Towards Green PHM: Adaptive Early Stopping for Sustainable Neural Architecture Search in Industrial Applications

Solís-Martín, David[1] and Galán-Páez, Juan[2] and Borrego-Díaz, Joaquín[3]

[1,2,3] *University of Seville, Seville, Spain*
dsolis@us.es, juangalan@us.es, jborrego@us.es

ABSTRACT

Neural Architecture Search has revolutionized Prognostics and Health Management, yet adoption is often hindered by the massive carbon footprint generated during the evaluation of candidate architectures. To address this sustainability challenge, this work introduces a Green AI framework that significantly reduces the energy consumption of such searches through an intelligent and adaptive early stopping mechanism. The approach utilizes Prototypical Networks for regression to extrapolate learning curves from partial data, predicting final model performance early in the training process.

A distinct sustainability advantage of using Prototypical Networks lies in the inherent data efficiency and superior generalization capabilities of the architecture. By leveraging metric learning, the framework avoids the energy intensive process of training task specific predictors from scratch. Instead, it enables few shot transfer across diverse domains, minimizing the total computational overhead of the search process. Furthermore, a key contribution of this framework is the dynamic adaptation of the decision logic as the optimization process evolves. By utilizing a decision tree classifier that adjusts thresholds based on the progression of the search, the system becomes increasingly selective and prioritizes computational resources for the most promising candidates.

The proposed framework was validated across sixty one thousand learning curves from fifty diverse datasets. Experimental results demonstrate a drastic reduction in total computational hours, achieving 57% of decrease in training time while maintaining high diagnostic fidelity. The system consistently identified top tier model configurations, reaching a mean selection rank of 0.9 across tested industrial scenarios. These results prove that high performance industrial intelligence can be achieved without the prohibitive environmental costs typically associated with large scale architecture optimization.

1. INTRODUCCION

While Machine Learning (ML) has fundamentally reshaped data-driven industries, its application within Prognostics and Health Management (PHM) represents a sophisticated domain characterized by high-dimensional, time-varying, and often imbalanced industrial datasets. By leveraging diverse ML paradigms—ranging from classical statistical learning to advanced deep architectures—PHM frameworks facilitate robust condition-based monitoring and predictive diagnostics. This integration enables the early detection of subtle anomalies and the accurate estimation of Remaining Useful Life (RUL), directly translating to optimized maintenance strategies, minimized unscheduled downtime, and extended asset longevity. Beyond these technical gains, modern ML-driven PHM serves as a strategic bridge between raw sensor data and actionable intelligence, providing engineers with the diagnostic transparency required for reliable, expert-driven decision-making in critical environments (Borrego-Díaz & Galán-Páez, 2022).

The efficacy of ML models in industrial contexts is fundamentally dictated by the selection of appropriate algorithms and their underlying configurations, which determine the robustness of feature extraction and subsequent predictive performance. While established paradigms—ranging from classical ensembles like Random Forests to advanced deep architectures like Transformers—offer diverse capabilities, the manual selection of models and the tuning of associated hyperparameters (e.g., kernel types, regularization terms, or learning rates) remains a labor-intensive process requiring significant domain expertise. To address these limitations and move beyond iterative trial and error, Automated Machine Learning (AutoML) has emerged as a robust framework for streamlining model design. By systematically exploring expansive search spaces, AutoML identifies optimal pipelines that balance predictive accuracy with industrial resource constraints, ensuring an objective, repeatable, and efficient path toward high-performing PHM solutions.

Despite its potential to revolutionize industrial diagnostics, the automated search for optimal machine learning pipelines

is notoriously resource-intensive. Early benchmarks in Neural Architecture Search (NAS), such as NASNet (Zoph, Vasudevan, Shlens, & Le, 2018) and AmoebaNet-A (Real, Aggarwal, Huang, & Le, 2019), required thousands of GPU days to running high-end hardware continuously for several years to produce a single optimized model. While these figures represent the upper bound of deep learning complexity, the broader challenge of AutoML remains significant: balancing exhaustive search spaces with the real-time constraints of industrial deployment. For PHM applications, where models must often be updated or retrained to account for evolving machine degradation, reducing this computational footprint is critical. Developing efficient search strategies is essential for making automated PHM solutions both economically viable and practically accessible for diverse engineering environments.

Bayesian Optimization (BO) provides a robust probabilistic framework for optimizing objective functions that are computationally expensive or lack a closed-form analytical expression. By constructing a surrogate model—most commonly a Gaussian Process (GP)—BO approximates the performance of the system surface based on a limited set of prior evaluations. An acquisition function then strategically identifies the next configuration to test, effectively balancing the exploration of unknown regions with the exploitation of known high-performing areas. This targeted approach typically requires significantly fewer iterations than exhaustive methods like grid or random search.

Within the context of Prognostics and Health Management, BO serves as a systematic alternative to traditional manual tuning, which historically relied on expert-driven analysis of learning curves to guide model selection and early-stopping criteria (Klein, Falkner, Springenberg, & Hutter, 2017a; Domhan, Springenberg, & Hutter, 2015). Its utility spans the entire ML spectrum, from optimizing classical hyperparameter sets (e.g., regularization terms and kernel bandwidths in SVMs (Snoek, Larochelle, & Adams, 2012)) to modern AutoML and Neural Architecture Search (NAS). For instance, frameworks like NASBOT (Kandasamy, Neiswanger, Schneider, Poczos, & Xing, 2018) demonstrate how BO can be adapted to navigate complex, non-Euclidean search spaces, making it a vital tool for developing high-fidelity, resource-efficient PHM solutions.

The integration of AutoML and BO into PHM frameworks directly addresses the dual imperatives of the "Sustainability in/by PHM" track. By automating the discovery of high-performance diagnostic models, these methodologies optimize asset lifecycles and reduce industrial waste through precise predictive maintenance—embodying Sustainability by PHM. Simultaneously, by replacing traditional, brute-force search methods with strategically efficient probabilistic frameworks, this approach significantly lowers the carbon footprint and energy consumption of the AI development process itself, fulfilling the criteria for Sustainability in PHM. This research demonstrates how lean, data-driven architectures can serve as a technical and environmental catalyst, aligning industrial reliability with the broader Sustainable Development Goals (SDGs) by balancing high-fidelity prognostic accuracy with minimized computational demand.

