Remaining useful life prediction with limited run-to-failure data: a Bayesian ensemble approach combining mode-dependent RVM and similarity

Zhuyi Li[a], Xianbo Xiang[b], Shuai Liu[c], Yiming Wan*[a,]

[a]School of Artificial Intelligence and Automation, Key Laboratory of Image Processing and Intelligent Control, Ministry of Education of China, Huazhong University of Science and Technology, Wuhan, 430074, Hubei, China
[b]School of Naval Architecture and Ocean Engineering, State Key Laboratory of Intelligent Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, 430074, Hubei, China
[c]National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing, 100071, China

Abstract

Accurate prediction of remaining useful life (RUL) is crucial for predictive maintenance of industrial systems. Although data-driven RUL prediction methods have received considerable attention, they typically require massive run-to-failure (R2F) data which is often unavailable in practice. If not properly addressed, training with a limited number of R2F trajectories not only leads to large errors in RUL prediction, but also causes difficulty in quantifying the prediction uncertainty. To address the above challenge, this paper proposes a Bayesian ensemble RUL prediction method that combines mode-dependent relevance vector machine (RVM) and trajectory similarity. Firstly, the proposed approach clusters historical R2F trajectories of unequal lengths into different degradation modes, and constructs RVM and similarity based predictions with improved accuracy by using mode-dependent libraries of kernel functions and similar trajectories. Secondly, the proposed Bayesian ensemble scheme fuses the RVM and similarity based predictions, and quantifies the associated prediction uncertainty even though the number of historical R2F trajectories are limited. In two case studies involving bearings and batteries, using only 11 and 16 R2F trajectories as training data, respectively, the proposed method reduces the mean absolute percentage error of RUL prediction by more than 20% compared to three existing methods.

Keywords: Remaining useful life prediction, mode-dependent prediction, relevance vector machine, trajectory similarity, Bayesian ensemble.

1. Introduction

Prognostics and health management (PHM) is a rapidly growing field that has received significant attention from both academia and industry due to the imperative to accurately predict the monitored equipment's remaining useful life (RUL) for predictive maintenance [1–4]. Accurate RUL prediction is critical to optimize system performance, reduce downtime, and lower maintenance costs [1–4].

Existing RUL prediction methods can be categorized into three categories: physical model-based, data-driven, and hybrid methods [1, 4, 5]. Physical model-based approaches involve establishing first-principle models that describe the complex degradation mechanisms [6–10]. However, developing accurate mathematical models for most practical degradation processes can be challenging due to limited knowledge of degradation physics. Without requiring first-principle models, data-driven methods leverage various machine learning models to predict RUL [11–15]. The conventional machine learning models for RUL prediction include support vector machine [11, 12], relevance vector machine (RVM) [13], and Gaussian process regression [14, 15], et al. With recent progress in deep learning, long short-term memory networks [16, 17], convolution neural networks (CNN) [18, 19], and recurrent neural networks [20] have emerged in the field of PHM, aiming to iteratively learn the mapping function from degradation features to RUL [21]. Although these deep learning models have demonstrated superior prediction accuracy in recent literature [22], they usually require a large amount of training data and are lack of interpretability, which restricts their applications in industrial practice where run-to-failure (R2F) data is often limited [23]. The hybrid approach integrates multiple methods to overcome the limitation of each individual method, which has received wide attention in recent years. For example, the machine learning model and empirical exponential degradation model are combined in [5, 24] to further improve RUL prediction accuracy. However, existing hybrid methods in literature still do not fully address the issue of limited R2F data.

As a data-driven RUL prediction method, the similarity based approach has the advantages of strong interpretability and simple implementation [3, 25]. It relies on the assumption that the online test trajectory follows a similar trend to that of a HT if they both experience similar operating conditions. By introducing a similarity metric, the similarity based method first searches along each historical trajectory (HT) to locate a segment that exhibits high similarity to the online test trajectory. The next step is to determine RUL as the duration from the end of the best matching segment to the failure point of the corre-

Email address: ywan@hust.edu.cn (Yiming Wan*)

sponding HT [3, 25].

The similarity based RUL prediction algorithms have continued to develop and improve prediction accuracy. Recent relevant research has focused on improving similarity evaluation, uncertainty quantification, and model prediction aggregation. Several techniques have been proposed for improved similarity evaluation, including dynamic time warping [26], maximum mean discrepancy [27–30] and the combination of multiple similarity calculations [31–34]. To quantify the prediction uncertainty, kernel density estimation [34, 35] and Weibull distribution [27, 36] have been employed to determine the distribution of RUL and provide appropriate confidence intervals. Ensemble algorithms, including multiscale similarity ensemble [37], and support vector regression and trajectory similarity integration [38], have also been introduced to enhance the robustness and accuracy of RUL prediction. Despite these advancements, the similarity based methods still face the following limitations:

i) The existing literature often assumes the availability of abundant HTs and a sufficient number of similar trajectories [39, 40]. In practice, however, the number of complete HTs is typically limited [2, 3], and the similarity between HTs and online trajectories is often low. This adversely affects RUL prediction accuracy, and presents a challenge to quantify the prediction uncertainty.
ii) The majority of similarity based RUL prediction methods fail to fully leverage distinctive characteristics of different degradation modes which represent the overall trends of degradation trajectories under a spectrum of operating conditions. Instead, the conventional similarity based method locally evaluates degree of similarity between the HTs and the online test trajectory. While two trajectories may exhibit high local similarity at a specific time instant, they may still belong to different degradation modes, resulting in significant discrepencies in their future trends. This implies that the conventional similarity based method, which does not distinguish between degradation modes, can lead to considerable prediction errors.

To address the above limitations of existing similarity based prediction methods, we propose a mode-dependent RVM-similarity based Bayesian ensemble method in this paper. Firstly, we perform offline mode clustering of HTs of unequal lengths using time feature vectors of the same dimension associated with equally spaced samples along each trajectory. For each degradation mode, we also construct a trajectory library and the corresponding kernel function library. Then, we identify the degradation mode of the test trajectory online, and establish mode-dependent RVM and similarity based HI predictions. These two predictions are adaptively fused by a Bayesian regression approach to determine RUL with uncertainty quantification.

The main contributions are summarized as follows:

i) We propose a Bayesian framework that fuses the RVM and similarity based HI predictions in an adaptive ensemble. This Bayesian ensemble approach not only addresses the limitation of inadequate long-term prediction capability of RVM, but also mitigates performance loss of the similarity based prediction due to lack of highly similar HTs. In addition, this Bayesian ensemble approach generates an appropriate confidence interval for the predicted RUL without necessitating an extensive amount of historical data. Consequently, the proposed Bayesian ensemble method improves prediction accuracy and enhances the ability to generalize in the presence of limited historical R2F data.
ii) We propose mode-dependent prediction with improved accuracy by distinguishing different degradation modes. By doing so, our proposed mode-dependent similarity based prediction excludes dissimilar HTs, and our proposed mode-dependent RVM based prediction selects more suitable values of hyperparameters within the kernel library that align with the identified degradation mode.

The remainder of this paper is structured as follows: Section 2 presents the proposed mode-dependent RVM-similarity based Bayesian ensemble algorithm. Using the PHM2012 bearing dataset and the Toyota battery dataset, we illustrate the advantage of our approach by extensive comparisons with some existing approaches in literature in Sections 3 and 4. Some concluding remarks are provided in Section 5.

2. Methodology

In this section, we propose the mode-dependent RVM-similarity based Bayesian ensemble RUL prediction method whose flowchart is shown in Figure 1. Before applying our proposed algorithm, the health indicator (HI) is extracted from the original condition monitoring data to reflect the degradation status [40], which is not the focus of this paper. In the offline phase, HTs of extracted HIs are clustered into different modes using K-means clustering. Each cluster of HTs is regarded as a trajectory library, and their RVM modeling determines the associated kernel functions as a kernel library corresponding to the specific trajectory library. In the online phase, mode identification of the online HI trajectory is performed to determine the corresponding trajectory library and kernel library for mode-dependent predictions using RVM and trajectory similarity. Finally, the Bayesian ensemble approach is leveraged to aggregate the RVM and similarity based predictions and quantify the aggregated RUL distribution.

2.1. Offline mode clustering of historical trajectories

The degradation trajectories of a system are diverse due to the effect of different operating conditions or various fault factors [41]. For example, as depicted in Figure 2, multiple degradation trajectories of bearings can be roughly classified into slow mode (e.g., trajectory 1-1) and rapid mode (e.g., trajectory 1-2). In the offline phase, we cluster HTs into different degradation modes. This allows the use of mode-dependent prediction models to improve the accuracy of RUL predictions later in Sections 2.3 and 2.4.

The celebrated K-means algorithm is adopted to cluster the HTs, and its basic idea is to assign each trajectory to the cluster

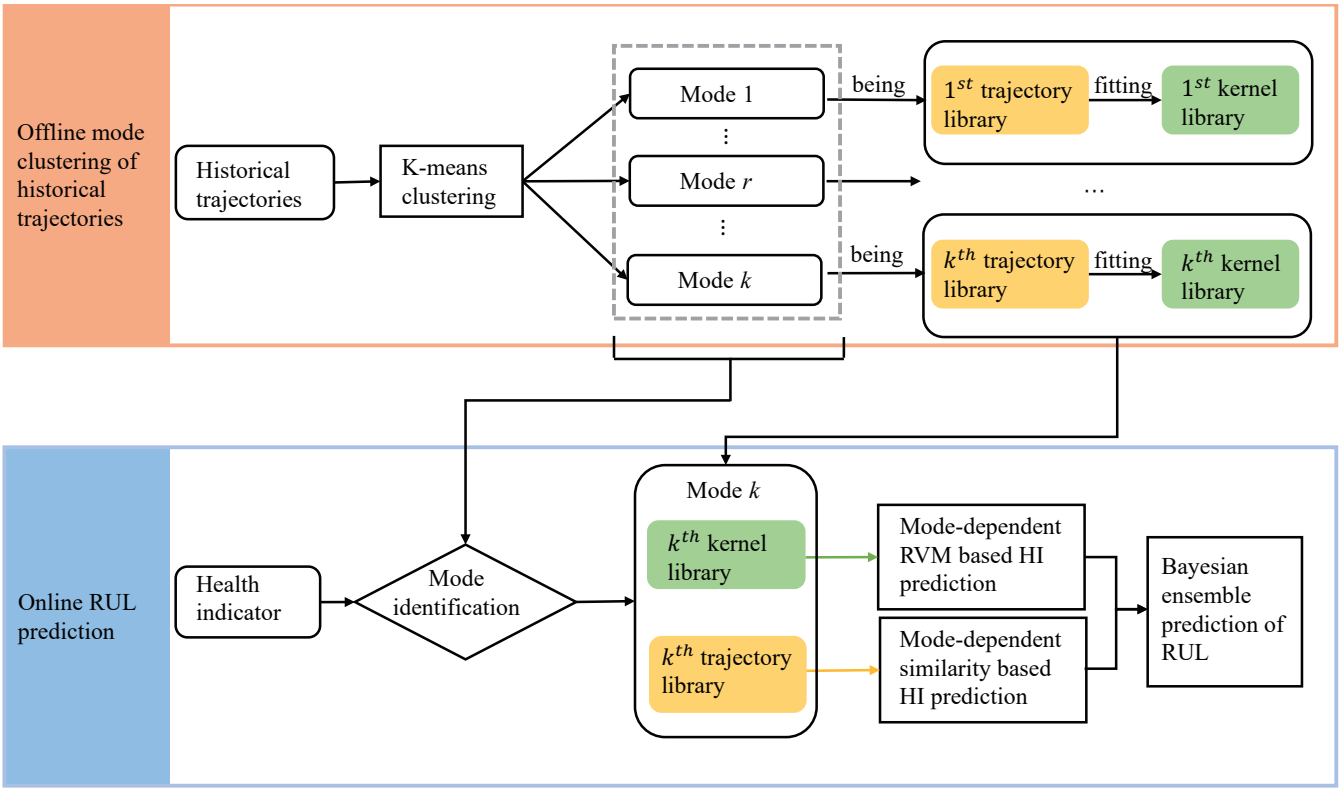

Figure 1: The flowchart of the proposed approach

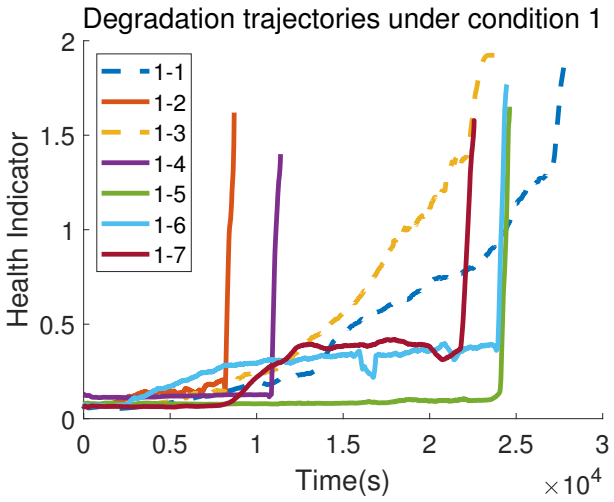

Figure 2: The bearing degradation trajectories.

represented by the nearest cluster center [42]. In the existing RUL literature, K-means clustering is often used in trajectory segment [17, 43], i.e., dividing one trajectory into healthy or damaged stages. In contrast, we focus on the mode clustering of multiple trajectories in this paper. A key difficulty in clustering multiple trajectories is coping with their unequal lengths. This issue cannot be solved by simply clipping or extrapolating trajectories as either option results in loss or error of degradation information.

