Constraint-Guided Learning of Data-Driven Health Indicator Models: An Application on Bearings

Yonas Tefera[1, 2, 3, 4], Quinten Van Baelen[1, 2, 3], Maarten Meire[1, 2, 3], Stijn Luca[5] and Peter Karsmakers[1, 2, 3]

[1] *KU Leuven, Dept. of Computer Science, ADVISE-DTAI, Kleinhoefstraat 4, B-2440 Geel, Belgium*

[2] *Leuven.AI - KU Leuven institute for AI*

[3] *Flanders Make @ KU Leuven*
{*yonas.tefera, quinten.vanbaelen, maarten.meire, peter.karsmakers*}*@kuleuven.be*

[4] *Addis Ababa University, School of Electrical and Computer Engineering, Addis Ababa, Ethiopia*
yonas.yehualaeshet@aau.edu.et

[5] *Ghent University, Department of Data Analysis and Mathematical Modelling, Coupure Links 653, 9000 Gent, Belgium*
stijn.luca@ugent.be

ABSTRACT

This paper presents a constraint-guided deep learning (DL) framework to develop physically consistent health indicators (HIs) in bearing prognostics and health management. Conventional data-driven approaches often lack physical plausibility, while physics-based models are limited by incomplete knowledge of complex systems. To address this, we integrate domain knowledge into DL models via constraints, ensuring monotonicity, bounding output ranges between 1 and 0 (representing healthy to failed states, respectively), and maintaining consistency between signal energy trends and HI estimates. Using constraints eliminates the need for complex loss term balancing to incorporate domain knowledge. The constraint-guided gradient descent algorithm (CGGD) is used to train a DL model that satisfies specific constraints. Using time-frequency representations of accelerometer signals from the pronostia and XJTU-SY bearing datasets, the model learned using constraints generates more accurate and reliable representations of bearing health compared to conventional methods. It produces smoother degradation profiles that align with the expected physical behavior. Model performance is assessed using three metrics: trendability, robustness, and consistency. When compared to a conventional baseline model, the model learned using constraints shows a significant improvement in all three metrics. Another baseline incorporated the monotonicity behavior directly into the loss function using a soft-ranking approach. While this approach outperforms the model learned using constraints in trendability, due to its explicit monotonicity enforcement, the model learned with constraints performed better in robustness and consistency, providing stable and interpretable HI estimates over time. The ablation study confirms the importance of each constraint: the monotonicity constraint improves trendability, the boundary constraint ensures consistency, and the energy–HI consistency constraint enhances robustness. These findings demonstrate the effectiveness of CGGD in producing reliable and physically meaningful HIs for bearing prognostics and health management, offering a promising direction for future prognostic applications.

1. INTRODUCTION

Prognostics and health management (PHM) is a domain that focuses on the management of the health of engineering systems and their critical components. Bearings are critical components in many industrial machines and are often exposed to significant stress and wear (Cubillo, Perinpanayagam, & Esperon-Miguez, 2016). As such, effective PHM is essential to prevent unexpected failures, boost operational efficiency, and extend the service life of the machine. To achieve this, it is crucial to develop robust techniques for monitoring the health of the bearings. In industrial settings, a variety of diagnostic and prognostic methods are applied, often using open-source datasets (Su & Lee, 2023). These methods generally focus on two key objectives (Zhou et al., 2022): (i) detecting early-stage bearing faults to prevent catastrophic failures, and (ii) tracking and characterizing how the bearing's health

https://doi.org/10.36001/IJPHM.2025.v16i2.4268

condition evolves over time. This work focuses on the latter.

For this purpose, the construction of HIs is a crucial step in assessing and predicting the condition of assets (Biggio & Kastanis, 2020). The goal is to extract effective features from sensor signals to establish a stable HI that reflects the degradation of the asset over time. Preferably, this HI should have a fixed range, such as 1 to 0, where 1 represents a healthy condition and 0 indicates a faulty one. This allows for the definition of a threshold that can trigger an alarm if the HI falls below it, providing an early warning before the system progresses into critical failure.

The modeling approaches used in the construction of HIs for bearings are broadly categorized into two types: physics-based methods and data-driven methods (Li, Zhang, Li, & Si, 2024). Physics-based methods (Xu, Kohtz, Boakye, Gardoni, & Wang, 2023) involve analyzing and modeling the physical failure mechanisms of bearings using fundamental principles. When these mechanisms or relevant domain knowledge are well understood and the parameters of these physical models can be accurately estimated from measurement data, physics-based methods can provide precise, generalizable and interpretable HI values. However, in most cases, particularly for complex engineering systems, the physical failure mechanisms and domain knowledge may be incomplete or scarcely available. In this situation, the established physical models must make simplifications or assumptions, which leads to a poor representative models (Lei et al., 2016). Thus, as an alternative, data-driven methods can be applied (Jieyang et al., 2023), which are particularly promising for tackling complex processes that are not entirely understood or when physics-based methods are too computationally demanding. Data-driven methods have relatively high data collection requirements and do not require extensive consideration of the physical meaning of the data (Lu, Wang, Zhang, & Gu, 2024). Although data-driven models can achieve high accuracy in fitting observed data, they often lack physical realism or plausibility when interpolating or extrapolating beyond the available labeled data, leading to poor generalization. It is clear that both physics-based and data-driven methods have their unique strengths and weaknesses. As a result, a sensible strategy is to combine these two types of methods to take advantage of their respective merits through a hybrid approach (Liao & Köttig, 2014). Unlike the typical combination found in hybrid methods, embedding physical knowledge into data-driven models holds promise to guide models towards generating physically consistent results (Von Rueden et al., 2023). Accordingly, there is a need to train data-driven machine learning (ML) models by incorporating physical or domain knowledge (Karniadakis et al., 2021; Meng, Seo, Cao, Griesemer, & Liu, 2022) to improve the interpretability of the model. This domain knowledge often comes in the form of physical models, constraints, dependencies, and known valid ranges of features (Muralidhar, Islam, Marwah, Karpatne, & Ramakrishnan, 2018). There are various methods to incorporate specific physical or domain information into ML models, enabling the development of robust physics-informed ML systems. These methods include physics-informed data augmentation (Xiong, Fink, Zhou, & Ma, 2023), architecture design (Nascimento & Viana, 2019; Chen & Liu, 2021), residual modeling (Willard, Jia, Xu, Steinbach, & Kumar, 2022), and modifying the loss function to include a physics-inspired regularization term (J. Wang, Li, Zhao, & Gao, 2020; Raissi, Perdikaris, & Karniadakis, 2019). In the latter methods, the optimization procedure is adapted such that during learning models that are not consistent with the domain knowledge are penalized to guide learning towards a model that makes a trade-off between explaining the measurements and being consistent with domain knowledge. Adding domain knowledge to learning can be considered as a regularization method which can reduce the need for labeled data, shrink the search space during model optimization, and enhance the model's generalizability to unseen scenarios (Li et al., 2024). Using this strategy, the domain knowledge described by some formal representation, such as an inequality constraint, is transferred to the parameter values of a (black box) data-driven model. In our opinion, this strategy is one of the most convenient ways to embed knowledge in data-driven models; therefore, it is selected for designing HI models in this work.

In prior works, physics-inspired constraints were considered by adding an additional regularization term to the loss function. Such a term requires a proper trade-off hyperparameter, which requires careful balancing with the other terms in the objective function. Especially when more constraints are considered, which all require their own hyperparameter, such balancing can become cumbersome. In this work, a unified framework proposed by Van Baelen and Karsmakers (2023) used to include domain knowledge in a data-driven approach to estimate the HI for bearings. This framework does not require a tuning of the hyperparameters of the different regularization terms and can be easily extended with the inclusion of additional domain-knowledge inspired constraints. More specifically, in this work the following constraints are considered: monotonicity, bounded ranges from fully healthy to failure states, and a characteristic degradation pattern informed by energy trends. This approach allows HI estimates to remain within defined ranges and adhere to the actual observed degradation patterns of a bearing, without the need to rely on a predefined shape of the degradation function. The proposed method will be experimentally tested and compared with two baseline HI learning models.

The remainder of this paper is organized as follows. In Section 2, we present the proposed novel methodology that discusses the training approach, the proposed constraints, and the baseline methods used. Section 3 discusses the experimental setup starting with the general HI construction procedure, description of the datasets, the pre-processing proce-

dure, the model architecture used, the data set partitioning procedure and the evaluation metrics. In Section 4, we describe the implementation of our proposed approach and discuss the results. In addition, a comparison of the results with a conventional baseline analysis is implemented. An ablation study is presented in Section 5 to discuss performance changes by slightly modifying the proposed model. Finally, in Section 6, we conclude by summarizing the results of our experiments.

2. METHODOLOGY

The proposed method for constructing a bearing HI model leverages domain knowledge to guide the training process, resulting in a robust and accurate model. Incorporating domain knowledge during training is expected to reduce the reliance on labeled data. Unlike other approaches in the literature, domain knowledge in this work is not integrated by adding a regularization term to the loss function. Instead, the method employs the CGGD framework proposed in (Van Baelen & Karsmakers, 2023), which facilitates solving constrained non-linear optimization problems. Specifically, this framework is utilized to train a DL model by minimizing a loss function while ensuring that the model satisfies certain constraints. These constraints represent domain knowledge. In this way, domain knowledge can be embedded into the resulting DL model. First, the application of the CGGD framework to develop DL-based HI models is described. Next, a detailed explanation of the domain-knowledge inspired constraints are provided along with their implementation within the CGGD framework. Finally, the baseline methods used for comparison with the proposed approach are explained.

2.1. Constraint Guided Learning of a DL based HI Model

DL architectures commonly used in PHM (Rezaeianjouybari & Shang, 2020) include convolutional neural networks (CNN), autoencoders (AE), deep belief networks (DBN), recurrent neural networks (RNN) and generative adversarial networks (GAN). In systems where obtaining representative labeled data is challenging, the use of unsupervised learning methods, such as AE architectures, are particularly beneficial (Hoffmann Souza, da Costa, & de Oliveira Ramos, 2023). Although the proposed method is agnostic to the model architecture and learning procedure, in this work a convolutional autoencoder (CAE) model architecture learned by using a reconstruction loss is adopted as a starting point. As will be explained later in Section 3, acceleration signals are transformed into a time-frequency representation denoted as $X \in \mathbb{R}^{D\times T}$ to be used as an input to the model.

In its original formulation, the CAE model is trained to construct a compact latent representation $z \in \mathbb{R}^{D'}$ of input signals X with $D' \ll D$ using an encoder function $\mathcal{E}$. This process occurs without significant loss of information. Given z, a decoder $\mathcal{D}$ reconstructs the input to $\hat{X}$, aiming to closely resemble the original signal X. During learning, the model parameters are determined using the following objective (Vincent et al., 2010):

$$\min_{\boldsymbol{\theta}_\mathcal{E},\boldsymbol{\theta}_\mathcal{D}} \mathcal{L}_{\text{reconn}}(\boldsymbol{X},\boldsymbol{\theta}_\mathcal{E},\boldsymbol{\theta}_\mathcal{D}), \tag{1}$$

where $\boldsymbol{X}$ denotes the set of all input samples, $\mathcal{L}_{\text{reconn}}(\boldsymbol{X},\boldsymbol{\theta}_\mathcal{E},\boldsymbol{\theta}_\mathcal{D}) = \sum_{X\in\boldsymbol{X}} \|X - \mathcal{D}(\mathcal{E}(X))\|_2^2$ denotes the reconstruction loss, and $\boldsymbol{\theta}_\mathcal{E}$ and $\boldsymbol{\theta}_\mathcal{D}$ are the parameters of the encoder $\mathcal{E}$ and decoder $\mathcal{D}$ models, respectively. The model is trained using data segments from a healthy bearing in operation, resulting in a minimal reconstruction error for this healthy state. In contrast, in scenarios where the bearing is faulty, the reconstruction process becomes less accurate, leading to a larger reconstruction error. This increased error serves as an indicator of bearing health degradation. Once the CAE parameters are learned, the HI estimate for an input X is calculated using $f_{\text{HI}}^{\text{CAE}}(X) = -\|X - \mathcal{D}(\mathcal{E}(X))\|_2$ which computes the reconstruction error and will have a decreasing trend over time.

Using the CGGD framework, the optimization problem provided in Eq. (1) can be turned into a constrained optimization task as:

$$\begin{aligned} \min_{\boldsymbol{\theta}_\mathcal{E},\boldsymbol{\theta}_\mathcal{D}} \quad & \mathcal{L}_{\text{reconn}}(\boldsymbol{X},\boldsymbol{\theta}_\mathcal{E},\boldsymbol{\theta}_\mathcal{D}) \\ \text{s.t.} \quad & C_i(\boldsymbol{X},\boldsymbol{\theta}_\mathcal{E},\boldsymbol{\theta}_\mathcal{D}), \quad \text{for } i = 1,\ldots,M. \end{aligned} \tag{2}$$

where $C_i : \mathbb{R}^n \to \mathbb{R}$ is the i-th constraint, and M is used to denote the number of constraints incorporated in the model. To apply CGGD, a direction $\text{dir}_\mathbf{C}$ needs to be defined for each constraint individually. This direction, when used to update the model, will result in a local improvement with respect to the constraint considered. For example, the constraint $C : \mathbb{R}^2 \to \mathbb{R} : (x_1, x_2) \mapsto x_1 - x_2$ with $C(x_1, x_2) \leq 0$ can be given a possible direction by $\text{dir}_\mathbf{C} = (\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$. Consider, for example, an intermediate solution $(x_1, x_2) = (3, 1)$ which does not satisfy the constraint. Observe that when the gradient of a gradient descent-based optimizer is replaced by the direction $\text{dir}_\mathbf{C}$, $(x_1, x_2) = (3, 1) - \eta(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$, that the updated solution, for a large enough η, will satisfy the constraint as x_1 can be decreased and x_2 can be increased until they at least have an equal value.