This work introduces an optimized AutoML framework specifically applied in PHM applications. By implementing a novel performance estimation strategy based on Metric Learning, this research leverages early learning curves to prune suboptimal trainings during the BO process. This approach addresses critical bottlenecks identified in existing literature (Solis-Martin, Galan-Paez, & Borrego-Diaz, 2024; Solís-Martín, Galán-Páez, & Borrego-Díaz, 2025) through the following methodological advancements:

- Threshold-free Decision Making: Traditional early-stopping methods rely on fixed performance thresholds to terminate training. We introduce a decision tree classifier that dynamically evaluates whether a candidate model trajectory is likely to surpass the current best-in-class performance, eliminating the need for arbitrary manual thresholds.
- Flexible Temporal Observation: Unlike earlier estimators constrained to fixed observation windows, our framework is designed to process an arbitrary number of training epochs. This allows for adaptive early-stopping decisions that respond to varying convergence rates typical of complex industrial datasets.
- Metric Learning for Curve Extrapolation: We replace conventional sequence models with prototypical networks adapted for regression. This metric learning approach treats historical learning curves as a support set, providing superior extrapolation capabilities and intrinsic uncertainty estimates based on support set variation.
- Interpretable Rule-based Framework: The proposed architecture synthesizes metric learning predictions with validation metrics and temporal data through an interpretable decision-making layer. This ensures that model-selection logic remains transparent for safety-sensitive PHM environments.

These contributions establish a sophisticated, resource-aware framework for model discovery. By significantly reducing the computational overhead of finding optimal diagnostics, this work advances the goals of Sustainability in PHM while maintaining the high-fidelity predictive accuracy required for asset health monitoring.

The source code and datasets used in this work are publicly available at `https://github.com/dasolma/pndt`.

2. RELATED WORK

The evolution of early stopping mechanisms in AutoML reveals distinct methodological paradigms, each addressing specific aspects of computational efficiency while exhibiting particular limitations. This section presents a systematic comparison between established approaches and the proposed framework.

The foundational work of Baker et al. (Baker, Gupta, Raskar, & Naik, 2018) established performance-based early stopping through direct final performance prediction, requiring practitioners to define appropriate threshold values for termination decisions. This approach introduces the challenge of threshold selection, which significantly impacts both computational efficiency and solution quality.

The Hyperband algorithm (Li, Jamieson, DeSalvo, Rostamizadeh, & Talwalkar, 2017) overcomes this limitation through resource-based termination, removing the need for explicit thresholds by adopting adaptive resource allocation. However, it does not use curve estimators and instead relies solely on the current performance of each training configuration. The BOHB framework (Falkner, Klein, & Hutter, 2018) represented a synthesis of Bayesian optimization principles with Hyperband resource allocation, yet continued to operate within fixed temporal constraints and limited historical information utilization.

Klein et al. (Klein, Falkner, Springenberg, & Hutter, 2017b) advanced the field by introducing Bayesian uncertainty quantification in learning curve prediction, providing probabilistic estimates of model performance. However, their methodology maintained reliance on predetermined observation windows and did not systematically integrate uncertainty measures into the stopping criteria. Recent work by Egele et al. (Egele, Mohr, Viering, & Balaprakash, 2024) demonstrated the potential for extreme early stopping through single-epoch evaluation. While computationally efficient, this approach lacks the sophistication necessary for complex optimization scenarios and provides no uncertainty quantification. Adriaensen et al. (Adriaensen, Rakotoarison, Müller, & Hutter, 2024) further advanced the field through the development of efficient Bayesian learning curve extrapolation using prior-data fitted networks, emphasizing probabilistic modeling techniques for improved prediction accuracy.

The analysis of existing methodologies reveals four fundamental limitations that constrain their effectiveness in complex industrial optimization scenarios. Threshold dependency remains a critical issue, as most approaches require manual parameter tuning. Temporal inflexibility limits adaptability, as predetermined observation windows may not align with the natural learning dynamics of different architectures. Information underutilization represents a significant inefficiency, as valuable historical data from previous evaluations is typically discarded. Finally, uncertainty neglect in decision-making processes leads to suboptimal stopping decisions, particularly where prediction confidence varies significantly.

The current work addresses these limitations through four key innovations designed for the rigors of PHM. Automated decision-making eliminates threshold dependency by employing decision tree classifiers that learn optimal stopping criteria from training data. Flexible temporal processing enables the framework to operate with arbitrary numbers of observed epochs, adapting to the natural learning characteristics of different models rather than imposing fixed constraints.

Metric learning integration systematically incorporates historical learning curves from previous evaluations of the Bayesian Optimization process, enabling the framework to leverage accumulated knowledge throughout the optimization process. This approach transforms each evaluation from an isolated decision into a component of a comprehensive learning system. Uncertainty-aware termination explicitly computes and integrates prediction certainty measures into the stopping criteria, providing more robust decision-making capabilities that account for prediction confidence, ensuring the reliability required for prognostic systems.

3. DESCRIPTION OF THE PROPOSAL

Figure 1 summarizes the overall process of BO using the proposed approach. The workflow begins when the BO algorithm selects a candidate model configuration to be evaluated (label A). At each training epoch, the system evaluates the termination criteria to ensure computational efficiency and resource sustainability.

First, it checks if the standard convergence or maximum epoch limits have been reached (label B). If these standard criteria are not met, the proposed learning curve extrapolation policy is triggered (label C). Within this policy, the Metric Learning module (label D) estimates the final performance of the current model based on its partial training behavior, while simultaneously computing an uncertainty measure for this extrapolation (label E).

These values, combined with the current validation accuracy and the best BO performance observed so far, serve as input features for the Decision Tree Classifier (label F). This classifier determines the final action: if a Stop is recommended, the training is aborted, and the BO process generates a new candidate configuration (label A). Otherwise, the training proceeds to the next epoch (label G). This mechanism ensures that industrial resources are not wasted on unpromising models, directly contributing to the Sustainability in PHM paradigm.

The following sections detail each key component of the proposed framework, including the mathematical formulation of the metric learning approach and the architectural details of the decision-making classifier.

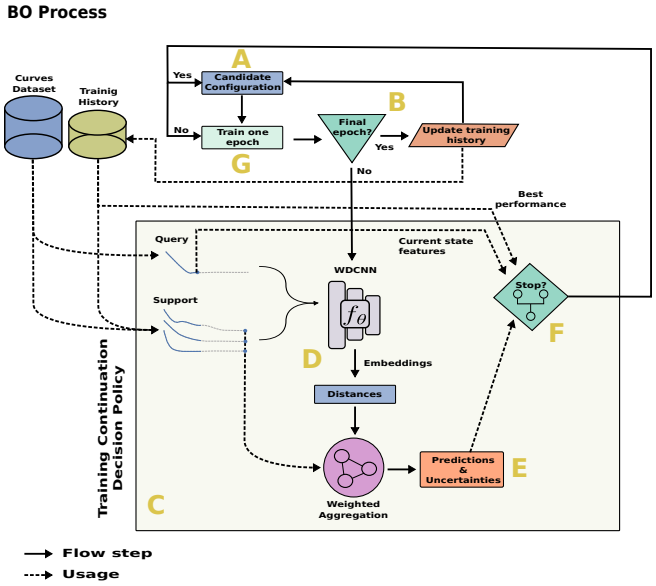

Figure 1. Overview of the proposed BO framework with metric learning-based early stopping for AutoML in PHM applications.

