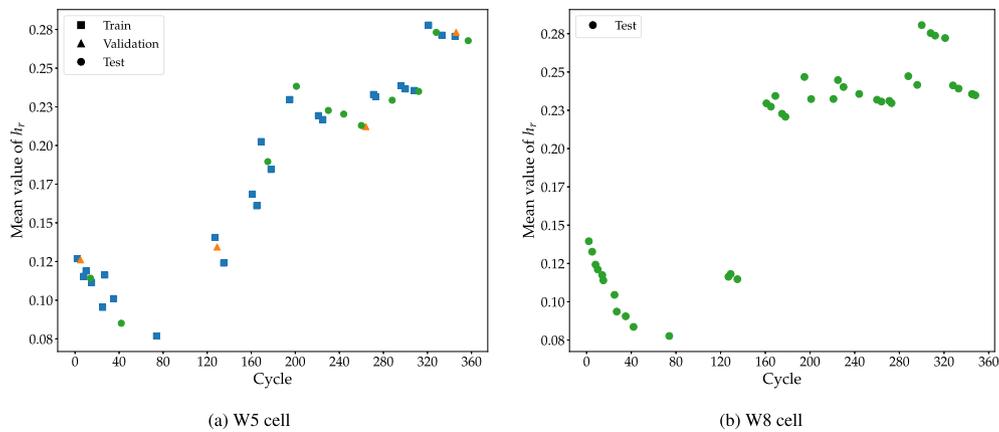


Fig. 18. Mean value of h_r versus cycle number for cell W5 (left) and W8 (right), where the mean h_r is calculated as the SOC varies from 80 % to 20 % within each cycle.

In summary, while the current results using laboratory-based data are promising, addressing these real-world challenges remains an important next step. By integrating more diverse data sources, we aim to extend the capabilities of our physics-informed model toward a robust and practical battery management solution.

8. Conclusion

This study presents a physics-informed encoder–decoder framework to model the multi-timescale dynamics of lithium-ion batteries. By separating slow-changing states, such as state of charge (SOC) and state of health (SOH), from fast voltage dynamics, the model captures the batteries’ multi-timescale behavior, increasing interpretability compared to conventional machine learning methods that often act as black boxes. The integration of physics-guided architecture and physics-informed loss functions enables direct mapping of the latent space to quantifiable battery states without requiring explicit SOC and SOH labels, using only SOC boundary conditions (BCs), thereby addressing the common challenge of the unavailability of labeled data. We validated the model using Stanford’s open-source lithium-ion battery ageing dataset, which features dynamic discharge profiles aimed at mimicking electric vehicle driving patterns. The model is trained on sparse intermittent cycle data, which further reduces the need for SOC BCs, as the model effectively learns SOC and SOH dynamics without requiring BCs for every cycle. The model shows consistent performance across both single-cell and multi-cell scenarios, accurately estimating SOC and SOH while providing reliable voltage predictions up to 400 s ahead. Future work will focus on improving model accuracy by, e.g., incorporating thermal effects into the model, applying transfer learning techniques to adapt the model to field data, and extending the framework to pack-level battery systems.

CRediT authorship contribution statement

Tushar Desai: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Alexander J. Gallo:** Writing – review & editing, Methodology, Conceptualization. **Riccardo M.G. Ferrari:** Writing – review & editing, Supervision, Resources, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Tushar Desai reports financial support was provided by AB Volvo. Dr. Riccardo Ferrari reports a relationship with AB Volvo that includes:

funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This document is the result of the research project funded by Volvo AB (Sweden), and NWO (Netherlands) under TKI-HTSM grant 21.0056.

Appendix A. Model architecture

See Tables A.7 and A.8 for the model architecture.

Table A.7
Encoder model architecture.

Layer	Layer type	Parameters	Stage
1	Input layer	0	First
2	1D convolution	63	First
3	1D convolution	80	First
4	1D convolution	112	First
5	1D convolution	220	First
6	1D convolution	315	First
7	Flatten	0	First
8(a)	Dense layer (h_r)	488	First
9(a)	Output layer (h_r)	9	First
8(b)	Dense layer (h_{soc})	488	First
9(b)	Output layer (h_{soc})	9	First
8(c)	Dense layer (\hat{h}_{cap})	488	First
9(c)	Output layer (\hat{h}_{cap})	9	First
10	Padding layer	0	Transition
11	Masking layer	0	Transition
12	1D convolution	153	Second
13	1D convolution	155	Second
14	1D convolution	252	Second
15	Flatten	0	Second
16	Dense layer	40	Second
17	Output layer (h_{cap})	6	Second
Total trainable parameters:			2887

Table A.8
Decoder model architecture.