To provide greater flexibility, this work employs a CGGD based CAE (CCAE), where the HI value is not derived from the reconstruction error, as in conventional CAEs, but is instead defined as a learnable function of the encoding $\mathcal{E}(X)$. This will be denoted as $f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X))$ with the learnable parameters $\boldsymbol{\theta}_{\text{HI}}$.

2.2. Constraints

In this section, the constraints applied in this work are described to ensure that the predicted bearing HI values accurately reflect their degradation over time. These constraints are designed to represent domain knowledge to ultimately improve the accuracy and robustness of the model even in situations where annotated data is scarce.

2.2.1. Monotonic Degradation Constraint

When a bearing is put into operation, it undergoes inevitable wear, which progressively worsens with time. This degradation should be reflected in the predicted HI, which is expected to decrease monotonically over time. To enforce this constraint, the HI estimates should be penalized if they deviate from the expected monotonic trend based on the time location of the corresponding input data samples. For inputs X_t measured at time step t and the corresponding health state predictions $f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_t))$, the monotonic degradation constraint can be expressed as follows:
For $t_1 < t_2$, hence X_{t_1} a measurement taken before X_{t_2} then it must hold that $f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_{t_1})) > f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_{t_2}))$. To enforce such a constraint with CGGD, a direction function needs to be defined, which is done by comparing the ranks of $f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_t))$ (from high to low value) estimates with their corresponding ranks in time (from early to later). This is calculated as follows:

$$\begin{aligned}\text{dir}_{\text{mono}}(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}) = \\ \text{rank}_{\text{desc}}(X_t, \boldsymbol{X}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}) - \text{rank}_{\text{asc}}(\text{time}(X_t), \boldsymbol{t}),\end{aligned} \quad (3)$$

where $\boldsymbol{t}$ is a set of timestamps aligned with the samples in set $\boldsymbol{X}$, X_t is an element from set $\boldsymbol{X}$, $\text{time}(X_t)$ a function that extract the time at which the sample X_t was measured, $\text{rank}_{\text{desc}}(\cdot)$ is a function that outputs the rank of $f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_t))$ when compared to all other HI estimates of the elements in $\boldsymbol{X}$ when ranked in descending order, $\text{rank}_{\text{asc}}(\cdot)$ is a function that outputs the rank of t among all values in $\boldsymbol{t}$ sorted in ascending order.

As time progresses, it is expected that the corresponding HI estimates decrease. A positive value for dir_{mono} implies that the HI estimate is ranked too high, suggesting that a decrease in the HI estimate is necessary. In contrast, a negative value implies a lower rank, indicating that the HI estimate should be increased. A value of zero means that the HI estimate matches its expected rank. A significant deviation in the ranking of a sample from the desired position will have a greater impact on the direction than a smaller deviation. By incorporating this monotonic degradation constraint into the HI estimation process, the model ensures that the predicted HI values decrease monotonically over time, aligning with the expected degradation pattern of the bearing.

2.2.2. Energy-HI Consistency Constraint

Building on the monotonic degradation constraint, we expect that while the HI values should decrease over time, the difference between the HI values of two consecutive samples should not vary significantly unless a substantial change in signal energy occurs. To enforce this concept, a constraint is introduced that ensures that if two samples have similar energy levels, their HI estimates should also be close in value.

We define this relationship mathematically as follows:

$$f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_{t_0})) - \alpha\Delta \leq f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_{t_i})) < f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_{t_0})), \quad (4)$$

where $\Delta = \max(\kappa, |E(X_{t_i}) - E(X_{t_0})|)$ represents the normalized total energy[1] difference between the two samples at times t_0 and t_i where $t_i > t_0$, $\alpha > 0$ is a hyperparameter that controls the sensitivity of this constraint and $0 < \kappa < 1$. κ is a parameter that introduces flexibility in HI predictions for samples with similar energy values. When the energy difference is minimal, κ allows a margin of variation in the predicted HI.

Based on the predicted HI of the model, the subsequent update direction of the energy-HI consistency constraint is determined as follows:

$$\begin{aligned}&\text{dir}_{\text{ene}}(X_t, X_{t_0}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}) = \\ &\begin{cases} 1, & f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_t)) > f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_{t_0})), \\ 0, & -\alpha\Delta \leq f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_t)) - f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_{t_0})) < 0, \\ -1, & \text{otherwise.} \end{cases}\end{aligned} \quad (5)$$

By penalizing discrepancies between the energy difference and the HI difference, we encourage the model to maintain a consistent relationship between these two variables. This approach helps prevent large fluctuations in the HI values and ensures that the model accurately reflects the gradual degradation process of the bearing.

2.2.3. HI Boundary Constraint

When predicting the HI of a bearing, it is convenient that the values remain within a normalized range: a fully healthy state is represented by a value of $ub = 1$, while a failure state corresponds to a value of $lb = 0$. To enforce this condition, boundary constraints are enforced during the training process to ensure that all HI predictions fall within this defined range.

In addition to the broader bound ranges, stricter bound ranges are also applied during certain phases of the bearing's life cycle. In the initial $a\%$ (e.g. $a = 10$) of its operation, the HI value is expected to be at least $b_a < ub$ (e.g. $b_a = 0.9$), ensuring that the new bearings operate in near-optimal health. In contrast, in the final $b\%$ (e.g. $b = 5$) of its operation,

[1]The energy $E(\cdot)$ is calculated by summing all squared elements in X.

the HI is expected not to exceed $b_b > lb$ (e.g. $b_b = 0.05$), indicating that the bearing is close to failure due to significant degradation.

Based on the predicted HI, the subsequent direction of the upper and lower boundary constraints is determined as follows:

$$\mathrm{dir}_{\text{upper}}(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}) = \begin{cases} 1, & \text{if } f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_t)) > ub, \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathrm{dir}_{\text{lower}}(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}) = \begin{cases} -1, & \text{if } f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X_t)) < lb, \\ 0, & \text{otherwise,} \end{cases}$$

where lb and ub are the lower and upper bounds, respectively, within which the HI predictions should lie.

When $\mathrm{dir}_{\text{upper}}$ is 1, it means that the HI estimate is greater than ub, which requires a reduction in subsequent updates. When $\mathrm{dir}_{\text{lower}}$ is -1, the HI estimate is below lb and should be increased. A zero in both $\mathrm{dir}_{\text{upper}}$ and $\mathrm{dir}_{\text{lower}}$ indicates that the HI estimate is within the specified bounds and that no adjustments are necessary. By implementing these boundary constraints, it is ensured that the HI predictions accurately reflect the bearing condition, from optimal functionality to failure.

2.3. Implementing the Constraints in the CGGD Framework

This section provides a comprehensive explanation on how the CGGD framework can be used to train the HI estimator model, denoted as $f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(\cdot))$. As is indirectly indicated in Eq, (2), a multi-head network model is used. One head decodes the encoded input to calculate the reconstruction loss (to calculate $\mathcal{L}_{\text{reconn}}$) and another head is used for the HI prediction ($f_{\text{HI}}^{\text{CCAE}}$) based on the encoded input. Both will use the same encoder, and the input of each head is thus given by $\mathcal{E}(X)$.

At the core of the CGGD optimization procedure, the update of the model parameters is defined in Eq. (6). Here θ_j is the value of a learnable weight on iteration j and θ_{j+1} the value of the same learnable weight in the next iteration, R_{mono}, R_{ene}, R_{upper}, and R_{lower} are the rescale factors, which control the relative weight of the different constraints compared to the loss function, for the monotonic degradation constraint, the energy-HI consistency constraint, the upper bound on the HI, and the lower bound on the HI, respectively, η is the learning rate, F_{MH} is a function that balances the gradients of the different heads appropriately, $\nabla_{\mathcal{E}}$ is the gradient with respect to the latent space determined by the encoder $\mathcal{E}$, and $0 < \epsilon < 1$ to ensure there is a minimum step size in case the gradients from the reconstruction loss to the encodings are very small. A small adaptation to standard CGGD can be observed as the variable F_{MH} is included. This is required as $\boldsymbol{\theta}_{\text{HI}}$ are only trained using the constraints, while $\boldsymbol{\theta}_{\mathcal{D}}$ are only trained with a loss function. Therefore, the first shared space by both objectives is the encoding space. In this space, the constraints should be prioritized over the reconstruction loss function, as the final model should satisfy the constraints on the training data. The constraints and the loss function will determine an update vector for the encoding space. In order to make sure that the update vector linked to the constraints dominates the model updating both update vectors are set to the same norm. This is accomplished by first setting the update vector created by the constraints to have a unit norm, after which it is multiplied by the norm of the update vector that is calculated based on the loss function. To have a unit norm for the constraints-based update vector, first the gradient of the direction function (dir) is multiplied by the automatic differentiation of the predicted HI ($f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X))$) for each encoding space dimension. Then, to let it have unit norm this update vector is multiplied by the rescale factor of the constraint and F_{MH} which is defined as:

$$F_{\text{MH}}\left(X_t, \mathrm{dir}(X_t, \cdot)\right) := \frac{\|\mathrm{dir}(X_t, \cdot)\|}{\left\|\nabla_{\mathcal{E}}\left[\mathrm{dir}(X_t, \cdot) f_{\text{HI}}^{\text{CCAE}}\left(\mathcal{E}\left(X_t\right)\right)\right]\right\|}.$$

After being rescaled with F_{MH}, the constraint update vector is multiplied by $\|\nabla_{\mathcal{E}} \mathcal{L}_{\text{reconn}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathcal{D}}\right)\|$, as can be seen in Eq. (6), to let both the constraint and loss update vector have the same norm.

To train the neural network, an off-the-shelve Adam optimizer is used by applying Eq. (6) iteratively for all learnable parameters. Observe that the partial derivatives of the loss function with respect to $\boldsymbol{\theta}_{\text{HI}}$ are 0 as they are independent from each other and the partial derivatives of the encodings with respect to $\boldsymbol{\theta}_{\mathcal{D}}$ are 0 as well as the weights of the decoder are independent from the encodings.

All rescale factors are assumed to be strictly greater than 1, as they represent the relative weight of the constraints compared to the loss function by the definition of CGGD. Different values can be assigned to individual rescale factors to prioritize certain constraints over others. Generally, the constraint with the largest rescale factor takes precedence over all other constraints and the loss function.

In this work, the rescale factor for the monotonic degradation constraint is dynamically adjusted for each individual prediction based on its deviation from the ground truth, within a bounded interval defined by $[R_{\text{mono\_lw}}, R_{\text{mono\_up}}]$. This interval ensures higher rescale factors for larger deviations between the predicted and true ranks and comparatively lower factors for smaller deviations. For a monotonicity direction ($\mathrm{dir}_{\text{mono}}(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})$) calculated for sample X_t, the monotonic rescale factor is calculated using Eq. (7). Here, batch_size-1 represents an upper bound on the absolute value of the values in $\mathrm{dir}_{\text{mono}}$. In particular, this upper bound can be

$$\theta_{j+1} := \theta_j - \eta \left(\frac{\partial \mathcal{L}_{\text{reconn}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathcal{D}}\right)}{\partial \theta_j} + \max\left\{ \left\| \nabla_{\mathcal{E}} \mathcal{L}_{\text{reconn}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathcal{D}}\right) \right\|, \epsilon \right\} \frac{\partial f_{\text{HI}}^{\text{CCAE}}\left(\mathcal{E}\left(X_t\right)\right)}{\partial \theta_j} \right.$$
$$\Big[R_{\text{mono}} \operatorname{dir}_{\text{mono}}\left(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}\right) F_{\text{MH}}\left(X_t, \operatorname{dir}_{\text{mono}}(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})\right)$$
$$+ R_{\text{ene}} \operatorname{dir}_{\text{ene}}\left(X_t, X_{t_0}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}\right) F_{\text{MH}}\left(X_t, \operatorname{dir}_{\text{ene}}(X_t, X_{t_0}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})\right)$$
$$+ R_{\text{upper}} \operatorname{dir}_{\text{upper}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}\right) F_{\text{MH}}\left(X_t, \operatorname{dir}_{\text{upper}}(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})\right)$$
$$\left. + R_{\text{lower}} \operatorname{dir}_{\text{lower}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}\right) F_{\text{MH}}\left(X_t, \operatorname{dir}_{\text{lower}}(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})\right) \Big] \right) \tag{6}$$

attained when all samples are from the same run and the earliest sample is predicted as the last sample or the other way around.

$$R_{\text{mono}_t} = R_{\text{mono\_lw}} + \frac{\left(R_{\text{mono\_up}} - R_{\text{mono\_lw}}\right) \cdot \operatorname{dir}_{\text{mono}}(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})}{\text{batch\_size} - 1} \tag{7}$$

In terms of computational complexity, unlike a standard CAE that only computes the reconstruction loss after the forward pass, the proposed method incurs additional computational and memory overhead from enforcing the added constraints during training, as formulated in Eq. (6). These constraints are evaluated alongside the usual forward and backward propagation, and the extent of the overhead depends on their complexity. For example, enforcing the monotonicity constraint requires two sorting operations of lists with size equal to the batch size, each with a complexity of $\mathcal{O}(\text{batch size} \cdot \log(\text{batch size}))$, followed by a linear comparison, adding $\mathcal{O}(\text{batch size})$.