3.1. Metric Learning Curve Extrapolation

While existing literature in Prognostics and Health Management (PHM) has explored sequence-based architectures such as Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs) (Solis-Martin et al., 2024; Solís-Martín et al., 2025), and Transformers (Adriaensen et al., 2024) as estimators for learning curves, the potential of metric learning remains largely untapped. This work proposes the use of a Prototypical Network (PN) (Snell, Swersky, & Zemel, 2017) to address the challenges of learning curve extrapolation in resource-constrained environments.

Originally designed for few-shot classification, PNs assign query samples to the class whose prototype is closest within a learned embedding space. A prototype is defined as the centroid of feature embeddings for support examples belonging to a specific class. Formally, for a class c, the prototype p_c is computed as:

$$p_c = \frac{1}{K} \sum_{x_i \in S_c} f(x_i), \tag{1}$$

where $f(\cdot)$ denotes the embedding function, typically a neural network encoder, and S_c represents the set of support samples for class c. By mapping samples into a structured metric space, PNs ensure that similar instances are clustered together, providing an efficient mechanism for pattern recognition.

In this work, the PN framework is adapted from its traditional classification roots to solve a regression problem: estimating the final performance of a predictive model. Unlike hyperspherical variants that embed targets as directions (Mettes, Van der Pol, & Snoek, 2019), the proposed approach estimates a continuous target value by aggregating support information through a distance-based weighting mechanism in the embedding space.

Given a set of support samples $S = \{x_1, \ldots, x_K\}$ with corresponding known performance targets $Y_S = \{y_1, \ldots, y_K\}$, and a set of query samples $Q = \{q_1, \ldots, q_N\}$ representing partial learning curves, embeddings are generated using a shared encoder $f(\cdot)$:

$$e_q = f(q), \quad e_s = f(x), \quad \forall q \in Q,\, x \in S. \tag{2}$$

The distance between each query embedding e_{q_i} and all support embeddings e_{s_j} is calculated using a similarity metric:

$$d_{ij} = -\text{dist}(e_{q_i}, e_{s_j}), \quad \forall i \in [1, N],\, j \in [1, K]. \tag{3}$$

These distances are normalized into a probability distribution over the support set using a softmax function:

$$w_{ij} = \frac{\exp(d_{ij})}{\sum_{k=1}^{K} \exp(d_{ik})}, \tag{4}$$

where w_{ij} represents the weight of support point x_j for query q_i. To refine the extrapolation and emphasize high-confidence neighbors, a power-based accentuation hyperparameter γ is applied to the weights:

$$\tilde{w}_{ij} = \frac{w_{ij}^{\gamma}}{\sum_{k=1}^{K} w_{ik}^{\gamma}}. \tag{5}$$

The final prediction for each query is then derived as a weighted sum of the support targets:

$$\hat{y}_i = \sum_{j=1}^{K} \tilde{w}_{ij} \cdot y_j. \tag{6}$$

The encoder $f(\cdot)$ is trained by minimizing the Mean Squared Error (MSE) between the weighted prediction $\hat{y}_i$ (Equation 6) and the known final performance y_i of each query curve:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{7}$$

This choice is consistent with the regression formulation of the task and provides a differentiable objective that directly minimizes prediction error over the support-query structure of the PN.

This mechanism enables the model to extrapolate final performance metrics by leveraging a learned similarity structure. The choice of PNs for this task offers two distinct strategic advantages for PHM applications. First, the support set serves as a dynamic reference system, allowing the framework to incorporate the most recent training history without retraining the entire estimator. Second, by performing stochastic variations of the support set for a fixed query, the system can generate a distribution of outcomes. This allows for the calculation of an uncertainty measure, such as the standard deviation across predictions, providing a critical reliability metric for the subsequent decision-making layers.

For each training batch (episode), samples are selected dynamically with a specific truncation length t_{len} from a set of valid operational horizons. To ensure that similarity is evaluated under identical observation windows, both the Query Set Q (comprising a batch size of M instances) and the Support Set S (comprising K reference instances, or *shots*) are bounded by this same temporal constraint t_{len}.

3.2. Decision Tree Logic for Adaptive Early Stopping

The determination of whether to persist with or terminate the training of a model configuration is formulated as a supervised classification task. Rather than relying on rigid, manually defined thresholds, this work proposes a Decision Tree (DT) classifier to synthesize partial training trajectories into an optimal termination policy. This approach ensures that the decision-making process remains transparent and verifiable, a core requirement for Prognostics and Health Management (PHM) systems deployed in safety-critical industrial environments.

Each epoch of the training process is treated as a potential decision point, where a specific feature vector is extracted to represent the current state of a model training relative to the global optimization goal.

For fault diagnosis tasks, the validation objective is the categorical cross-entropy loss, while for RUL estimation tasks the Mean Squared Error (MSE) is employed. Although these metrics operate on different absolute scales, the decision-making framework remains agnostic to this heterogeneity by design. Both the Expected Improvement (EI) and Validation Improvement (VI) features are defined as relative ratios with respect to the best performance observed within the current BO run (Equations 8 and 9). This normalization ensures that all features fed to the Decision Tree classifier are dimensionless and scale-invariant, preserving comparability across task types without requiring an explicit unified metric. Consequently, the learned termination policy generalizes across both classification and regression PHM scenarios without modification.

Central to this logic is the Expected Improvement (EI) metric, which captures the predicted performance gain of the current model, as estimated by the PN, relative to the best-performing configuration found by the BO process thus far. It is defined as:

$$\text{EI} = \frac{\text{predicted performance} - \text{best performance}}{\text{best performance}} \tag{8}$$

To mitigate the impact of outliers and ensure numerical stability, EI values are clipped to the interval $[-1, 1]$. Complementing this is the Validation Improvement (VI) feature, which assesses the actual, observed performance of the model at the current epoch against the search benchmark:

$$\text{VI} = \frac{\text{validation performance} - \text{best performance}}{\text{best performance}} \tag{9}$$

Consistent with the treatment of EI, VI values are clipped. This provides the framework with a real-time assessment of the model's competitive standing within the broader AutoML search space.

