applied sciences

MDPI

Article

Multi-Objective Hyperparameter Optimization of Lithium-Ion Battery State-of-Health Prediction Models

Andrés Barrantes Boza [1], Juan José Montero Jiménez [1] and Rob Vingerhoeds [2,*]

[1] Escuela de Ingeniería Electromecánica, Tecnológico de Costa Rica, Oriental, P.O. Box 159-7050, Cartago 30101, Cartago, Costa Rica; an.barrantes@estudiantec.cr (A.B.B.); juan.montero@itcr.ac.cr (J.J.M.J.)
[2] Federation ONERA-ISAE-SUPAERO-ENAC, Université de Toulouse, 10 Avenue Marc Pélegrin, BP 54032, 31055 Toulouse, France
* Correspondence: rob.vingerhoeds@isae-supaero.fr

Abstract

Selecting machine learning models for state-of-health (SOH) prediction in predictive maintenance involves trade-offs among competing criteria, such as predictive accuracy, robustness to measurement noise, and computational requirements. Although model comparisons are commonly based primarily on prediction error, this approach may overlook characteristics relevant to practical deployment, particularly in resource-constrained systems. This work proposes a multi-objective hyperparameter optimization framework for SOH prediction based on the nondominated sorting genetic algorithm II (NSGA-II), in which three commonly used prediction models are evaluated and compared: support vector regression (SVR), long short-term memory (LSTM), and physics-informed neural networks (PINNs). For each model, NSGA-II optimizes model-specific hyperparameters with respect to validation RMSE, sensitivity to measurement noise, and inference time. Noise sensitivity is quantified as the relative increase in RMSE across six Gaussian-noise levels ranging from 5% to 30%. The framework is evaluated through a case study using the lithium-ion battery aging dataset from the NASA Prognostics Center of Excellence. The resulting Pareto-optimal solutions quantify the trade-offs involved in model selection and support practitioners in identifying configurations that match their specific requirements.

Keywords: predictive maintenance; hyperparameter optimization; trade-off analysis; multi-objective optimization; state of health; lithium-ion batteries; Pareto frontier

Check for updates

Academic Editors: Alexandru Stanciu and Rui Araújo

Received: 25 September 2026
Revised: 5 November 2026
Accepted: 10 November 2026
Published: 15 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Industrial assets are subject to degradation mechanisms that may ultimately lead to failure. Although asset failure is generally undesirable, its consequences can be particularly severe, motivating proactive measures to mitigate them. Predictive maintenance addresses this need by seeking to anticipate failures, thereby enhancing safety and reducing maintenance costs [1].

Predictive maintenance (PdM) relies on specialized systems that perform diagnostic and prognostic functions [2]. These systems support maintenance decision-making by enabling timely fault detection and failure prediction. However, their design and development pose challenges related to data quality, model selection, system integration, and real-world applicability.

In practice, model selection is often primarily based on predictive accuracy. However, this criterion alone may be insufficient for selecting models for predictive maintenance

systems. Other performance indicators, such as computational efficiency, generalization across varying operating conditions, and robustness to data imperfections, should also be considered when selecting a predictive maintenance model, but may be overlooked [3].

To address this limitation, this work investigates multi-objective hyperparameter optimization for state-of-health (SOH) prediction. Rather than selecting a single hyperparameter configuration solely with respect to prediction error, the proposed approach searches for configurations that represent different trade-offs among prediction error, sensitivity to noisy measurements, and inference time. Three SOH prediction approaches are considered: support vector regression (SVR), long short-term memory (LSTM), and physics-informed neural networks (PINNs).

The central hypothesis of this work is that hyperparameter selection should not be considered independently of the operational requirements of the predictive model. Different configurations of the same model architecture can exhibit substantially different predictive accuracy and sensitivity to measurement noise [4]. Consequently, optimizing the hyperparameters against multiple objectives can identify a set of non-dominated configurations. This avoids having to select a single configuration based on a single objective.

By considering multiple performance objectives, the framework enables practitioners to identify models that align with the specific requirements and constraints of their application. This multi-objective evaluation facilitates model selection based on multiple performance criteria, thereby supporting predictive maintenance solutions that better reflect application-specific requirements.

The main contributions of this work include the use of a genetic algorithm in a multi-objective hyperparameter optimization formulation in which multiple performance objectives are incorporated directly into the optimization process, yielding a set of non-dominated configurations that represent different trade-offs among the objectives. Additionally, synthetic perturbations are applied to the validation data to evaluate model sensitivity at six noise levels, ranging from 5% to 30%. From the final non-dominated configurations, representative configurations corresponding to favorable values for individual objectives are selected and evaluated using additional error and computational metrics, providing a broader characterization of the selected configurations beyond the objectives directly optimized.

The remainder of this paper is organized as follows: Section 2 reviews previous work and existing techniques for SoH prediction. Section 3 presents the theoretical background required to understand the proposed framework. Section 4 introduces the proposed multi-objective evaluation framework, which considers various aspects of model performance. Section 5 discusses the results and their implications for practical predictive maintenance applications. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2. Related Work

The trade-off between predictive accuracy and resource constraints makes model selection an important consideration [5]. No single method consistently outperforms others across different datasets and applications. Therefore, model comparison should consider multiple performance criteria when assessing SOH prediction models [3].

Existing studies have also examined trade-offs among performance criteria [6,7], conducted multi-metric analyses [8,9], and proposed model-selection frameworks [5]. Feature-importance analyses have likewise been investigated [10]. Furthermore, previous studies provide qualitative comparisons of the advantages and disadvantages of different SOH prediction models, offering insights into their suitability for specific applications [11,12].

2.1. Research Gap and Motivation

Despite the extensive literature on lithium-ion battery SOH prediction, model comparisons are often based on a single selected or optimized configuration for each architecture, with predictive error used as the primary evaluation criterion. This pattern is also evident in several literature reviews, where the model with the lowest RMSE is identified as the preferred model primarily on the basis of this metric [13]. While predictive accuracy provides a straightforward basis for initial comparison, relying on a single metric may be insufficient for application-specific model selection. Practical considerations, such as computational requirements in embedded systems, mobile devices, or industrial environments, may not be explicitly incorporated into the model-selection process [7].

When SOH estimators are considered for deployment beyond the laboratory setting, purely data-driven models can achieve high accuracy under controlled conditions yet learn false correlations under noisy operation, causing prediction failures on unseen assets, while physics–data fusion approaches remain challenged by sparse physical features in fragmented field data and by nonlinear deviations in late-aging stages [14].

A further limitation is that hyperparameter optimization and model comparison are often treated as separate problems. A model may appear to perform better based on its architecture, while its observed performance can also vary with the selected hyperparameters. Consequently, comparing architectures using a single selected configuration may not fully characterize the trade-offs that arise from different hyperparameter choices. Treating hyperparameter selection as a multi-objective problem provides a means of explicitly characterizing these trade-offs rather than reducing them to a single objective value.

Accordingly, this study proposes a systematic, multi-metric evaluation framework for selecting SOH prediction models in predictive maintenance applications, formulating hyperparameter selection as a multi-objective optimization problem. The optimization simultaneously minimizes validation RMSE, noise sensitivity, and inference time. The resulting non-dominated configurations provide a quantitative representation of the trade-offs among these objectives for SVR, LSTM, and PINN models. This formulation provides a quantitative basis for evaluating model configurations according to application-specific performance requirements rather than relying solely on a single performance criterion.

This work forms part of a broader research line on predictive maintenance system design. Previous work within this research line has addressed the conceptual phase of predictive maintenance system development, from gathering needs and stakeholder expectations to formalizing and prioritizing requirements and proposing a generic logical architecture [15]. Within the same research line, a decision support system based on Case-Based Reasoning has also been proposed to retrieve suitable models for different diagnostic and prognostic functions in predictive maintenance systems [16].

2.2. Current MOO Techniques in Hyperparameter Selection for ML

Hyperparameter optimization (HPO) is an important component of machine learning workflows, as model performance can depend strongly on the selection of hyperparameters [17]. Traditionally, HPO has been formulated as a single-objective problem, typically minimizing a performance metric such as prediction error. However, in many practical applications, including battery state-of-health (SOH) prediction, additional performance criteria or constraints may also be relevant when evaluating and selecting model configurations. This need has motivated the adoption of multi-objective optimization (MOO) techniques, which seek to approximate the set of Pareto-optimal solutions representing different trade-offs among competing objectives [18,19].

Several algorithmic approaches have been applied to HPO, ranging from simple search strategies to model-based and population-based methods. Random search pro-

vides a competitive baseline and can outperform grid search in high-dimensional search spaces, particularly when only a subset of hyperparameters has a substantial influence on performance [20].

More advanced approaches include Bayesian optimization, which builds a probabilistic surrogate model of the objective function to guide the selection of promising configurations, and evolutionary algorithms (EA), which use population-based search and can accommodate mixed and hierarchical search spaces [21]. Multi-fidelity methods, such as Hyperband, can further improve resource allocation by dynamically assigning computational resources to different configurations based on their observed performance [22].

Additionally, HPO approaches can be extended or adapted to optimize multiple performance criteria simultaneously [23]. Multi-objective formulations have also been adopted in adjacent battery-management problems, where competing operational objectives such as degradation, efficiency, and consumption are optimized jointly; however, such formulations do not address the selection of prediction-model configurations and selected metrics considered in this study [24].

Among state-of-the-art algorithms used for solving multi-objective optimization problems, NSGA-II is a widely used dominance-based genetic algorithm, which identifies non-dominated solutions while maintaining diversity along the approximated Pareto front [25]. Although such MOO methods provide established approaches for handling competing objectives, their systematic use for hyperparameter selection in lithium-ion battery SOH prediction remains limited in the reviewed studies [26,27].

3. Theoretical Basis

In this work, a trade-off is defined as a situation in which improving one performance criterion requires sacrificing performance in at least one other criterion. The proposed framework represents these criteria as explicit objective functions and identifies non-dominated configurations.