Layer	Layer type	Parameters
1	Modified RNN	31
Total trainable parameters:		31

Data availability

The data used in this study is from Stanford's open-source battery dataset [30]. The code is available upon request.

References

- Fuller TF, Doyle M, Newman J. Relaxation phenomena in lithium-ion-Insertion cells. *J Electrochem Soc* 1994;141:982.
- Fuller TF, Doyle M, Newman J. Simulation and optimization of the dual lithium ion insertion cell. *J Electrochem Soc* 1994;141:1.
- Atlung S, West K, Jacobsen T. Dynamic aspects of solid solution cathodes for electrochemical power sources. *J Electrochem Soc* 1979;126:1311–21.
- Miguel E, Plett GL, Trimboli MS, Oca L, Iraola U, Bekaert E. Review of computational parameter estimation methods for electrochemical models. *J Energy Storage* 2021;44:103388.
- Safari M, Morcrette M, Teyssoit A, Delacourt C. Multimodal physics-based aging model for life prediction of li-ion batteries. *J Electrochem Soc* 2008;156:A145.
- Arora P, Doyle M, White RE. Mathematical modeling of the lithium deposition overcharge reaction in lithium-ion batteries using carbon-based negative electrodes. *J Electrochem Soc* 1999;146:3543.
- Wood KN, Kazyak E, Chadwick AF, Chen K-H, Zhang J-G, Thornton K, et al. Dendrites and pits: untangling the complex behavior of lithium metal anodes through operando video microscopy. *ACS Cent Sci* 2016;2:790–801.
- Xu R, Yang Y, Yin F, Liu P, Cloetens P, Liu Y, et al. Heterogeneous damage in li-ion batteries: experimental analysis and theoretical modeling. *J Mech Phys Solids* 2019;129:160–83.
- Li J, Adewuyi K, Lotfi N, Landers RG, Park J. A single particle model with chemical/Mechanical degradation physics for lithium ion battery state of health (SOH) estimation. *Appl Energy* 2018;212:1178–90.
- Edge JS, O'Kane S, Prosser R, Kirkaldy ND, Patel AN, Hales A, et al. Lithium ion battery degradation: what you need to know. *Phys Chem Chem Phys* 2021;23:8200–21.
- O'Kane SEJ, Ai W, Madabattula G, Alonso-Alvarez D, Timms R, Sulzer V, et al. Lithium-ion battery degradation: how to model it. *Phys Chem Chem Phys* 2022;24:7909–22.
- Lai X, Wang S, Ma S, Xie J, Zheng Y. Parameter sensitivity analysis and simplification of equivalent circuit model for the state of charge of lithium-ion batteries. *Electrochim. Acta* 2020;330:135239.
- Campestrini C, Kosch S, Jossen A. Influence of change in open circuit voltage on the state of charge estimation with an extended kalman filter. *J Energy Storage* 2017;12:149–56.
- Fink O, Wang Q, Svensén M, Dersin P, Lee W-J, Ducoffe M. Potential, challenges and future directions for deep learning in prognostics and health management applications, eng. *Appl Artif Intell* 2020;92:103678.
- Li W, Sengupta N, Dechent P, Howey D, Annaswamy A, Sauer DU. Online capacity estimation of lithium-ion batteries with deep long short-term memory networks. *J. Power Sources* 2021;482:228863.
- Chemali E, Kollmeyer PJ, Preindl M, Emadi A. State-of-charge estimation of li-ion batteries using deep neural networks: a machine learning approach. *J. Power Sources* 2018;400:242–55.
- Li Y, Liu K, Foley AM, Zülke A, Bercebar M, Nanini-Maury E, et al. Data-driven health estimation and lifetime prediction of lithium-ion batteries: a review, renew. *Sustain Energy Rev* 2019;113:109254.
- Sulzer V, Mohtat P, Aitio A, Lee S, Yeh YT, Steinbacher F, et al. The challenge and opportunity of battery lifetime prediction from field data. *Joule* 2021;5:1934–55.
- Zhao H, Chen Z, Shu X, Shen J, Liu Y, Zhang Y. Multi-step ahead voltage prediction and voltage fault diagnosis based on gated recurrent unit neural network and incremental training. *Energy* 2023;266:126496.