Furthermore, the Expected Improvement Uncertainty (EIU) is incorporated to represent the confidence level of the extrapolation of the PN.

In addition to the core improvement metrics, the proposed framework incorporates dynamic signal features to better capture the transient behavior of the training process. These features are designed to enhance the ability of the DT classifier to distinguish between meaningful convergence and stochastic noise.

The first auxiliary feature is the Validation Velocity (VV), which measures the instantaneous rate of improvement by calculating the difference in performance between the current and the preceding training epoch:

$$\text{VV}_t = \text{validation performance}_t - \text{validation performance}_{t-1} \tag{10}$$

By monitoring the velocity of the learning curve, the framework can identify periods of stagnation or rapid gains that may not be fully captured by static performance values. This provides the decision logic with a temporal context, allowing it to recognize when a improvement of the model has plateaued.

The second feature addresses the challenge of validation noise through an Exponential Moving Average (EMA) of the performance signal. Industrial data and complex architectures often produce erratic validation curves with significant fluctuations. To mitigate this, a smoothed validation metric is computed:

$$\text{EMA}_t = \alpha \cdot \text{validation performance}_t + (1 - \alpha) \cdot \text{EMA}_{t-1} \tag{11}$$

where α is a smoothing factor derived from a predefined span (e.g., span=3). This EMA feature allows the classifier to base its termination decisions on a more stable representation of the model's progress, preventing premature stopping caused by temporary performance dips or outliers.

To guarantee that the decision tree generalizes effectively across different phases of the training lifecycle, the current epoch is normalized relative to the maximum expected trajectory length:

$$e' = \frac{e}{|\text{BC}|} \tag{12}$$

Where e is the epoch number, BC is the best curve found in the current BO iteration, and $|\cdot|$ return the length of the curve. This normalized temporal feature prevents the model from developing biases toward fixed absolute time steps, enabling scaling across experiments with varying resource budgets.

Furthermore, a critical contextual benchmark is introduced: the Epoch-Specific Baseline (BC_e). Rather than evaluating a model strictly against the final, absolute performance of the historical champion, this baseline measures the competitor's trajectory against the historical champion's performance at that exact same epoch. This ensures a fairer, time-aligned comparison during the early and intermediate stages of training.

The binary target for training the DT classifier is generated through a supervised assessment of this localized competitiveness. A configuration is labeled to continue if its current validation loss remains within a conservative 5% tolerance threshold of the epoch-specific baseline:

$$\text{continue} = \begin{cases} \text{True} & \text{if } CC_e \leq 1.05 \cdot BC_e \\ \text{False} & \text{otherwise} \end{cases} \tag{13}$$

Where CC_e is the validation error in the current epoch e. This localized target definition enables the framework to internalize optimal stopping boundaries based on current efficiency rather than retrospective hindsight. By effectively pruning non-competitive models early in their lifecycle without demanding unrealistic immediate parity with the final benchmark, the system maximizes computational efficiency and aligns with the sustainable AI practices prioritized in this conference track.

3.3. Tree prunning by leaf merging

To improve the interpretability and robustness of the decision tree used for early stopping, a post-processing step is applied to simplify its structure. The procedure recursively merges leaf nodes that predict the same class, effectively pruning unnecessary branches while preserving decision consistency.

Given a trained decision tree, the simplification algorithm performs a depth-first traversal starting from the root. At each internal node, it checks whether both child nodes are leaves and whether they predict the same class. If these conditions are met, the internal node is converted into a leaf node by aggregating the counts from its children and updating the number of samples accordingly.

To determine the predicted class at each leaf, a custom rule is applied based on a user-defined *negative class threshold* $\theta \in [0, 1]$. Specifically, let $p_{\texttt{False}}$ denote the proportion of samples at a node that belong to the negative class (i.e., `False`). The predicted class is assigned according to:

$$\text{class}(v) = \begin{cases} \texttt{False}, & \text{if } p_{\texttt{False}} > \theta, \\ \texttt{True}, & \text{otherwise.} \end{cases} \tag{14}$$

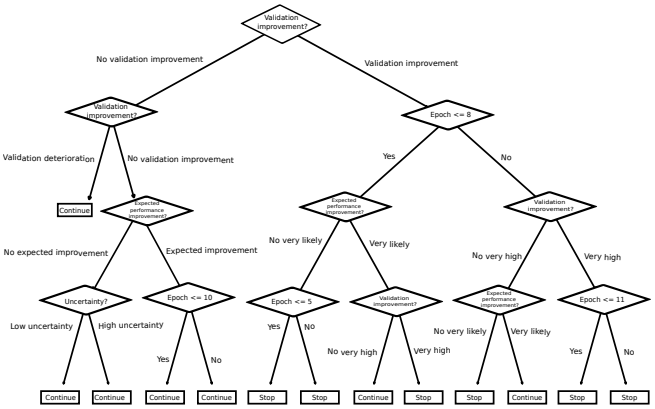

Figure 2. Example of an original tree without prunning.

The recursive simplification terminates when no further merges are possible. This process reduces model complexity, enhances interpretability, and stabilizes decisions near uncertain regions of the feature space.

This rule allows the user to enforce more conservative or optimistic stopping behavior depending on the application context. A higher value of θ biases the tree towards recommending continuation unless there is strong evidence to stop. Figure 2 shows an example of a tree without prunning, while Figures 3 and 4 display the reduced trees using $\theta = 0.6$ and $\theta = 0.8$, respectively. As can be seen, with $\theta = 0.8$ the behavior becomes more conservative, increasing the number of decisions to continue training.

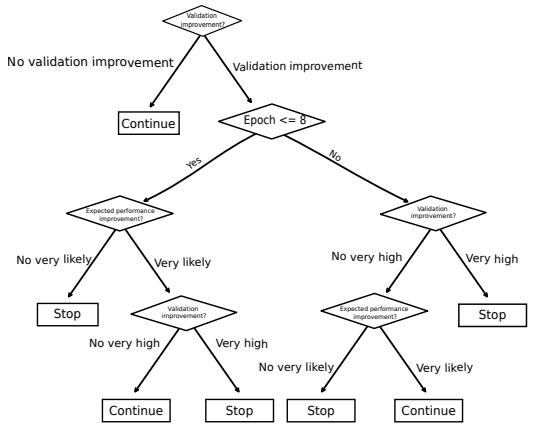

Figure 3. Pruned tree obtained with $\theta = 0.6$

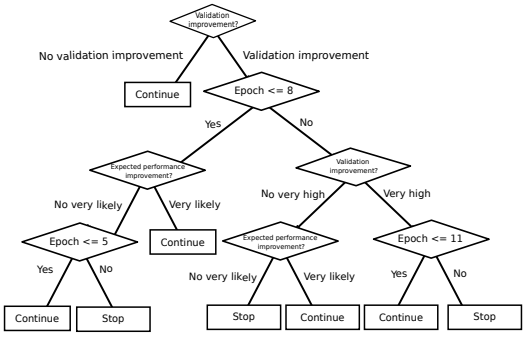

Figure 4. Pruned tree obtained with $\theta = 0.8$

4. EXPERIMENTAL SETUP

This section describes the experimental configuration used to evaluate the proposed uncertainty estimation approach. The evaluation is conducted using a large collection of training curves in the PHM context and involves the configuration of a neural network architecture along with the decision tree component for stopping criteria.