To cluster trajectories of unequal lengths, we introduce a novel method that does not directly use the HI values of each degradation trajectory as the clustering feature. Instead, the clustering feature of each trajectory $\mathcal{T}_i$ is extracted as a n-dimensional vector $\mathbf{t}^i$ of time indices corresponding to n equally-spaced HI values between the start of degradation and the end of life.

Detailed procedures for extracting the above clustering feature are as follows:

i) Data fitting for each HT

Along each trajectory $\mathcal{T}_i$, we collect data points $\{(t_j^i, c_j^i)\}_{j=1}^{n_i}$, where t_j^i represents the j^{th} inspection time along the trajectory $\mathcal{T}_i$, c_j^i denotes the corresponding HI value, and n_i is the number of data points. It is assumed that all the available trajectories start degradation at the same HI value c_d and end with the same HI value c_e. If such an assumption does not hold for the original degradation trajectories, we perform a normalized transformation before trajectory clustering. To accomplish this, the conventional min-max normalization is not applicable since the min/max HI values are not available beforehand for an online degradation process. Therefore, without requiring the knowledge of min/max HI values, we follow the idea of confidence value in [44] to construct the normalized HI which takes values within the range of $[0, 1]$:

$$\text{HI}_{\text{norm}} = \frac{2}{1 + e^{-a \cdot \text{HI}}} - 1, \tag{1}$$

where a is the scaling parameter defined as

$$a = -\frac{\ln(\frac{\text{HI}_{\text{def}}}{2-\text{HI}_{\text{def}}})}{\text{mean}(\text{HI}_{\text{healthy}})}, \tag{2}$$

with HI_{def} being a predefined HI value at the end of life, and $\text{mean}(\text{HI}_{\text{healthy}})$ representing the averaged HI during the healthy stage before degradation is detected. A higher value of the normalized HI in (1) indicates a higher degree of degradation.

ii) Feature extraction
To facilitate feature extraction, we first leverage the monotonic trend of the degradation process to construct a function (see Remark 1 for more explanations)

$$t = g_i(c) \tag{3}$$

from the data points $\{(t_j^i, c_j^i)\}_{j=1}^{n_i}$ (normalized if needed) for each trajectory $\mathcal{T}_i$, via piecewise linear interpolation. Then, on each trajectory $\mathcal{T}_i$, we select n samples which are equally spaced along the HI axis, i.e., $c_d, c_1, c_2, \cdots, c_{n-2}, c_e$, with the sampling interval

$$\Delta c = |c_1 - c_d| = |c_2 - c_1| = \cdots = |c_e - c_{n-2}| \tag{4}$$

and find their corresponding time indices $\{\delta t_j^i\}_{j=1}^{n}$ along each trajectory $\mathcal{T}_i$ as the clustering feature

$$\mathbf{t}^i = [t_d^i, \delta t_1^i, \delta t_2^i, \cdots, \delta t_{n-1}^i], \tag{5}$$

with $g_i(\cdot)$ defined in (3), and

$$t_d^i = g_i(c_d), \tag{6a}$$
$$\delta t_1^i = g_i(c_1) - g_i(c_d), \tag{6b}$$
$$\delta t_j^i = g_i(c_j) - g_i(c_{j-1}), \quad j = 2, \cdots, n-2, \tag{6c}$$
$$\delta t_{n-1}^i = g_i(c_e) - g_i(c_{n-2}). \tag{6d}$$

Take the battery degradation trajectory in Figure 3 as an example. We consider the knee point between the healthy and degraded stages as (t_d, c_d), and introduce an HI sampling interval Δc to divide the HI range $[c_d, c_e]$ into $n-1$ equally-spaced segments. The clustering feature for the m^{th} battery is $\mathbf{t}^m = [t_d^m, \delta t_1^m, \delta t_2^m, \cdots, \delta t_{n-1}^m]$. For example, the battery HI range spanning from $c_d = 1.045$ to $c_e = 0.887$ is divided into $n-1 = 4$ segments by choosing the HI sampling interval $\Delta c = 0.033$, then the calculation of $\mathbf{t}^m$ would consist of the time values associated with HIs $[1.045, 1.012, 0.979, 0.946, 0.913]$.

iii) Clustering of feature vectors
We perform mode clustering of all HTs by applying K-means clustering to the feature vectors $\{\mathbf{t}^i\}$.

By applying the above procedures with a fixed integer n, the extracted feature vector $\mathbf{t}^i$ has the same dimension n for a run-to-failure trajectory of any given length, and is able to capture the major information of the degradation process for mode clustering.

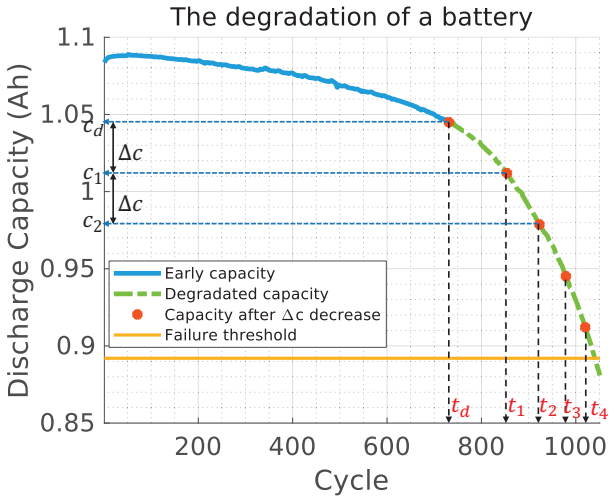

Figure 3: The battery degradation trajectory and its clustering feature.

To determine the number of clusters for the HTs, the gap statistic method in [45] or the silhouette method in [46] can be used, whose details are omitted and referred to [45, 46]. Each cluster r contains S_r HTs which form the HT library $\{\mathcal{T}_i^r\}_{i=1}^{S_r}$ under mode r.

Remark 1. *The above mode clustering method is limited to monotonic degradation processes, because the function in* (3) *allows only one time index t at a particular HI value c. If the degradation trajectory has a monotonic trend contaminated by stochastic non-monotonic variations, we may first obtain the smoothed monotonic degradation trend by filtering out the stochastic variations, and then apply our proposed mode clustering method. It is more challenging to clustering non-monotonically trended degradation trajectories, which is left to our future research.*

Remark 2. *With limited R2F data, K-means clustering has difficulty in accurately determining the number of degradation modes, but it can still ensure reasonably small within-cluster distances since K-means clustering is obtained by minimizing the sum of squared within-cluster distances [42]. This is sufficient for our proposed mode-dependent HI prediction presented later in Sections 2.3 and 2.4 to achieve the goal of selecting the most similar degradation mode and avoiding interference from dissimilar trajectories. Another issue with limited training data is that the online trajectory might have low similarity to all historical trajectories, which renders the similarity based prediction in Section 2.4 unreliable. This issue is effectively accommodated by the subsequent Bayesian ensemble step in Section 2.5: the similarity based prediction would be assigned with a lower weight in the learned Bayesian regression model in this case, while the RVM based prediction would dominate the Bayesian ensemble output.*

Remark 3. *The dimension n of each feature vector $\mathbf{t}^i$ should be sufficiently large such that the dynamic evolution pattern of*

a given degradation trajectory is fully characterized. This implies that we would obtain poor clustering performance and inaccurate RUL prediction if the dimension n of the clustering feature is too small. It should be also noted that a too-large n brings just a marginal performance improvement while introducing unnecessarily heavier computational burden.

2.2. Online mode identification

The online mode identification uses the clustering features mentioned in Section 2.1 to determine which mode an online trajectory belongs to. For this purpose, we follow the procedures as below:

i) Data fitting for the online trajectory
This step is the same as Step i) for the HTs in Section 2.1.

ii) Feature extraction of the online trajectory
Let c_t^{on} denote the online HI value at current time t. Using the sampling interval Δc in (4), there are f_n sampled HI values along the online trajectory, where f_n is

$$f_n = \lfloor \frac{c_t^{\text{on}} - c_d}{\Delta c} \rfloor \tag{7}$$

with $\lfloor \cdot \rfloor$ representing the floor function that outputs the greatest integer less than or equal to a given real number. By applying the same procedure as Step ii) for the HTs in Section 2.1, we obtain the extracted clustering feature of the online trajectory

$$\mathbf{t}^{\text{on}} = [t_d^{\text{on}}, \delta t_1^{\text{on}}, \cdots, \delta t_{f_n-1}^{\text{on}}]. \tag{8}$$

iii) Mode identification of the online trajectory
For each cluster r of the HTs, the center of the extracted feature vectors is represented by $\mathbf{c}_r$. It should be noted that each HT is a complete R2F trajectory whilst an online trajectory is just a partial R2F process. Then the center of the extracted feature vectors $\mathbf{c}_r$ associated with HTs has a higher dimension than that of the online feature vector $\mathbf{t}^{\text{on}}$ in (8). As such, the distance between $\mathbf{c}_r$ and $\mathbf{t}^{\text{on}}$ can be measured by

$$D_r = \|\mathbf{t}^{\text{on}} - \mathbf{c}_r(1:f_n)\|^2, \tag{9}$$

where $\mathbf{c}_r(1:f_n)$ represents the first f_n dimensional features of the cluster center of cluster r. Then, the mode κ of the online trajectory is determined as the one whose cluster center has the shortest distance to the online feature vector $\mathbf{t}^{\text{on}}$, i.e.,

$$\kappa = \arg\min_r D_r, \tag{10}$$

where D_r is defined in (9), and κ is the cluster to which the online trajectory belongs. The next mode identification is triggered whenever the change of the online HI value c_t^{on} with respect to the previous mode center exceeds the specified sampling interval Δc.

Through the above steps, we determine the corresponding trajectory library $\{\mathcal{T}_i^{\kappa}\}_{i=1}^{S_\kappa}$ under mode κ for the online trajectory.

2.3. Mode-dependent RVM based HI prediction

In this section, we first briefly review preliminaries of RVM, and then propose a mode-dependent RVM based RUL prediction method which adaptively selects its kernel function according to the identified degradation mode.