The prediction problem is formulated as a supervised one-step-ahead regression task. For forecast example i, the input $\mathbf{x}_i$ contains the voltage, current, temperature, and cycle information available through the preceding cycle, while the target y_i is the measured discharge capacity at the next cycle, expressed in ampere-hours (Ah). The LSTM uses a sequence of preceding-cycle feature vectors; the SVR and PINN use their specified cycle-level input representations. SOH is derived from capacity for reporting and is not the quantity optimized during training. Each prediction model approximates the mapping

$$\hat{y}_i = f_{\boldsymbol{\lambda}}(\mathbf{x}_i), \tag{1}$$

where $\boldsymbol{\lambda}$ denotes the hyperparameter configuration defining the model. The trainable model parameters are determined during the training process for each configuration $\boldsymbol{\lambda}$.

3.1. Prediction Model Descriptions

Several machine learning approaches have been investigated for lithium-ion battery SOH estimation. Data-driven models learn relationships between input variables and SOH directly from historical operational data, while physics-informed approaches additionally incorporate physical relationships into the learning formulation [28]. The three prediction approaches considered in this study (SVR, LSTM, and PINN) represent different ways of incorporating information into SOH prediction and therefore provide distinct model configurations for the proposed multi-objective evaluation framework.

Support vector machines (SVMs) are a family of supervised learning methods for classification and regression. The regression formulation is referred to as support vector regression (SVR) [29]. A feature mapping $\Phi(\mathbf{x})$ represents nonlinear relationships in a

transformed feature space. The corresponding kernel is $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, and the regression function can be expressed as

$$f(\mathbf{x}) = \boldsymbol{w}^T \Phi(\mathbf{x}) + b, \tag{2}$$

with $\boldsymbol{w}$ denoting the weight vector and b the bias term. SVR can be applied to datasets with relatively small sample sizes and can provide good generalization when the model is appropriately regularized [29].

In this study, the radial basis function (RBF) kernel is used to represent nonlinear relationships between the input features. It is defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \tag{3}$$

where γ controls the width of the RBF kernel. The SVR training problem with an ε-insensitive loss function is formulated as:

$$\min_{\boldsymbol{w}, b, \xi_i, \xi_i^*} \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \tag{4}$$

subject to the constraints:

$$\begin{cases} y_i - \boldsymbol{w}^T \Phi(\mathbf{x}_i) - b \leq \varepsilon + \xi_i, \\ \boldsymbol{w}^T \Phi(\mathbf{x}_i) + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, \ldots, N, \end{cases} \tag{5}$$

where C is the regularization parameter, ε defines the width of the ε-insensitive tube, and ξ_i and ξ_i^* are slack variables that account for deviations outside this tube.

Recurrent neural networks (RNNs) have been widely investigated for battery SOH prediction. LSTM is a recurrent neural network architecture designed to capture temporal dependencies through gated memory mechanisms and has been extensively applied to sequential battery degradation data [30].

LSTM networks introduce memory cells and gated mechanisms, including input, forget, and output gates, which regulate the flow and retention of information within the network. This structure facilitates the learning of long-term dependencies while mitigating the vanishing-gradient problem. These characteristics make LSTM suitable for modeling sequential dependencies in battery SOH prediction and have contributed to its widespread use for this task [31].

For a time step t with input X_t and previous hidden state H_{t-1}, the standard LSTM update equations for the forget gate f_t, input gate i_t, candidate cell state $\tilde{C}_t$, output gate o_t, cell state C_t, and hidden state H_t are presented in Equations (6)–(11), where W_f, W_i, W_c, and W_o denote the weight matrices, while b_f, b_i, b_c, and b_o denote the corresponding bias vectors. The function $\sigma(\cdot)$ represents the sigmoid activation function, and $\odot$ denotes the Hadamard (element-wise) product.

$$f_t = \sigma\left(W_f \cdot [H_{t-1}, X_t] + b_f\right) \tag{6}$$

$$i_t = \sigma(W_i \cdot [H_{t-1}, X_t] + b_i) \tag{7}$$

$$\tilde{C}_t = \tanh(W_c \cdot [H_{t-1}, X_t] + b_c) \tag{8}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{9}$$

$$o_t = \sigma(W_o \cdot [H_{t-1}, X_t] + b_o) \tag{10}$$

$$H_t = o_t \odot \tanh(C_t) \tag{11}$$

Physics-informed neural networks (PINNs) incorporate physical knowledge into data-driven learning by including physics-based relationships in the training formulation. In battery SOH prediction, this information can be represented through empirical degradation relationships and state-space dynamics, providing a physics-informed alternative to approaches based solely on input-output relationships [32–35] additionally, related hybrid physics–data prognostic frameworks have also been developed for mechanical components [36]. The applied formulation differs from detailed electrochemical models, such as pseudo-two-dimensional (P2D) Newman-type models, by using a reduced representation of the underlying battery dynamics.

Within this framework, we follow the dual-network architecture proposed in [37], the implementation separates the scalar prediction from an auxiliary dynamics network. The solution network $\mathcal{F}(\mathbf{z};\Phi)$ maps the standardized four-feature input to next-cycle capacity u. Automatic differentiation computes the derivatives of u with respect to the standardized feature coordinates; the last coordinate is the standardized cycle index. The dynamics network $\mathcal{G}(\mathbf{z}, u, u_t, \nabla_x u;\Theta)$ receives the input, output, and derivatives and estimates the temporal derivative. Both networks are multilayer perceptrons with sine activations in hidden layers.

The networks are trained using a composite loss function that incorporates data fidelity and physics-based constraints [37]:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \alpha\mathcal{L}_{\text{PDE}} + \beta\mathcal{L}_{\text{mono}} \tag{12}$$

where $\mathcal{L}_{\text{data}}$ is the batch mean squared error between standardized predicted and measured capacity , $\mathcal{L}_{\text{PDE}}$ is the squared residual $(u_t - \mathcal{G})^2$, and $\mathcal{L}_{\text{mono}}$ is the batch mean of u_t, penalizing positive capacity change with increasing cycle index . The parameters α and β weight these terms. In this implementation, the derivatives are with respect to standardized coordinates and the physics term is a learned residual constraint. The dynamics network contributes to training but is not called when producing deployed predictions, this implies that inference uses the solution network $\mathcal{F}$ alone. Also, this structure can facilitate transfer across different battery datasets or operating conditions by separating the representation of degradation dynamics from the SOH prediction network [37].

3.2. Multiobjective Optimization

Multiobjective optimization (MOO) addresses problems in which multiple, potentially competing objectives must be optimized simultaneously. Unlike single-objective optimization, which typically seeks a solution that minimizes or maximizes a single objective, MOO seeks a set of non-dominated solutions representing different trade-offs among the objectives. A solution is considered Pareto-optimal if no objective can be improved without degrading at least one other objective. The corresponding set of objective values forms the Pareto front in the objective space [38]. The general formulation of a multiobjective optimization problem can be expressed as:

$$\min_{\lambda\in\Omega}(f_1(\boldsymbol{\lambda}), f_2(\boldsymbol{\lambda}), \ldots, f_k(\boldsymbol{\lambda}))^\top \tag{13}$$

where $\Omega \subseteq \mathbb{R}^D$ denotes the feasible region of decision vectors, and $k \geq 2$ represents the number of objectives [38]. In the context of predictive-maintenance model selection, MOO provides a formulation for simultaneously considering competing performance criteria. This is relevant when model configurations must satisfy different performance requirements and resource constraints associated with their intended deployment [25].

Since all objectives are minimized, a solution $\boldsymbol{\lambda}^a$ is said to dominate a solution $\boldsymbol{\lambda}^b$ if

$$f_j(\boldsymbol{\lambda}^a) \leq f_j(\boldsymbol{\lambda}^b), \qquad \forall j, \tag{14}$$

with strict inequality for at least one objective. The resulting set of non-dominated decision vectors, referred to as the Pareto-optimal set, is defined as

$$\mathcal{P}_{\text{set}} = \left\{ \boldsymbol{\lambda} \in \Omega \,\middle|\, \nexists\, \boldsymbol{\lambda}' \in \Omega : \boldsymbol{\lambda}' \prec \boldsymbol{\lambda} \right\}. \tag{15}$$

The corresponding Pareto front in the objective space is given by

$$\mathcal{P}_{\text{front}} = F(\mathcal{P}_{\text{set}}). \tag{16}$$

4. Implementation

A multi-objective hyperparameter optimization framework was implemented for SVR, LSTM, and PINN. For each model family, a NSGA-II implementation using the pymoo Python library [39] identifies hyperparameter configurations that simultaneously minimize validation RMSE, sensitivity to measurement noise, and inference time. The optimization is performed independently for each model, and the resulting non-dominated configurations are compared using a common set of evaluation criteria.

4.1. Case Study

State of health (SOH) expresses the remaining battery capacity relative to a reference capacity. For retrospective reporting, it is defined as

$$\text{SOH}_i\ (\%) = \frac{C_i}{C_{\text{ref}}} \times 100, \qquad C_{\text{ref}} = \max_j C_j, \tag{17}$$

where C_i is the measured or predicted discharge capacity at cycle i and C_{ref} is the maximum measured discharge capacity over that battery's recorded life. SOH is therefore reported as a percentage. The models are trained to predict capacity (C_i), measured in Ah.

The study uses data from twelve lithium-ion batteries (B0005, B0006, B0007, B0018, B0045, B0046, B0047, B0048, B0053, B0054, B0055, and B0056) from the NASA Prognostics Center of Excellence (PCoE) dataset [40]. The dataset provides battery records in MATLAB .mat format, containing measurements collected over repeated charge and discharge cycles. These files were loaded using SciPy version 1.13.1 under Python version 3.12.3. Each loader constructs cycle-level records containing battery identifier, cycle index, measured discharge capacity , and mean voltage, current, and temperature. Rows with missing or non-finite required values are removed and the remaining rows are sorted by cycle. The feature vector used by the supplied implementations is $[V, I, T, t]$, where V, I, and T are cycle-level mean voltage, current, and temperature and t is cycle index. Measured discharge capacity in ampere-hours (Ah), is the model target. SOH is derived from capacity using Equation (17).