4.1. Dataset

To demonstrate the proposed methodology, we utilized a comprehensive dataset consisting of 61,000 learning curves generated from 50 distinct datasets (with 62 total tasks). The collection process was facilitated by the Python tool *phmd* (Solís-Martín, Galán-Páez, & Borrego-Díaz, 2025), which aggregates and standardizes open-source industrial data.

The repository encompasses the two primary problem archetypes in PHM:

1. **Fault Diagnosis (Classification):** Tasks aimed at identifying specific failure modes (e.g., inner vs. outer race bearing faults) from vibration or acoustic emission signals. Prominent examples included in our evaluation are the Case Western Reserve University (CWRU) bearing data and the IMS bearing dataset.
2. **Remaining Useful Life (RUL) Estimation (Regression):** Tasks focused on predicting the time-to-failure of a system based on the history of sensor readings.

4.2. Experimental framework

The experimental methodology is designed to ensure the generalizability of the proposed framework across a wide range of industrial scenarios. The evaluation utilizes a comprehensive pool of 50 datasets, partitioned into distinct subsets to facilitate the staged training of the metric learning and decision-making components.

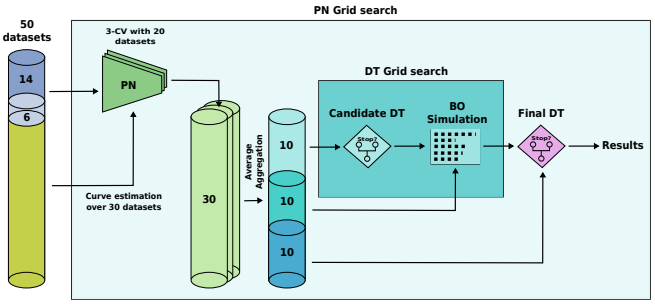

Figure 5. Hierarchical experimental framework and data partitioning strategy: The process initiates with a cross-validation phase on 20 datasets to train the PN. Performance estimations are then mean-aggregated across 30 datasets to provide robust inputs for the decision-making layer. This layer is developed through a three-stage pipeline consisting of Candidate DT training, BO simulation, and terminal evaluation on independent test sets to ensure the generalizability of the early-stopping policy.

The process begins with the development of the PN for learning curve estimation. This module is trained and validated using a 20-dataset subset, specifically partitioned into 14 for training and 6 for internal cross-validation. Once the curve estimator is optimized, it is deployed to generate performance estimations for the remaining 30 datasets. To ensure robustness, the estimations from the three networks trained during the cross-validation phase are mean-aggregated, producing the final performance predictions for the 30-dataset pool.

These estimations, alongside the historical learning curve data, are then utilized to generate the training data for the hierarchical decision-making layer. This phase involves a three-way split of the 30 datasets into groups of 10 datasets each:

- Candidate DT Training: The first 10 datasets are used to identify the optimal DT classifier. This model learns to map the PN performance estimates and auxiliary features, such as validation velocity and EMA, into a preliminary stopping policy.
- BO Simulation: The next 10 datasets serve as the environment for a BO simulation. During this stage, the Candidate DT is integrated into the AutoML loop to refine

the search process and evaluate the efficiency of early termination in real-time.

- Final DT and Results: The final 10 datasets are reserved for the terminal evaluation. A Final DT is consolidated using the insights from the BO simulation and tested against these unseen datasets to produce the final performance results.

By employing this stratified experimental setup, the framework is subjected to a rigorous validation process. This ensures that the early-stopping rules are not overfitted to specific architectures or datasets, but instead represent a generalized solution for resource-efficient model discovery in the PHM domain. This multi-stage validation reinforces the technical reliability and economic viability of the proposed approach, directly aligning with the sustainability objectives of the conference.

4.3. Parameterized WDCNN Encoder

The encoder architecture employed in this study is based on the Wide Deep Convolutional Neural Network (WDCNN) (Zhang, Peng, Li, Chen, & Zhang, 2017), a recognized benchmark in PHM for its ability to extract features from noisy signals. As illustrated in Figure 6, the network follows a hierarchical structure defined by the kernel size (K), stride (S), number of filters (F), and activation function (A).

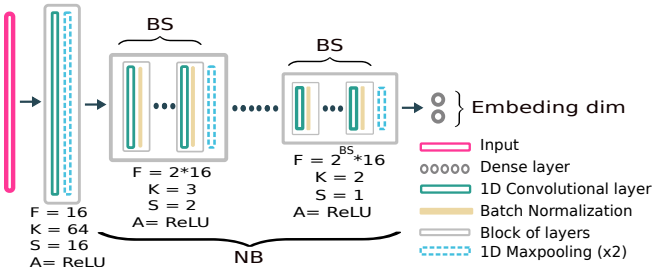

Figure 6. Parameterized WDCNN encoder architecture. The configuration is defined by the base number of filters (16), the number of convolutional blocks (NB), and the internal block size (BS), allowing the optimization framework to adapt the network depth and capacity for specific PHM tasks.

The defining feature of this architecture is the initial layer, which utilizes a wide kernel and a larger stride to capture high-frequency noise and long-term dependencies. Subsequent layers transition to smaller kernels (K=3) and a stride of S=1 to perform deeper feature extraction. To facilitate automated optimization via the proposed framework, the following structural hyperparameters are made configurable:

- Number of Blocks (NB): Determines the depth of the network, defining the hierarchy of feature learning.
- Block Size (BS): Defines the number of convolutional operations within each individual block.
- Embedding Dim (ED): Defines the size of the latent vector generated by the encoder.

These hyperparameters are explored during the experimentation phase, as shown in Figure 5, using a grid search. This parameterization allows for a systematic exploration of the architectural design space, balancing computational efficiency with representational capacity. Table 1 outlines the complete parameter ranges used to configure the WDCNN encoder architecture.

