Ensembles of Probabilistic LSTM Predictors and Correctors for Bearing Prognostics Using Industrial Standards

Venkat P. Nemani[1], Hao Lu[1], Adam Thelen[1], Chao Hu[1,2], and Andrew T. Zimmerman[3,4]

[1]Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA

[2]Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA

[3]Percēv LLC, Davenport, IA 52807, USA

[4]Grace Technologies, Davenport, IA 52807, USA

Abstract

Probabilistic prediction of the remaining useful life (RUL) of bearings is critically important, especially in an industrial setting where unplanned maintenance needs, unscheduled equipment downtime, or catastrophic failures can cost a company millions of dollars and threaten worker safety. Current research in the field of bearing prognostics clearly shows the advantage of a deep learning-based solution, but the reliability of purely data-driven predictions is questionable in harsh industrial environments with varying operational conditions. To make this work industrially relevant, we adopt ISO guidelines to determine bearing failure thresholds (specifically ISO 10816), which are defined in the velocity domain, while considering characteristic bearing fault frequencies defined by the geometry of each bearing. We propose a two-stage Long Short-Term Memory (LSTM) model ensemble which includes: (1) a predictor step to forecast and (2) a corrector step to offset the RUL prediction. Each LSTM model within the ensemble is customized to include a Gaussian layer that captures the aleatoric uncertainty in the forecasted parameter, and the ensemble of all the individual LSTM models provides the epistemic uncertainty in the RUL prediction. We demonstrate the implementation of the proposed model on the publicly available Xi'an Jiaotong University and Changxing Sumyoung Technology Co., Ltd. (XJTU-SY) bearing dataset and establish the superiority of the model, both in terms of accuracy as well as uncertainty quantification, when compared against other commonly used techniques in the field of bearing prognostics. The ensemble model tends to explore multiple functional/forecast modes providing better uncertainty estimates when compared to Bayesian counterparts.

Keywords: Bearing prognostics, LSTM, time series forecasting, probabilistic prediction, ensemble method.

1. Introduction

Prognostics and health management (PHM) technology has been receiving wide attention in recent years because of its potential to help reduce machine downtime, avoid catastrophic failure, and improve overall system reliability [1]–[3]. In the industrial environment, rolling element bearings are a predominant focus of PHM because of their presence in the rotating component of almost any critical piece of machinery [4]–[6]. The primary purpose of bearings is to reduce the rotational friction between multiple rotating parts while holding them in place. In an industrial setting, the bearings are often continuously operated under radial and/or axial loads and any catastrophic bearing failure may severely affect not just the bearing but also other connected components and/or

processed outputs, leading to costly downtime and equipment replacement. Therefore, detection of bearing faults [7], [8] and predicting the remaining useful life (RUL) of the bearings with a certain degree of confidence can empower the maintenance engineer to schedule maintenance well before bearing failure.

Predicting the RUL of bearings has typically been approached in one of two ways: (1) by using a model-based approach where bearing failure mechanisms are modeled using mathematical constructs and (2) by using a data-driven approach where the failure data of a previous set of bearings will be used to train an offline model. In both cases, the generated model can be used to predict the RUL of a similar bearing at a given point in time.

A micro-level model-based approach to RUL prediction requires prior knowledge of a bearing's failure mechanisms and their explicit modeling [9]. This level of understanding of the physics of bearing degradation can lead to very accurate RUL estimates, but modeling extremely non-linear failure mechanisms, such as excessive loading, breakdown of lubrication, contamination, and bearing currents [10], along with the wide variation in bearing operating conditions, can severely limit the application of model-based approaches. On the other hand, a macro-level model-based prognostic approach includes simplification of the represented system by defining a certain relationship between the input variables, the state variables, and the system output. Previous research in this domain includes the use of the Kalman filter (and its derivatives) [11]–[17] and particle filter (PF) [18]–[20]. Notably, Singleton et al. [11] uses an exponential form state equation to predict the bearing RUL using extended Kalman filter. Li et al. [19] has proposed an improved exponential model where the first prediction time is adaptively determined, and the PF technique is used to reduce the errors associated with the stochastic noise. Qian et al. [20] combines two-time scales by integrating phase space warping and a Paris crack growth model with PF to effectively predict the bearing RUL.

Data-driven approaches do not require prior knowledge about bearing failure mechanisms and can provide an estimate of bearing RUL that grows in credibility as more learning data is collected. However, the accuracy of the data-driven approach is heavily dependent on the amount of failure data available and is subject to typical reliability issues (such as overfitting) that present themselves frequently in modern data science. Machine learning techniques such as artificial neural networks (ANNs) [21]–[24], support/relevance vector machine (S/RVM) [25]–[29] are a few data-driven approaches often used in this domain of research. Recently, deep learning techniques are becoming more prominent due to their learning capability at multiple levels [30]. Among these, convolutional neural networks (CNN) [31]–[35] and recurrent neural networks (RNN) [23], [24] are gaining increased popularity due to their ability to store temporal information, which can be particularly useful in predicting the bearing health condition. Guo et al. [24] was the first to construct a bearing health indicator based on a feature selection criterion and used the health indicator to train a recurrent neural network (RNN). Wang et al. [36] developed a new framework of recurrent convolutional neural networks (RCNN) combined with variational inference to determine probabilistic RUL prediction. Peng et al. [37] proposed a Bayesian deep-learning-based method for uncertainty quantification in the field of prognostics.