- Hong J, Wang Z, Yao Y. Fault prognosis of battery system based on accurate voltage abnormality prognosis using long short-term memory neural networks. *Appl Energy* 2019;251:113381.
- Gedon D, Wahlström N, Schön TB, Ljung L. Deep state space models for nonlinear system identification. *IFAC-PapersOnLine* 2021;54:481–86.
- Masti D, Bemporad A. Learning nonlinear state-space models using autoencoders. *Automatica* 2021;129:109666.
- Champion K, Lusch B, Kutz JN, Brunton SL. Data-driven discovery of coordinates and governing equations. *Proc Natl Acad Sci USA* 2019;116:22445–51.
- Chen J, Feng X, Jiang L, Zhu Q. State of charge estimation of lithium-ion battery using denoising autoencoder and gated recurrent unit recurrent neural network. *Energy* 2021;227:120451.
- Wu L, Zhang Y. Attention-based encoder-decoder networks for state of charge estimation of lithium-ion battery. *Energy (Oxf)* 2023;268:126665.
- Bian C, He H, Yang S, Huang T. State-of-charge sequence estimation of lithium-ion battery based on bidirectional long short-term memory encoder-decoder architecture. *J. Power Sources* 2020;449:227558.
- Wei M, Ye M, Wang Q, Xinxi-Xu JPT. Remaining useful life prediction of lithium-ion batteries based on stacked autoencoder and Gaussian mixture regression. *J Energy Storage* 2022;47:103558.
- Dang W, Liao S, Yang B, Yin Z, Liu M, Yin L, et al. An encoder-decoder fusion battery life prediction method based on Gaussian process regression and improvement. *J Energy Storage* 2023;59:106469.
- Desai T, Ferrari RMG. Separating Multiscale battery dynamics and predicting multi-step ahead voltage simultaneously through a data-driven approach. In: *VPPC*; 2023. p. 1–6.
- Pozzato G, Allam A, Onori S. Lithium-ion battery aging dataset based on electric vehicle real-driving profiles. *Data Brief* 2022;41:107995.
- Pozzato G, Allam A, Pulvirenti L, Negroita GA, Paxton WA, Onori S. Analysis and key findings from real-world electric vehicle field data. *Joule* 2023;7:2035–2053.
- von Bülow F, Wassermann M, Meisen T. State of health forecasting of lithium-ion batteries operated in a battery electric vehicle fleet. *J Energy Storage* 2023;72:108271.
- von Bülow F, Meisen T. A review on methods for state of health forecasting of lithium-ion batteries applicable in real-world operational conditions. *J Energy Storage* 2023;57:105978.
- Mori H. Transport, collective motion, and Brownian motion. *Progr Theoret Phys* 1965;33:423–55.
- Khalil HK. *Nonlinear systems*. Prentice-Hall; 2002.
- Sulzer V, Mohtat P, Pannala S, Siegel JB. Accelerated battery lifetime simulations using adaptive inter-cycle extrapolation algorithm. *J Electrochem Soc* 2021;168:120531.
- Wehmeyer C, Noé F. Time-lagged autoencoders: deep learning of slow collective variables for molecular kinetics. *J Chem Phys* 2018;148:241703.
- Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. *Adv Neural Inf Process Syst* 2014;27.
- Wiewel S, Becher M, Thuerey N. Latent space physics: towards learning the temporal evolution of fluid flow. *Comput Graph Forum* 2019;38:71–82.
- Kiranyaz S, Avci O, Abdeljaber O, Ince T, Gabbouj M, Inman DJ. 1D convolutional neural networks and applications: a survey. *Mech Syst Signal Process* 2021;151:107398.
- TensorFlow. Time distributed layer; 2015. <https://www.tensorflow.org>. [Accessed: 2024-5-22].
- Ribeiro AH, Tiels K, Umenberger J, Schön TB, Aguirre LA. On the smoothness of nonlinear system identification. *Automatica* 2020;121:109158.
- Beintema GI, Schoukens M, Tóth R. Deep subspace encoders for nonlinear system identification. *Automatica* 2023;156:111210.