Parameter	Values
Embedding Dim (ED)	{8, 16}
Block Size (BS)	{1,2, 3}
Number of Blocks (NB)	{1,2}

Table 1. Parameter ranges for the WDCNN encoder architecture configuration.

4.4. Decision Tree Training and Cost-Aware Optimization

The DT component undergoes a systematic grid search to optimize the architectural parameters that govern model complexity and class balance. This optimization focuses on two primary hyperparameters: the maximum tree depth, which prevents overfitting by controlling the hierarchy of the rules, and the positive class weight, which balances the importance assigned to Continuation decisions versus Stop decisions.

A specialized weighted accuracy metric is employed as the objective function to reflect the asymmetric risks inherent in PHM. The scoring function is defined as a linear combination of class-specific accuracies:

$$\text{score} = \frac{2}{5} \cdot \text{Accuracy}_{\text{Stop}} + \frac{3}{5} \cdot \text{Accuracy}_{\text{Continue}} \qquad (15)$$

The asymmetric weighting in Equation 15 reflects the inherent cost asymmetry of the early-stopping decision in industrial PHM. A false stop (terminating a promising configuration) irreversibly removes a candidate from the search space, whereas a false continue (retaining a suboptimal configuration) incurs only a bounded computational overhead before the next evaluation epoch. Formally, if c_s and c_c denote the costs of misclassifying a Stop and a Continue decision respectively, the weight ratio $w_c/w_s = 3/2$ implies that missing a top-tier configuration is considered 1.5 times more costly than unnecessary computation.

The parameter search space encompasses tree depths ranging from 2 to 9, providing a balance between model interpretability and expressiveness. The positive class weights are explored within the range of 10 to 90, while the negative class weight remains fixed at unity. This range allows for a sys-

tematic investigation of class imbalance mitigation strategies, ensuring the policy remains robust even when stopping opportunities are frequent.

For each parameter combination, a Decision Tree is trained and a BO simulation is executed, as described in Section 4. This simulation-based validation ensures that the selected hyperparameters prioritize long-term search efficiency and diagnostic accuracy over simple classification metrics. The final model is selected based on its cross-validated performance, ensuring a termination policy that generalizes across diverse industrial datasets.

4.5. Strategy Simulation and Performance Metrics

To validate the practical utility of the proposed decision-making framework, a strategy simulation is conducted. This process replays historical BO traces to evaluate how the learned decision tree policy would have performed if deployed during the original search. The simulation logic, as implemented in the experimental pipeline, focuses on three primary dimensions: temporal efficiency, resource conservation, and model fidelity.

The simulation iterates through unseen experimental units, evaluating the termination criteria at every epoch of the training process. A key feature of this simulation is the implementation of a robustness mechanism known as a patience counter. To prevent premature termination due to transient fluctuations in the learning curve, the system requires three consecutive Stop signals from the classifier before officially aborting the training process. Once this threshold is met, the remaining epochs for that specific model configuration are marked as avoided, and the BO process immediately moves to the next candidate.

The impact of the strategy is measured through the following metrics:

- Epochs Avoided: The total number of training iterations eliminated by the early-stopping policy. This is calculated as the difference between the full training duration (U_{real}) and the actual epochs run before pruning (E_{run}).
- Avoided Training Time: A temporal projection of energy savings, estimated by scaling the original training time by the ratio of avoided epochs:

$$T_{avoided} = \sum \left(\frac{T_{total}}{U_{real}} \times (U_{real} - E_{run}) \right) \quad (16)$$

- Performance Score: A ratio comparing the final validation loss of the best model found using the pruning strategy against the global optimum found by the original exhaustive search.
- Selection Rank: This metric evaluates the quality of the final model choice by identifying its ordinal rank among all evaluated candidates within a specific experiment. A rank of zero indicates that the strategy successfully identified the absolute best model, while higher values quantify the distance from the global optimum in the architectural search space.

A composite score is finally computed to provide a holistic view of the effectiveness of the framework, balancing the performance score and the time-saving score with equal weighting. This evaluation ensures that the proposed framework not only accelerates the discovery of PHM models but also preserves the high-fidelity prognostic capabilities required for industrial health monitoring, directly contributing to the technical and environmental sustainability of the system.

5. RESULTS AND DISCUSSION

The experimental evaluation of the proposed metric learning-based early stopping framework reveals a substantial reduction in computational overhead while preserving high diagnostic fidelity. Table 2 summarizes the performance metrics for the top-performing PN+DT configurations identified during the hierarchical validation process.

Table 2. Top-5 performance metrics for representative experimental units within the test subset.

Validation Score	Test Mean Rank	Saved Test Time (%)
0.7580	0.9458	57%
0.7575	1.0688	57%
0.7571	1.0846	57%
0.7569	0.9328	57%
0.7568	0.9418	57%

A core finding of this study is the high efficiency of the termination policy. In the most balanced configuration, the framework achieved a test mean rank near to 1.0. This indicates that the selected model architecture was consistently the top-tier or near-optimal candidate compared to the global optimum identified by exhaustive search. Crucially, this high-fidelity selection was achieved while avoiding between a 57% of the total training time.

To provide a concrete quantification of the sustainability claims, an energy consumption estimate is derived based on the experimental hardware and the epoch budget of the search process. All experiments were conducted on a single NVIDIA GTX 1080 Ti GPU, with a Thermal Design Power (TDP) of 250 W. Each Bayesian Optimization experiment evaluates up to 100 candidate configurations over a maximum of 100 epochs each, yielding a total budget of 10,000 epochs per experiment. Assuming a conservative epoch duration of 30 s, the baseline energy consumption per experiment is approximately 20.8 Wh.

The overhead introduced by the Prototypical Network training phase amounts to 60 epochs (3 cross-validation folds $\times$

20 training epochs), representing only 0.6% of the total epoch budget, and is fully amortized within a single BO run. Accounting for this overhead, the proposed early-stopping framework reduces energy consumption to approximately 9.1 Wh per experiment at the observed 57% time savings, yielding a net saving of 11.7 Wh per experiment. Across the 10 test datasets, the estimated total energy saving amounts to approximately 117 Wh. This figure represents a reduction of 56% in GPU energy consumption relative to an exhaustive search baseline, directly quantifying the environmental benefit of the proposed framework in industrial AutoML scenarios.

It should be noted that these estimates are derived from the nominal TDP value and a fixed epoch duration, and therefore represent a conservative upper bound on actual energy consumption. Real-world values may differ depending on GPU utilization rate and dataset characteristics. Exact measurements could be obtained using energy-monitoring tools such as `CodeCarbon` or `nvidia-smi` in future deployments.