Let $\{t, c\}$ denote a pair of the time t and its corresponding HI value c, which belongs to the available dataset $\{t_i, c_i\}_{i=1}^N$. The underlying input-output relationship is

$$c = f(t) + \epsilon, \tag{11}$$

where $\epsilon = \mathcal{N}(0, \sigma^2)$ is a Gaussian noise. In RVM, the nonlinear function $f(t)$ is expressed as a weighted sum of basis functions given by kernels, i.e.,

$$f(t) = \sum_{i=1}^{N} w_i k(t, t_i) + b, \tag{12}$$

where $k(t, t_i)$ is the kernel function describing the correlation between t and t_i, and b is the bias.

Define the input and output data as $\mathbf{t} = \{t_i\}_{i=1}^N$ and $\mathbf{c} = \{c_i\}_{i=1}^N$. According to (11), the conditional probability of the output data $\mathbf{c}$ is expressed as [42, 47]

$$p(\mathbf{c}|\mathbf{t}, w, \sigma^2) = \prod_{i=1}^{N+1} p(c_i|t_i, w_i, \sigma^2) = \prod_{i=1}^{N+1} \mathcal{N}(c_i|f(t_i), \sigma_\epsilon^2), \tag{13}$$

where w_{N+1} presents the bias b, and $w = (w_1, w_2, ..., w_{N+1})$. The prior distribution of w is set to the following isotropic Gaussian distribution [42, 47]:

$$p(w|\boldsymbol{\alpha}) = \prod_{i=1}^{N+1} \mathcal{N}(w_i|0, \alpha_i^{-1}), \tag{14}$$

where α_i is the precision parameter for w_i, and $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_{N+1}\end{bmatrix}^\top$. According to the Bayes rule, the posterior distribution of w is derived as [42, 47]

$$p(w|\mathbf{c}, \mathbf{t}, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\mathbf{c}|\mathbf{t}, w, \sigma^2)p(w|\boldsymbol{\alpha})}{p(\mathbf{c}|\mathbf{t}, \sigma^2)} = \mathcal{N}(w|\mu_w, \Sigma_w), \tag{15}$$

where μ_w and Σ_w are defined as

$$\mu_w = \Sigma \Phi^\top B \mathbf{c}, \ \Sigma_w = (A + \Phi^\top B \Phi)^{-1}, \tag{16a}$$

$$A = \text{diag}(\alpha_1, \alpha_2, ..., \alpha_{N+1}), \ B = \sigma^{-2} I_N, \tag{16b}$$

$$\Phi = \begin{bmatrix} K(\mathbf{t}, t_1) & K(\mathbf{t}, t_2) & \cdots & K(\mathbf{t}, t_N) & \mathbf{1}_{N\times 1}, \end{bmatrix}, \tag{16c}$$

$$K(\mathbf{t}, t_i) = \begin{bmatrix} k(t_1, t_i) & k(t_2, t_i) & \cdots & k(t_N, t_i) \end{bmatrix}^\top, \tag{16d}$$

with $\text{diag}(\alpha_1, \alpha_2, ..., \alpha_{N+1})$ being a diagonal matrix with $\alpha_1, \alpha_2, \cdots, \alpha_{N+1}$ being its diagonal elements. The hyperparameters σ^2 and $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_{N+1}\end{bmatrix}^\top$ in the above formulas are determined by maximizing the marginal likelihood function [42, 47]

$$(\hat{\boldsymbol{\alpha}}, \hat{\sigma}^2) = \arg\max_{\boldsymbol{\alpha}, \sigma^2} \ln p(\mathbf{c}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2), \tag{17}$$

where $\hat{\boldsymbol{\alpha}}$ and $\hat{\sigma}^2$ represent the estimated $\boldsymbol{\alpha}$ and σ^2. To reduce the computational cost, the sequential sparse Bayesian learning

algorithm [48] is adopted to estimate $\hat{\boldsymbol{\alpha}}$ and $\hat{\sigma}^2$. If the estimate of α_i is infinity, the posterior distribution of the corresponding weight w_i has zero mean and zero variance, i.e., $w_i = 0$. In this case, the kernel functions associated with zero w_i do not contribute to the target output, hence are pruned. Therefore, the target output is only decided by the kernel functions corresponding to the nonzero w_i's. The inputs corresponding to these nonzero w_i's are known as Relevance Vectors (RVs).

At a future time instant t^*, the RVM based HI prediction $c_{t^*}^{\text{RVM}}$ follows the posterior distribution [42, 47]

$$p(c_{t^*}^{\text{RVM}}|t^*, \mathbf{t}, \mathbf{c}) = \mathcal{N}\left(\hat{c}_{t^*}^{\text{RVM}}, (\sigma_{t^*}^{\text{RVM}})^2\right), \tag{18a}$$

$$\hat{c}_{t^*}^{\text{RVM}} = \mu_w^\top K(\mathbf{t}, t^*), \tag{18b}$$

$$(\sigma_{t^*}^{\text{RVM}})^2 = \hat{\sigma}^2 + K(\mathbf{t}, t^*)^\top \Sigma_w K(\mathbf{t}, t^*), \tag{18c}$$

with μ_w, Σ_w, and $K(\mathbf{t}, t^*)$ defined in (16).

According to (18b) and (18c), the choice of kernel function has a significant impact on the predicted HI at a given time. Therefore, it is essential to assess the effect of different kernel functions on the RVM based prediction. Using the bearing system as an example, we compare the performance of the RVM based predictions using different kernel functions under two degradation modes. In this paper, we adopt two kernel functions as follows:

i) Polynomial kernel function $k(t, t_i) = (t - t_i)^p$;

ii) Exponential kernel function $k(t, t_i) = \exp\left(\frac{|t-t_i|}{\lambda^2}\right)$.

Figure 4 illustrates the effect of polynomial and exponential kernels under different degradation modes. Under the slow mode, the polynomial kernel performs better than the exponential kernel. However, under the rapid degradation mode, the exponential kernel outperforms the polynomial kernel. Therefore, it is both necessary and effective to choose a suitable kernel function for a specific degradation mode. Different degradation modes are distinguished via offline mode clustering in Section 2.1 and online mode identification in Section 2.2.

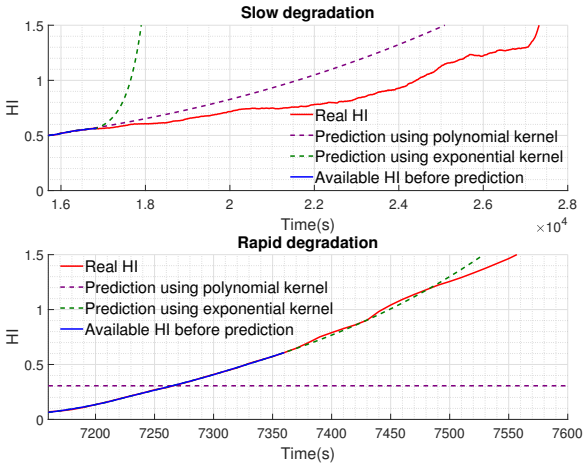

Figure 4: Comparison of RVM based predictions using different kernel functions under different degradation modes

For each degradation mode, we establish an appropriate kernel library by including a set of appropriate kernel functions of different types or parameters. To this end, we first design a collection of kernels, including polynomial and exponential kernels, covering a wide range of degrees and widths. A wider range of degrees or widths generally implies a higher expressive power of the kernel library. Subsequently, these kernel functions are employed to predict the RUL of the trajectories in our trajectory libraries. Finally, for each HT, we include the first m kernel functions with the smallest sum of errors in the kernel library corresponding to its mode. If a trajectory library contains n HTs, the resulting kernel library for that mode will consist of no more than mn kernel functions.

For online prediction at time t, we first select the optimal prediction kernel from the corresponding kernel library by evaluating the root mean square error (RMSE) Δ of the online trajectory over the sliding window $[t - t_{\text{on}} + 1, t]$

$$\Delta = \sqrt{\frac{1}{t_{\text{on}}} \sum_{i=t-t_{\text{on}}+1}^{t} (c_i^{\text{on}} - \hat{c}_i^{\text{RVM}})^2}, \tag{19}$$

where c_i^{on} is the HI of online trajectory and $\hat{c}_i^{\text{RVM}}$ represents the RVM based prediction at time i. By doing so, the kernel function with the minimum RMSE in the corresponding kernel library is used in online prediction.

2.4. Mode-dependent similarity based HI prediction

The conventional similarity based method suffers from dissimilar HTs [39]. To address this issue, we leverage the identified degradation mode in Section 2.2, and perform the similarity based RUL prediction using only the trajectory library corresponding to the identified degradation mode of the online trajectory. By doing so, we reduce the interference of dissimilar HTs in the other degradation modes.

In Section 2.2, we determine the mode κ of online trajectory and the corresponding trajectory library $\{\mathcal{T}_j^\kappa\}_{j=1}^{S_\kappa}$. To compute the similarity between the online trajectory and a particular HT $\mathcal{T}_l^\kappa$, we need to shift the online trajectory step by step towards the end of $\mathcal{T}_l^\kappa$. Specifically, we compute the distance between two vectors of HI values, i.e.,

$$d_p^l = \sqrt{\left\|\mathbf{c}_t^{\text{on}} - \mathbf{c}_p^l\right\|_2^2}, \quad t_{\text{on}} \le p \le t_{end}^l, \tag{20}$$

where t_f^l represents the failure time of the HT $\mathcal{T}_l^\kappa$,

$$\mathbf{c}_t^{\text{on}} = \left[c_{t-t_{\text{on}}+1}^{\text{on}} \quad \cdots \quad c_t^{\text{on}}\right]^\top \tag{21}$$

is the HI values of the online trajectory over a recent time window $[t - t_{\text{on}} + 1, t]$ of length t_{on}, and

$$\mathbf{c}_p^l = \left[c_{p-t_{\text{on}}+1}^l \quad \cdots \quad c_p^l\right]^\top \tag{22}$$

over a different time window $[p - t_{\text{on}} + 1, p]$ moves along the HT $\mathcal{T}_l^\kappa$ as p varies. For the considered HT $\mathcal{T}_l^\kappa$, we find its best-matching segment which gives the shortest distance to the online trajectory, as depicted in Figure 5. For each HT $\mathcal{T}_l^\kappa$, the

best-matching segment ends at the point

$$q_l = \arg\min_{t_{\text{on}} \le p \le t_f^l} d_p^l. \tag{23}$$

With $\{q_l\}_{l=1}^{S_\kappa}$ computed for all HTs within the trajectory library of the identified mode κ, we predict HI at a future time t^* as the weighted average of the HTs of the identified mode κ, i.e.,

$$c_{t^*}^{\text{TS}} = \sum_{l=1}^{S_\kappa} w_l c_{t^*-t+q_l}^l, \tag{24}$$

where $c_{t^*}^{\text{TS}}$ represents the similarity based HI prediction at time t^*, $c_{t^*-t+q_l}^l$ is the HI value of the l^{th} HT at time $t^* - t + q_l$ that matches with the predicted HI $\hat{c}_{t^*}^{\text{TS}}$, and w_l is the weight assigned to the l^{th} HT. S_κ is the number of HTs in the trajectory library corresponding to the identified degradation mode κ of the online trajectory. If the recorded l^{th} HT ends before the future time $t^* - t + q_l$, it does not have the HI value $c_{t^*-t+q_l}^l$ used in (24). To circumvent this issue, we perform linear extrapolation to determine $c_{t^*-t+q_l}^l$ for the l^{th} HT.