When applying CGGD, the added cost includes computing the gradient of the constraint with respect to the latent space, backpropagating the constraint loss, calculating the norm of these gradients and performing a linear number of multiplications and additions before standard network backpropagation. Overall, the additional complexity is proportional to the type and number of constraints used in the model.

2.4. Comparison Baselines

To evaluate the effectiveness of the proposed CGGD-based approach, it was compared to two baseline methods. The first baseline is a standard CAE method in which the encoder and decoder parameters are learned according to Eq. (1). Hence, there is no additional head $f_{\text{HI}}^{\text{CCAE}}$ and there are no added constraints.

The second baseline incorporates the monotonic degradation property using a regularization term in the loss function of the model, unlike the proposed CCAE method, which applies it as a constraint during learning. However, if conventional ranking operations are used for the monotonic degradation loss function, they will create discrete, piecewise-constant outputs, like integer ranks, that are not differentiable. This poses challenges for gradient backpropagation in DL due to null or undefined derivatives. To overcome these issues, the method proposed by Blondel, Teboul, Berthet, and Djolonga (2020) was used.

The method casts the ranking as a projection onto the permutahedron (the convex hull of all permutation vectors). As a result, it creates projection operators that are differentiable, making them suitable for formal analysis. For a given input vector ($\boldsymbol{x}$), the soft rank $r_\varepsilon(\boldsymbol{x})$ is computed as:

$$r_\varepsilon(\boldsymbol{x}) = Proj_{\mathcal{P}}(-\boldsymbol{x}/\varepsilon)$$

where $\mathcal{P}$ is the permutahedron and $\varepsilon > 0$ is a regularization strength that controls approximation smoothness.

For the predicted HIs ($f_{\text{HI}}(\boldsymbol{X})$) of the set $\boldsymbol{X}$, the soft-rank loss is calculated as:

$$\mathcal{L}_{\text{soft-rank}} = \frac{1}{2} \| r_\varepsilon(\text{rank}_{\text{asc}}(\boldsymbol{t})) - r_\varepsilon(f_{\text{HI}}(\boldsymbol{X})) \|_2^2, \tag{8}$$

where $\boldsymbol{t}$ is a set of timestamps aligned with the samples in the set $\boldsymbol{X}$, $\text{rank}_{\text{asc}}(\boldsymbol{t})$ outputs the rank of $\boldsymbol{t}$ sorted in ascending order which represents the true ranks and $r_\varepsilon(\cdot)$ provides the soft-ranks for the input vectors.

Then, the total loss function of the model is calculated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reconn}} + \lambda \mathcal{L}_{\text{soft-rank}}, \tag{9}$$

where λ is a trade-off hyperparameter that balances the reconstruction and the soft-rank term.

This second baseline, which incorporates monotonicity as a regularization term in the loss function using the soft-rank approach, is called the soft-rank loss function based CAE (SR-CAE) method.

3. EXPERIMENTAL SETUP: BEARING HI ESTIMATION

This section discusses the datasets, the preprocessing and partitioning approach, the model architecture, the setting of the hyperparameters, and the relevant evaluation metrics used in this work.

3.1. HI Construction Procedure

The framework for building a predictive model involves several steps, starting with collecting data from monitoring signals, such as vibration, to reflect the health of the equipment. This data is then pre-processed to enhance its quality for analysis. An appropriate model is then chosen based on the characteristics of the data and the prognostic needs, with a specific architecture designed to detect bearing degradation patterns. Finally, the model estimates the HIs of bearing health, providing insight into degradation over time.

3.2. Bearing Degradation Datasets

To evaluate the effectiveness of the proposed approach, we conducted experiments using two benchmark bearing degradation datasets: the pronostia and the XJTU-SY datasets. Accelerometer measurements from the benchmarks that capture the progression of bearing degradation were used as input to estimate the HI. A brief overview of each dataset is provided below.

3.2.1. Pronostia Bearing Dataset

One of the datasets used is the IEEE PHM 2012 Prognostic challenge dataset, collected from the pronostia platform (Patrick Nectoux, Ramasso, & Brigitte Chebel-Morello, 2012). The bearings on this platform are tested under various loads and rotational speeds, comprising 17 run-to-failure datasets of rolling element bearings.

Table 1 provides an overview of the operating conditions along with the total number of bearing runs available for each condition. To perform accelerated degradation tests in a few hours, a high-level radial force was applied that exceeded the maximum dynamic load of the bearings. During these tests, the rotating speed of each bearing was maintained at a stable level. Two accelerometers and a thermocouple were employed to capture vibration signals and bearing temperatures. Vibration signals were recorded using two accelerometers positioned along the vertical and horizontal axes, with a sampling frequency of $25.6\,\text{kHz}$ and 2560 samples (that is, $1/10\,\text{s}$) collected every 10 seconds.

Since the bearings were subjected to natural degradation, we expect that the degradation patterns will vary between samples. Furthermore, little is known about the specific nature and origin of degradation (for example, whether it involves balls, inner or outer races, or cages), necessitating the application of data-driven techniques. Failures in any component, ball, rings, or cage could occur simultaneously. The useful life of a bearing is considered to end when the amplitude of the vibration signal exceeds $20\,\text{g}$. In this work, only accelerometer data from the provided data set were used as input to estimate the HI.

Table 1. Summary of the pronostia dataset.

Conditions	Load (N)	Speed (rpm)	Total number of Bearings
1	4000	1800	7
2	4200	1650	7
3	4500	1300	3

3.2.2. XJTU-SY Bearing Dataset

The second dataset used in this work is the XJTU-SY bearing dataset, provided by the Institute of Design Science and Basic Components at Xi'an Jiaotong University (XJTU) (B. Wang, Lei, Li, & Li, 2018). This dataset contains run-to-failure data for 15 rolling element bearings, tested under various loads and rotational speeds.

Table 2 provides an overview of the operating conditions along with the total number of bearing runs available for each condition. The testbed was designed to perform accelerated degradation tests under different radial forces and rotational speeds. The radial load was applied to the bearing housing using a hydraulic loading system.

Vibration signals were collected using two accelerometers mounted in the vertical and horizontal directions. Data were sampled at a frequency of $25.6\,\text{kHz}$, with 32768 samples (equivalent to $1.28\,\text{s}$ of data) recorded every minute. The dataset also specifies the cause of each bearing failure, including various types of fault such as inner race wear, cage fracture, outer race wear, and outer race fracture.

Table 2. Summary of the XJTU-SY dataset.

Conditions	Load (N)	Speed (rpm)	Total number of Bearings
1	10000	2400	5
2	11000	2250	5
3	12000	2100	5

3.3. Preprocessing

Various vibration analysis techniques can be applied to preprocess the raw signals from an accelerometer (Vishwakarma, Purohit, Harshlata, & Rajput, 2017). In this work a similar approach to Meire, Brijder, Dekkers, and Karsmakers (2022) was used, where rich features were calculated that do not include prior knowledge about the bearings. The raw acceleration signals are first converted into log-mel spectrograms before being fed into the machine learning model. For the pronostia dataset, 0.1-second recordings (2560 samples) are captured every 10 seconds. For the XJTU-SY dataset, 1.28 seconds of recordings (32768 samples) are captured every minute. In both cases, we use matching window and hop sizes

equal to the recording length (i.e. 0.1 s for pronostia, 1.28 s for XJTU-SY). From the vertical and horizontal accelerometer axes, we extracted 128 mel frequency bands, applied logarithmic scaling, and stacked the two resulting spectrograms. This yields a $(128, 2)$ 2D input frame for each time window.

A sample mel spectrogram extracted from the pronostia dataset, corresponding to the first bearing operating under condition 3, is shown in Figure 1. At the start of each bearing run, their is a transient behavior likely caused by the initial start-up of the setup. To eliminate this effect, we visually inspect and remove a small section of samples at the beginning of each bearing operation.

Finally, we normalize the data from each operating condition to achieve zero mean and unit variance across the mel frequency bands, treating each individual axis separately. This normalization is based on the mean and standard deviation calculated from the training set. This normalization helps ensure consistency between different runs under the same conditions.

By transforming raw vibration signals into log-mel spectrograms and applying the appropriate preprocessing steps, we obtain a consistent set of input features to train machine learning models for bearing HI estimation.

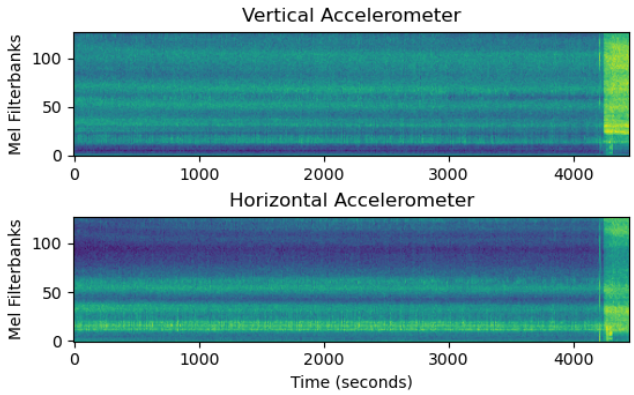

Figure 1. The two axes mel spectral features of a bearing.

3.4. Model Architecture

The standard CAE architecture used in this study, illustrated in Figure 2, consists of an encoder with four convolutional layers containing 64, 32, 32, and 16 filters, respectively, followed by a fully connected layer with 16 neurons. The decoder consists of a fully connected layer and 4 deconvolutional layers with 16, 32, 32, and 64 filters, ending with a final deconvolutional layer with two filters, corresponding to the two-axis accelerometer data as the output layer. A batch normalization layer follows each convolutional layer, except the final one. All convolutional layers employ ReLu activation functions and the dense layers use linear activation functions. The convolutional filters used are 1D with a kernel size of 3 and move with a stride of 2. In CCAE a second head is added on top of the encoder output (upper head in Figure 2) to estimate the HI. This second head is composed of three fully connected layers that have 16, 8, and 4 neurons, and the final layer providing a single HI estimate. The hyperparameters used in this model include the Adam optimizer with a learning rate of 1×10^{-3}, an early stopping criterion with a patience of 10 epochs, a batch size of 64. To ensure a standardized comparison of the methodologies, neither regularization nor dropout techniques are implemented in the architecture used.

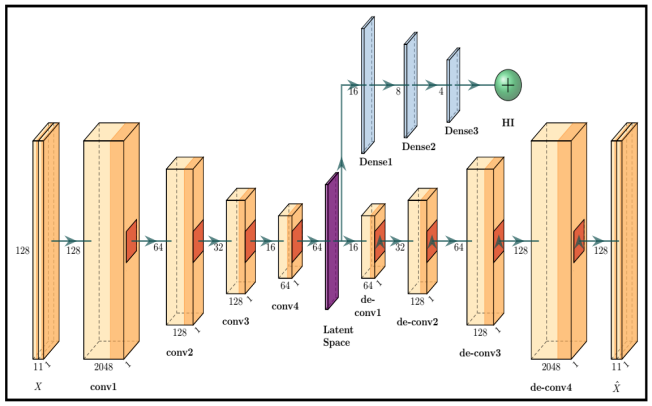

Figure 2. The CCAE Model Architecture.

The final architecture and hyperparameters are selected after conducting multiple experiments to fine-tune them. This is done through a systematic evaluation of various configurations based on reconstruction error and validation accuracy.

3.5. Dataset Partitioning

After extracting the log-mel spectral features, training batches are constructed in a structured and balanced manner. Each run from the training set contributes to the overall batch, with the number of samples drawn from each run proportional to its total number of available samples. This ensures that all runs are fairly represented in the training process. To avoid sampling bias, where batches may contain data from only specific segments of a run, samples are drawn from across the entire run, from start to end. To ensure greater variability and temporal coverage, each run is divided into three segments, and samples are drawn from each segment. These segments are:

1. Healthy state: The first segment comprises the initial 10% of the run, assumed to reflect a fully healthy state.
2. Slight degradation phase: The second segment includes samples from 10% to 95% of the run, representing a phase of slight degradation.
3. Sharp degradation phase: The final segment, expected to begin roughly after 95% of the samples in a run, characterizes rapid degradation leading up to apparent failure.

Batches are constructed by randomly sampling from each of the three segments of the training runs. To ensure representation across different degradation stages, each batch consists of approximately 20% samples from the healthy state, 70% from the gradual degradation phase, and 10% from the sharp degradation phase. This balanced composition improves the model's ability to learn across all stages of degradation. To evaluate performance and ensure robustness, each experiment is repeated 10 times with different random seeds for batch sampling and model initialization. In each run, data from the training bearings are combined and shuffled, with 75% used for training and 25% reserved for validation.

For the pronostia dataset, the first two bearing runs under each operating condition are used for training and the remaining runs are reserved for testing as shown in Table 3, in accordance with the original PHM 2012 challenge protocol (Patrick Nectoux et al., 2012).

Table 3. Train-test bearing split for each operating condition in the pronostia dataset.