Figure 1 illustrates discharge-phase voltage, current, and temperature signals for batteries B0018, B0048, and B0056, all from the held-out test split. These measurements are summarized per cycle to obtain the mean voltage, mean current, and mean temperature used as model inputs .

The split is performed by battery. The shared split routine permutes batteries separately within each of three predefined groups and assigns two batteries per group to training, one to validation, and one to testing. The resulting split is train: B0005, B0006, B0045, B0046, B0054, B0053; validation: B0007, B0047, B0055; test: B0018, B0048, B0056. Cycles from a battery occur in only one partition. Validation data are used for hyperparameter selection, and test batteries are held out until final evaluation. All models perform

one-step-ahead prediction, using measurements through cycle $t-1$ to predict capacity at cycle t.

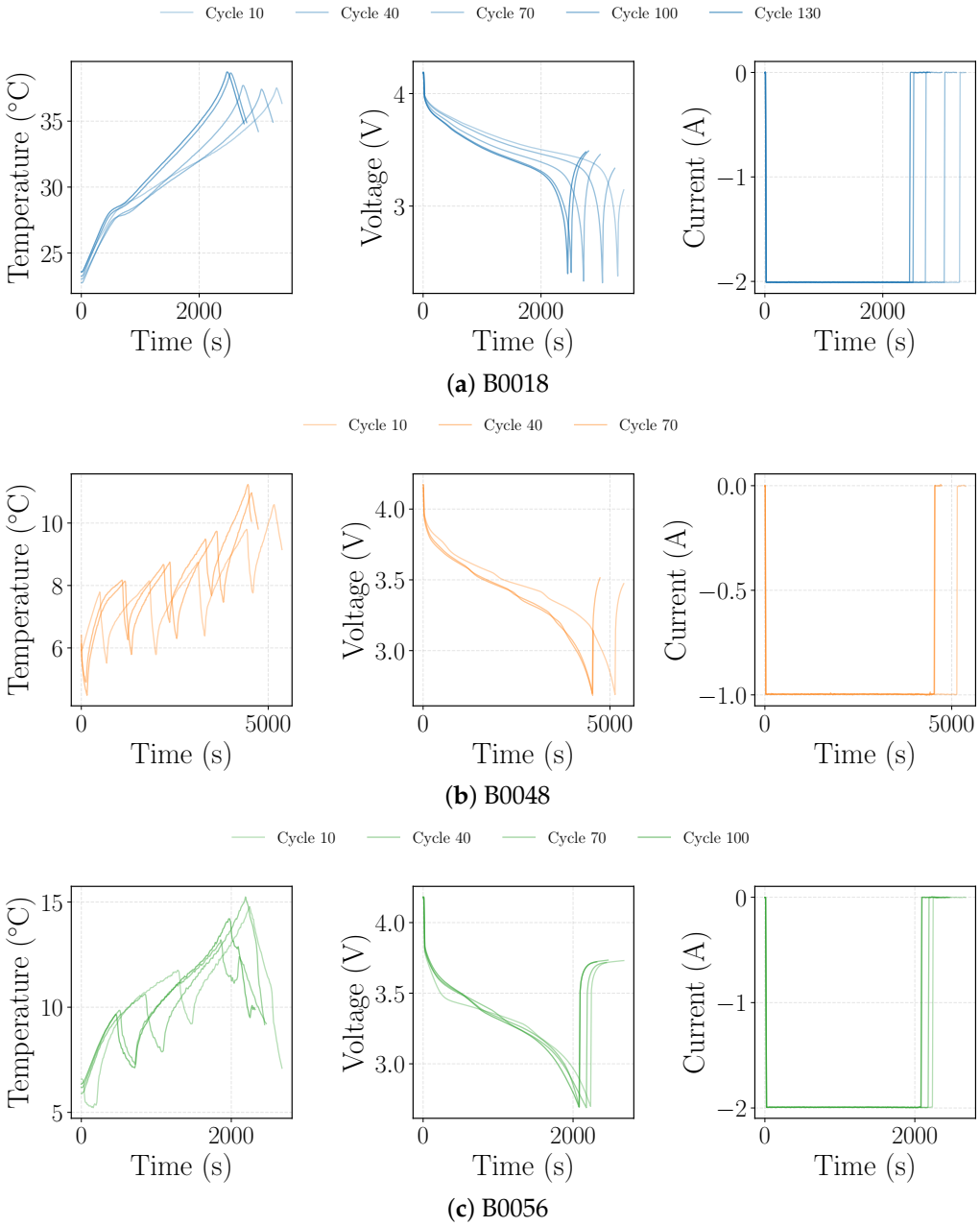


Figure 1. Discharge-phase input signals for batteries B0018 , B0048, and B0056. Each panel shows temperature, voltage, and current as functions of time for sampled discharge cycles .

For each candidate, input and target standardization parameters are fitted using training data forecast rows. For input feature j, the standardized feature values are calculated as

$$z_{ij} = \frac{x_{ij} - \mu_{j,\text{train}}}{\sigma_{j,\text{train}}}. \tag{18}$$

The population standard deviation is used. Capacity targets are standardized in the same way and then inverse-transformed to Ah before RMSE and other capacity metrics are calculated.

Table 1 summarizes the battery-group assignments, data partitions, cycle counts, and resulting one-step-ahead forecast-pair counts, together with the listed operating conditions.

It makes explicit which batteries contribute to training, validation, and held-out testing; each forecast pair uses the preceding cycle as input and the next cycle's capacity as its target.

Table 1. Battery groups, data partitions, and summary statistics for the NASA PCoE dataset.

Group	Battery	Split	Cycles	Pairs	Ambient T (°C)	Discharge I (A)	Cutoff V (V)
G1	B0005	train	168	167	24	2	2.7
	B0006	train	168	167	24	2	2.5
	B0007	val	168	167	24	2	2.2
	B0018	test	132	131	24	2	2.5
G2	B0045	train	70	69	4	1	2.0
	B0046	train	69	68	4	1	2.2
	B0047	val	69	68	4	1	2.5
	B0048	test	69	68	4	1	2.7
G3	B0053	train	55	54	4	2	2.0
	B0054	train	102	101	4	2	2.2
	B0055	val	102	101	4	2	2.5
	B0056	test	102	101	4	2	2.7

The validation partition is used for candidate scoring and model selection. The held-out test partition is reserved for final evaluation. SVR is fitted on the training partition during search and refitted on the combined training and validation data for final testing. LSTM and PINN candidates are trained for at most 200 epochs with patience 20, retaining the checkpoint with the lowest validation loss.

The hyperparameter search spaces and fixed model settings used by NSGA-II are listed in Table 2. The models use the same cycle-level feature types and predict measured discharge capacity. NSGA-II variables are mapped by model-specific decoders: SVR's three positive parameters and the LSTM learning rate are represented in $\log_{10}$ space; LSTM hidden units, batch size, and sequence length are rounded to integers; PINN widths and batch size are decoded as powers of two, learning rates and loss weights are $\log_{10}$-coded, and layer counts are rounded to integers.

Following the multi-objective formulation presented in Section 3, the optimization problem is defined as

$$\min_{\boldsymbol{\theta}\in\Omega} \mathbf{F}(\boldsymbol{\theta}) = [f_1(\boldsymbol{\theta}), f_2(\boldsymbol{\theta}), f_3(\boldsymbol{\theta})], \tag{19}$$

where f_1, f_2, and f_3 denote the validation RMSE, the noise-sensitivity coefficient, and the inference time, respectively. All three objectives are minimized. Each model family is optimized independently, and the discrete choice among model families is not included in the decision vector. The resulting non-dominated configurations are subsequently compared using a common set of evaluation criteria. The first objective is the validation root mean squared error. Let y_i and $\hat{y}_i$ denote the measured and predicted capacity in Ah for held-out forecast example i. The mean measured capacity is defined as

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i. \tag{20}$$

$$f_1(\boldsymbol{\theta}) = \mathrm{RMSE}_{\mathrm{val}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{21}$$

computed on the clean validation set. Minimizing f_1 favors configurations that generalize to held-out batteries in the development partition .

Equation (21) defines the first objective as the square root of the mean squared capacity-prediction error on clean validation examples; lower values indicate more accurate predictions on that set.

Table 2. Model configurations and hyperparameter search spaces used by NSGA-II.

Model	Parameter	Physical Range/Configuration
SVR	C	$[10^{-2}, 10^{3}]$
	γ	$[10^{-6}, 10^{1}]$
	ϵ	$[10^{-3}, 10^{-1}]$
LSTM	Hidden units H	$H \in \{16, \ldots, 256\}$
	Learning rate η	$[10^{-5}, 10^{-1}]$
	Batch size B	$B \in \{8, \ldots, 128\}$
	Sequence length S	$S \in \{1, \ldots, 30\}$
	LSTM layers	1
	Output dropout p	0.2
PINN	Solution hidden dim. H_U	$\{16, 32, 64, 128, 256\}$
	Dynamics hidden dim. H_G	$\{16, 32, 64, 128, 256\}$
	Learning rate η	$[10^{-4}, 10^{-2}]$
	Dynamics learning rate η_G	$[10^{-4}, 10^{-2}]$
	PDE weight α	$[10^{-4}, 1]$
	Monotonicity weight β	$[10^{-6}, 10^{-2}]$
	Solution layers L_U	$\{2, 3, 4\}$
	Dynamics layers L_G	$\{2, 3, 4\}$
	Batch size B	$\{8, 16, 32, 64\}$

Let $X_{\text{val}} \in \mathbb{R}^{N \times D}$ denote the validation input matrix in physical feature units and let σ_j denote the standard deviation of feature j calculated from eligible training examples . For a noise level s from the configured set $\mathcal{S}$, expressed as a percentage of the training-set feature scale, and an independent realization d, the perturbation applied to feature j is drawn from

$$\epsilon_{i,j}^{(d,s)} \sim \mathcal{N}\left(0, \left(\frac{s}{100}\sigma_j\right)^2\right), \tag{22}$$

where i indexes examples, j indexes input features, and d indexes the noise realization. Perturbations are applied only to voltage, current, and temperature; cycle index and capacity target are unchanged. The noisy validation input is given by

$$X_{\text{val}}^{(d,s)} = X_{\text{val}} + \boldsymbol{\epsilon}_{d,s}. \tag{23}$$

Equation (22) specifies zero-mean Gaussian perturbations scaled by each training feature's standard deviation, and Equation (23) applies those perturbations to the validation inputs for each noise realization and level.