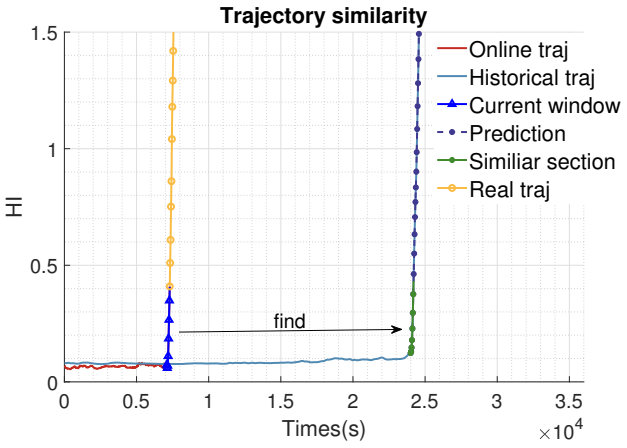

Figure 5: Illustration of trajectory similarity

To determine weights in (24), we formulate a non-negative least squares (NNLS) problem which is expressed as [49, 50]

$$\min_{\mathbf{w} \ge 0} \left\| \mathbf{C}\mathbf{w} - \mathbf{c}_t^{\text{on}} \right\|_2^2, \tag{25}$$

where $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_{S_\kappa} \end{bmatrix}^\top$ is the weight vector, $\mathbf{C} = \left[\mathbf{c}_{q_1}^1 \mathbf{c}_{q_2}^2 \cdots \mathbf{c}_{q_{S_\kappa}}^{S_\kappa} \right]$ is a matrix of size $t_{\text{on}} \times S_\kappa$ with $\mathbf{c}_p^l$ defined in (22), and $\mathbf{c}_t^{\text{on}}$ is defined in (21).

2.5. Bayesian ensemble for RUL prediction

In this section, we fuse the aforementioned two HI predictions within a Bayesian ensemble scheme to predict RUL. Firstly, we adopt a Bayesian regression approach to compute the fused HI prediction with uncertainty quantification. Then, the estimated RUL and its uncertainty interval are determined by comparing the fused HI prediction with the predefined failure threshold.

To fuse the RVM based HI prediction c_t^{RVM} in (18) and the similarity based HI prediction c_t^{TS} in (24) at time t, the following Bayesian regression model is used:

$$c_t = \eta^\top z_t + v_t, \; z_t = \begin{bmatrix} c_t^{\text{RVM}} \\ c_t^{\text{TS}} \end{bmatrix}, \tag{26}$$

where the model output is the fused HI prediction c_t, η is the model coefficient vector, and v_t represents the output noise. As a common practice in Bayesian regression, we assume that both the prior distribution of η and the noise distribution of v_t are Gaussian with zero mean, i.e.,

$$p(\eta|\boldsymbol{\alpha}_\eta) = \mathcal{N}\left(\eta \left| 0, \boldsymbol{\alpha}_\eta^{-1} \right.\right), \; p(v_t) = \mathcal{N}\left(v_t \left| 0, \beta_v^{-1} \right.\right), \tag{27}$$

with the diagonal matrix $\boldsymbol{\alpha}_\eta \in \mathbb{R}^{2\times 2}$ and the scalar β_v in (27) being the hyperparameters.

Before computing the fused HI prediction, we first determine the hyperparameters $\boldsymbol{\alpha}_\eta$ and β_v in (27) by maximizing the marginal likelihood function, i.e.,

$$(\hat{\boldsymbol{\alpha}}_\eta, \hat{\beta}_v) = \arg\max_{\boldsymbol{\alpha}_\eta, \beta_v} \ln p\left(\mathbf{c}_t^{\text{on}} \left| \mathbf{Z}_t, \boldsymbol{\alpha}_\eta, \beta_v \right.\right), \tag{28}$$

using the output data $\mathbf{c}_t^{\text{on}}$ defined in (21) and the input data

$$\mathbf{Z}_t = \begin{bmatrix} \hat{c}_{t-t_{\text{on}}+1}^{\text{RVM}} & c_{t-t_{\text{on}}+1}^{\text{TS}} \\ \vdots & \vdots \\ \hat{c}_t^{\text{RVM}} & c_t^{\text{TS}} \end{bmatrix} \tag{29}$$

over a recent time window $[t - t_{\text{on}} + 1, t]$, with $\hat{c}_t^{\text{RVM}}$ being the predicted mean in (18b) from the RVM based method. More details for formulating the above marginal likelihood maximization problem are referred to Section 3.5 in [42]. Since solving the non-convex problem (28) repeatedly at each time instant can be computationally prohibited for online RUL prediction, we adopt a suboptimal approach instead by performing grid search over a predefined range of $(\boldsymbol{\alpha}_\eta, \beta_v)$, which works well for the case studies in Sections 3 and 4.

With the learned parameters $\hat{\boldsymbol{\alpha}}_\eta$ and $\hat{\beta}_v$ in (28), the posterior distribution of the model coefficient η is derived as [42]

$$\begin{aligned} &p(\eta|\mathbf{c}_t^{\text{on}}, \mathbf{Z}_t) = \mathcal{N}(\eta \left| \mu_{\eta,t}, \Sigma_{\eta,t} \right.), \\ &\mu_{\eta,t} = \hat{\beta}_v \Sigma_\eta \mathbf{Z}_t^\top \mathbf{c}_t^{\text{on}}, \; \Sigma_{\eta,t} = (\hat{\beta}_v \mathbf{Z}_t^\top \mathbf{Z}_t + \hat{\boldsymbol{\alpha}}_\eta \mathbf{I}_2)^{-1}, \end{aligned} \tag{30}$$

with $\mathbf{c}_t^{\text{on}}$ and $\mathbf{Z}_t$ defined in (21) and (29), respectively.

Given the input z_{t^*} at a future time t^*, the Bayesian regression model (26) predicts the fused HI by its posterior predictive distribution [42]

$$\begin{aligned} p\left(c_{t^*} \left| z_{t^*}, \mathbf{c}_t^{\text{on}}, \mathbf{Z}_t \right.\right) &= \int p\left(c_{t^*} \left| z_{t^*}, \eta, \mathbf{c}_t^{\text{on}}, \mathbf{Z}_t \right.\right) p(\eta|\mathbf{c}_t^{\text{on}}, \mathbf{Z}_t) d\eta \\ &= \int \mathcal{N}\left(c_{t^*} \left| \eta^\top z_{t^*}, \hat{\beta}_v^{-1} \right.\right) \mathcal{N}\left(\eta \left| \mu_{\eta,t}, \Sigma_{\eta,t} \right.\right) d\eta \\ &= \mathcal{N}\left(c_{t^*} \left| \mu_c(z_{t^*}), \sigma_c^2(z_{t^*}) \right.\right) \end{aligned} \tag{31}$$

which is derived using (27) and (30), with

$$\mu_c(z_{t^*}) = \mu_{\eta,t}^\top z_{t^*}, \; \sigma_c^2(z_{t^*}) = \hat{\beta}_v^{-1} + z_{t^*}^\top \Sigma_{\eta,t} z_{t^*}. \tag{32}$$

It should be noted that within the input vector $z_{t^*} = \left[c_{t^*}^{\mathrm{RVM}} \quad c_{t^*}^{\mathrm{TS}}\right]^\top$, the RVM based HI prediction $c_{t^*}^{\mathrm{RVM}}$ is probabilistic, i.e., $c_{t^*}^{\mathrm{RVM}} \sim \mathcal{N}\left(c_{t^*}^{\mathrm{RVM}} \left| \hat{c}_{t^*}^{\mathrm{RVM}}, (\sigma_{t^*}^{\mathrm{RVM}})^2 \right.\right)$ given in (18), whilst the similarity based HI prediction $c_{t^*}^{\mathrm{TS}}$ is deterministic. Considering such an uncertain element $c_{t^*}^{\mathrm{RVM}}$ within the input z_{t^*}, the fused HI prediction c_{t^*} actually follows the marginal posterior predictive distribution

$$p\left(c_{t^*} \left| c_{t^*}^{\mathrm{TS}}, \mathbf{c}_t^{\mathrm{on}}, \mathbf{Z}_t \right.\right) = \int p\left(c_{t^*} \left| z_{t^*} = \begin{bmatrix} c_{t^*}^{\mathrm{RVM}} \\ c_{t^*}^{\mathrm{TS}} \end{bmatrix}, \mathbf{c}_t^{\mathrm{on}}, \mathbf{Z}_t \right.\right) \mathcal{N}\left(c_{t^*}^{\mathrm{RVM}} \left| \hat{c}_{t^*}^{\mathrm{RVM}}, (\sigma_{t^*}^{\mathrm{RVM}})^2 \right.\right) dc_{t^*}^{\mathrm{RVM}}, \tag{33}$$

which is marginalized with respect to $c_{t^*}^{\mathrm{RVM}}$. As indicated in (31) and (32), the posterior predictive distribution $p\left(c_{t^*} \left| z_{t^*}, \mathbf{c}_t^{\mathrm{on}}, \mathbf{Z}_t \right.\right) = \mathcal{N}\left(c_{t^*} \left| \mu_c(z_{t^*}), \sigma_c^2(z_{t^*}) \right.\right)$ has a complicated dependence on the uncertain input $c_{t^*}^{\mathrm{RVM}}$. Hence it is intractable to derive an analytical closed-form solution for the integral in (33). To address this issue, we resort to the Monte-Carlo based numerical approximation:

$$\begin{aligned} p\left(c_{t^*} \left| c_{t^*}^{\mathrm{TS}}, \mathbf{c}_t^{\mathrm{on}}, \mathbf{Z}_t \right.\right) &\approx \frac{1}{N_s} \sum_{i=1}^{N_s} p\left(c_{t^*} \left| z_{t^*}(i) = \begin{bmatrix} c_{t^*}^{\mathrm{RVM}}(i) \\ \hat{c}_{t^*}^{\mathrm{TS}} \end{bmatrix}, \mathbf{c}_t^{\mathrm{on}}, \mathbf{Z}_t \right.\right) \\ &= \frac{1}{N_s} \sum_{i=1}^{N_s} \mathcal{N}\left(c_{t^*} \left| \mu_c(z_{t^*}(i)), \sigma_c^2(z_{t^*}(i)) \right.\right) \end{aligned} \tag{34}$$

where $\left\{c_{t^*}^{\mathrm{RVM}}(i)\right\}_{i=1}^{N_s}$ are independent samples from the probability distribution $\mathcal{N}\left(c_{t^*}^{\mathrm{RVM}} \left| \hat{c}_{t^*}^{\mathrm{RVM}}, (\sigma_{t^*}^{\mathrm{RVM}})^2 \right.\right)$. With derivations detailed in Appendix Appendix A, we obtain the marginal posterior predictive distribution $p\left(c_{t^*} \left| c_{t^*}^{\mathrm{TS}}, \mathbf{c}_t^{\mathrm{on}}, \mathbf{Z}_t \right.\right)$ as

$$p\left(c_{t^*} \left| c_{t^*}^{\mathrm{TS}}, \mathbf{c}_t^{\mathrm{on}}, \mathbf{Z}_t \right.\right) = \mathcal{N}\left(c_{t^*} \left| \overline{\mu}_{c,t^*}, \overline{\sigma}_{\mathrm{c},\mathrm{t}^*}^2 \right.\right), \tag{35a}$$

$$\overline{\mu}_{c,t^*} = \frac{1}{N_s} \sum_{i=1}^{N_s} \mu_c(z_{t^*}(i)), \tag{35b}$$

$$\overline{\sigma}_{c,t^*}^2 = \frac{1}{N_s} \sum_{i=1}^{N_s} \sigma_c^2(z_{t^*}(i)) + \left(\mu_c(z_{t^*}(i)) - \bar{\mu}_{c,t^*}\right)^2, \tag{35c}$$

with $z_{t^*}(i)$ defined in the first equation of (34). The mean $\overline{\mu}_{c,t^*}$ in (35b) can be regarded as the point HI prediction at the future time t^*. Then, the 95% confidence interval $[\underline{c}_{t^*}, \overline{c}_{t^*}]$ of the fused HI prediction is calculated as

$$\overline{c}_{t^*} = \overline{\mu}_{c,t^*} + 1.96\overline{\sigma}_{c,t^*}, \ \underline{c}_{t^*} = \overline{\mu}_{c,t^*} - 1.96\overline{\sigma}_{c,t^*}. \tag{36}$$

When the predicted HI reaches a predefined failure threshold, we determine that the monitored system fails and can no longer work. Therefore, at the inspection time t_k the predicted RUL and its uncertainty interval $[\underline{\mathrm{RUL}}(t_k), \overline{\mathrm{RUL}}(t_k)]$ are determined as

$$\mathrm{RUL}(t_k) = \min\{t^* | \overline{\mu}_{c,t^*} \geq c_{\mathrm{th}}\} - t_k, \tag{37a}$$

$$\overline{\mathrm{RUL}}(t_k) = \min\{t^* | \underline{c}_{t^*} \geq c_{\mathrm{th}}\} - t_k, \tag{37b}$$

$$\underline{\mathrm{RUL}}(t_k) = \min\{t^* | \overline{c}_{t^*} \geq c_{\mathrm{th}}\} - t_k, \tag{37c}$$

with c_{th} being the predefined failure threshold.