Condition	Training bearings	Test bearings
1	Bearing1_1, Bearing1_2	Bearing1_3, Bearing1_4 Bearing1_5, Bearing1_6 Bearing1_7
2	Bearing2_1, Bearing2_2	Bearing2_3, Bearing2_4 Bearing2_5, Bearing2_6 Bearing2_7
3	Bearing3_1, Bearing3_2	Bearing3_3

The training set for each condition is relatively small due to the limited number of runs available. Furthermore, fault patterns and bearing lifetimes can vary significantly, even under identical conditions, complicating the HI estimation. This variability is illustrated in Figure 3, which shows the degradation processes of seven bearings that operate under a similar condition. Although the energy of these bearings is expected to follow a generally increasing trend, the progression towards failure varies considerably among them. Notably, certain bearings, such as Bearing1_2 and Bearing1_3 (denoted as B2 and B3 respectively in Figure 3), exhibit large energy fluctuations rather than a smooth transition.

For the XJTU-SY dataset, three bearings with different types of failures (e.g., inner race, outer race, and cage faults) are used for training, while two are reserved for testing under each operating condition, as shown in Table 4. Similarly to the pronostic dataset, failure times vary significantly, even among bearings operating under similar conditions. In addition, some bearings show considerable fluctuations in their degradation signals, which underscores the difficulty in learning a consistent and robust HIs across diverse and varied failure behaviors.

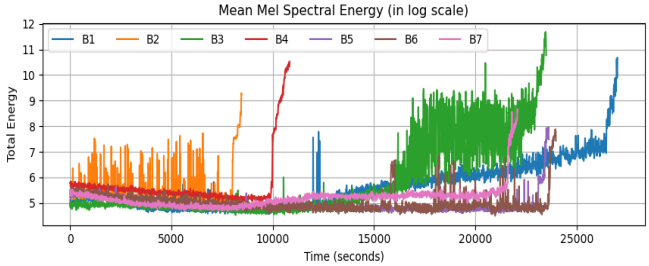

Figure 3. Mean mel energy progression over time for pronostia bearings in operational condition 1.

Table 4. Train-test bearing split for each operating condition in the XJTU-SY dataset.

Condition	Training bearings	Test bearings
1	Bearing1_2, Bearing1_4	Bearing1_1, Bearing1_3 Bearing1_5
2	Bearing2_1, Bearing2_5	Bearing2_2, Bearing2_3 Bearing2_4
3	Bearing3_2, Bearing3_4	Bearing3_1, Bearing3_3 Bearing3_5

3.6. HI Evaluation Metrics

The effectiveness of the constructed HI estimates are assessed using three key metrics: trendability, robustness, and consistency (Lei et al., 2018).

3.6.1. Trendability

As operating time increases, components are expected to gradually degrade. Consequently, the degradation trend of an HI should correlate with the operating time. The trendability metric is used as a quantitative measure to evaluate how well HI reflects changes in the condition of a machine over time. To evaluate this correlation in nonlinear degradation trends, we use the Spearman coefficient as the trendability metric, defined as follows (Carino, Zurita, Delgado, Ortega, & Romero-Troncoso, 2015):

$$\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t}) = 1 - \frac{6\sum_{i=1}^{N}(\mathrm{rank}(f_{\mathrm{HI}}(X_i)) - \mathrm{rank}(t_i))^2}{N(N^2-1)}, \tag{10}$$

where N is the number of samples, $\mathrm{rank}(f_{\mathrm{HI}}(X_i))$ and $\mathrm{rank}(t_i)$ are the ranks of the estimated HI values and the corresponding time value, respectively. The value of $\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t})$ ranges from -1 to 1, approaching either end when there is a strong positive or negative correlation between HI and time. $\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t}) = 1$ means the health indicator is perfectly increasing with time (positive trend). $\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t}) = -1$ means the health indicator is perfectly decreasing with time (negative trend) and $\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t}) = 0$ indicates that there is no trend. The aim of the constraint is to achieve a value of -1.

3.6.2. Robustness

A suitable HI should also be robust against inherent interference while maintaining a smooth degradation trend. This property is quantified by the robustness metric (Zhang, Zhang, & Xu, 2016), defined as:

$$\text{Rob}(f_{\text{HI}}(\boldsymbol{X}), \boldsymbol{t}) = \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\left|\frac{f_{\text{HI}}(X_i) - f^s_{\text{HI}}(t_i, \boldsymbol{X}, \boldsymbol{t})}{f_{\text{HI}}(X_i)}\right|\right), \tag{11}$$

where $f^s_{\text{HI}}(t_i, \boldsymbol{X}, \boldsymbol{t})$ is a smoothed HI value at t_i. The smoothing is performed in this work using locally weighted regression (LOESS) (Cleveland & Devlin, 1988; Duong et al., 2018) where each smoothed value is determined using neighboring data points within a specified range.

3.6.3. Consistency

Consistency refers to the degree of correlation among multiple HIs. When examining different HI estimates from a single unit, it is expected that they will exhibit some level of correlation because they all reflect the same degradation process. This metric is especially useful in situations where multiple predictions are made, as it enables the assessment of the consistency between these predictions.
Mosallam, Medjaher, and Zerhouni (2016) proposed a consistency metric based on the pairwise symmetrical uncertainty, defined as:

$$\text{Con}(f_{\text{HI}}(\boldsymbol{X_1}), f_{\text{HI}}(\boldsymbol{X_2})) = \frac{2I(f_{\text{HI}}(\boldsymbol{X_1}), f_{\text{HI}}(\boldsymbol{X_2}))}{\text{H}(f_{\text{HI}}(\boldsymbol{X_1})) + \text{H}(f_{\text{HI}}(\boldsymbol{X_2}))}, \tag{12}$$

Where, $I(f_{\text{HI}}(\boldsymbol{X_1}), f_{\text{HI}}(\boldsymbol{X_2}))$ represents the mutual information and $\text{H}(f_{\text{HI}}(\boldsymbol{X_1}))$ and $\text{H}(f_{\text{HI}}(\boldsymbol{X_2}))$ denote the entropies of $f_{\text{HI}}(\boldsymbol{X_1})$ and $f_{\text{HI}}(\boldsymbol{X_2})$, respectively. The output value, $\text{Con}(f_{\text{HI}}(\boldsymbol{X_1}), f_{\text{HI}}(\boldsymbol{X_2}))$, is normalized to a range between 0 and 1. A higher value indicates a greater similarity between the two HIs, suggesting a stronger consistency. For multiple HI estimates, pairwise consistency values are computed and the final consistency metric is determined by taking the mean and standard deviation of these pairwise results.

4. EVALUATION OF HI ESTIMATION METHODS: EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the experimental results, starting with a comparative analysis of the proposed CCAE method against two baselines: the standard CAE and the SR-CAE methods. This is followed by a detailed discussion of all methods based on the performance metrics described in Section 3.6. When comparing results, if the models show the same mean performance metric, the one with a smaller standard deviation is considered better.

4.1. General Comparative Analysis

The overall performance of each approach is visually assessed by analyzing the results of the pronostia bearings in condition 3 across the three HI estimation methods. This analysis provides a general overview that provides insight into the effectiveness of each technique in estimating bearing health degradation. In all HI estimation plots presented in this work, the smoothing (red line) is done using LOESS.

Figure 4 illustrates the degradation patterns using the standard CAE HI estimation method. Here, HI estimates might not decrease consistently, even in the training data, highlighting challenges related to the generalizability of this approach. Furthermore, the reconstruction error varies considerably across different bearings, despite operating under similar conditions. This results in varying HI values, making it difficult to establish a reliable correlation between these values and the actual state of bearing health.

Figure 5 presents the HI estimates using the SR-CAE method. This is an alternative approach to enforce monotonicity by regularizing the loss function. In this method, the influences of both the reconstruction and the soft-rank losses on the gradients of the CAE architecture are equally weighted, with the hyperparameter λ set to 1 in Eq. (9). Although this method demonstrates much improved monotonic degradation behavior, there remains considerable variation in HI values between different bearings. Notably, even in the training sets, substantial differences are observed at both the beginning of the operation and near the failure points.

In the CCAE method, different rescale factors are applied to the constraints: [1.25, 1.5] for the monotonicity constraint, 1.5 for the energy-HI consistency constraint and 2.0 for both the upper and lower HI boundary constraints. Prioritizing boundary constraints ensures that the predicted HI values remain within the interval [0, 1]. Lowering the priority on boundary constraints increases the chances that the predictions fall outside of this range. The HI estimates based on CCAE provide several advantages over the baseline methods, as shown in Figure 6. It produces a smoother degradation profile, indicating a more consistent decline in bearing health over time. The HI values range from 1 (indicating a fully healthy state) to 0 (indicating complete failure), which aligns well with the expected physical degradation process. In addition, the degradation curves correspond to typical bearing degradation, exhibiting a steady decline during the initial phase followed by rapid deterioration as the bearing approaches failure. As a result, the CCAE approach provides a more accurate and reliable representation of bearing health compared to the baseline methods.

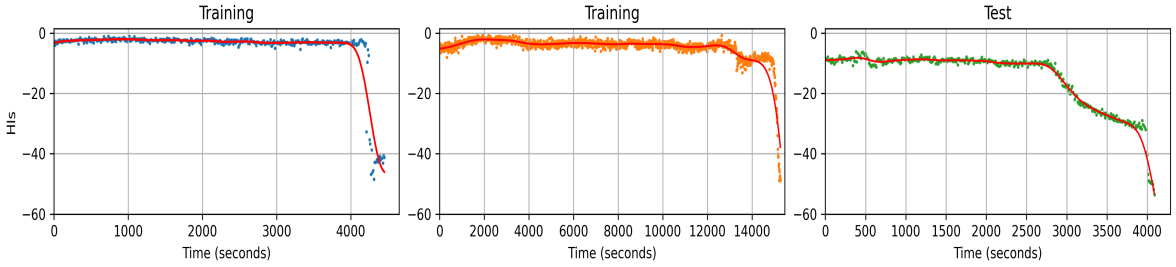

Figure 4. Standard CAE based HI estimates for condition 3 pronostia bearings.

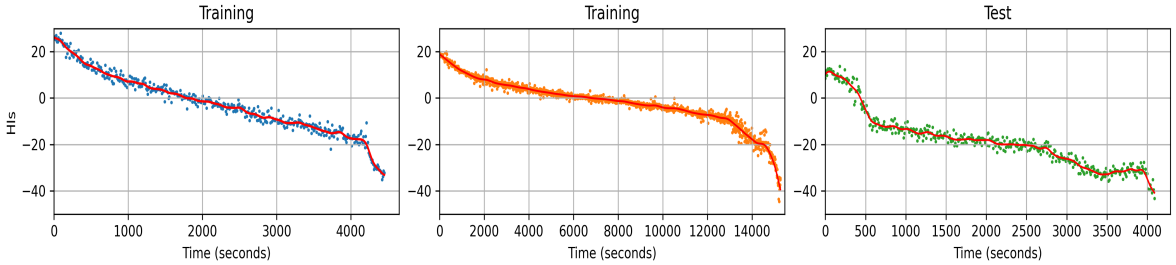

Figure 5. SR-CAE based HI estimates for condition 3 pronostia bearings.

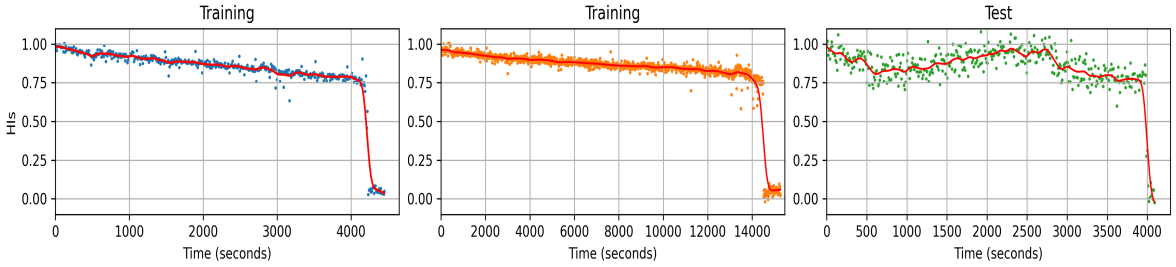

Figure 6. CCAE based HI estimates for condition 3 pronostia bearings.

4.2. Comparison of Standard CAE and CCAE Methods

The standard CAE model is trained using the initial 10% of the training bearing data, which roughly represents the healthy phase of the bearing operation. During this phase, the model reconstruction error is close to zero. However, as the bearing operation time increases, the reconstruction error increases, indicating a degradation in bearing health and a progression towards failure.

Table 5 presents the performance comparison between the standard CAE and the proposed CCAE method across all Pronostia training and test bearings. The CCAE method shows significant improvements, over 76% in trendability and consistency, and nearly 65% in robustness, when averaged across all bearings. These results indicate that CCAE more effectively captures the underlying degradation trends, providing robust, consistent, and repeatable HI estimates over time. In some cases, while the standard CAE method might yield superior results, these observations may not fully capture the overall performance of the method. Specifically, for Bearing_2_3, the trendability metric value for the standard CAE method is -0.732, which is significantly lower than the -0.092 obtained for CCAE. Although this value suggests a better decreasing correlation of HI with time, the HI value actually increases as the bearing approaches failure, contradicting the expected full-run decreasing trend, as depicted in Figure 7. In contrast, the CCAE method, despite having a lower trendability metric, accurately reflects the expected decrease in HI values as the bearing approaches failure, as shown in Figure 8. In addition, for Bearing_2_3 the HI value for the healthiest condition is close to -50, approximately 2500 seconds into its operation, and at the end. In contrast, Bearing_2_5 approaches an HI value of -50 near failure. This inconsistency, where one bearing's healthiest HI value is close to another bearing's failure point, underscores the challenge of obtaining a generally representative HI estimates using the standard CAE method.