The correlation analysis between the validation simulation score and the mean selection rank provides further evidence of the framework stability. A negative correlation of approximately -0.12 indicates that higher validation simulation scores generally correspond to lower, and therefore superior, ordinal ranks on test. This relationship confirms that the decision making policy does not achieve computational savings at the expense of model quality. Instead, the framework consistently directs the search toward the most promising architectural candidates, ensuring that even with aggressive early stopping, the final selected models remain at the top of the performance hierarchy. This consistency across diverse datasets demonstrates that the DT captures the underlying patterns of successful learning curves, allowing for a reliable reduction in search time without losing the global optimum.

Table 3 benchmarks the proposed framework against two established resource-based early-stopping strategies: Hyperband (HB) (Li et al., 2017) and BOHB (Falkner et al., 2018). To evaluate the performance of BOHB, we implemented a Tabular Benchmark Simulator. When the Bayesian optimizer (TPE) explores the continuous space, its suggestions are mapped to the nearest evaluated experiment using a Euclidean distance metric. We opted for this tabular approach instead of a surrogate model (Surrogate Benchmark) to guarantee the empirical fidelity of the learning curves and preserve the true stochastic behavior of the training runs, thereby avoiding the prediction biases typically introduced by regression models.

The comparative analysis presented in Table 3 evaluates the trade-off between the selection rank and the avoided training time across various resource-allocation strategies. While BOHB yields the highest training time reduction at 73.1%, it introduces a performance penalty, resulting in a higher test selection rank of 1.196 ± 0.072. In contrast, the proposed integrated configuration ($BO + PN + DT$) minimizes the test selection rank to an optimal 0.999 ± 0.128 while still maintaining a substantial time reduction of 56.9%. This confirms that incorporating prototypical networks for learning curve extrapolation effectively preserves diagnostic fidelity without sacrificing computational efficiency.

Table 3. Performance comparison of different early-stopping strategies across evaluation settings.

	R_{test}	$T_{avoided}$
BO	*0.000(±0.000)*	*0.000(±0.000)*
BOHB	1.196(±0.072)	**0.731(±0.003)**
HB	1.185(±0.039)	0.638(±0.017)
BO+DT	1.081(±0.147)	0.569(±0.023)
BO+PN+DT	**0.999(±0.128)**	0.569(±0.022)

Table 4 displays the grid search optimization for the structural hyperparameters of the WDCNN encoder across different embedding dimensions (E_{dim}), numbers of convolutional blocks (NB), and internal block sizes (BS). The empirical results indicate that deeper configurations ($NB = 2$) operating with a compact embedding space ($E_{dim} = 8$) achieve significant stability in training, minimizing R_{train} to 0.127 ± 0.995 under a block size of $BS = 3$. For generalization on the test set, both the ($E_{dim} = 8, NB = 2, BS = 3$) and ($E_{dim} = 16, NB = 1, BS = 2$) setups deliver the lowest selection rank ($R_{test} = 0.942$), showing an optimal balance between feature capacity and noise robustness in industrial environments.

Table 4. Grid search evaluation of the parameterized WDCNN encoder structural hyperparameters on training and testing selection ranks.

E_{dim}	NB	BS	R_{train}	R_{test}
8	1	1	0.630(±0.993)	0.946(±0.189)
		2	0.718(±0.966)	0.958(±0.178)
		3	0.707(±0.976)	1.085(±0.149)
	2	1	0.157(±0.989)	0.931(±0.163)
		2	0.152(±0.982)	1.069(±0.143)
		3	**0.127(±0.995)**	**0.942(±0.193)**
16	1	1	0.148(±0.983)	0.947(±0.124)
		2	**0.144(±0.985)**	0.953(±0.127)
		3	0.130(±0.993)	1.103(±0.113)
	2	1	0.721(±0.932)	1.073(±0.144)
		2	0.632(±0.857)	0.977(±0.181)
		3	0.573(±0.992)	1.128(±0.161)

The relative importance of the features utilized by the early-stopping decision tree is detailed in Table 5. The empirical metrics heavily dominate the splitting criteria, where the Exponential Moving Average (EMA) leads, followed closely by the epoch-specific baseline (BC_e). Together, these two features account for over 92% of the total model decision weight. While predictive uncertainty (PU) and expected improvement (EI) exhibit low average importance scores within

individual trees, reflecting that many optimized trees do not select them as primary splitting criteria, this does not imply that these features are dispensable at the system level.

Table 5. Feature importance scores derived from the optimized Decision Tree classifier for early-stopping decisions.

Feature	Importance
EMA	0.5056(±0.0589)
BC_e	0.4222(±0.0667)
VV	0.0479(±0.0088)
e'	0.0119(±0.0057)
VI	0.0111(±0.0087)
PU	0.0011(±0.0013)
EI	0.0002(±0.0004)

As the ablation study in Table 6 shows, their inclusion during training consistently improves the test selection rank, specially when both are available. This discrepancy suggests that PU and EI contribute selectively in boundary cases where empirical trend metrics alone are insufficient for a confident termination decision, a benefit that is not captured by average feature importance scores but is reflected in the overall generalization performance.

Table 6. Ablation study evaluating the impact of Prototypical Network metrics on test selection rank and avoided training time.

	R_{test}	$T_{avoided}$
W/PN	1.491(±0.072)	0.593(±0.002)
EI	1.474(±0.048)	0.589(±0.010)
PU	1.022(±0.133)	0.570(±0.021)
EI+PU	0.907(±0.147)	0.565(±0.022)

To further characterize the behaviour of the proposed framework, Table 7 disaggregates the Mean Selection Rank by task type and network architecture. Regarding task type, fault detection datasets yield a lower mean rank (0.894) than RUL datasets (1.132), suggesting that the early-stopping policy generalises more reliably on classification-oriented curves, whose convergence patterns are typically more regular and easier to extrapolate. The higher rank observed for RUL tasks may reflect the greater variability in degradation trajectories across different machinery and operating conditions, which poses a harder extrapolation problem for the Prototypical Network.

The breakdown by architecture reveals a more pronounced dispersion. Recurrent networks (RNN) achieve the lowest mean rank (0.354), followed by MSCNN (0.759) and FCN (1.195). Transformer-based architectures present the highest mean rank (1.578), which may be attributed to their slower and less monotonic convergence dynamics during the early training epochs, making early termination decisions less reliable. These differences suggest that the convergence behaviour of each architecture influences the difficulty of the early-stopping problem, and future work could explore termination based on architecture policies to further improve selection fidelity.

Table 7. Mean Selection Rank (R_{test}) disaggregated by task type and network architecture for the proposed BO+PN+DT framework.