When the historical data is rather limited or even unavailable, the proposed combination of RVM and trajectory similarity is expected to produce an improved prediction accuracy compared to just using a single prediction model or a deep learning model. The reasons are as follows:

- When the run-to-failure data is rather limited, the offline mode clustering of historical trajectories still works (see Remark 2), and the two predictions from RVM and trajectory similarity take complementary roles, hence their combination produces an improved performance than using any one of them alone. On one hand, the proposed ensemble method selects historical degradation trajectories with sufficient similarity to the online data such that the long-term prediction accuracy is enhanced compared to using RVM alone. On the other hand, if all historical degradation trajectories have a low level of similarity to the online data, the similarity based prediction alone is unreliable. In this case, with the above Bayesian ensemble method, the similarity based prediction would be assigned with a lower weight, while the RVM based prediction takes the dominant role in the fused prediction. The fusion weights assigned to these two predictions are automatically determined by Bayesian learning as in (28)–(30).
- When no historical run-to-failure data is available, the offline mode clustering of historical trajectories presented in Section 2.1 is omitted, and the similarity based prediction cannot work. In this case, the proposed ensemble method can still provide RVM based prediction, which leverages the online data before the current time instant to construct the RVM model.
- When the historical data is unavailable or rather limited, the proposed ensemble method still works as explained above, whilst the deep learning based method might fail to construct a reliable prediction model with such a small amount of training data.

Remark 4. *The training window length t_{on} is the number of HI samples of the test trajectory used to determine the model parameters involved in mode-dependent RVM/similarity based predictions and Bayesian regression based fusion. Selecting the training window length t_{on} involves a tradeoff as follows. With a larger length t_{on}, the training window contains more information of the global degradation dynamics, but also includes more outdated data which may not reflect the future degradation trend. On the other hand, using a smaller length t_{on}, the training phase focuses on updated information that is more relevant to future degradation, but could be sensitive to noisy variations within a rather local window. Therefore, t_{on} should be neither too large nor too small.*

3. Case study: PHM2012 bearing dataset

In this section, we use the PHM2012 bearing dataset to verify the effectiveness of the proposed algorithm. First, the dataset

Table 1: Operating conditions in the PHM2012 bearing dataset.

Operating conditions	Load (N)	Rotating speed (rpm)	Degradation trajectories
1	4000	1800	1-1, 1-2, 1-3, 1-4, 1-5, 1-6, 1-7
2	4200	1650	2-1, 2-2, 2-3, 2-4, 2-5, 2-6, 2-7
3	5000	1500	3-1, 3-2, 3-3

and the offline preparations are described. Then, the performance comparisons of different RUL prediction algorithms are illustrated in detail.

3.1. Data description and experimental setting

The PHM2012 bearing dataset [51] was utilized in the IEEE PHM 2012 Data Challenge organized by FEMTO-ST. The experimental platform consisted of two main components: the rotating part and the loading part. The rotating section comprised a motor with 250W power and a maximum speed of 2830rpm, which ensured that the second shaft rotated at a fixed speed of 2000rpm. The loading component consisted of a pneumatic jack that provided a dynamic load of 4000N on the bearing. The collected degradation data for bearings primarily include vibration and temperature measurements. Two miniature accelerometers were placed vertically on the outer ring of each bearing. Every 10 seconds, the vibration measurements are collected at a sampling frequency of 25.6kHz. We solely relied on the vibration data in this experiment. Table 1 summarizes the seventeen bearings employed, under three distinct working conditions.

To evaluate the proposed method in the case of limited data, we consider 9 scenarios using 9 degradation trajectories (i.e., 1-1, 1-2, 1-3, 1-4, 1-5, 1-6, 1-7, 2-1, 3-2) from Table 1. Each scenario selects one of the above trajectories as the test data, and uses the other 8 degradation trajectories as the training data, which is listed in Table 2.

3.2. Offline preparation

The offline preparation involves four steps, including HI extraction, HI denoising, HI normalization, and degradation mode clustering.

In this paper, the HI for the bearing is obtained from the vibration signal using the Support Vector Data Description method [52]. Specifically, it constructs a hypershere in the feature space such that most training samples in the healthy condition are located within the hypershere and a weighted sum of the hypershere radius and outliers' penalty is minimized. The HI of a given vibration feature vector is then defined to be its distance to the center of the above hypershere. The trajectory of extracted HIs generally exhibits a monotonically increasing trend.

After HI extraction, noises in the extracted HIs need to be filtered before the HI normalization step. To this end, we apply the SGolay filter [53] which utilizes local polynomial least squares fitting in the time domain. The advantage of the filter is that it effectively eliminates noise while preserving the shape of the degradation trajectory.

Following the HI denoising step, we employ (1) and (2) to normalize the HI values such that they are within the same range. In this case, we set $\mathrm{HI}_{\mathrm{def}} = 0.94$ and determine $\mathrm{mean}(\mathrm{HI}_{\mathrm{healthy}})$ to be the mean of the first 100 sampling points of each bearing.

As the last step in offline preparation, mode clustering in Section 2.1 is applied. The HTs are divided into two modes, and we construct a trajectory library for each modes. Specifically, bearings 1-1 and 1-3 belong to the slow degradation mode, while the other bearings degrade rapidly.

3.3. Results and discussions

In this subsection, the bearing HI/RUL predictions are calculated every 20 seconds, with tuning parameters as below:

- Dimension n of extracted feature vector for mode clustering: Its influence on model performance is explained in Remark 3. We choose $n = 10$ in this case study.
- Training window length t_{on} for mode-dependent RVM/similarity based prediction and Bayesian regression based fusion: Its influence on model performance is explained in Remark 4. The training window is set to include $t_{\mathrm{on}} = 20$ HI samples here.
- Hyperparameters of kernel functions used in mode-dependent RVM based prediction: A rich set of hyperparameter values in the kernel library generally implies a higher expressive power, but requires a heavier computational cost. For the rapid degradation mode, we choose exponential kernels with widths $\{22, 23, 24, \cdots, 30\}$ and polynomial kernels with degrees $\{0.1, 0.2, 0.3, \cdots, 4\}$; for the slow degradation mode, we choose exponential kernels with widths $\{5, \sqrt{50}, \sqrt{75}, \cdots, \sqrt{550}, 22, 23, \cdots, 26\}$ and polynomial kernels with degrees $\{0.5, 1, 1.5, \cdots, 20\}$.

Even though further fine tuning of the above parameters is not performed, our proposed Bayesian ensemble method achieves improved results discussed as follows.

Figure 6 shows the HI prediction results using mode-dependent RVM, trajectory similarity and Bayesian ensemble methods. It can be seen that the Bayesian ensemble prediction is more accurate than the mode-dependent RVM based prediction, due to enhancing long-term prediction accuracy by exploiting similar historical trajectories. The mode-dependent similarity based method achieves the best performance in Figure 6(a) but gives the worst performance in Figure 6(b), because the test bearing 3-2 in Figure 6(a) has relatively high similarity with some HTs, and the test bearing 1-1 in Figure 6(b) has rather low similarity with all HTs. This shows that accuracy of

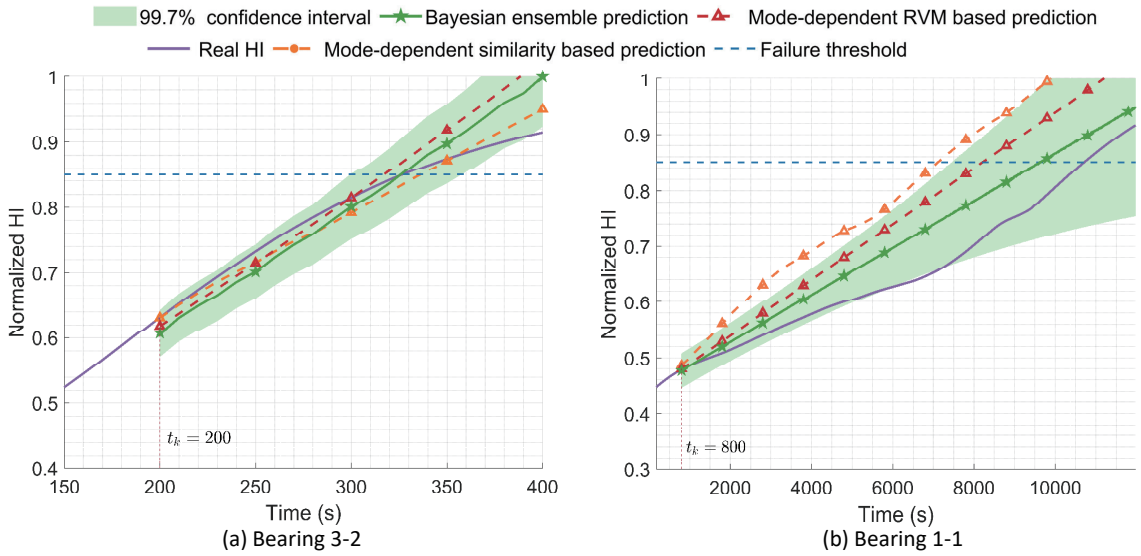

Figure 6: The HI predictions using mode-dependent RVM based, mode-dependent similarity based, and Bayesian ensemble approaches for bearings 3-2 and 1-1.

the mode-dependent similarity based method highly relies on degree of similarity between the test trajectory and HTs. In comparison, the Bayesian ensemble prediction has more advantage if the test trajectory has low similarity with HTs. As illustrated in Figure 6(b), the Bayesian ensemble prediction achieves the highest accuracy due to assigning a low belief to the poorly performed similarity based prediction.

To demonstrate the superiority of the proposed ensemble algorithm, extensive comparisons are made with the hybrid RVM [5], CNN [54], and hybrid BNN [24] based methods. Figure 7 illustrates the RUL predictions of the first three methods applied to bearings 1-1, 1-4, 1-5, 1-7, 2-1, and 2-2, respectively. In each subfigure of Figure 7, the RUL prediction starts after detecting the fault. In Figure 7, smaller deviation from the real RUL (blue solid line) implies better prediction performance. It can be seen that both the CNN and hybrid RVM based methods (blue and yellow dashed lines) give large prediction errors, which is due to not explicitly coping with limited training data. In comparison, the overall performance of our proposed Bayesian ensemble method (purple dashed line) is much better.

To quantitatively evaluate the RUL prediction accuracy of these five algorithms, relative error (RE), mean absolute percentage error (MAPE), mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R2) are used as the assessment criteria, i.e.,

$$\mathrm{RE}(t_i) = \frac{|\widehat{\mathrm{RUL}}(t_i) - \mathrm{RUL}(t_i)|}{\mathrm{RUL}(t_i)} \times 100\%, \tag{38a}$$

$$\mathrm{MAPE} = \frac{1}{N_p}\sum_{i=1}^{N_p} \mathrm{RE}(t_i), \tag{38b}$$

$$\mathrm{MAE} = \frac{1}{N_p}\sum_{i=1}^{N_p} |\widehat{\mathrm{RUL}}(t_i) - \mathrm{RUL}(t_i)|, \tag{38c}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p} \left(\widehat{\mathrm{RUL}}(t_i) - \mathrm{RUL}(t_i)\right)^2}, \tag{38d}$$

$$\mathrm{R2} = 1 - \frac{\sum_{i=1}^{N_p} \left(\widehat{\mathrm{RUL}}(t_i) - \mathrm{RUL}(t_i)\right)^2}{\sum_{i=1}^{N_p} \left(\overline{\mathrm{RUL}} - \mathrm{RUL}(t_i)\right)^2}, \tag{38e}$$

where $\widehat{\mathrm{RUL}}(t_i)$ and $\mathrm{RUL}(t_i)$ are the predicted RUL and the real RUL at the inspection time t_i, N_p is the number of observations, and $\overline{\mathrm{RUL}}$ represents the sample average of the true RULs. A lower MAPE implies higher prediction accuracy.