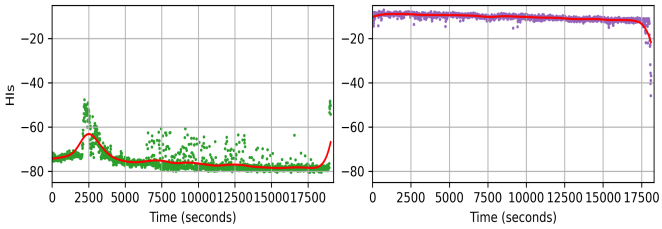

Figure 7. Standard CAE based HI estimates for Bearing_2_3 and Bearing_2_5.

Further insights into performance discrepancies are provided by examining bearings from condition 1 in figures 9 and 10. Figure 3 shows that Bearing_1_3 experiences significant energy fluctuations after 16,000 seconds. These fluctuations are reflected in the HI estimates of both methods. However, in the case of standard CAE, the HI values increase as the failure approach, contrary to expectations of a decreasing trend. However, CCAE minimizes these fluctuations and maintains HI values that align with the expected degradation trajectory. For Bearing_1_7, the trendability metric for the standard CAE HI estimation approach achieves -0.610, outperforming the CCAE score of 0.339. Despite this higher trendability score for CCAE, a notable advantage is that its HI values are bounded within a range of 0 to 1, providing a more interpretable measure of health status.

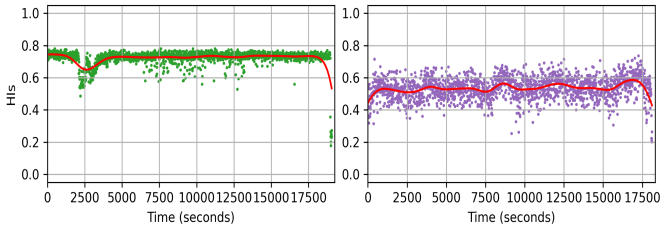

Figure 8. CCAE based HI estimates for Bearing_2_3 and Bearing_2_5.

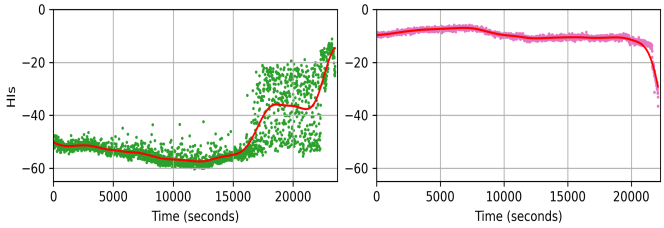

Figure 9. Standard CAE based HI estimates for Bearing_1_3 and Bearing_1_7.

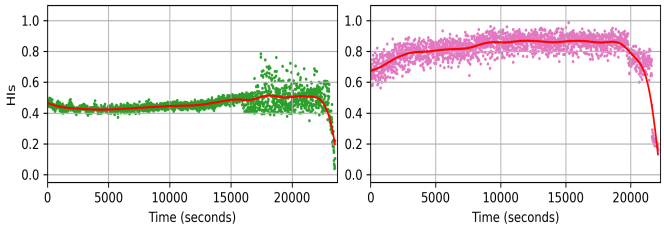

Figure 10. CCAE based HI estimates for Bearing_1_3 and Bearing_1_7.

Table 6 shows the same comparison of standard CAE and CCAE methods for the XJTU-SY dataset. Here, CCAE outperforms the standard CAE with even larger margins: more than 86% improvement in trendability and consistency, and up to 100% in robustness. These results highlight CCAE's superior performance on this dataset, further confirming its ability to generate reliable and consistent HI estimates across different operational conditions and bearing failure modes.

In general, the CCAE method improves on the standard CAE HI estimation approach by providing more consistent HI estimates. It is generalizable and robust, handling noisy and

Table 5. Performance comparison of standard CAE and CCAE-based HI estimation across all pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent testing bearings.

Bearings	Standard CAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	-0.180 ± 0.026	$\mathbf{0.954 \pm 0.001}$	$\mathbf{0.985 \pm 0.012}$	$\mathbf{-0.991 \pm 0.005}$	0.946 ± 0.003	0.921 ± 0.070
Bearing_1_2	-0.464 ± 0.055	0.940 ± 0.003	$\mathbf{0.877 \pm 0.101}$	$\mathbf{-0.963 \pm 0.019}$	$\mathbf{0.941 \pm 0.005}$	0.723 ± 0.244
Bearing_1_3	0.324 ± 0.027	0.936 ± 0.003	0.691 ± 0.199	$\mathbf{-0.392 \pm 0.394}$	$\mathbf{0.946 \pm 0.016}$	$\mathbf{0.856 \pm 0.091}$
Bearing_1_4	-0.772 ± 0.020	0.942 ± 0.001	$\mathbf{0.929 \pm 0.056}$	$\mathbf{-0.785 \pm 0.109}$	$\mathbf{0.948 \pm 0.016}$	0.814 ± 0.147
Bearing_1_5	0.597 ± 0.077	$\mathbf{0.972 \pm 0.002}$	0.772 ± 0.171	$\mathbf{0.316 \pm 0.247}$	0.946 ± 0.006	$\mathbf{0.922 \pm 0.073}$
Bearing_1_6	0.446 ± 0.117	0.916 ± 0.004	0.883 ± 0.089	$\mathbf{0.157 \pm 0.227}$	$\mathbf{0.922 \pm 0.009}$	$\mathbf{0.938 \pm 0.059}$
Bearing_1_7	$\mathbf{-0.629 \pm 0.121}$	$\mathbf{0.966 \pm 0.002}$	0.899 ± 0.083	0.339 ± 0.248	0.948 ± 0.004	$\mathbf{0.949 \pm 0.046}$
Bearing_2_1	-0.568 ± 0.004	0.927 ± 0.001	0.915 ± 0.025	$\mathbf{-0.983 \pm 0.000}$	$\mathbf{0.943 \pm 0.004}$	$\mathbf{0.922 \pm 0.070}$
Bearing_2_2	0.397 ± 0.030	0.948 ± 0.002	0.944 ± 0.025	$\mathbf{-0.981 \pm 0.001}$	$\mathbf{0.949 \pm 0.006}$	$\mathbf{0.959 \pm 0.040}$
Bearing_2_3	$\mathbf{-0.732 \pm 0.005}$	$\mathbf{0.975 \pm 0.008}$	0.602 ± 0.224	-0.092 ± 0.337	0.879 ± 0.091	$\mathbf{0.743 \pm 0.184}$
Bearing_2_4	0.381 ± 0.108	0.926 ± 0.002	0.816 ± 0.208	$\mathbf{-0.665 \pm 0.143}$	$\mathbf{0.930 \pm 0.008}$	$\mathbf{0.937 \pm 0.066}$
Bearing_2_5	$\mathbf{-0.856 \pm 0.051}$	$\mathbf{0.921 \pm 0.002}$	0.850 ± 0.142	-0.054 ± 0.488	0.874 ± 0.034	$\mathbf{0.896 \pm 0.082}$
Bearing_2_6	0.683 ± 0.016	0.915 ± 0.005	0.864 ± 0.117	$\mathbf{-0.220 \pm 0.462}$	$\mathbf{0.917 \pm 0.009}$	$\mathbf{0.907 \pm 0.083}$
Bearing_2_7	-0.472 ± 0.027	0.846 ± 0.009	0.507 ± 0.346	$\mathbf{-0.506 \pm 0.320}$	$\mathbf{0.879 \pm 0.107}$	$\mathbf{0.600 \pm 0.365}$
Bearing_3_1	-0.134 ± 0.088	0.930 ± 0.004	0.533 ± 0.343	$\mathbf{-0.947 \pm 0.009}$	$\mathbf{0.938 \pm 0.005}$	$\mathbf{0.946 \pm 0.057}$
Bearing_3_2	-0.565 ± 0.078	0.925 ± 0.003	$\mathbf{0.982 \pm 0.012}$	$\mathbf{-0.955 \pm 0.008}$	$\mathbf{0.943 \pm 0.004}$	0.956 ± 0.045
Bearing_3_3	$\mathbf{-0.737 \pm 0.045}$	$\mathbf{0.966 \pm 0.001}$	0.850 ± 0.102	-0.599 ± 0.127	0.918 ± 0.013	$\mathbf{0.852 \pm 0.143}$

Table 6. Performance comparison of standard CAE and CCAE-based HI estimation across all XJTU-SY bearings. Shaded rows indicate training bearings; unshaded rows represent testing bearings.

Bearings	Standard CAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_2	-0.232 ± 0.190	0.573 ± 0.041	0.962 ± 0.026	$\mathbf{-0.892 \pm 0.102}$	$\mathbf{0.931 \pm 0.008}$	$\mathbf{0.963 \pm 0.035}$
Bearing_1_4	-0.828 ± 0.039	0.750 ± 0.064	0.851 ± 0.106	$\mathbf{-0.915 \pm 0.084}$	$\mathbf{0.935 \pm 0.006}$	$\mathbf{0.923 \pm 0.068}$
Bearing_1_1	0.181 ± 0.397	0.673 ± 0.082	$\mathbf{0.852 \pm 0.110}$	$\mathbf{-0.417 \pm 0.307}$	$\mathbf{0.891 \pm 0.033}$	0.824 ± 0.139
Bearing_1_3	0.615 ± 0.170	0.726 ± 0.056	0.823 ± 0.079	$\mathbf{-0.004 \pm 0.471}$	$\mathbf{0.895 \pm 0.031}$	$\mathbf{0.866 \pm 0.130}$
Bearing_1_5	0.742 ± 0.245	0.704 ± 0.085	$\mathbf{0.770 \pm 0.098}$	$\mathbf{-0.380 \pm 0.280}$	$\mathbf{0.911 \pm 0.020}$	0.645 ± 0.273
Bearing_2_1	-0.034 ± 0.117	0.463 ± 0.054	0.895 ± 0.086	$\mathbf{-0.838 \pm 0.021}$	$\mathbf{0.970 \pm 0.004}$	$\mathbf{0.946 \pm 0.064}$
Bearing_2_5	-0.105 ± 0.051	0.695 ± 0.048	0.737 ± 0.215	$\mathbf{-0.902 \pm 0.036}$	$\mathbf{0.932 \pm 0.006}$	$\mathbf{0.935 \pm 0.073}$
Bearing_2_2	$\mathbf{-0.227 \pm 0.241}$	0.742 ± 0.093	0.728 ± 0.203	-0.125 ± 0.412	$\mathbf{0.928 \pm 0.007}$	$\mathbf{0.924 \pm 0.049}$
Bearing_2_3	-0.089 ± 0.090	0.653 ± 0.114	0.774 ± 0.213	$\mathbf{-0.135 \pm 0.120}$	$\mathbf{0.908 \pm 0.012}$	$\mathbf{0.938 \pm 0.049}$
Bearing_2_4	0.145 ± 0.430	0.828 ± 0.048	0.554 ± 0.331	$\mathbf{-0.120 \pm 0.477}$	$\mathbf{0.926 \pm 0.021}$	$\mathbf{0.769 \pm 0.167}$
Bearing_3_2	-0.200 ± 0.332	0.521 ± 0.075	0.945 ± 0.046	$\mathbf{-0.932 \pm 0.036}$	$\mathbf{0.940 \pm 0.003}$	$\mathbf{0.958 \pm 0.039}$
Bearing_3_4	-0.148 ± 0.155	0.827 ± 0.088	0.866 ± 0.125	$\mathbf{-0.937 \pm 0.020}$	$\mathbf{0.942 \pm 0.009}$	$\mathbf{0.967 \pm 0.033}$
Bearing_3_1	0.365 ± 0.199	0.590 ± 0.108	0.857 ± 0.101	$\mathbf{-0.139 \pm 0.190}$	$\mathbf{0.779 \pm 0.131}$	$\mathbf{0.905 \pm 0.135}$
Bearing_3_3	$\mathbf{0.002 \pm 0.066}$	0.533 ± 0.089	0.833 ± 0.136	0.047 ± 0.211	$\mathbf{0.799 \pm 0.121}$	$\mathbf{0.857 \pm 0.178}$
Bearing_3_5	0.237 ± 0.220	0.541 ± 0.106	0.514 ± 0.359	$\mathbf{0.184 \pm 0.156}$	$\mathbf{0.842 \pm 0.104}$	$\mathbf{0.590 \pm 0.285}$

unseen test data much better.

4.3. Comparison of Soft-Rank Loss Function based CAE and CCAE Methods

Table 7 provides a detailed comparison of the performance of the SR-CAE HI estimation method compared to the CCAE method across all pronostia bearings. The results indicate that the SR-CAE method surpasses the CCAE method in trendability for nearly 95% of bearings. Conversely, the CCAE method outperforms the SR-CAE method in robustness for nearly 95% of the bearings and exhibits greater consistency for nearly 90% of the bearings with a smaller overall variance.

The advantage of the SR-CAE method in trendability comes from its objective function, which minimizes the reconstruction loss while enforcing a decreasing monotonicity in the HI estimates without constraints, resulting in a more monotonic degradation pattern. However, CCAE demonstrates superior robustness and consistency, showing less variability in HI estimates. This suggests more stable degradation trends and smoother transitions in HI values over time. Furthermore, for multiple HI estimates, the results consistently provide stable and replicable values.