Let E_0 denote the RMSE on the clean validation set , and let $\overline{\mathrm{E}}_s$ denote the mean noisy-data RMSE at noise level s , averaged over the configured noise realizations. For each noise level, the relative RMSE increase is calculated with respect to the clean-data RMSE. The second objective is the arithmetic mean of these relative increases over the six configured noise levels, $\mathcal{S} = \{5, 10, 15, 20, 25, 30\}$ percent:

$$f_2(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \left[\frac{\overline{\mathrm{E}}_s - \mathrm{E}_0}{|\mathrm{E}_0|}\right], \tag{24}$$

Equation (24) defines f_2 as the mean relative RMSE change across the configured noise levels; its sign indicates whether average error increased or decreased relative to the clean

validation result. Noise is added to voltage, current, and temperature, while cycle index and target capacity are unchanged.

The third objective , f_3, is inference time. It is measured with a shared timing routine that uses a batch size of 1, sets PyTorch to one CPU thread, makes 10 warm-up calls, then records 30 blocks of 20 forward passes. It returns the median of the 30 per-call block means, converted to milliseconds. Timing excludes feature extraction, data loading, training, and framework start-up. SVR timing uses the PyTorch support-vector predictor; LSTM timing uses the recurrent model; PINN timing measures only the solution network $\mathcal{F}$, because the dynamics network is used as a training constraint but is not part of the deployed prediction path. For a final reported configuration, timing is repeated using the first held-out test example under the same protocol.

$$f_3(\boldsymbol{\theta}) = T_{\text{CPU}}(\boldsymbol{\theta}). \tag{25}$$

This model-forward latency is hardware- and runtime-dependent, and does not establish real-time performance on an embedded battery-management system. Model size, parameter count, and training time are complementary deployment indicators but are not included in the three-objective vector. The recorded execution environment is an AMD Ryzen 5 5600 6-Core processor with Python 3.12.3, PyTorch 2.6.0, and pymoo 0.6.2 [39].

Minimizing f_3 is relevant to on-board battery management systems, where predictions must be produced within strict latency constraints. The measured values should be interpreted as comparative forward-pass times under the stated evaluation setup.

Each candidate is fit on training batteries and scored on validation batteries. LSTM training minimizes standardized-capacity mean squared error with Adam; its mini-batches preserve row order. PINN training uses Adam optimizers for the solution and dynamics networks, with the separately searched learning rates and the loss in the PINN description above. Neural candidates run for at most 200 epochs with early-stopping patience 20, and the best validation-loss checkpoint is used for scoring. For all model families, the generalization gap, in Ah, is calculated as

$$g = \text{RMSE}_{\text{train}} - \text{RMSE}_{\text{val}}. \tag{26}$$

A candidate is rejected if $|g| \geq 0.10$ Ah, assigning 10^6 penalty values to all three objectives.

The optimization is performed independently for each model family using NSGA-II from the `pymoo` [39] library, with real-valued random sampling, simulated binary crossover (probability 0.9, distribution index 15), polynomial mutation (probability $1/n_{\text{var}}$, distribution index 15), duplicate elimination, population size 100, over 50 generations. The Pareto archive is extracted from all feasible evaluated designs following its elitist nature. The three searches receive the same evaluation budget.

At each NSGA-II generation, the parent population $\mathcal{P}_t$ produces an offspring population $\mathcal{Q}_t$. The combined population $\mathcal{R}_t = \mathcal{P}_t \cup \mathcal{Q}_t$ is ranked according to non-dominated sorting and crowding distance, and the next population $\mathcal{P}_{t+1}$ is selected from the resulting fronts.

4.2. Additional Evaluation Metrics

In addition to the three optimization objectives, the final test predictions are summarized using mean absolute error (MAE), mean absolute percentage error (MAPE), and the coefficient of determination (R^2). Let y_i and $\hat{y}_i$ denote the measured and predicted capacity in Ah for held-out forecast example i, and let $\bar{y}$ be the mean measured capacity defined

in Equation (20). The metrics are calculated over all held-out examples pooled across the test batteries:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{27}$$

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{|y_i|}, \tag{28}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}. \tag{29}$$

MAE is reported in Ah, while MAPE is reported in percent. R^2 compares the model's squared prediction error with the error from predicting the pooled test-set mean; values closer to 1 indicate better fit. The denominator safeguards in Equations (28) and (29) avoid division by values below 10^{-12}. Model size and parameter count are auxiliary implementation-specific indicators. Furthermore, PINN counts and size include the dynamics network, whereas inference latency in Equation (25) measures only the solution network.

5. Results and Discussion

The results are organized into three parts. First, the Pareto fronts are analyzed to identify the trade-offs among validation RMSE, noise sensitivity, and inference time for the SVR, PINN, and LSTM models. Second, the hypervolume trajectories are examined to assess the convergence behavior of the three NSGA-II searches. Third, representative non-dominated evaluated configurations are compared using balance-oriented selection criteria. All three optimization objectives are formulated as minimization objectives; therefore, lower values indicate better performance for the corresponding metric.

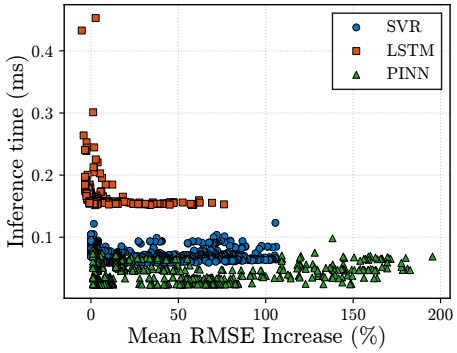
5.1. Pareto Frontiers and Objective Trade-Offs

The multi-objective optimization generated a set of feasible non-dominated configurations for each model family. The resulting fronts should be interpreted as sets of alternative compromises.

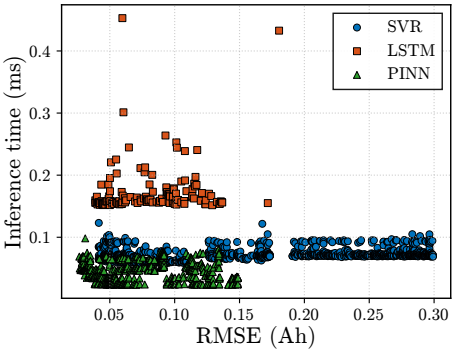
Figure 2 plots the validation objectives for the feasible non-dominated designs: panel (a) relates mean RMSE increase to inference latency, (b) relates clean validation RMSE to latency, (c) relates clean RMSE to noise sensitivity, and (d) shows the three objectives jointly. Lower values are preferred on each axis; a negative mean RMSE increase means that the averaged noisy-data RMSE was lower than the clean-data RMSE under this specific perturbation protocol. The figure shows that the three model families occupy distinct regions. LSTM includes low-RMSE and low-coefficient configurations but has longer measured forward latency. SVR achieves comparatively low RMSE over a narrower latency range, with noticeable variation in its noise sensitivity. PINN has the lowest measured latency, in part because the inference path times only the solution network and excludes its auxiliary dynamics network; RMSE also varies among configurations with similar latency.

The RMSE–inference-time projection in Figure 2b emphasizes the separation in measured forward latency. SVR and PINN configurations are substantially faster than LSTM configurations. Within each family, similar latencies can correspond to different RMSE values; thus latency alone is insufficient to infer prediction quality. The broad family-level separation and within-family spread describe these implementations and search spaces.

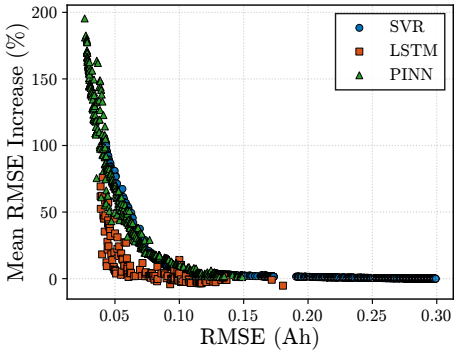
The robustness-related projections in Figure 2a,c show that noise sensitivity varies substantially among non-dominated solutions. Configurations close to zero have smaller average relative error changes under noise, but this does not imply low clean-data RMSE.