Table 2 and Figure 8 show the performance evaluation results of the six methods over 40% $\sim$ 90% of the bearing lifetime. As shown in Table 2, the hybrid RVM and BNN methods give the worst performance for most test bearings. Although the CNN method achieves the smallest MAE and RMSE for bearings 1-1 and 1-3, its prediction accuracy varies drastically for the other test bearings. Note that the smallest MAPE/MAE/RMSE is obtained by one of our proposed methods: mode-dependent RVM based prediction, mode-dependent similarity based prediction, and Bayesian ensemble method. The R2 index is computed over 80% $\sim$ 100% of the bearing lifetime. The hybrid RVM, CNN, and hybrid BNN methods are not listed in Table 2 because each of these methods gives negative R2 values for more than 5 test bearings. This implies that their corresponding prediction models are even worse than just using the averaged true

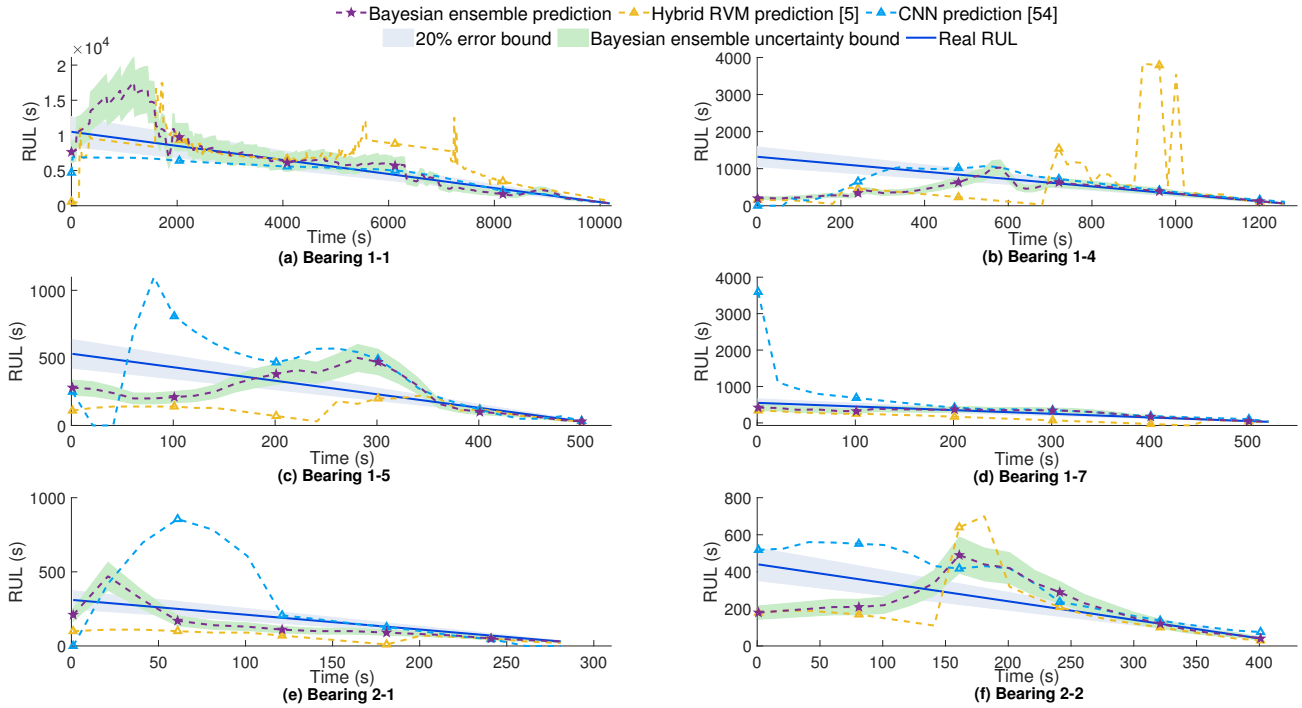

Figure 7: Comparison of RUL prediction with three different approaches for bearings 1-1, 1-4, 1-5, 1-7, 2-1, and 2-2.

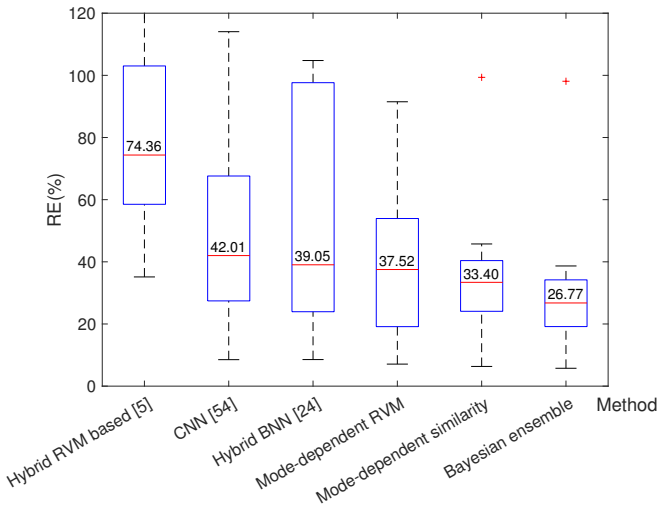

Figure 8: Boxplots reflecting the dispersion of REs of the five methods applied to the PHM2012 bearing dataset. The horizontal red line indicates the median RE.

RUL over the future time window. In comparison, our proposed Bayesian ensemble method gives positive R2 values for all test bearings, and achieves the largest averaged R2 value. It should be also be noted that Figure 8 further depicts the dispersion of REs generated by each method for all test bearings over the considered life span. It can be seen that the REs of the proposed Bayesian ensemble method have the lowest dispersion, which again illustrates the performance superiority of our proposed method.

The MAPE and RE of the proposed Bayesian ensemble method are the smallest in most cases, as expected. The superior performance of the ensemble algorithm is attributed to the following reasons:

i) By combining the similarity and RVM based predictions, the proposed Bayesian ensemble method not only addresses the limitation of RVM's inadequate long-term prediction capability by incorporating prior knowledge of similar HTs, but also mitigates performance loss of the similarity based prediction due to lack of highly similar HTs.

ii) The hybrid RVM based method in [5] suffers from high computational cost due to a wide selection range of kernel function parameters. Additionally, relying solely on RVs for degradation trend prediction, especially for long-term forecasts, may fail to capture the latest trend and result in poor RUL prediction. In contrast, the proposed method establishes mode-specific kernel libraries using HT data, significantly reducing the number of kernel functions. This computational efficiency makes our method more suitable for real-time computations compared to the hybrid approach in [5].

iii) The CNN and hybrid BNN methods in [24, 54] require massive training data to enhance generalization. However, in this scenario, the training data size is limited to 11 HTs, which may not be sufficient for stable and accurate RUL predictions. In contrast, the proposed method relies less on the amount of training data, and the trajectory similarity compensates for the long-term prediction capability of

Table 2: Performance comparisons of RUL prediction methods on PHM2012 bearing dataset. The best performance metric in each column is highlighted in bold.

Methods	Test bearings												Averaged
	1-1	1-2	1-3	1-4	1-5	1-6	1-7	2-1	2-2	2-5	3-2	3-3	
Performance metric: MAPE (%)													
Hybrid RVM [5]	66.60	61.30	96.28	163.46	35.14	80.38	78.46	55.72	42.64	109.77	70.26	131.09	82.59
CNN [54]	**8.84**	75.77	8.52	20.59	56.80	42.06	35.30	41.96	34.27	359.34	114.05	59.46	71.41
Hybrid BNN [24]	20.23	39.15	75.79	27.66	90.51	37.07	17.93	294.27	636.94	104.77	38.94	8.55	115.98
Mode-dependent RVM	32.75	45.14	34.05	**6.34**	**19.62**	45.75	**14.40**	35.65	**29.95**	99.36	34.32	28.54	35.49
Mode-dependent similarity	9.23	**31.44**	**7.09**	46.19	43.61	**18.11**	52.46	**20.21**	55.41	129.83	91.50	20.18	43.77
Bayesian ensemble	19.13	31.77	22.02	9.86	38.67	34.37	19.20	31.51	34.02	**98.10**	**21.84**	**5.75**	**30.52**
Performance metric: MAE													
Hybrid RVM [5]	239.47	15.07	224.21	71.71	9.40	9.00	15.73	7.89	10.25	14.63	4.00	14.50	52.99
CNN [54]	**32.57**	15.46	**22.59**	11.28	13.46	5.63	6.61	5.97	7.57	47.77	6.75	4.26	14.99
Hybrid BNN [24]	91.42	**7.64**	234.80	**3.72**	13.27	5.62	3.72	30.05	178.42	18.13	21.81	0.83	50.79
Mode-dependent RVM	126.19	11.21	108.62	4.53	**4.27**	5.78	**3.60**	5.11	**6.25**	**16.55**	2.40	2.33	24.74
Mode-dependent similarity	40.78	8.36	23.21	25.41	10.53	**2.22**	10.67	**2.67**	12.17	22.18	5.60	1.67	**13.79**
Bayesian ensemble	64.11	8.43	72.40	5.79	8.67	4.78	4.40	4.44	7.75	**16.55**	**1.80**	**0.50**	16.63
Performance metric: RMSE													
Hybrid RVM [5]	295.75	18.93	226.64	123.48	13.94	9.00	16.76	9.15	17.55	**20.33**	4.00	23.15	64.89
CNN [54]	**39.59**	17.69	**25.28**	13.95	17.13	6.52	6.94	13.40	9.82	63.95	6.86	5.00	18.84
Hybrid BNN [24]	167.75	**9.00**	262.67	**4.77**	16.18	6.80	**5.29**	42.45	399.12	37.28	32.64	1.08	82.08
Mode-dependent RVM	142.36	13.86	135.39	9.38	**6.54**	6.13	5.44	6.02	**8.26**	21.52	2.61	2.65	30.01
Mode-dependent similarity	57.26	11.28	34.18	30.37	13.87	**2.40**	11.37	**3.46**	15.86	29.56	5.76	2.31	**18.14**
Bayesian ensemble	73.89	11.40	99.74	9.43	12.01	5.41	5.55	5.12	10.67	22.07	**2.14**	**0.71**	21.51
Performance metric: R2													
Mode-dependent RVM	0.41	0.54	0.58	1.00	0.54	-0.58	0.95	0.01	0.84	-0.70	0.15	0.50	0.35
Mode-dependent similarity	**0.90**	**0.93**	**0.99**	0.77	**0.94**	**0.65**	-0.73	**0.67**	0.19	-0.35	-3.15	0.85	0.22
Bayesian ensemble	0.84	0.92	0.96	**0.99**	0.78	0.23	**0.73**	0.28	**0.89**	**0.80**	**0.43**	**0.90**	**0.73**

RVM, resulting in more accurate RUL predictions.

4. Case study: Toyota battery dataset

In this section, we apply the proposed Bayesian ensemble algorithm to battery RUL prediction, which further demonstrates its potential applicability to different types of engineering systems. We utilize capacity and cycle information from the Toyota battery dataset considered in [55] to complete the degenerated trajectory characterization.

4.1. Data description and experimental setting

The Toyota battery dataset includes data for 124 commercial LFP/graphite cells under 72 different fast-charging conditions. And the dataset contains measurements of charge and discharge currents, voltage, capacity and cycles. Capacity is used to quantify the state-of-health of batteries, which serves as the HI for RUL prediction.

Fig. 9 depicts the capacity degradation of batteries whose nominal capacities are 1.1Ah. The cycle life is defined as the cycle number at which the battery capacity drops to 0.88 Ah.