The long short-term memory (LSTM) architecture is a special class of RNN that has the ability to store long-term feature dependencies, and it is also being explored for prognostic applications

[38]–[41]. Mao et al. [42] used CNN to extract bearing degradation features which are then fed into an LSTM model for RUL prediction. Although many of these deep learning methods show promising results, these models often consist of a large number of parameters, requiring extensive computational resources and time even for making predictions, particularly if Bayesian methods are involved for uncertainty quantification. The scalability of such models, especially in an embedded industrial internet of things (IIoT) platform or soft-sensor applications [43], is not clear. To this end, we attempt to advance the current state-of-the-art in bearing prognostics in the following ways:

1) We use the International Organization of Standardization (ISO 10816) set standards for industrial machines to determine the end of life (EOL) for bearings as opposed to traditional heuristic approaches of using maximum or mean vibration amplitude. The ISO standards, which often evaluate excessive vibration in terms of velocity units like inches per second (ips), define "excessive" vibration from an industrial standpoint which could be quite different from what is seen in a lab-based experiment. Particularly, a lab-based experiment can allow for a catastrophic bearing failure but this is not the case in an industrial setting where a catastrophic failure can cost millions of dollars.
2) We extract velocity domain root mean square (RMS) features while accounting for characteristic bearing fault frequencies. These features are then used to determine the first prediction time (FPT) and to train the proposed model. Simultaneously, we also extract features from both the time and frequency domains of acceleration, velocity, and jerk vibration signals, which are used to train other correlation-based models, such as CNNs, for comparison. Similar to the approach presented in Ref. [24], a total of twenty-four features are selected based on their Pearson correlation coefficient and monotonicity which indicates the variation of the bearing health condition with time.
3) We develop a simple and scalable ensemble of lightweight deep LSTM networks (EnLSTM) that can provide a probabilistic prediction of RUL. As opposed to complex and heavy parameter deep learning models, our proposed model uses multiple lightweight models to enable embeddability on vibration measuring sensors for online prognostics of bearing failure. A simple data augmentation technique is used during the training phase of the LSTM networks to improve the accuracy and robustness of RUL prediction.
4) We propose a two-step algorithm consisting of (a) a predictor step (EnPLSTM): which forecasts a selected feature to a certain threshold for an initial RUL prediction, (b) a corrector step (EnCLSTM): which corrects the prediction of the EnPLSTM, and (c) temporal fusion: which weighs in the predictions from the recent past to make a smoother final prediction. Each individual LSTM model provides aleatoric uncertainty of predictions through the use of a custom Gaussian layer. These LSTM models, when combined to form an ensemble, can help estimate the epistemic uncertainty in bearing health forecasts. This method ensures robust RUL predictions that are less sensitive to measurement noise and also provide consistent predictions that do not vary vastly between successive measurements.
5) Several metrics for quantifying uncertainty are used to compare the proposed model with other probabilistic methods such as optimized PF and Bayesian-like Monte-Carlo (MC) Dropout [44]. We also investigate how the ensemble EnPLSTM model works by demonstrating that the training of each individual PLSTM model takes a different optimization route. We show how exploring the functional/forecast modes provides a better measure of uncertainty.

The rest of the paper is organized as follows. In section 2, we first formally introduce the overall methodology followed by a discussion on relevant feature extraction while considering characteristic bearing fault frequencies. Following this, we present an LSTM architecture with the inclusion of a Gaussian layer to account for aleatoric uncertainty in the feature forecast and use this to develop the EnPLSTM and EnCLSTM. In section 3, we implement the proposed model on a publicly available bearing dataset from accelerated degradation experiments and establish the superiority of our model when compared to other deterministic as well as probabilistic contemporary models. Particularly in section 3, we reason why the ensemble predictor and corrector work well for the bearing dataset by identifying scenarios where the model would fail.

2. Methodology

In the introduction, we established that probabilistic RUL prediction of rolling element bearings is critically important for scheduling maintenance. In this section, we describe the technical approach for achieving confident RUL predictions. We first describe the features extracted from the vibration signals and then detail the proposed EnP/CLSTM algorithm after which we briefly present model-based approaches like particle filter, similarity, and exponential/quadratic regression, and a CNN data-driven approach. The detailed flow chart of the proposed bearing prognostic algorithm is shown in Figure *1* for the proposed architecture. In a later section, we show the advantage of the proposed method by using a case study of a run-to-failure bearing dataset.