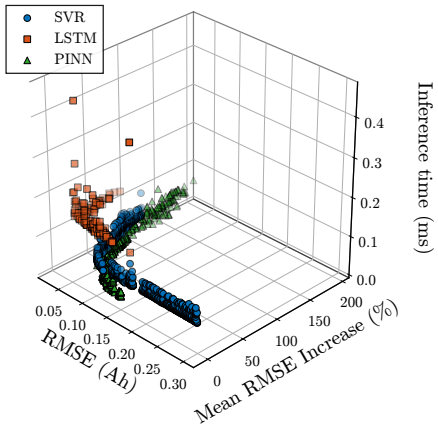
(**a**) mean RMSE increase vs. inference time



(**b**) RMSE vs. inference time



(**c**) RMSE vs. mean RMSE increase



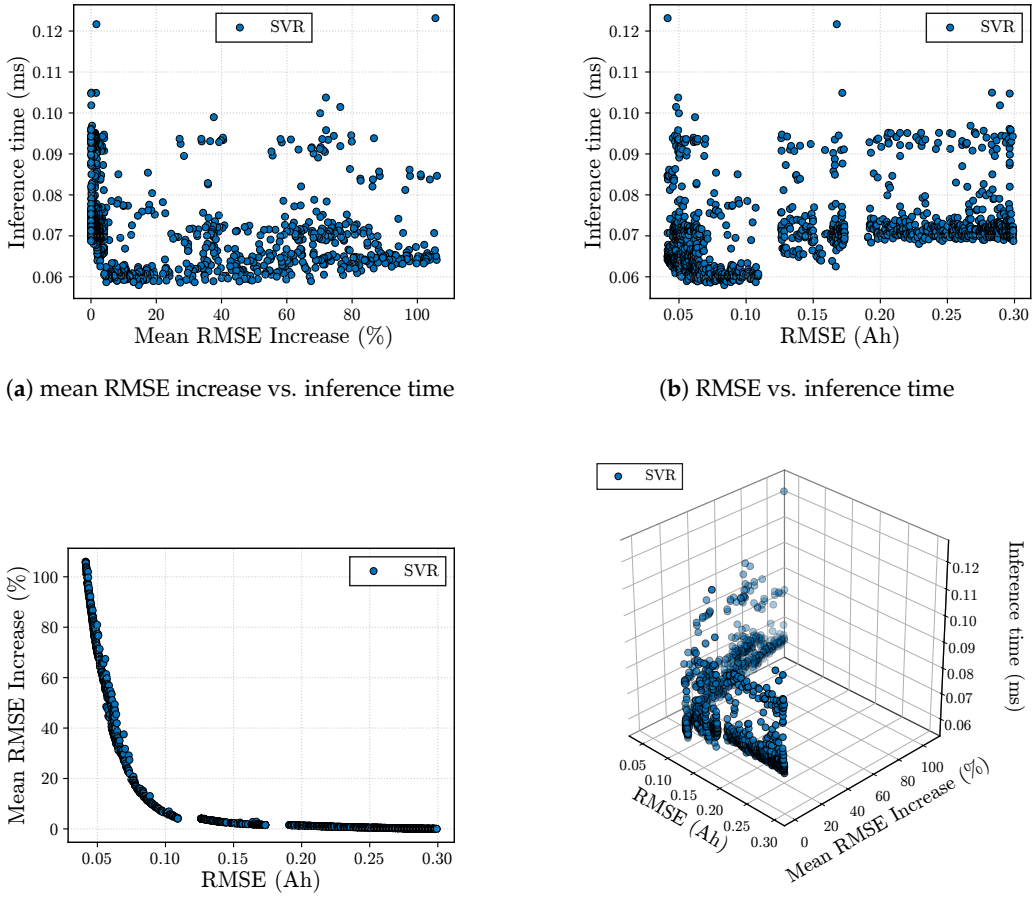
(**d**) Three-objective Pareto sets

Figure 2. Combined Pareto sets for SVR , PINN, and LSTM: pairwise objective projections and three-dimensional objective-space representation. Lower values are preferred for all objectives.

The SVR projections in Figure 3 show a pronounced two-objective trade-off between RMSE and the mean RMSE increase. In Figure 3c, the lowest-RMSE configurations, with RMSE between approximately 0.04 and 0.10 Ah, exhibit mean RMSE increases ranging from approximately 6.59% to 106.01%. Conversely, configurations with mean RMSE increases close to zero have markedly higher RMSE values. This distribution indicates that, within the evaluated SVR search space, lower validation RMSE is generally associated with higher sensitivity to noisy inputs.

Figure 3b contains a dense group of configurations near the low-RMSE and low-inference-time region. Most of these solutions exhibit similar inference times within the latency range shown in the figure. This concentration suggests that several hyperparameter combinations provide similar computational performance while retaining comparatively low prediction error.

The robustness–inference-time projection in Figure 3a shows substantial variation in robustness over a comparatively narrow latency interval. Consequently, selecting an SVR configuration only from RMSE and inference time may conceal a considerable difference in behavior under noisy inputs.



(**a**) mean RMSE increase vs. inference time

(**b**) RMSE vs. inference time

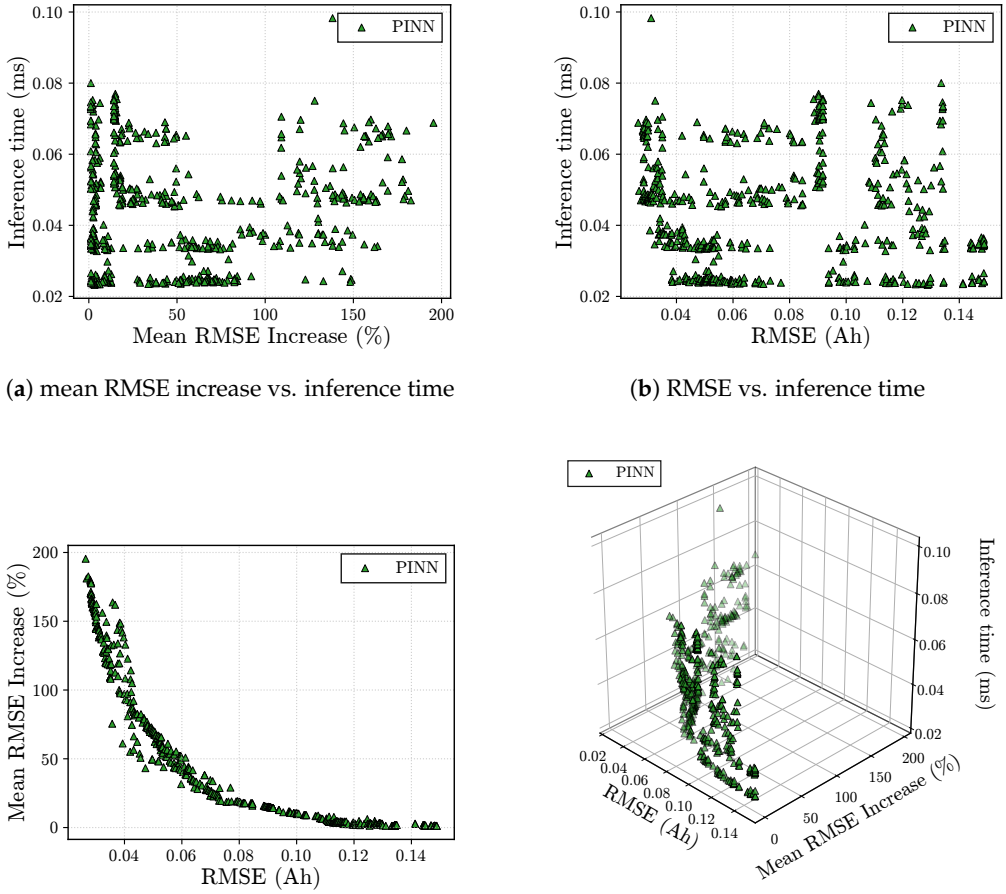
(**c**) RMSE vs. mean RMSE increase

(**d**) Three-objective Pareto set

Figure 3. SVR Pareto set: pairwise objective projections and three-dimensional objective-space representation.

The PINN front in Figure 4 spans validation RMSE values of approximately 0.026–0.149 Ah, a narrower range than the approximately 0.041–0.299 Ah observed for SVR. Figure 4b shows substantial variation in RMSE among configurations with similar inference times. The plotted configurations therefore do not establish a monotonic accuracy–latency relationship: a longer inference time does not consistently correspond to a lower prediction error.

Figure 4c shows a trade-off between clean-data accuracy and sensitivity to the specified input perturbations. Configurations in the low-RMSE region exhibit a broad range of mean RMSE increases, whereas several configurations with higher clean-data RMSE have smaller relative increases. A small noise-sensitivity coefficient should therefore be interpreted together with the clean-data RMSE, since a small relative change does not necessarily indicate accurate predictions. Figure 4a also shows that configurations with similar inference times can differ substantially in noise sensitivity; inference time alone is therefore insufficient to characterize their response to noisy inputs.



(**a**) mean RMSE increase vs. inference time

(**b**) RMSE vs. inference time

(**c**) RMSE vs. mean RMSE increase

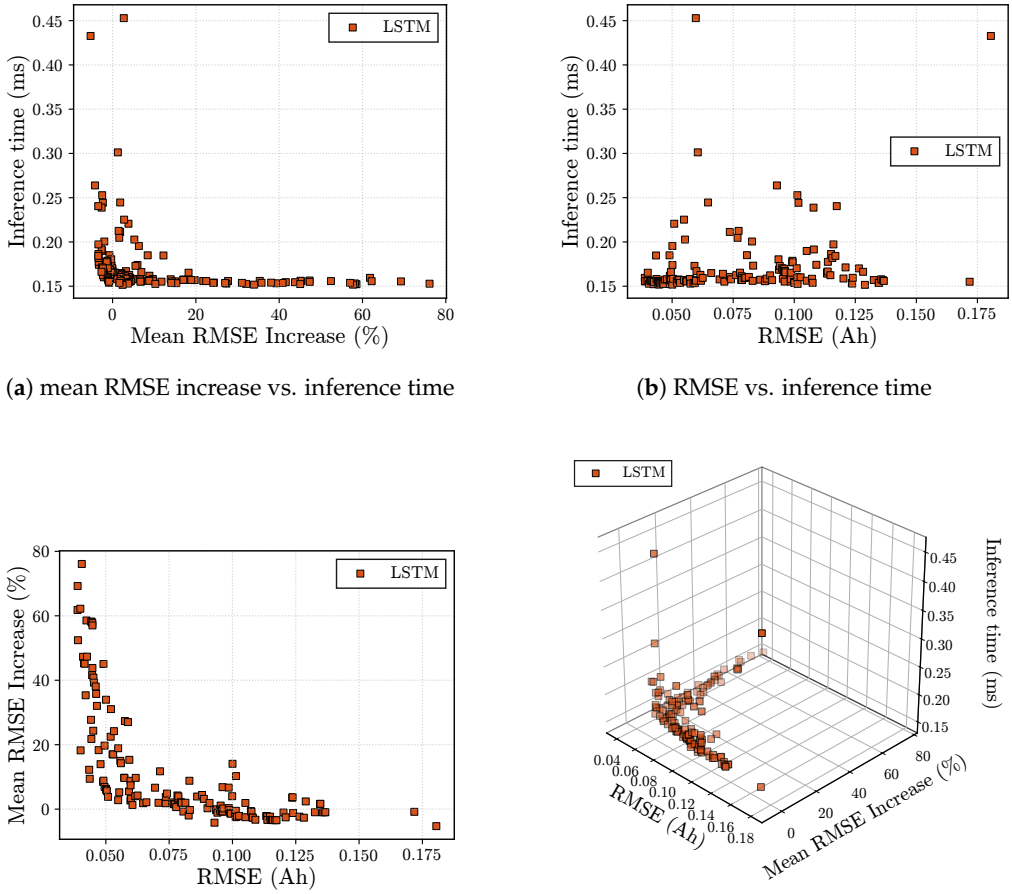
(**d**) Three-objective Pareto set

Figure 4. PINN Pareto set: pairwise objective projections and three-dimensional objective-space representation.