To evaluate RUL prediction accuracy using limited historical data, we set up a challenging scenario as follows. Firstly, 5 degradation trajectories are randomly selected for testing. Secondly, only 16 degradation trajectories including two degradation modes are used as historical data, and each degradation mode contains 8 trajectories.

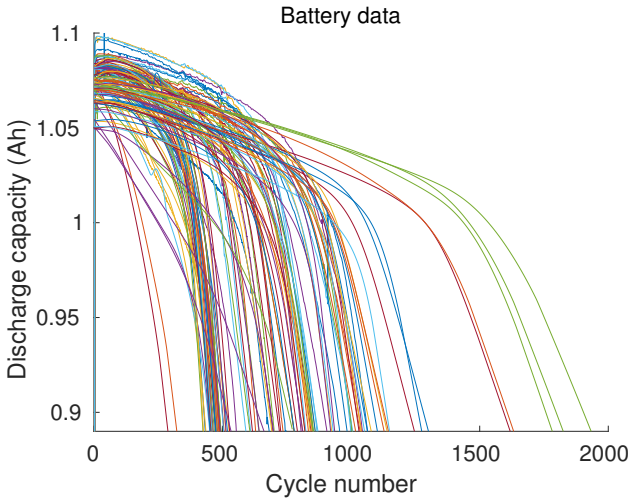

Figure 9: Discharge capacity of LFP/graphite cells

4.2. Offline preparation

We follow the procedures in Section 3.2 to process battery data offline. Since the battery capacity servers as the HI and each degradation trajectory starts at 1.1Ah and ends at 0.88Ah, the normalization step is not required. During the offline phase, mode clustering is performed, where the number of clusters is determined to be 2.

Table 3: Performance comparisons of RUL prediction methods on Tokyo battery dataset. The best performance metric in each column is highlighted in bold.

Methods	Test cells								Averaged
	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell7	Cell 8	
Performance metric: MAPE (%)									
Hybrid RVM [5]	**8.50**	20.41	**7.91**	85.94	68.48	93.00	16.67	13.96	39.36
CNN [54]	63.44	43.14	73.82	15.69	**11.90**	13.14	11.07	27.56	32.47
Hybrid BNN [24]	20.60	16.88	80.28	18.63	73.18	82.17	15.37	25.03	41.52
Mode-dependent RVM	24.05	9.09	23.78	12.46	15.80	32.81	**9.41**	21.70	18.64
Mode-dependent similarity	25.89	8.71	37.67	**10.75**	15.20	**7.33**	29.00	19.76	19.29
Bayesian ensemble	9.31	**5.73**	11.27	11.60	13.55	16.20	10.62	**14.83**	**11.64**
Performance metric: MAE									
Hybrid RVM [5]	**4.70**	11.57	**3.79**	188.13	123.27	191.54	19.05	23.92	70.75
CNN [54]	37.53	34.19	43.01	26.62	**18.86**	26.89	9.23	54.50	31.35
Hybrid BNN [24]	15.70	27.87	50.04	27.51	54.56	50.63	19.84	35.31	35.18
Mode-dependent RVM	16.70	9.11	17.43	26.54	29.28	59.36	9.99	28.02	24.55
Mode-dependent similarity	14.91	5.84	20.43	**15.20**	22.95	**13.53**	27.36	35.01	19.40
Bayesian ensemble	5.98	**3.80**	8.24	21.38	23.03	28.74	**8.72**	**21.46**	**15.17**
Performance metric: RMSE									
Hybrid RVM [5]	**5.22**	13.00	**4.38**	286.04	169.96	290.24	30.63	37.45	104.61
CNN [54]	43.17	39.93	49.39	31.40	23.05	46.15	**9.53**	72.77	39.42
Hybrid BNN [24]	18.34	39.13	67.94	33.97	65.75	62.35	24.15	60.75	46.55
Mode-dependent RVM	22.38	13.49	25.19	44.17	41.07	141.52	18.94	45.87	44.08
Mode-dependent similarity	17.07	6.27	24.85	**17.72**	**26.44**	**20.59**	39.29	54.41	25.83
Bayesian ensemble	8.88	**4.16**	13.03	31.59	29.88	52.92	12.56	**26.84**	**22.48**
Performance metric: R2									
Mode-dependent RVM	0.19	0.83	-0.08	0.70	0.65	-3.17	0.77	0.52	0.05
Mode-dependent similarity	0.53	0.96	-0.05	**0.95**	**0.85**	**0.91**	0.03	0.33	0.56
Bayesian ensemble	**0.87**	**0.98**	**0.71**	0.85	0.81	0.40	**0.90**	**0.84**	**0.80**

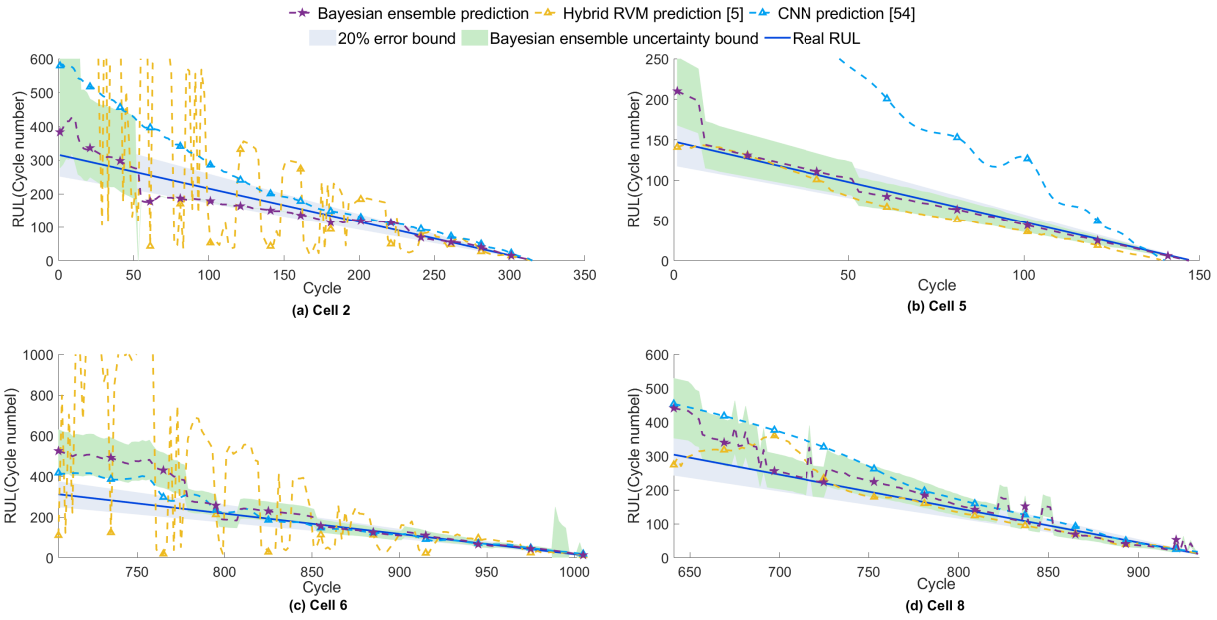

Figure 10: Comparison of RUL prediction results with three different approaches for battery cells 2, 5, 6, and 8.

4.3. Results and discussions

As describe in the first paragraph of Section 3.3, the same tuning parameters are involved. For this battery dataset, the dimension n of the extracted feature vector for mode clustering

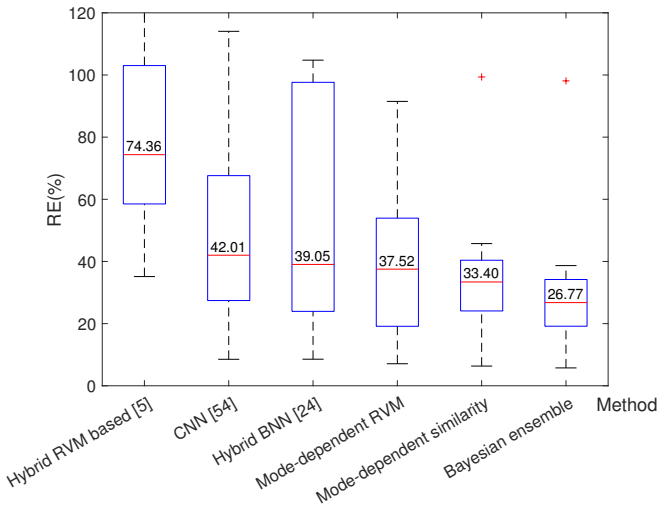

Figure 11: Boxplots reflecting the dispersion of REs of the five methods applied to the Tokyo battery dataset. The horizontal red line indicates the median RE.

and the training window length t_{on} are still set to $n = 10$ and $t_{\text{on}} = 100$, respectively. Since the clustered degradation modes of battery cells do not have distinct difference as the rapid and slow modes of the bearing degradation in Section 3, the kernel libraries of cell degradation modes are set to be the same: each kernel library includes exponential kernels with widths $\{5, \sqrt{50}, \sqrt{75}, \cdots, \sqrt{550}, 22, 23, \cdots, 30\}$ and polynomial kernels with degrees $\{0.5, 1, 1.5, \cdots, 16\}$.

Most literature start making the RUL prediction of batteries at 75% of their life. However, this specific moment is unknown a priori in practice. Therefore, we start battery RUL prediction when its capacity drops to 95% of the nominal capacity, i.e., 1.045Ah. The battery RUL is predicted every two cycles, using the hybrid RVM [5], CNN [54], hybrid BNN [24] based methods, as well as the proposed mode-dependent RVM/similarity based methods and the Bayesian ensemble method again.

As depicted in Figure 10, the proposed ensemble prediction method demonstrates superior accuracy and stability compared to the RVM-model hybrid and CNN methods in [5, 54].

The performance metrics defined in (38) are utilized to assess the prediction results, as presented in Table 3 and Figure 11. The MAPEs and REs of the implemented methods over $10\% \sim 85\%$ of the cycle life are evaluated for performance comparison. It can be seen that the CNN method in [54] and the hybrid BNN method in [24] perform poorly due to limited training data. In addition, although the hybrid RVM based method in [5] achieves the most accurate prediction for Cells 1 and 3, its prediction accuracy varies drastically among different cells, as listed in Table 3. In terms of the R2 index, the hybrid RVM, CNN, and hybrid BNN methods are not included in Table 3 again, because each of these methods gives negative R2 values for 3 test cells. In contrast, our proposed methods, especially the Bayesian ensemble approach gives the largest averaged R2 value. As depicted in Figure 11, our proposed ensemble method yields the best overall performance, i.e., consistently producing low REs with the least dispersion.

5. Conclusion

This paper proposes a novel mode-dependent RVM-similarity based Bayesian ensemble algorithm for RUL prediction. With offline mode clustering and online mode identification, mode-dependent RVM and similarity based predictions are constructed. These prediction results are later fused in a Bayesian ensemble scheme that produces a fused prediction with uncertainty quantification. In the case study, the proposed algorithm as well as three existing methods are tested using limited data, where only 11 and 16 historical R2F trajectories are used as training data in the two case studies of bearings and batteries, respectively. The superior performance achieved by our proposed method on the PHM2012 bearing dataset and Toyota battery dataset demonstrates its potential applicability to a wide range of industrial systems under limited R2F data.

Still our proposed Bayesian ensemble method has one limitation: it requires each historical degradation trajectory to cover the entire range from the healthy status to the failure point. However, due to various maintenance activities that prevent failures, it is common practice that only early-stage or partial degradation data are available. Therefore, our future work will focus on accurate RUL prediction using a limited number of incomplete degradation trajectories.