Figure 11 provides a validation of these results. The improved monotonicity of the HI values is notably apparent in Bearing_2_3 when using the SR-CAE method, as opposed to CCAE which was shown in Figure 8. However, significant variability in HI values is observed in both Bearing_2_3 and Bearing_2_5 along their lifetime with the SR-CAE method, unlike the CCAE method which usually provided bounded HI values between 0 and 1. Such disparities complicate the generalizability of the SR-CAE method when compared to the CCAE method.

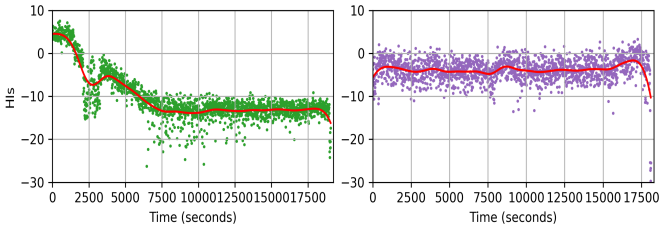

Figure 11. SR-CAE based HI estimates for Bearing_2_3 and Bearing_2_5.

Similarly, Table 8 shows the performance comparison between SR-CAE and CCAE for the XJTU-SY dataset. SR-CAE again leads in trendability, outperforming CCAE in over 86% of the bearings. However, CCAE exhibits superior robustness in all bearings and achieves better consistency in over 93% of them, maintaining a lower overall variance. These findings reinforce the strength of CCAE in generating reliable and smooth HIs under varying conditions.

5. ABLATION STUDY

In this section, we analyze the effects of modifying various aspects of the complete implementation of the CCAE method. Specifically, we examine the impact of individual constraints within the CCAE method, the influence of different constraint rescale factors, and the effect of replacing the monotonicity constraint with a soft-rank loss function in the CCAE method. Given the consistent performance of CCAE across both bearing datasets in earlier experiments, the ablation studies in this section are conducted using only the pronostia dataset to ensure a clear and concise presentation.

5.1. Impact of Constraints on CCAE

In this section, we examine the impact of each constraint in the CCAE implementation by systematically excluding one constraint at a time and evaluating the resulting performance. The experiments devised for this study are the following:

1. **CCAE_EB**: The monotonicity constraint is excluded. However, the energy-HI consistency and boundary constraints are retained, with rescale factors of 1.5 and 2.0, respectively.
2. **CCAE_MB**: The energy-HI consistency constraint is excluded. However, monotonicity and boundary constraints are retained, with rescale factors of [1.25, 1.5] and 2.0, respectively.
3. **CCAE_ME**: The boundary constraints are excluded, leaving only the monotonicity and energy-HI consistency constraints in the CCAE method, with rescale factors of [1.25, 1.5] and 1.5, respectively.

The results of this experiment, presented in Table 9, highlight the importance of the different constraints in the CCAE method. The monotonicity constraint significantly enhances the trendability of the HI estimates, as its primary objective is to enforce the progressive health degradation of the bearings. The boundary constraint plays a crucial role in maintaining the consistency of the HI estimates by ensuring that they remain within the defined range of [0, 1], leading to reliable HI estimates across multiple experiments. Furthermore, the energy-HI consistency constraint contributes to both robustness and consistency. Its formulation penalizes high fluctuations in HI predictions that do not align with the energy progression of the signal, thereby providing a smooth and reliable degradation trend over time.

These findings are further supported by the summarized results in Figure 12, which illustrates the average performance across the three key metrics, trendability, robustness, and consistency, when each constraint is individually removed from the full CCAE model, resulting in a semi-constrained variant. The figure presents the performance difference (Δ) between the fully constrained CCAE and its semi-constrained variants, where negative values indicate a drop in perfor-

Table 7. Performance comparison of SR-CAE and CCAE-based HI estimation across all pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent testing bearings.

Bearings	SR-CAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	$\mathbf{-0.998 \pm 0.000}$	0.895 ± 0.009	0.915 ± 0.047	-0.991 ± 0.005	$\mathbf{0.946 \pm 0.003}$	$\mathbf{0.921 \pm 0.070}$
Bearing_1_2	$\mathbf{-0.989 \pm 0.001}$	0.859 ± 0.021	$\mathbf{0.860 \pm 0.076}$	-0.963 ± 0.019	$\mathbf{0.941 \pm 0.005}$	0.723 ± 0.244
Bearing_1_3	$\mathbf{-0.418 \pm 0.147}$	0.906 ± 0.011	0.762 ± 0.113	-0.392 ± 0.394	$\mathbf{0.946 \pm 0.016}$	$\mathbf{0.856 \pm 0.091}$
Bearing_1_4	$\mathbf{-0.921 \pm 0.025}$	0.867 ± 0.047	$\mathbf{0.844 \pm 0.149}$	-0.785 ± 0.109	$\mathbf{0.948 \pm 0.016}$	0.814 ± 0.147
Bearing_1_5	$\mathbf{-0.537 \pm 0.174}$	0.552 ± 0.099	0.913 ± 0.067	0.316 ± 0.247	$\mathbf{0.946 \pm 0.006}$	$\mathbf{0.922 \pm 0.073}$
Bearing_1_6	$\mathbf{-0.500 \pm 0.130}$	0.812 ± 0.054	0.901 ± 0.114	0.157 ± 0.227	$\mathbf{0.922 \pm 0.009}$	$\mathbf{0.938 \pm 0.059}$
Bearing_1_7	$\mathbf{-0.868 \pm 0.021}$	0.636 ± 0.026	0.930 ± 0.055	0.339 ± 0.248	$\mathbf{0.948 \pm 0.004}$	$\mathbf{0.949 \pm 0.046}$
Bearing_2_1	$\mathbf{-0.995 \pm 0.001}$	0.896 ± 0.008	0.825 ± 0.113	-0.983 ± 0.000	$\mathbf{0.943 \pm 0.004}$	$\mathbf{0.922 \pm 0.070}$
Bearing_2_2	$\mathbf{-0.997 \pm 0.000}$	0.893 ± 0.005	0.917 ± 0.057	-0.981 ± 0.001	$\mathbf{0.949 \pm 0.006}$	$\mathbf{0.959 \pm 0.040}$
Bearing_2_3	$\mathbf{-0.672 \pm 0.054}$	$\mathbf{0.938 \pm 0.034}$	0.666 ± 0.201	-0.092 ± 0.337	0.879 ± 0.091	$\mathbf{0.743 \pm 0.184}$
Bearing_2_4	$\mathbf{-0.839 \pm 0.040}$	0.627 ± 0.048	0.789 ± 0.167	-0.665 ± 0.143	$\mathbf{0.930 \pm 0.008}$	$\mathbf{0.937 \pm 0.066}$
Bearing_2_5	0.009 ± 0.156	0.710 ± 0.086	0.878 ± 0.128	$\mathbf{-0.054 \pm 0.488}$	$\mathbf{0.874 \pm 0.037}$	$\mathbf{0.896 \pm 0.082}$
Bearing_2_6	$\mathbf{-0.500 \pm 0.201}$	0.774 ± 0.046	0.829 ± 0.161	-0.220 ± 0.462	$\mathbf{0.917 \pm 0.009}$	$\mathbf{0.907 \pm 0.083}$
Bearing_2_7	$\mathbf{-0.803 \pm 0.071}$	0.845 ± 0.112	0.423 ± 0.332	-0.506 ± 0.320	$\mathbf{0.879 \pm 0.107}$	$\mathbf{0.600 \pm 0.365}$
Bearing_3_1	$\mathbf{-0.988 \pm 0.002}$	0.805 ± 0.004	0.927 ± 0.032	-0.947 ± 0.009	$\mathbf{0.938 \pm 0.005}$	$\mathbf{0.946 \pm 0.057}$
Bearing_3_2	$\mathbf{-0.995 \pm 0.001}$	0.838 ± 0.012	0.932 ± 0.056	-0.955 ± 0.008	$\mathbf{0.943 \pm 0.004}$	$\mathbf{0.956 \pm 0.045}$
Bearing_3_3	$\mathbf{-0.966 \pm 0.005}$	0.884 ± 0.017	0.805 ± 0.135	-0.599 ± 0.127	$\mathbf{0.918 \pm 0.013}$	$\mathbf{0.852 \pm 0.143}$

Table 8. Performance comparison of SR-CAE and CCAE-based HI estimation across all XJTU-SY bearings. Shaded rows indicate training bearings; unshaded rows represent testing bearings.

Bearings	SR-CAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_2	$\mathbf{-0.999 \pm 0.000}$	0.923 ± 0.009	0.877 ± 0.064	-0.892 ± 0.102	$\mathbf{0.931 \pm 0.008}$	$\mathbf{0.963 \pm 0.035}$
Bearing_1_4	$\mathbf{-0.998 \pm 0.000}$	0.896 ± 0.013	0.801 ± 0.096	-0.915 ± 0.084	$\mathbf{0.935 \pm 0.006}$	$\mathbf{0.923 \pm 0.068}$
Bearing_1_1	$\mathbf{-0.742 \pm 0.086}$	0.855 ± 0.053	0.816 ± 0.133	-0.417 ± 0.307	$\mathbf{0.891 \pm 0.033}$	$\mathbf{0.824 \pm 0.139}$
Bearing_1_3	$\mathbf{-0.797 \pm 0.068}$	0.845 ± 0.043	0.821 ± 0.210	-0.004 ± 0.471	$\mathbf{0.895 \pm 0.031}$	$\mathbf{0.866 \pm 0.130}$
Bearing_1_5	$\mathbf{-0.606 \pm 0.131}$	0.832 ± 0.041	$\mathbf{0.718 \pm 0.219}$	-0.380 ± 0.280	$\mathbf{0.911 \pm 0.020}$	0.645 ± 0.273
Bearing_2_1	$\mathbf{-0.962 \pm 0.004}$	0.745 ± 0.031	0.894 ± 0.076	-0.838 ± 0.021	$\mathbf{0.970 \pm 0.004}$	$\mathbf{0.946 \pm 0.064}$
Bearing_2_5	$\mathbf{-0.980 \pm 0.007}$	0.816 ± 0.027	0.868 ± 0.076	-0.902 ± 0.036	$\mathbf{0.932 \pm 0.006}$	$\mathbf{0.935 \pm 0.073}$
Bearing_2_2	$\mathbf{-0.211 \pm 0.242}$	0.809 ± 0.059	0.839 ± 0.184	-0.125 ± 0.412	$\mathbf{0.928 \pm 0.007}$	$\mathbf{0.924 \pm 0.049}$
Bearing_2_3	$\mathbf{-0.203 \pm 0.145}$	0.773 ± 0.082	0.844 ± 0.118	-0.135 ± 0.120	$\mathbf{0.908 \pm 0.012}$	$\mathbf{0.938 \pm 0.049}$
Bearing_2_4	0.084 ± 0.299	0.711 ± 0.112	0.710 ± 0.252	$\mathbf{-0.120 \pm 0.477}$	$\mathbf{0.926 \pm 0.021}$	$\mathbf{0.769 \pm 0.167}$
Bearing_3_2	$\mathbf{-0.941 \pm 0.007}$	0.704 ± 0.016	0.943 ± 0.029	-0.932 ± 0.036	$\mathbf{0.940 \pm 0.003}$	$\mathbf{0.958 \pm 0.039}$
Bearing_3_4	-0.925 ± 0.006	0.893 ± 0.072	0.945 ± 0.038	$\mathbf{-0.937 \pm 0.020}$	$\mathbf{0.942 \pm 0.009}$	$\mathbf{0.967 \pm 0.033}$
Bearing_3_1	$\mathbf{-0.332 \pm 0.146}$	0.760 ± 0.079	0.813 ± 0.102	-0.139 ± 0.190	$\mathbf{0.779 \pm 0.131}$	$\mathbf{0.905 \pm 0.135}$
Bearing_3_3	$\mathbf{-0.159 \pm 0.116}$	0.796 ± 0.112	0.454 ± 0.305	0.047 ± 0.211	$\mathbf{0.799 \pm 0.121}$	$\mathbf{0.857 \pm 0.178}$
Bearing_3_5	$\mathbf{-0.227 \pm 0.444}$	0.665 ± 0.031	0.475 ± 0.318	0.184 ± 0.156	$\mathbf{0.842 \pm 0.104}$	$\mathbf{0.590 \pm 0.285}$

Table 9. Impact of constraints on CCAE performance on all pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent testing bearings.