Figure 5 confirms that the LSTM configurations incur the largest inference times, with measured latencies spanning approximately 0.152–0.453 ms and many configurations near 0.15–0.20 ms. This higher computational cost is accompanied by access to a low-RMSE region and by mean RMSE increases that are generally lower than those of the most accurate SVR and PINN configurations. The LSTM family therefore occupies an accuracy- and robustness-oriented region of the combined objective space, but at a clear latency penalty.

The relationship between RMSE and inference time in Figure 5b does not follow a monotonic trade-off relationship. Some of the longest inference times do not correspond to the lowest RMSE. Hence, the figures do not support the general conclusion that a larger or slower LSTM necessarily predicts more accurately. Figure 5c shows a clearer tendency for the mean RMSE increase to decrease as RMSE increases, although substantial dispersion remains in the low-RMSE region. The recurrent memory gates may contribute to filtering short-term perturbations, but the present results do not isolate a gate-specific regularization effect. Demonstrating such a mechanism would require controlled ablation experiments or an analysis linking the number of units, layers, parameters, and gate activations to performance under each noise level.

Taken together, the model-specific fronts show that architecture-level conclusions and hyperparameter-level conclusions must be separated. The combined front supports the conclusion that LSTM, SVR , and PINN occupy different latency–accuracy–robustness regimes.



(**a**) mean RMSE increase vs. inference time

(**b**) RMSE vs. inference time

(**c**) RMSE vs. mean RMSE increase

(**d**) Three-objective Pareto set

Figure 5. LSTM Pareto set: pairwise objective projections and three-dimensional objective-space representation.

5.2. Hypervolume Convergence

The convergence behavior of the optimization was evaluated using the hypervolume indicator, the volume dominated by the non-dominated solutions relative to a reference point. To report the hypervolume, a single fixed reference point was shared by all three families, calculated in the objective units from all feasible evaluation-history points. Each coordinate is the observed maximum plus 5% of that objective's observed range. The cumulative archive includes all feasible candidates evaluated up to each generation, so its hypervolume is non-decreasing by construction . An increase indicates improvement or expanded coverage relative to this reference point, but does not establish global convergence.

Figure 6 shows an upward trajectory for each cumulative archive. The PINN curve starts near 0.94 and approaches 0.96; the SVR curve rises from about 0.87 to 0.91; and the LSTM curve increases from roughly 0.78 to 0.83 across the plotted 0–50 generations. PINN and SVR gain most of their plotted hypervolume early and then flatten, whereas LSTM continues with smaller increases later in the run. Under the shared reference point, the plotted ordering indicates that the PINN archive covers the largest dominated volume on the joint objective scale; it does not show that PINN is best on every objective or universally preferable. Because hypervolume is calculated from a cumulative archive, non-decreasing curves are expected, and flattening indicates little or no additional expansion at the recorded updates, indicating little additional hypervolume improvement during the recorded updates.

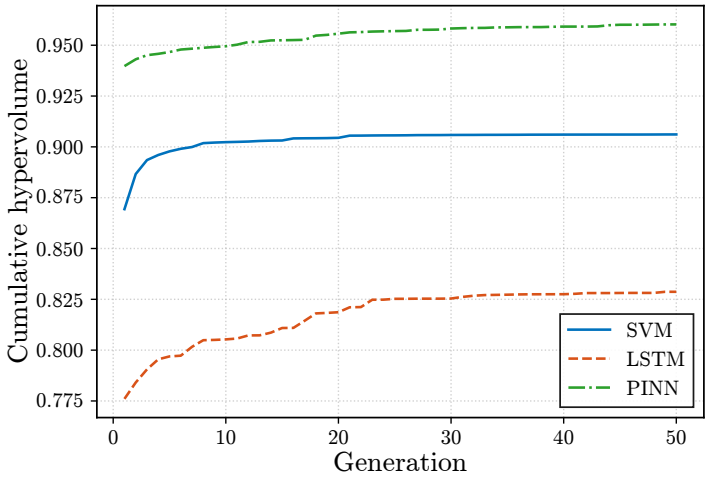


Figure 6. Combined cumulative hypervolume trajectories for the SVR , PINN, and LSTM NSGA-II searches, calculated using the common fixed reference point.

5.3. Representative Pareto Configurations

Although each Pareto front contains multiple non-dominated configurations, a smaller set of representative solutions is useful for detailed comparison. The intended criteria are minimum test RMSE, minimum test mean RMSE increase under noise, minimum test inference time, and minimum Euclidean distance to the ideal point. The distance-based choice is called balanced; it gives equal geometric weight to the three scaled objectives, not equal weight in their raw units. The obtained results can be found in Tables 3 and 4.

Table 3. Test performance of selected Pareto configurations.

Model	Selection Criterion	RMSE (Ah)	Mean RMSE Increase (%)	Inference (ms)
SVR	Balanced	0.0799	5.9011	0.0697
	Minimum RMSE	0.0649	50.8558	0.0621
	Minimum noise sensitivity	0.2067	2.8000×10^{-5}	0.0790
	Minimum inference time	0.0806	23.4675	0.0586
LSTM	Balanced	0.0676	4.4153	0.1564
	Minimum RMSE	0.0501	98.8890	0.1539
	Minimum noise sensitivity	0.1352	−3.6647	0.1673
	Minimum inference time	0.0698	44.3541	0.1499
PINN	Balanced	0.0925	1.7455	0.0244
	Minimum RMSE	0.0587	66.1320	0.0319
	Minimum noise sensitivity	0.1615	−3.2454	0.0342
	Minimum inference time	0.1568	−3.1862	0.0235

Within these results, the LSTM minimum-RMSE row has the lowest test RMSE (0.050109 Ah), while the PINN minimum-inference row has the lowest measured latency (0.023498 ms). Negative noise-sensitivity coefficients for the LSTM and PINN mean that, averaged over the six specified test noise levels and three draws per level, noisy-data RMSE was below clean-data RMSE for those rows. This does not imply zero error or general robustness to other noise distributions.

Table 4. Hyperparameter configurations for selected Pareto solutions.

Model	Hyperparameter	Balanced	Min. RMSE	Min. Noise Sensitivity	Min. Inference Time
SVR	C	0.0310	8.7009	0.0105	12.2234
	γ	0.0732	0.2847	1.4022×10^{-6}	1.0460
	ε	0.0999	0.0623	0.0461	0.0858
LSTM	Hidden units	16	16	49	16
	Learning rate	0.0033	0.0685	0.0036	0.0829
	Batch size	90	13	79	101
	Sequence length	6	1	5	1
PINN	Batch size	64	8	32	64
	Hidden width H_U	128	128	16	16
	Hidden width H_G	16	16	16	64
	Layers L_U	2	2	2	2
	Layers L_G	4	4	3	2
	Learning rate η	0.0039	0.0097	0.0037	0.0034
	Learning rate η_G	0.0006	0.0009	0.0006	0.0014
	α	0.7316	0.1859	0.7055	0.1819
	β	3.6300×10^{-6}	0.0037	3.4800×10^{-6}	0.0062

The representative results identify distinct performance profiles. The LSTM minimum-RMSE configuration has the lowest held-out error, while the PINN minimum-inference configuration has the lowest measured latency. The noise-sensitivity values include negative test-set relative changes for the selected LSTM and PINN configurations, reflecting lower average RMSE under the evaluated noisy inputs than under clean inputs . No single representative configuration minimizes all three objectives simultaneously.

The representative hyperparameters show how selected settings vary across objective priorities. For the SVR, the balanced, minimum-RMSE , minimum-noise-sensitivity, and minimum-inference configurations use different combinations of C , γ , and ε ; the low-noise-sensitivity solution uses a particularly small γ. The held-out metrics in Table 3 show that this selection criterion does not imply the lowest clean-data RMSE.

For the LSTM, all four selected configurations use 16 hidden units except the minimum-noise-sensitivity solution, which uses 49. Sequence lengths range from one to six cycles, and batch sizes vary from 13 to 101. The learning rate and batch size therefore distinguish several selected settings even when their hidden-unit counts match.

For the PINN, the balanced and minimum-RMSE configurations share widths of 128/16 and layer counts of 2/4, but differ in batch size, learning rates, and loss weights. The minimum-noise-sensitivity and minimum-inference configurations use different width and layer combinations, with different values of α and β.

These differences are important when interpreting the PINN results because the evaluated architecture contains separate components for the solution and dynamics representations. The physics-related computations contribute to the training process, whereas deployment inference uses the solution network to produce the SOH prediction. Therefore, the low inference times obtained by the PINN configurations should be interpreted as a property of the evaluated PINN formulation and deployment procedure rather than as a general property of PINNs. Changes to the physical formulation or to the deployment architecture could modify this relationship.