Appendix A. Derivation for the fused HI prediction in (35)

Using (34), we derive (35) as follows. Firstly, the predicted mean $\overline{\mu}_{c,t^*}$ is expressed as

$$\begin{aligned}\overline{\mu}_{c,t^*} &= \frac{1}{N_s}\sum_{i=1}^{N_s}\int c_{t^*}\mathcal{N}\left(c_{t^*}\left|\mu_c(z_{t^*}(i)),\sigma_c^2(z_{t^*}(i))\right.\right)dc_{t^*} \\ &= \frac{1}{N_s}\sum_{i=1}^{N_s}\mu_c(z_{t^*}(i))\end{aligned} \tag{A.1}$$

which gives (35b). Secondly, the predicted variance $\overline{\sigma}^2_{c,t^*}$ is derived as

$$\begin{aligned}\overline{\sigma}^2_{c,t^*} &= \int\left(c_{t^*}-\overline{\mu}_{c,t^*}\right)^2\left[\frac{1}{N_s}\sum_{i=1}^{N_s}\mathcal{N}\left(c_{t^*}\left|\mu_c(z_{t^*}(i)),\sigma_c^2(z_{t^*}(i))\right.\right)\right]dc_{t^*} \\ &= \frac{1}{N_s}\sum_{i=1}^{N_s}\int\left(c_{t^*}-\overline{\mu}_{c,t^*}\right)^2\mathcal{N}\left(c_{t^*}\left|\mu_c(z_{t^*}(i)),\sigma_c^2(z_{t^*}(i))\right.\right)dc_{t^*}.\end{aligned} \tag{A.2}$$

Then we have (35c) by substituting

$$\begin{aligned}\left(c_{t^*}-\overline{\mu}_{c,t^*}\right)^2 &= \left[\left(c_{t^*}-\mu_c(z_{t^*}(i))\right)+\left(\mu_c(z_{t^*}(i))-\overline{\mu}_{c,t^*}\right)\right]^2 \\ &= \left(c_{t^*}-\mu_c(z_{t^*}(i))\right)^2+\left(\mu_c(z_{t^*}(i))-\overline{\mu}_{c,t^*}\right)^2 \\ &\quad +2\left(c_{t^*}-\mu_c(z_{t^*}(i))\right)\left(\mu_c(z_{t^*}(i))-\overline{\mu}_{c,t^*}\right)\end{aligned} \tag{A.3}$$

into (A.2).

References

[1] M. S. Kan, A. C. Tan, J. Mathew, A review on prognostic techniques for non-stationary and non-linear rotating systems, Mechanical Systems and Signal Processing 62 (2015) 1–20.

[2] M. H. Lipu, M. Hannan, A. Hussain, M. Hoque, P. J. Ker, M. H. M. Saad, A. Ayob, A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations, Journal of Cleaner Production 205 (2018) 115–133.

[3] B. Xue, H. Xu, X. Huang, K. Zhu, Z. Xu, H. Pei, Similarity-based prediction method for machinery remaining useful life: A review, The International Journal of Advanced Manufacturing Technology 121 (2022) 1501–1531.

[4] T. Xia, Y. Dong, L. Xiao, S. Du, E. Pan, L. Xi, Recent advances in prognostics and health management for advanced manufacturing paradigms, Reliability Engineering & System Safety 178 (2018) 255–268.

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

[6] L. Cui, X. Wang, H. Wang, J. Ma, Research on remaining useful life prediction of rolling element bearings based on time-varying kalman filter, IEEE Transactions on Instrumentation and Measurement 69 (2019) 2858–2867.

[7] H. Hanachi, J. Liu, A. Banerjee, Y. Chen, A. Koul, A physics-based modeling approach for performance monitoring in gas turbine engines, IEEE Transactions on Reliability 64 (2014) 197–205.

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

[9] N. Li, Y. Lei, J. Lin, S. X. Ding, An improved exponential model for predicting remaining useful life of rolling element bearings, IEEE Transactions on Industrial Electronics 62 (2015) 7762–7773.

[10] D. Pan, J.-B. Liu, J. Cao, Remaining useful life estimation using an inverse Gaussian degradation model, Neurocomputing 185 (2016) 64–72.

[11] L. Chen, Y. Zhang, Y. Zheng, X. Li, X. Zheng, Remaining useful life prediction of lithium-ion battery with optimal input sequence selection and error compensation, Neurocomputing 414 (2020) 245–254.

[12] Y. Li, X. Huang, C. Zhao, P. Ding, A novel remaining useful life prediction method based on multi-support vector regression fusion and adaptive weight updating, ISA Transactions 131 (2022) 444–459.

[13] Y. Zhang, B. Guo, Online capacity estimation of lithium-ion batteries based on novel feature extraction and adaptive multi-kernel relevance vector machine, Energies 8 (2015) 12439–12457.

[14] R. R. Richardson, M. A. Osborne, D. A. Howey, Battery health prediction under generalized conditions using a Gaussian process transition model, Journal of Energy Storage 23 (2019) 320–328.

[15] B. Sun, Y. Li, Z. Wang, Y. Ren, Q. Feng, D. Yang, An improved inverse Gaussian process with random effects and measurement errors for RUL prediction of hydraulic piston pump, Measurement 173 (2021) 108604.

[16] J. Wu, K. Hu, Y. Cheng, H. Zhu, X. Shao, Y. Wang, Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network, ISA Transactions 97 (2020) 241–250.

[17] J. Liu, F. Lei, C. Pan, D. Hu, H. Zuo, Prediction of remaining useful life of multi-stage aero-engine based on clustering and LSTM fusion, Reliability Engineering & System Safety 214 (2021) 107807.

[18] A. Kumar, C. Gandhi, Y. Zhou, G. Vashishtha, R. Kumar, J. Xiang, Improved CNN for the diagnosis of engine defects of 2-wheeler vehicle using wavelet synchro-squeezed transform (WSST), Knowledge-Based Systems 208 (2020) 106453.

[19] B. Yang, R. Liu, E. Zio, Remaining useful life prediction based on a double-convolutional neural network architecture, IEEE Transactions on Industrial Electronics 66 (2019) 9521–9530.

[20] W. Yu, I. Y. Kim, C. Mechefske, Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme, Mechanical Systems and Signal Processing 129 (2019) 764–780.

[21] T. Wang, D. Guo, X.-M. Sun, Remaining useful life predictions for turbofan engine degradation based on concurrent semi-supervised model, Neural Computing and Applications 34 (2022) 5151–5160.

[22] F. Zeng, Y. Li, Y. Jiang, G. Song, A deep attention residual neural network-based remaining useful life prediction of machinery, Measurement 181 (2021) 109642.

[23] H. Li, W. Wang, Z. Li, L. Dong, Q. Li, A novel approach for predicting tool remaining useful life using limited data, Mechanical Systems and Signal Processing 143 (2020) 106832.

[24] J. Liang, H. Liu, N.-C. Xiao, A hybrid approach based on neural network and double exponential model for remaining useful life prediction, Expert Systems with Applications (2024) 123563.

[25] T. Wang, Trajectory Similarity based Prediction for Remaining Useful Life Estimation, Ph.D. thesis, University of Cincinnati, 2010.

[26] Z. Que, Z. Xu, A data-driven health prognostics approach for steam turbines based on XGBoost and DTW, IEEE Access 7 (2019) 93131–93138.

[27] H. Cai, X. Jia, J. Feng, W. Li, L. Pahren, J. Lee, A similarity based methodology for machine prognostics by using kernel two sample test, ISA Transactions 103 (2020) 112–121.

[28] H. Cai, J. Feng, W. Li, Y.-M. Hsu, J. Lee, Similarity-based particle filter for remaining useful life prediction with enhanced performance, Applied Soft Computing 94 (2020) 106474.

[29] X. Jia, Data Suitability Assessment and Enhancement for Machine Prognostics and Health Management Using Maximum Mean Discrepancy, Ph.D. thesis, University of Cincinnati, 2018.

[30] X. Jia, H. Cai, Y. Hsu, W. Li, J. Feng, J. Lee, A novel similarity-based method for remaining useful life prediction using kernel two sample test, in: Proceedings of the Annual Conference of the PHM Society, volume 11, 2019.

[31] J. Lyu, R. Ying, N. Lu, B. Zhang, Remaining useful life estimation with multiple local similarities, Engineering Applications of Artificial Intelligence 95 (2020) 103849.

[32] M. Gu, Y. Chen, Two improvements of similarity-based residual life prediction methods, Journal of Intelligent Manufacturing 30 (2019) 303–315.

[33] S. Zhou, X. Xu, Y. Xiao, W. Chang, S. Qian, X. Pan, Remaining useful life prediction with similarity fusion of multi-parameter and multi-sample based on the vibration signals of diesel generator gearbox, Entropy 21 (2019) 861.

[34] C.-G. Huang, H.-Z. Huang, W. Peng, T. Huang, Improved trajectory similarity-based approach for turbofan engine prognostics, Journal of Mechanical Science and Technology 33 (2019) 4877–4890.

[35] L. Cui, X. Wang, H. Wang, H. Jiang, Remaining useful life prediction of rolling element bearings based on simulated performance degradation dictionary, Mechanism and Machine Theory 153 (2020) 103967.

[36] Z. Liang, J. Gao, H. Jiang, X. Gao, Z. Gao, R. Wang, A similarity-based method for remaining useful life prediction based on operational reliability, Applied Intelligence 48 (2018) 2983–2995.

[37] T. Xia, J. Shu, Y. Xu, Y. Zheng, D. Wang, Multiscale similarity ensemble framework for remaining useful life prediction, Measurement 188 (2022) 110565.

[38] Y. Yang, Y. Guo, Z. Huang, N. Chen, L. Li, Y. Jiang, N. He, Research on the milling tool wear and life prediction by establishing an integrated predictive model, Measurement 145 (2019) 178–189.

[39] Y. Liu, X. Hu, W. Zhang, Remaining useful life prediction based on health index similarity, Reliability Engineering & System Safety 185 (2019) 502–510.

[40] T. Xia, Y. Song, Y. Zheng, E. Pan, L. Xi, An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation, Computers in Industry 115 (2020) 103182.

[41] M. Hou, D. Pi, B. Li, Similarity-based deep learning approach for remaining useful life prediction, Measurement 159 (2020) 107788.

[42] C. M. Bishop, Pattern recognition and machine learning, Springer, 2006.

[43] J. Singh, A. Darpe, S. P. Singh, Bearing remaining useful life estimation using an adaptive data-driven model based on health state change point identification and K-means clustering, Measurement Science and Technology 31 (2020) 085601.

[44] J. Singh, A. Darpe, S. Singh, Bearing damage assessment using Jensen-Rényi divergence based on EEMD, Mechanical Systems and Signal Processing 87 (2017) 307–339.

[45] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63 (2001) 411–423.

[46] A. M. Bagirov, R. M. Aliguliyev, N. Sultanova, Finding compact and well-separated clusters: Clustering using silhouette coefficients, Pattern Recognition 135 (2023) 109144.

[47] M. E. Tipping, Sparse Bayesian learning and the relevance vector ma-

chine, Journal of Machine Learning Research 1 (2001) 211–244.

[48] M. E. Tipping, A. C. Faul, Fast marginal likelihood maximisation for sparse Bayesian models, in: Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 2003, pp. PMLR R4:276–283.

[49] R. Bro, S. De Jong, A fast non-negativity-constrained least squares algorithm, Journal of Chemometrics 11 (1997) 393–401.

[50] S. Foucart, D. Koslicki, Sparse recovery by means of nonnegative least squares, IEEE Signal Processing Letters 21 (2014) 498–502.

[51] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, C. Varnier, PRONOSTIA: An experimental platform for bearings accelerated degradation tests., in: Proceedings of IEEE International Conference on Prognostics and Health Management, 2012, pp. 1–8.

[52] T. Benkedjouh, K. Medjaher, N. Zerhouni, S. Rechak, Fault prognostic of bearings by using support vector data description, in: Proceedings of 2012 IEEE Conference on Prognostics and Health Management, 2012, pp. 1–7.

[53] S. J. Orfanidis, Introduction to Signal Processing, Prentice-Hall, Inc., 1995.

[54] G. Sateesh Babu, P. Zhao, X.-L. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, in: Proceedings of the 21st International Conference on Database Systems for Advanced Applications, 2016, pp. 214–228.

[55] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis, et al., Data-driven prediction of battery cycle life before capacity degradation, Nature Energy 4 (2019) 383–391.