		$\overline{R}_{test}$
Task	Fault Detection	0.894
	RUL	1.132
Architecture	RNN	**0.354**
	MSCNN	0.759
	FCN	1.195
	Transformer	1.578

6. CONCLUSIONS

This work presented a Green AI framework for resource-efficient Neural Architecture Search in Prognostics and Health Management, addressing the critical sustainability bottleneck introduced by the exhaustive evaluation of candidate configurations in AutoML pipelines. The proposed approach integrates three synergistic components: a Prototypical Network adapted for regression-based learning curve extrapolation, a dynamically updated Decision Tree classifier for adaptive early stopping, and a leaf-merging pruning strategy that preserves interpretability in safety-critical environments.

The experimental evaluation over 61,000 learning curves spanning 50 diverse PHM datasets validated the effectiveness of the framework across both fault diagnosis and remaining useful life estimation tasks. The integrated BO+PN+DT configuration achieved a mean test selection rank of 0.999 ± 0.128, demonstrating near-optimal model recovery while avoiding 56.9% of total training time. This result represents a SOTA result over resource-based baselines: while BOHB achieved a higher time reduction (73.1%), it did so at the cost of a degraded selection rank (1.196 ± 0.072), confirming that the proposed framework does not sacrifice diagnostic fidelity in exchange for computational savings.

The ablation study revealed that incorporating the uncertainty and and expected improvement metrics generated by the Prototypical Network, consistently improves generalization performance. Although these features exhibit low average importance within individual trees, their contribution at the system level is evidenced by the reduction in mean selection rank from 1.022 (PN without EI) to 0.907 (PN with both metrics).

The feature importance analysis further highlighted that the Exponential Moving Average (EMA) and the epoch-specific baseline (BC_e) collectively account for over 92% of the decision weight, confirming that robust, noise-resilient representations of the training trajectory are essential for reliable early stopping. The analysis by task type and architecture indicated that fault detection tasks and recurrent architectures benefit most from the proposed policy, while Transformer-

based models—characterized by slower and less monotonic convergence—present a harder extrapolation problem, suggesting a direction for future work.

From an energy perspective, the framework reduces GPU energy consumption by an estimated 56% relative to an exhaustive search baseline, yielding a net saving of approximately $117\,\mathrm{Wh}$ across the ten test datasets. These figures, derived under conservative assumptions with a single NVIDIA GTX 1080 Ti, demonstrate that high-performance industrial intelligence can be achieved without the prohibitive environmental costs typically associated with large-scale architecture optimization, directly fulfilling the dual objectives of Sustainability in PHM and Sustainability by PHM.

Future work will explore architecture-aware termination policies tailored to the distinct convergence dynamics of each model family, as well as the integration of energy-monitoring tools such as `CodeCarbon` for precise, hardware-level sustainability accounting. A particularly promising direction concerns the improvement of the Prototypical Network's predictive accuracy: since the ablation study demonstrated that its uncertainty metrics contribute meaningfully to selection fidelity, enhancing the extrapolation quality of the PN would naturally increase the influence of PU and EI as splitting features within the decision tree. Complementarily, the cost-sensitive optimization of the DT could be extended to explicitly reward tree configurations that actively exploit PN-derived features, for instance by incorporating a regularization term that penalizes solutions in which these features are absent from the splitting criteria. Extensions to multi-fidelity search spaces and online adaptation of the prototypical support set represent further promising directions for advancing lean, reliable AutoML in industrial prognostic systems.

DATA AVAILABILITY STATEMENT

The dataset used in this work are publicly available. To access the dataset, the tool PHMD (Solís-Martín et al., 2025) was used.

ACKNOWLEDGMENTS

Grant PID2023-147198NB-I00 funded by MICIU/AEI/10.13039/501100011033 (Agencia Estatal de Investigación) and by FEDER, UE.

REFERENCES

Adriaensen, S., Rakotoarison, H., Müller, S., & Hutter, F. (2024). Efficient bayesian learning curve extrapolation using prior-data fitted networks. *Advances in Neural Information Processing Systems*, *36*.

Baker, B., Gupta, O., Raskar, R., & Naik, N. (2018). *Accelerating neural architecture search using performance prediction.*

Borrego-Díaz, J., & Galán-Páez, J. (2022). Explainable artificial intelligence in data science. *Minds and Machines*, *32*(3), 485–531. doi: 10.1007/s11023-022-09603-z

Domhan, T., Springenberg, J. T., & Hutter, F. (2015). Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Proceedings of the 24th international conference on artificial intelligence* (p. 3460–3468). AAAI Press.

Egele, R., Mohr, F., Viering, T., & Balaprakash, P. (2024). The unreasonable effectiveness of early discarding after one epoch in neural network hyperparameter optimization. *Neurocomputing*, 127964.

Falkner, S., Klein, A., & Hutter, F. (2018). Bohb: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning* (pp. 1437–1446).

Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., & Xing, E. P. (2018). Neural architecture search with bayesian optimisation and optimal transport. *Advances in neural information processing systems*, *31*.

Klein, A., Falkner, S., Springenberg, J., & Hutter, F. (2017b). Learning curve prediction with bayesian neural networks. In *Proceedings of the international conference on learning representations.*

Klein, A., Falkner, S., Springenberg, J. T., & Hutter, F. (2017a). Learning curve prediction with bayesian neural networks. In *International conference on learning representations.*

Li, L., Jamieson, K. G., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. In *Iclr (poster)* (p. 53).

Mettes, P., Van der Pol, E., & Snoek, C. (2019). Hyperspherical prototype networks. *Advances in neural information processing systems*, *32*.

Real, E., Aggarwal, A., Huang, Y., & Le, Q. V. (2019). Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 4780–4789).

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, *30*.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, *25*.

Solis-Martin, D., Galan-Paez, J., & Borrego-Diaz, J. (2024). Bayesian model selection pruning in predictive maintenance. In *International conference on hybrid artificial intelligence systems* (pp. 263–274).

Solís-Martín, D., Galán-Páez, J., & Borrego-Díaz, J. (2025). A model for learning-curve estimation in efficient neural architecture search and its application in predictive health maintenance. *Mathematics*, *13*(4), 555.

Solís-Martín, D., Galán-Páez, J., & Borrego-Díaz, J. (2025). Phmd: An easy data access tool for prognosis and health management datasets. *SoftwareX*, *29*, 102039. doi: https://doi.org/10.1016/j.softx.2025.102039

Zhang, W., Peng, G., Li, C., Chen, Y., & Zhang, Z. (2017). A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, *17*(2), 425.

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 8697–8710).