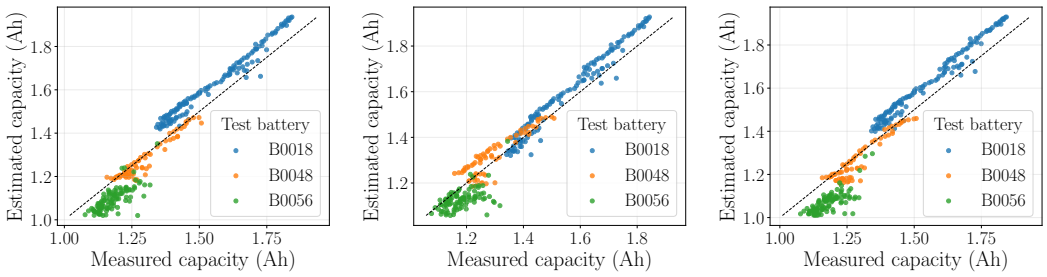
The reported comparisons illustrate why predictive error alone is insufficient for evaluating SOH prediction models intended for condition-based monitoring. Configurations selected for low error, robustness, and latency can have different performance on the other objectives. The Pareto set makes these trade-offs explicit rather than treating the lowest-RMSE configuration as universally preferable.

The balanced configuration provides an intermediate point between these objectives. Such a solution need not minimize any individual objective, but it can provide a compromise that satisfies multiple requirements to different degrees.

5.4. Prediction Diagnostics

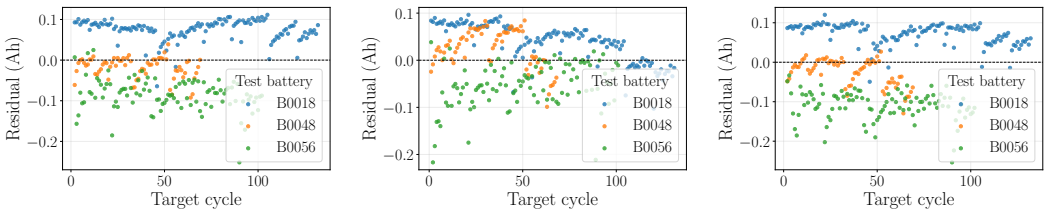
Figure 7 compares measured and predicted discharge capacity for the selected SVR, PINN, and LSTM models on held-out batteries. The dashed diagonal marks ideal one-to-one agreement; systematic displacement from it indicates prediction bias, while scatter around it reflects prediction error. Figure 8 plots predicted-minus-measured capacity against target cycle for the same models. Its zero-error reference line makes bias and cycle-dependent residual patterns visible; residuals distributed around zero without a systematic trend indicate more consistent errors across battery life.

The battery-specific plots compare measured and predicted capacity-derived SOH across cycles. Figure 9 shows the trajectories for held-out battery B0018, Figure 10 shows them for B0048, and Figure 11 shows them for B0056. Comparing each predicted trajectory with its measured counterpart illustrates how well the selected models track the observed degradation pattern on each test battery.



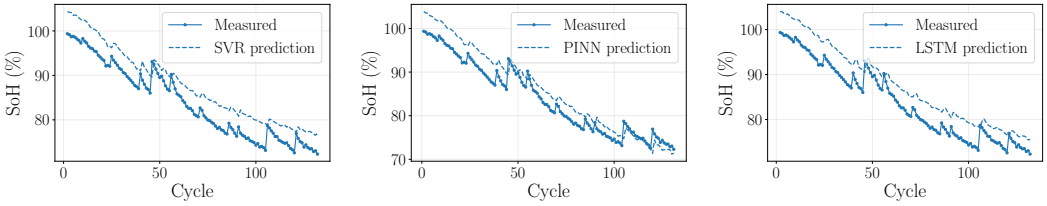
(**a**) SVR (**b**) PINN (**c**) LSTM

Figure 7. Parity plots for the selected models on the held-out batteries.



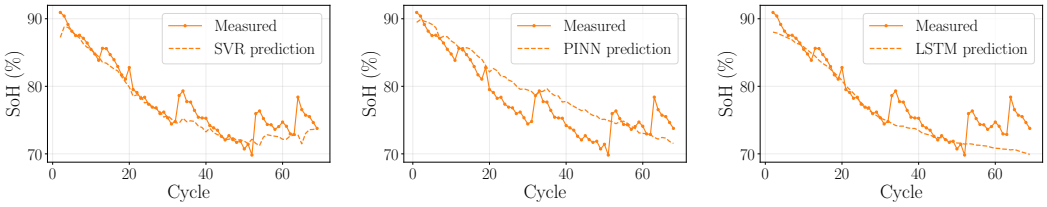
(**a**) SVR (**b**) PINN (**c**) LSTM

Figure 8. Residuals by target cycle for the selected models on the test batteries.



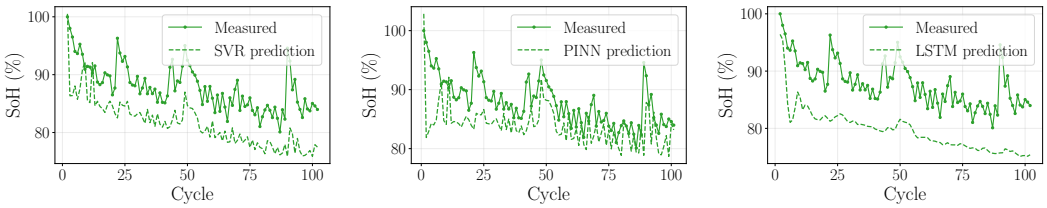
(**a**) SVR (**b**) PINN (**c**) LSTM

Figure 9. Measured and predicted capacity-derived SOH by cycle for held-out battery B0018.



(**a**) SVR (**b**) PINN (**c**) LSTM

Figure 10. Measured and predicted capacity-derived SOH by cycle for held-out battery B0048.



(**a**) SVR (**b**) PINN (**c**) LSTM

Figure 11. Measured and predicted capacity-derived SOH by cycle for held-out test battery B0056.

The additional held-out metrics for the balanced configurations are summarized in Table 5.

Table 5. Additional held-out test metrics for the exploratory balanced configurations.

Model	MAE (Ah)	MAPE (%)	R^2	Model Size (KB)	Parameters
SVR	0.0649	4.8806	0.8455	11.6641	598
LSTM	0.0559	4.2421	0.8894	5.5664	1425
PINN	0.0690	5.1666	0.7931	5.8203	1490

As defined in Section 4.2, MAE, MAPE, and R^2 are calculated over all held-out forecast examples pooled across the three test batteries. Among the exploratory balanced rows, LSTM has the smallest MAE and MAPE and the largest R^2, while PINN has higher errors and a lower R^2. The model-size and parameter-count values follow the model-specific evaluator definitions in Section 4.2; notably, the PINN measures include its auxiliary dynamics network, which is excluded from inference timing.

6. Conclusions and Future Work

This study presented a multi-objective framework for evaluating hyperparameter configurations of lithium-ion battery SOH prediction models based on predictive error, sensitivity to measurement noise, and inference time. NSGA-II was applied to the SVR, LSTM, and PINN models to identify non-dominated configurations and characterize the trade-offs among these objectives.

The evaluated configurations demonstrate that improvements in one performance criterion do not necessarily translate into improvements in the others. The LSTM achieved the lowest test RMSE (0.0501 Ah) among the minimum-RMSE configurations. For the balanced configurations, it also obtained the lowest MAE (0.0559 Ah) and MAPE (4.2421%), together with the highest R^2 (0.8894). In contrast, the PINN minimum-inference configuration achieved the shortest measured inference time (0.0235 ms).

Across the test batteries, the parity plots compare measured and predicted capacity, residuals are predominantly positive for B0018, indicating capacity overestimation, and negative for B0056, indicating underestimation; B0048 exhibits smaller residuals overall, although deviations vary across cycles. The capacity-derived SOH predictions reproduce

the overall degradation trajectories of all three batteries, but smooth out local fluctuations, particularly in B0056. The consistent residual sign across model families suggests that these biases are associated with battery-specific characteristics rather than a single model architecture.

The results demonstrate the value of considering multiple optimization objectives jointly when selecting SOH prediction models for PdM. The resulting Pareto sets support application-dependent selection according to the relative importance of these criteria. The reported comparisons are limited to the evaluated configurations, model implementations, search spaces, NASA PCoE batteries, and CPU architecture. Thus, they provide a comparative assessment of the evaluated model configurations under the same dataset, search budget, and hardware, offering quantitative evidence to guide application-specific model selection for PdM.

Future work should extend the evaluation to additional battery datasets, chemistries, cell formats, and operating conditions, particularly to assess generalization to previously unseen conditions. The robustness analysis could incorporate experimentally measured sensor-error distributions and additional perturbation mechanisms. Feature-importance and feature-selection analyses could further clarify the influence of input representations on the optimization objectives. Finally, alternative PINN formulations and physical constraints should be investigated to assess their effects on predictive performance and computational requirements.

Author Contributions: Conceptualization, A.B.B. and J.J.M.J.; methodology, A.B.B., J.J.M.J. and R.V.; software and validation, A.B.B.; writing, A.B.B., J.J.M.J. and R.V.; supervision, project administration and funding acquisition, J.J.M.J. and R.V. All authors have read and agreed to the published version of the manuscript.

Funding: Tecnológico de Costa Rica (VIE-project 1341032) and ISAE-SUPAERO.

Data Availability Statement: The raw battery measurements are available from the NASA PCoE dataset cited in this article. The codes used in this study are available from the authors upon request.