Bearings	CCAE_EB			CCAE_MB			CCAE_ME		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	-0.523 ± 0.075	0.923 ± 0.005	$\mathbf{0.995 \pm 0.004}$	-0.985 ± 0.003	0.904 ± 0.012	0.958 ± 0.036	$\mathbf{-0.994 \pm 0.002}$	$\mathbf{0.991 \pm 0.001}$	0.862 ± 0.087
Bearing_1_2	-0.557 ± 0.054	0.930 ± 0.009	0.877 ± 0.147	-0.876 ± 0.027	0.915 ± 0.006	$\mathbf{0.935 \pm 0.060}$	$\mathbf{-0.966 \pm 0.011}$	$\mathbf{0.988 \pm 0.002}$	0.495 ± 0.293
Bearing_1_3	-0.188 ± 0.320	0.875 ± 0.131	0.814 ± 0.112	$\mathbf{-0.775 \pm 0.205}$	0.923 ± 0.020	$\mathbf{0.911 \pm 0.052}$	-0.064 ± 0.300	$\mathbf{0.956 \pm 0.011}$	0.909 ± 0.069
Bearing_1_4	-0.704 ± 0.133	0.909 ± 0.027	0.855 ± 0.107	$\mathbf{-0.710 \pm 0.224}$	0.950 ± 0.010	$\mathbf{0.895 \pm 0.072}$	-0.629 ± 0.177	$\mathbf{0.972 \pm 0.005}$	0.862 ± 0.090
Bearing_1_5	0.230 ± 0.240	0.918 ± 0.020	0.864 ± 0.187	$\mathbf{-0.237 \pm 0.228}$	0.934 ± 0.011	$\mathbf{0.864 \pm 0.142}$	-0.046 ± 0.296	$\mathbf{0.974 \pm 0.005}$	0.838 ± 0.140
Bearing_1_6	0.034 ± 0.153	0.892 ± 0.017	$\mathbf{0.927 \pm 0.082}$	$\mathbf{-0.433 \pm 0.142}$	0.896 ± 0.031	0.860 ± 0.167	0.028 ± 0.123	$\mathbf{0.968 \pm 0.007}$	0.833 ± 0.181
Bearing_1_7	0.407 ± 0.172	0.923 ± 0.010	$\mathbf{0.939 \pm 0.062}$	$\mathbf{-0.658 \pm 0.204}$	0.935 ± 0.011	0.907 ± 0.075	-0.143 ± 0.388	$\mathbf{0.973 \pm 0.004}$	0.929 ± 0.055
Bearing_2_1	-0.622 ± 0.060	0.897 ± 0.006	$\mathbf{0.993 \pm 0.005}$	$\mathbf{-0.991 \pm 0.002}$	0.937 ± 0.005	0.868 ± 0.128	-0.983 ± 0.003	$\mathbf{0.975 \pm 0.004}$	0.671 ± 0.224
Bearing_2_2	-0.598 ± 0.045	0.900 ± 0.006	$\mathbf{0.992 \pm 0.005}$	$\mathbf{-0.995 \pm 0.001}$	0.944 ± 0.009	0.901 ± 0.093	-0.987 ± 0.005	$\mathbf{0.977 \pm 0.006}$	0.863 ± 0.123
Bearing_2_3	-0.264 ± 0.308	0.817 ± 0.169	$\mathbf{0.747 \pm 0.168}$	$\mathbf{-0.402 \pm 0.192}$	$\mathbf{0.858 \pm 0.097}$	0.652 ± 0.197	-0.323 ± 0.201	0.853 ± 0.100	0.682 ± 0.191
Bearing_2_4	-0.638 ± 0.377	0.905 ± 0.027	$\mathbf{0.893 \pm 0.076}$	-0.715 ± 0.186	0.925 ± 0.011	0.746 ± 0.254	$\mathbf{-0.765 \pm 0.120}$	$\mathbf{0.942 \pm 0.012}$	0.732 ± 0.230
Bearing_2_5	0.166 ± 0.306	0.808 ± 0.080	$\mathbf{0.918 \pm 0.084}$	$\mathbf{-0.180 \pm 0.283}$	0.869 ± 0.044	0.864 ± 0.127	-0.130 ± 0.318	$\mathbf{0.916 \pm 0.045}$	0.856 ± 0.139
Bearing_2_6	-0.168 ± 0.322	0.883 ± 0.025	0.899 ± 0.104	-0.387 ± 0.287	0.905 ± 0.014	0.847 ± 0.139	$\mathbf{-0.571 \pm 0.211}$	$\mathbf{0.937 \pm 0.008}$	$\mathbf{0.923 \pm 0.081}$
Bearing_2_7	-0.486 ± 0.234	0.882 ± 0.080	$\mathbf{0.563 \pm 0.287}$	$\mathbf{-0.611 \pm 0.262}$	0.841 ± 0.127	0.393 ± 0.410	-0.518 ± 0.277	$\mathbf{0.892 \pm 0.088}$	0.474 ± 0.353
Bearing_3_1	-0.484 ± 0.038	0.915 ± 0.011	$\mathbf{0.969 \pm 0.033}$	$\mathbf{-0.981 \pm 0.003}$	0.932 ± 0.006	0.945 ± 0.046	-0.959 ± 0.008	$\mathbf{0.983 \pm 0.003}$	0.904 ± 0.104
Bearing_3_2	-0.496 ± 0.027	0.917 ± 0.003	$\mathbf{0.992 \pm 0.006}$	$\mathbf{-0.991 \pm 0.002}$	0.951 ± 0.006	0.923 ± 0.063	-0.974 ± 0.003	$\mathbf{0.986 \pm 0.004}$	0.915 ± 0.062
Bearing_3_3	-0.234 ± 0.317	0.910 ± 0.016	0.898 ± 0.101	$\mathbf{-0.769 \pm 0.151}$	0.855 ± 0.030	$\mathbf{0.949 \pm 0.040}$	-0.673 ± 0.231	$\mathbf{0.954 \pm 0.012}$	0.928 ± 0.069

mance resulting from the removal of a specific constraint. The results reveal that not every constraint contributes positively to all metrics. Specifically, the monotonicity constraint improves trendability and robustness, but enforcing it across varying operational runs slightly degrades consistency. In contrast, the boundary constraints positively influences trendability and consistency by keeping the HI values within a valid range, though they may slightly reduce robustness in trying to satisfy the bounds. The energy–HI consistency constraint significantly enhances robustness and consistency, but can negatively affect trendability since the energy progression does not always follow a strictly monotonic decline. Overall, while these constraints are not fully complementary, each plays a critical role in shaping meaningful, stable, and interpretable HI estimates. Their combined effect enables the CCAE framework to model bearing degradation patterns more effectively with improved reliability and generalization.

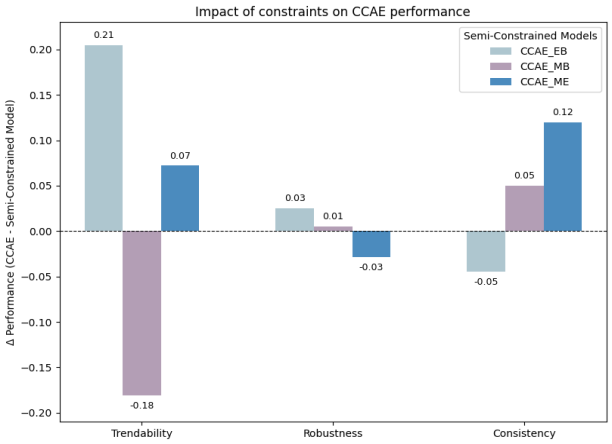

Figure 12. Summary of the impact of individual constraints on CCAE performance metrics.

5.2. Effects of Rescale Factors on CCAE

This section examines the impact of the selection of the rescale factors on the performance of CCAE. The objective of this experiment is to assess the stability of the CCAE method, demonstrating that small changes to the rescale factors do not lead to significant performance fluctuations. Building on the results from Section 4.2, where the rescale factors [1.25, 1.5], 1.5, 2.0, and 2.0 (referred as RF_C1) were applied to the monotonicity, energy-HI consistency, upper and lower boundary constraints, we conduct an additional experiment to evaluate the effect of small variations in these factors. To this end, we examine an alternative set of rescale factors: [1.05, 1.25], 1.25, 1.25, and 1.25 (referred as RF_C2).

The results of this experiment, presented in Table 10, highlight the performance differences of the CCAE method when using different constraint rescale factors. The findings indicate that in terms of trendability, RF_C2 slightly outperforms RF_C1 in 58.82% of the bearings. Both models show minimal variability, with the largest observed differences being -0.322 ± 0.228 for RF_C1 and -0.092 ± 0.337 for RF_C2 in Bearing_2_3. When comparing robustness, RF_C1 outperforms RF_C2 in 58.82% of the bearings. This suggests a slight overall advantage for RF_C1. Furthermore, the variability between the models for robustness remains minimal, with the largest differences being 0.922 ± 0.018 for RF_C1 and 0.879 ± 0.107 for RF_C2 in Bearing_2_7. In terms of consistency, RF_C2 demonstrates superior performance over RF_C1 in 64.71% of the bearings. This improvement is attributed to the larger boundary constraint rescale factor used in RF_C2. The largest differences in consistency are observed

in Bearing_1_6, where RF_C1 achieves 0.839 ± 0.119 and RF_C2 has a value of 0.938 ± 0.059.

Overall, the results indicate that the performance of both approaches is comparable and that the method is not sensitive to small changes in the rescale factors. In principle, the research factors prioritize the constraints. For example, the constraints with the highest rescale factor will be satisfied first. Hence, they only need to define an order. Their exact value is not that important. However, very large values might cause numerical instabilities. Therefore, we recommend using rescale factors in the range of [1, 2].

5.3. Soft Rank based CCAE

In this approach, the modification of CCAE involves replacing the monotonic degradation constraint with a soft-ranking in the loss function. The influence of the reconstruction and the soft-rank losses on the gradients of the model architecture are equally weighted, with a value of λ set to 1 in Eq. (9). In addition, both the energy–HI consistency constraint and the boundary constraints (upper and lower bounds) are retained, with rescale factors set to 1.5 and 2.0, respectively. This experiment investigates whether enforcing monotonic behavior through the loss function yields different outcomes compared to enforcing it as an explicit constraint.

Table 11 presents the performance comparison between the soft-rank CCAE variant and the fully constrained CCAE model across all bearings. The results indicate that the fully constrained CCAE achieves slightly better performance in trendability and consistency for approximately 53.3% of the bearings. A more pronounced advantage is observed in robustness, where CCAE outperforms the soft-rank variant in nearly 95% of the bearings, although the performance margins remain modest. Overall, the experimental findings suggest that incorporating domain-specific behavior through explicit constraints leads to marginal but consistent improvements over embedding the same behavior as a loss function. However, when it is not feasible to encode domain knowledge as a constraint, including it as a loss term still provides a significant performance benefit over models that overlook domain information entirely. Therefore, integrating domain knowledge, whether as constraints or loss functions, during training is crucial to fully leverage the predictive capabilities of the model.

6. CONCLUSION

In conclusion, this study successfully demonstrates the potential of a constraint-guided DL framework, specifically CCAE, to develop physically consistent health indicators for bearing PHM. By incorporating domain knowledge through monotonicity, boundary and energy-HI consistency constraints, the CCAE addresses the limitations of conventional data-driven methods. The experimental results show that CCAE is significantly better than the standard CAE, achieving 65% higher robustness and 75% higher consistency. It also outperforms the SR-CAE baseline with a 95% improvement in robustness and a 90% improvement in consistency considering the pronostia dataset. Although the SR-CAE method excels in trendability metrics for 95% of bearings due to the explicit use of monotonicity in model training, CCAE performs better than CAE for 75% of the bearings. Even marginally better results are also observed for other benchmark XJTU-SY dataset. Moreover, the CCAE generates HI outputs that are bounded within a specified range and reliably represent the bearing's health state. Its degradation profiles are smoother and closely align with expected physical degradation patterns, underscoring the value of incorporating domain knowledge constraints. The ablation study further confirms that the monotonicity constraint enhances trendability, the boundary constraint ensures consistency, and the energy-HI consistency constraint improves robustness. Furthermore, our findings indicate that minor changes in the rescale factor of these constraints do not substantially affect the performance of CCAE. These findings underscore the potential of employing representative constraints in the proposed DL framework to generate reliable bearing HIs. Future research could explore the potential for further increase in framework performance by incorporating additional domain-specific constraints, investigate its application to other areas with tailored domain constraints, and examine its potential for RUL prediction. In addition, integrating feature attribution methods could enhance interpretability by identifying which input components influence the learned HIs. This approach could also be extended to handle more complex degradation scenarios, building on the methodology presented here. Overall, this research offers a promising direction for future prognostic applications, improving the reliability and effectiveness of asset health management strategies.

REFERENCES

Biggio, L., & Kastanis, I. (2020). Prognostics and health management of industrial assets: Current progress and road ahead. *Frontiers in Artificial Intelligence*, *3*, 578613.

Blondel, M., Teboul, O., Berthet, Q., & Djolonga, J. (2020). Fast differentiable sorting and ranking. *37th International Conference on Machine Learning, ICML 2020*, *PartF16814*, 927–936.

Carino, J. A., Zurita, D., Delgado, M., Ortega, J. A., & Romero-Troncoso, R. J. (2015). Remaining useful life estimation of ball bearings by means of monotonic score calibration. *Proceedings of the IEEE International Conference on Industrial Technology*, *2015-June*, 1752–1758.

Chen, J., & Liu, Y. (2021). Probabilistic physics-guided machine learning for fatigue data analysis. *Expert Systems*

Table 10. Effects of rescale factors on CCAE performance on all pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent testing bearings.