- Glort X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Teh YW, Titterton M, editors. *Proceedings of the thirteenth international conference on artificial intelligence and statistics, volume 9 of proceedings of machine learning research, PMLR, Chia Laguna Resort, Sardinia, Italy; 2010*. p. 249–56.
- Wang S, Sankaran S, Wang H, Perdikaris P. An expert's guide to training physics-informed neural networks. *arXiv:2308.08468 [cs.LG]*; 2023.
- Werbos PJ. Backpropagation through time: what it does and how to do it. *Proc IEEE* 1990;78:1550–60.
- Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: *Proceedings of the 27th international conference on neural information processing systems - Volume 2, NIPS'14; Cambridge, MA, USA: MIT Press; 2014*. p. 3104–12.
- Kingma DP, Ba J. Adam: a method for stochastic optimization. In: *International conference for learning representations; 2014*.
- Geslin A, van Vlijen B, Cui X, Bhargava A, Asinger PA, Braatz RD, et al. Selecting the appropriate features in battery lifetime predictions. *Joule* 2023;7:1956–65.
- Li W, Rentemeister M, Badeda J, Jöst D, Schulte D, Sauer DU. Digital twin for battery systems: cloud battery management system with online state-of-charge and state-of-health estimation. *J Energy Storage* 2020;30:101557.
- Yang S, Zhang Z, Cao R, Wang M, Cheng H, Zhang L, et al. Implementation for a cloud battery management system based on the CHAIN framework. *Energy AI* 2021;5:100088.
- Ha S, Onori S. COBRAPRO: an open-source software for the Doyle-Fuller-Newman model with co-simulation parameter optimization framework. *J Electrochem Soc* 2024;171:090522.
- le Roux FA, Bergveld HJ, Donkers MCF. Improved parameter estimation of the Doyle-Fuller-Newman model by incorporating temperature dependence. *IFAC-PapersOnLine* 2023;56:6136–41.
- Reniers JM, Mulder G, Howey DA. Review and performance comparison of Mechanical-chemical degradation models for lithium-ion batteries. *J Electrochem Soc* 2019;166:A3189.
- Beelen H, Bergveld HJ, Donkers MCF. Joint estimation of battery parameters and state of charge using an extended Kalman filter: a single-parameter tuning approach. *IEEE Trans Control Syst Technol* 2021;29:1087–101.
- Hide C, Moore T, Smith M. Adaptive kalman filtering for low-cost INS/GPS. *J Navig* 2003;56:143–52.
- Zahid T, Xu K, Li W, Li C, Li H. State of charge estimation for electric vehicle power battery using advanced machine learning algorithm under diversified drive cycles. *Energy (Oxf)* 2018;162:871–82.
- Lu J, Xiong R, Tian J, Wang C, Hsu C-W, Tsou N-T, et al. Battery degradation prediction against uncertain future conditions with recurrent neural network enabled deep learning. *Energy Storage Mater* 2022;50:139–51.
- Li J, Wang Y, Ferrari RMG, Swevers J, Ding F. Online lithium-ion battery modeling and state of charge estimation via concurrent state and parameter estimation. *IFAC-PapersOnLine* 2024;58:462–67.
- Andersson JAE, Gillis J, Horn G, Rawlings JB, Diehl M. CasADi: a software framework for nonlinear optimization and optimal control. *Math Program Comput* 2019;11:1–36.
- Gardner DJ, Reynolds DR, Woodward CS, Balos CJ. Enabling new flexibility in the SUNDIALS Suite of nonlinear and Differential/algebraic equation solvers. *ACM Trans Math Softw* 2022;48:1–24.
- Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, et al. SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. *ACM Trans Math Softw* 2005;31:363–96.
- Moy K, Ganapathi D, Geslin A, Chueh W, Onori S. Synthetic duty cycles from real-world autonomous electric vehicle driving. *Cell Rep Phys Sci* 2023;4:101536.