Conflicts of Interest: The authors declare no conflicts of interest

References

1. Duan, S.; Sun, J.; Yu, Z.; Liu, S. Uncertain data driven predictive maintenance: A cost-oriented implementation method on aircraft system. *Reliab. Eng. Syst. Saf.* **2025**, *264*, 111278. https://doi.org/10.1016/j.ress.2025.111278.
2. Xu, G.; Jiang, P.; Ren, W.; Li, Y.; Chen, Z. The Research on Multi-Objective Maintenance Optimization Strategy Based on Stochastic Modeling. *Machines* **2025**, *13*, 633. https://doi.org/10.3390/machines13080633.
3. Li, W.; Li, T. Comparison of deep learning models for predictive maintenance in industrial manufacturing systems using sensor data. *Sci. Rep.* **2025**, *15*, 23545. https://doi.org/10.1038/s41598-025-08515-z.
4. Salinas-Camus, M.; Goebel, K.; Eleftheroglou, N. A comprehensive review and evaluation framework for data-driven prognostics: Uncertainty, robustness, interpretability, and feasibility. *Mech. Syst. Signal Process.* **2025**, *237*, 113015. https://doi.org/10.1016/j.ymssp.2025.113015.
5. Arena, S.; Florian, E.; Sgarbossa, F.; Sølvsberg, E.; Zennaro, I. A conceptual framework for machine learning algorithm selection for predictive maintenance. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108340. https://doi.org/10.1016/j.engappai.2024.108340.
6. Li, C.; Han, X.; Zhang, Q.; Li, M.; Rao, Z.; Liao, W.; Liu, X.; Liu, X.; Li, G. State-of-health and remaining-useful-life estimations of lithium-ion battery based on temporal convolutional network-long short-term memory. *J. Energy Storage* **2023**, *74*, 109498. https://doi.org/10.1016/j.est.2023.109498.
7. Zhao, J.; Han, X.; Wu, Y.; Wang, Z.; Burke, A.F. Opportunities and challenges in transformer neural networks for battery state estimation: Charge, health, lifetime, and safety. *J. Energy Chem.* **2025**, *102*, 463–496. https://doi.org/10.1016/j.jechem.2024.11.011.
8. Lyu, Z.; Jin, Z.; Li, X.; Wang, H.; Wu, L.; Chen, Y. From tradition to innovation: Evolution and trade-offs of lithium-ion battery state of health estimation methods. *J. Energy Storage* **2026**, *144*, 119730. https://doi.org/10.1016/j.est.2025.119730.

9. Eleftheriadis, P.; Gangi, M.; Leva, S.; Valdes Rey, A.; Groppo, E.; Grande, L. Comparative study of machine learning techniques for the state of health estimation of Li-Ion batteries. *Electr. Power Syst. Res.* **2023**, *235*, 110889. https://doi.org/10.1016/j.epsr.2024.110889.
10. Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. In *Feature Extraction: Foundations and Applications*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–25. https://doi.org/10.1007/978-3-540-35488-8_1.
11. Zhu, T.; Ran, Y.; Zhou, X.; Wen, Y. A Survey on Intelligent Predictive Maintenance (IPdM) in the Era of Fully Connected Intelligence. *IEEE Commun. Surv. Tutor.* **2026**, *28*, 633–671. https://doi.org/10.1109/comst.2025.3567802.
12. Samanta, A.; Williamson, S. Machine Learning-based Remaining Useful Life Prediction Techniques for Lithium-ion Battery Management Systems: A Comprehensive Review. *IEEJ J. Ind. Appl.* **2023**, *12*, 563–574. https://doi.org/10.1541/ieejjia.22004793.
13. Aydın, C.; Evrentuğ, B. Evaluation of predictive maintenance efficiency with the comparison of machine learning models in machining production process in brake industry. *PeerJ Comput. Sci.* **2025**, *11*, e2999. https://doi.org/10.7717/peerj-cs.2999.
14. Wang, W.; Tang, X.; Li, M.; Sun, C.; Hasanien, H.M.; Gu, X.; Zadeh, M.; Molinas, M. Cross-Vehicle Battery Health Estimation Through Physics-Guided Two-Stage Residual Learning From Real-World Electric Vehicle Data. *IEEE Trans. Ind. Inform.* **2026**, 1–12. https://doi.org/10.1109/TII.2026.3725383.
15. Montero-Jiménez, J J.; Vingerhoeds, R. A System Engineering Approach to Predictive Maintenance Systems: From needs and desires to logical architecture. In Proceedings of the 2019 International Symposium on Systems Engineering (ISSE), Edinburgh, UK, 1–3 October 2019. https://doi.org/10.1109/ISSE46696.2019.8984559.
16. Montero Jiménez, J.J.; Vingerhoeds, R.; Grabot, B. Enhancing predictive maintenance architecture process by using ontology-enabled Case-Based Reasoning. In Proceedings of the 7th IEEE International Symposium on System Engineering, Vienna, Austria, 13 September–13 October 2021. https://doi.org/10.1109/ISSE51541.2021.9582535.
17. Bischl, B.; Binder, M.; Lang, M.; Pielok, T.; Richter, J.; Coors, S.; Thomas, J.; Ullmann, T.; Becker, M.; Boulesteix, A.L.; et al. Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges. *arXiv* **2021**, arXiv:2107.05847. https://doi.org/10.48550/arXiv.2107.05847.
18. Jin, Y. *Multi-Objective Machine Learning*; Studies in Computational Intelligence; Springer: Berlin/Heidelberg, Germany, 2006. https://doi.org/10.1007/11399346.
19. Jin, Y.; Sendhoff, B. Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies. *IEEE Trans. Syst. Man, Cybern. Part C (Appl. Rev.)* **2008**, *38*, 397–415. https://doi.org/10.1109/TSMCC.2008.919172.
20. Bergstra, J. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
21. He, X.; Zhao, K.; Chu, X. AutoML: A survey of the state-of-the-art. *Knowl.-Based Syst.* **2021**, *212*, 106622. https://doi.org/10.1016/j.knosys.2020.106622.
22. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *arXiv* **2018**, arXiv:1603.06560. https://doi.org/10.48550/arXiv.1603.06560.
23. Karl, F.; Pielok, T.; Moosbauer, J.; Pfisterer, F.; Coors, S.; Binder, M.; Schneider, L.; Thomas, J.; Richter, J.; Lang, M.; et al. Multi-Objective Hyperparameter Optimization in Machine Learning—An Overview. *ACM Trans. Evol. Learn. Optim.* **2023**, *3*, 16. https://doi.org/10.1145/3610536.
24. Mei, J.; Li, Z.; Song, K.; Meng, X.; Wu, H.; Tang, X.; Hasanien, H.M.; Li, Y.; Sun, C. SOH-Disparity-Aware Energy Management for Multi-Stack Fuel Cells Using Enhanced Soft Actor-Critic Reinforcement Learning. *IEEE Trans. Transp. Electrif.* **2026**, 1
25. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. https://doi.org/10.1109/4235.996017.
26. Lu, Z.; Cheng, R.; Jin, Y.; Tan, K.C.; Deb, K. Neural Architecture Search as Multiobjective Optimization Benchmarks: Problem Formulation and Performance Assessment. *arXiv* **2023**, arXiv:2208.04321. https://doi.org/10.48550/arXiv.2208.04321.
27. Pfisterer, F.; Schneider, L.; Moosbauer, J.; Binder, M.; Bischl, B. YAHPO Gym—An Efficient Multi-Objective Multi-Fidelity Benchmark for Hyperparameter Optimization. In Proceedings of the First International Conference on Automated Machine Learning, Baltimore, MD, USA, 25–27 July 2022; Volume 188, pp. 3/1–39. https://doi.org/10.1109/TTE.2026.3710578.
28. Montero Jimenez, J.J.; Schwartz, S.; Vingerhoeds, R.; Grabot, B.; Salaün, M. Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *J. Manuf. Syst.* **2020**, *56*, 539–557. https://doi.org/10.1016/j.jmsy.2020.07.008.
29. Patil, S.; Havaldar, S.M.; R K, B.; Mathad, S.; Patil, K.R. Lithium-ion Battery State of Health Estimation Using Support Vector Regression(SVR). In Proceedings of the 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Ballari, India, 2–3 November 2023; pp. 1–6. https://doi.org/10.1109/AIKIIE60097.2023.10390075.
30. Zou, Y.; Lin, Z.; Li, D.; Liu, Z. Advancements in Artificial Neural Networks for Health Management of Energy Storage Lithium-Ion Batteries: A Comprehensive Review. *J. Energy Storage* **2023**, *73*, 109069. https://doi.org/10.1016/j.est.2023.109069.
31. Lin, J.; Yan, G.; Wang, C. Li-ion battery state of health prediction based on long short-term memory recurrent neural network. *J. Phys. Conf. Ser.* **2021**, *2010*, 012133. https://doi.org/10.1088/1742-6596/2010/1/012133.

32. Patel, K.; Gosala, V.; Stender, M.; Braun, M.; Ehlers, S. Predicting remaining useful life of lithium-ion batteries: A review of degradation mechanisms and open-source data availability. *Future Batter.* **2025**, *8*, 100124. https://doi.org/10.1016/j.fub.2025.100124.
33. Navidi, S.; Thelen, A.; Li, T.; Hu, C. Physics-informed machine learning for battery degradation diagnostics: A comparison of state-of-the-art methods. *Energy Storage Mater.* **2024**, *68*, 103343. https://doi.org/10.1016/j.ensm.2024.103343.
34. Arif, N.A.; Mekhilef, S.; Seyedmahmoudian, M.; Stojcevski, A. Degradation in Li-ion batteries and capacity decline under different cycling conditions: A comprehensive review. *J. Energy Storage* **2026**, *154*, 121270. https://doi.org/10.1016/j.est.2026.121270.
35. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* **2019**, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.
36. Lu, Q.; Zhu, D.; Li, M. Research on Remaining Useful Life Prediction of Bearing with Digital Twin and TSMixer. *Appl. Sci.* **2026**, *16*, 6762. https://doi.org/10.3390/app16136762.
37. Wang, F.; Zhai, Z.; Zhao, Z.; Di, Y.; Chen, X. Physics-informed neural network for lithium-ion battery degradation stable modeling and prognosis. *Nat. Commun.* **2024**, *15*, 4332. https://doi.org/10.1038/s41467-024-48779-z.
38. Miettinen, K. *Nonlinear Multiobjective Optimization*; International Series in Operations Research & Management Science; Springer: Boston, MA, USA, 1999; Volume 12. https://doi.org/10.1007/978-1-4615-5563-6.
39. Blank, J.; Deb, K. pymoo: Multi-Objective Optimization in Python. *IEEE Access* **2020**, *8*, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567.
40. Saha, B.; Goebel, K. Battery Data Set. NASA Prognostics Data Repository, NASA Ames Research Center, 2007. Available online: https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository (accessed on 23 July 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.