Bearings	RF_C1			RF_C2		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	$\mathbf{-0.993 \pm 0.002}$	0.944 ± 0.004	0.908 ± 0.071	-0.991 ± 0.005	$\mathbf{0.946 \pm 0.003}$	$\mathbf{0.921 \pm 0.070}$
Bearing_1_2	$\mathbf{-0.971 \pm 0.008}$	0.940 ± 0.004	$\mathbf{0.804 \pm 0.111}$	-0.963 ± 0.019	$\mathbf{0.941 \pm 0.005}$	0.723 ± 0.244
Bearing_1_3	-0.313 ± 0.408	$\mathbf{0.954 \pm 0.006}$	$\mathbf{0.864 \pm 0.118}$	$\mathbf{-0.392 \pm 0.394}$	0.946 ± 0.016	0.856 ± 0.091
Bearing_1_4	-0.741 ± 0.188	$\mathbf{0.952 \pm 0.007}$	$\mathbf{0.878 \pm 0.093}$	$\mathbf{-0.785 \pm 0.109}$	0.948 ± 0.016	0.814 ± 0.147
Bearing_1_5	0.331 ± 0.146	$\mathbf{0.948 \pm 0.006}$	$\mathbf{0.946 \pm 0.056}$	$\mathbf{0.316 \pm 0.247}$	0.946 ± 0.006	0.922 ± 0.073
Bearing_1_6	$\mathbf{0.108 \pm 0.146}$	$\mathbf{0.925 \pm 0.017}$	0.839 ± 0.119	0.157 ± 0.227	0.922 ± 0.009	$\mathbf{0.938 \pm 0.059}$
Bearing_1_7	0.359 ± 0.148	$\mathbf{0.950 \pm 0.005}$	0.920 ± 0.073	$\mathbf{0.339 \pm 0.248}$	0.948 ± 0.004	$\mathbf{0.949 \pm 0.046}$
Bearing_2_1	$\mathbf{-0.984 \pm 0.002}$	0.939 ± 0.005	$\mathbf{0.936 \pm 0.078}$	-0.983 ± 0.000	$\mathbf{0.943 \pm 0.004}$	0.922 ± 0.070
Bearing_2_2	-0.981 ± 0.005	0.942 ± 0.003	0.950 ± 0.038	$\mathbf{-0.981 \pm 0.001}$	$\mathbf{0.949 \pm 0.006}$	$\mathbf{0.959 \pm 0.040}$
Bearing_2_3	$\mathbf{-0.322 \pm 0.228}$	$\mathbf{0.917 \pm 0.017}$	0.737 ± 0.177	-0.092 ± 0.337	0.879 ± 0.091	$\mathbf{0.743 \pm 0.184}$
Bearing_2_4	-0.640 ± 0.211	0.924 ± 0.009	0.850 ± 0.168	$\mathbf{-0.665 \pm 0.143}$	$\mathbf{0.930 \pm 0.008}$	$\mathbf{0.937 \pm 0.066}$
Bearing_2_5	$\mathbf{-0.165 \pm 0.351}$	0.871 ± 0.030	0.886 ± 0.100	-0.054 ± 0.488	$\mathbf{0.874 \pm 0.037}$	$\mathbf{0.896 \pm 0.082}$
Bearing_2_6	-0.035 ± 0.291	$\mathbf{0.919 \pm 0.006}$	0.879 ± 0.147	$\mathbf{-0.220 \pm 0.462}$	0.917 ± 0.009	$\mathbf{0.907 \pm 0.083}$
Bearing_2_7	-0.474 ± 0.267	$\mathbf{0.922 \pm 0.018}$	0.553 ± 0.324	$\mathbf{-0.506 \pm 0.320}$	0.879 ± 0.107	$\mathbf{0.600 \pm 0.365}$
Bearing_3_1	$\mathbf{-0.948 \pm 0.010}$	$\mathbf{0.941 \pm 0.009}$	0.936 ± 0.063	-0.947 ± 0.009	0.938 ± 0.005	$\mathbf{0.946 \pm 0.057}$
Bearing_3_2	-0.948 ± 0.013	0.942 ± 0.005	0.938 ± 0.051	$\mathbf{-0.955 \pm 0.008}$	$\mathbf{0.943 \pm 0.004}$	$\mathbf{0.956 \pm 0.045}$
Bearing_3_3	-0.554 ± 0.229	$\mathbf{0.922 \pm 0.009}$	$\mathbf{0.925 \pm 0.063}$	$\mathbf{-0.599 \pm 0.127}$	0.918 ± 0.013	0.852 ± 0.143

Table 11. Soft Rank CCAE Vs. CCAE based performance evaluations on all pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent testing bearings.

Bearings	Soft Rank CCAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	-0.893 ± 0.031	0.909 ± 0.003	$\mathbf{0.978 \pm 0.020}$	$\mathbf{-0.991 \pm 0.005}$	$\mathbf{0.946 \pm 0.003}$	0.921 ± 0.070
Bearing_1_2	-0.959 ± 0.006	0.898 ± 0.018	$\mathbf{0.783 \pm 0.014}$	$\mathbf{-0.963 \pm 0.019}$	$\mathbf{0.941 \pm 0.005}$	0.723 ± 0.244
Bearing_1_3	-0.298 ± 0.066	0.846 ± 0.025	0.849 ± 0.017	$\mathbf{-0.392 \pm 0.394}$	$\mathbf{0.946 \pm 0.016}$	$\mathbf{0.856 \pm 0.091}$
Bearing_1_4	$\mathbf{-0.838 \pm 0.090}$	0.825 ± 0.072	$\mathbf{0.835 \pm 0.045}$	-0.785 ± 0.109	$\mathbf{0.948 \pm 0.016}$	0.814 ± 0.147
Bearing_1_5	$\mathbf{0.258 \pm 0.105}$	0.928 ± 0.014	0.906 ± 0.081	0.316 ± 0.247	$\mathbf{0.946 \pm 0.006}$	$\mathbf{0.922 \pm 0.073}$
Bearing_1_6	0.161 ± 0.062	0.908 ± 0.022	0.922 ± 0.046	$\mathbf{0.157 \pm 0.227}$	$\mathbf{0.922 \pm 0.009}$	$\mathbf{0.938 \pm 0.059}$
Bearing_1_7	$\mathbf{0.232 \pm 0.048}$	0.901 ± 0.019	0.937 ± 0.064	0.339 ± 0.248	$\mathbf{0.948 \pm 0.004}$	$\mathbf{0.949 \pm 0.046}$
Bearing_2_1	-0.905 ± 0.040	0.842 ± 0.025	0.917 ± 0.018	$\mathbf{-0.983 \pm 0.000}$	$\mathbf{0.943 \pm 0.004}$	$\mathbf{0.922 \pm 0.070}$
Bearing_2_2	-0.950 ± 0.052	0.887 ± 0.012	$\mathbf{0.972 \pm 0.006}$	$\mathbf{-0.981 \pm 0.001}$	$\mathbf{0.949 \pm 0.006}$	0.959 ± 0.040
Bearing_2_3	$\mathbf{-0.172 \pm 0.083}$	0.859 ± 0.016	$\mathbf{0.862 \pm 0.035}$	-0.092 ± 0.337	$\mathbf{0.879 \pm 0.091}$	0.743 ± 0.184
Bearing_2_4	-0.627 ± 0.206	0.869 ± 0.015	0.929 ± 0.048	$\mathbf{-0.665 \pm 0.143}$	$\mathbf{0.930 \pm 0.008}$	$\mathbf{0.937 \pm 0.066}$
Bearing_2_5	$\mathbf{-0.138 \pm 0.153}$	0.817 ± 0.039	0.863 ± 0.035	-0.054 ± 0.488	$\mathbf{0.874 \pm 0.037}$	$\mathbf{0.896 \pm 0.082}$
Bearing_2_6	$\mathbf{-0.278 \pm 0.177}$	0.856 ± 0.027	0.832 ± 0.163	-0.220 ± 0.462	$\mathbf{0.917 \pm 0.009}$	$\mathbf{0.907 \pm 0.083}$
Bearing_2_7	-0.480 ± 0.140	$\mathbf{0.887 \pm 0.009}$	$\mathbf{0.716 \pm 0.074}$	$\mathbf{-0.506 \pm 0.320}$	0.879 ± 0.107	$\mathbf{0.600 \pm 0.365}$
Bearing_3_1	$\mathbf{-0.964 \pm 0.003}$	0.893 ± 0.022	0.932 ± 0.026	-0.947 ± 0.009	$\mathbf{0.938 \pm 0.005}$	$\mathbf{0.946 \pm 0.057}$
Bearing_3_2	-0.914 ± 0.042	0.888 ± 0.009	$\mathbf{0.963 \pm 0.006}$	$\mathbf{-0.955 \pm 0.008}$	$\mathbf{0.943 \pm 0.004}$	0.956 ± 0.045
Bearing_3_3	$\mathbf{-0.670 \pm 0.134}$	0.910 ± 0.067	0.841 ± 0.038	-0.599 ± 0.127	$\mathbf{0.918 \pm 0.013}$	$\mathbf{0.852 \pm 0.143}$

with Applications, *168*, 114316.

Cleveland, W. S., & Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, *83*(403), 596–610.

Cubillo, A., Perinpanayagam, S., & Esperon-Miguez, M. (2016). A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. *Advances in Mechanical Engineering*, *8*(8), 1687814016664660.

Duong, B. P., Khan, S. A., Shon, D., Im, K., Park, J., Lim, D. S., ... Kim, J. M. (2018). A reliable health indicator for fault prognosis of bearings. *Sensors (Switzerland)*, *18*(11).

Hoffmann Souza, M. L., da Costa, C. A., & de Oliveira Ramos, G. (2023). A machine-learning based data-oriented pipeline for Prognosis and Health Management Systems. *Computers in Industry*, *148*(January), 103903.

Jieyang, P., Kimmig, A., Dongkun, W., Niu, Z., Zhi, F., Jiahai, W., ... Ovtcharova, J. (2023). A systematic review of data-driven approaches to fault diagnosis and early warning. *Journal of Intelligent Manufacturing*, *34*(8), 3277–3304.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440.

Lei, Y., Li, N., Gontarz, S., Lin, J., Radkowski, S., & Dybala, J. (2016). A model-based method for remaining useful life prediction of machinery. *IEEE Transactions on reliability*, *65*(3), 1314–1326.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834.

Li, H., Zhang, Z., Li, T., & Si, X. (2024). A review on physics-informed data-driven remaining useful life prediction: Challenges and opportunities. *Mechanical Systems and Signal Processing*, *209*(December 2023), 111-120.

Liao, L., & Köttig, F. (2014). Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, *63*(1), 191–207.

Lu, W., Wang, Y., Zhang, M., & Gu, J. (2024). Physics guided neural network: Remaining useful life prediction of rolling bearings using long short-term memory network through dynamic weighting of degradation process. *Engineering Applications of Artificial Intelligence*, *127*(PB), 107350.

Meire, M., Brijder, R., Dekkers, G., & Karsmakers, P. (2022). Accelerometer-Based Bearing Condition Indicator Estimation Using Supervised Adaptive DSVDD. *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, *14*(1), 1–10.

Meng, C., Seo, S., Cao, D., Griesemer, S., & Liu, Y. (2022). When physics meets machine learning: A survey of physics-informed machine learning. *arXiv preprint arXiv:2203.16797*.

Mosallam, A., Medjaher, K., & Zerhouni, N. (2016). Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, *27*, 1037–1048.

Muralidhar, N., Islam, M. R., Marwah, M., Karpatne, A., & Ramakrishnan, N. (2018). Incorporating Prior Domain Knowledge into Deep Neural Networks. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 36–45.

Nascimento, R. G., & Viana, F. A. (2019). Fleet prognosis with physics-informed recurrent neural networks. *Structural Health Monitoring 2019: Enabling Intelligent Life-Cycle Health Management for Industry Internet of Things (IIOT) - Proceedings of the 12th International Workshop on Structural Health Monitoring*, *2*, 1740–1747.

Patrick Nectoux, K. M., Rafael Gouriveau, Ramasso, E., & Brigitte Chebel-Morello, e. a. (2012). PRONOSTIA: An Experimental Platform for Bearings Accelerated Degradation Tests. *IEEE International Conference on Prognostics and Health Management, PHM'12., Jun 2012, Denver, Colorado, United States.*, 1-8.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707.

Rezaeianjouybari, B., & Shang, Y. (2020). Deep learning for prognostics and health management: State of the art, challenges, and opportunities. *Measurement: Journal of the International Measurement Confederation*, *163*, 107929.

Su, H., & Lee, J. (2023). Machine learning approaches for diagnostics and prognostics of industrial systems using open source data from phm data challenges: a review. *arXiv preprint arXiv:2312.16810*.

Van Baelen, Q., & Karsmakers, P. (2023). Constraint guided gradient descent: Training with inequality constraints with applications in regression and semantic segmentation. *Neurocomputing*, *556*, 126636.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, *11*(12).

Vishwakarma, M., Purohit, R., Harshlata, V., & Rajput, P. (2017). Vibration Analysis & Condition Monitoring

for Rotating Machines: A Review. *Materials Today: Proceedings*, *4*(2), 2659–2664.

Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., ... Schuecker, J. (2023). Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering*, *35*(1), 614–633.

Wang, B., Lei, Y., Li, N., & Li, N. (2018). A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, *69*(1), 401–412.

Wang, J., Li, Y., Zhao, R., & Gao, R. X. (2020). Physics guided neural network for machining tool wear prediction. *Journal of Manufacturing Systems*, *57*(October), 298–310.

Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2022). Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. *ACM Computing Surveys*, *55*(4), 1–34.

Xiong, J., Fink, O., Zhou, J., & Ma, Y. (2023). Controlled physics-informed data generation for deep learning-based remaining useful life prediction under unseen operation conditions. *Mechanical Systems and Signal Processing*, *197*.

Xu, Y., Kohtz, S., Boakye, J., Gardoni, P., & Wang, P. (2023). Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges. *Reliability Engineering & System Safety*, *230*, 108900.

Zhang, B., Zhang, L., & Xu, J. (2016). Degradation Feature Selection for Remaining Useful Life Prediction of Rolling Element Bearings. *Quality and Reliability Engineering International*, *32*(2), 547–554.

Zhou, H., Huang, X., Wen, G., Lei, Z., Dong, S., Zhang, P., & Chen, X. (2022). Construction of health indicators for condition monitoring of rotating machinery: A review of the research. *Expert Systems with Applications*, *203*(March), 117297.