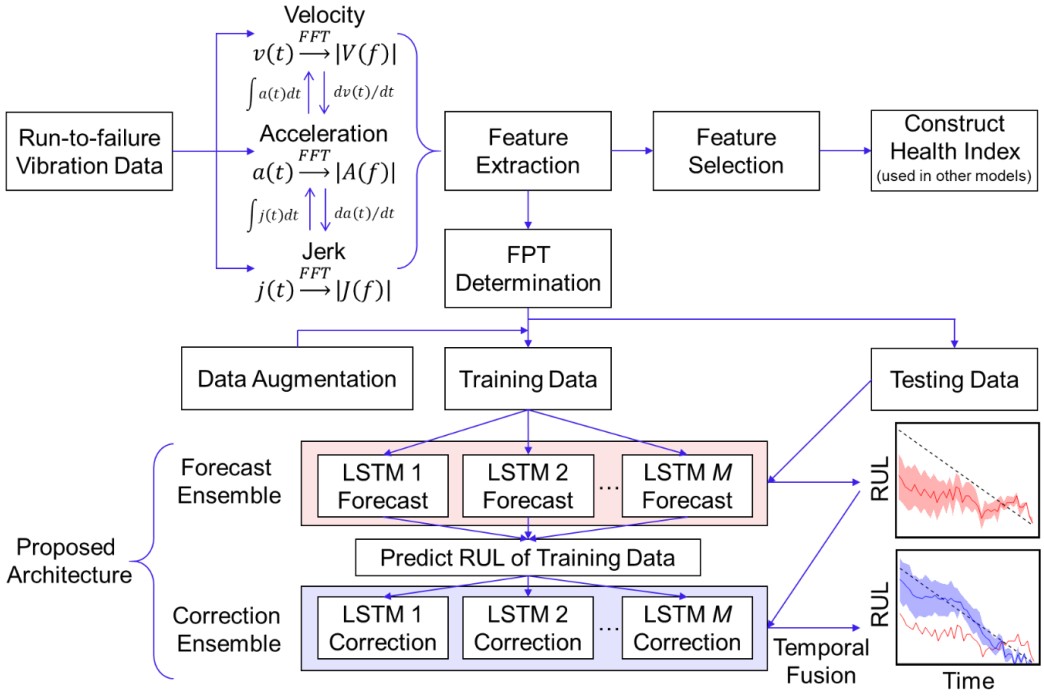

Figure 1: Schematic of the proposed bearing prognostic algorithm

The state of bearing health is often captured through vibration measurements collected in the radial direction. Bearing defects can usually be classified as either (1) single-point defects or (2) generalized roughness [45]. The former type of defect is localized, such as a spall or a pit, on an otherwise smooth bearing surface, producing four different characteristic fault frequencies.

Generalized roughness arises when larger areas of the bearing component surfaces become coarse, irregular, or deformed.

In this study, we limit our observations to single/multi-point defects, where the characteristic fault frequencies [46], [47] are functions of rotational speed and can be obtained for flaws in the outer race $BPFO = \frac{\omega N}{2}\left(1 - \frac{B}{P}\cos\phi\right)$, inner race $BPFI = \frac{\omega N}{2}\left(1 + \frac{B}{P}\cos\phi\right)$, on one of the ball bearings $BSF = \frac{\omega P}{2B}\left(1 - \frac{B^2}{P^2}\cos^2\phi\right)$, or in the cage $FTF = \frac{\omega}{2}\left(1 - \frac{B}{P}\cos\phi\right)$. Here ω is the shaft rotational speed in Hz, B is the ball diameter, P is the pitch diameter, ϕ is the contact angle, and N is the number of balls. A bearing with a particular defect shows harmonics of the corresponding fault frequency, and discrepancies arise whenever there is slippage. Moreover, when the fault is sufficiently pronounced, the vibrations are accompanied by sidebands around these characteristic frequencies. We, therefore, consider a frequency band around each fault frequency (see Figure 5) to capture the fault signatures.

2.1 Feature Extraction in Velocity Domain

Most academic bearing run-to-failure datasets provide vibration data in the acceleration domain whereas the ISO standards for defining end-of-life or alarm amplitudes are in the velocity domain [48]–[50]. This is because the magnitude of a signal in the acceleration domain increases with the frequency of that signal, whereas velocity provides a more stable representation of energy that is independent of shaft or rotational speed. Moreover, the vibration in the velocity domain is less susceptible to amplifier overloads that typically show up in the high-frequency domain which can compromise the fidelity of low-frequency signals [51]. To this end, we propose the bearing be considered unusable or require immediate maintenance if the overall velocity RMS in the frequency range of $0.2\omega - 12.8$ kHz (for a sampling frequency of $sf = 25.6$ kHz) for a single-sided fast Fourier transform (FFT) spectrum exceeds a certain threshold. According to the ISO standards [50], the threshold value varies with the type of application, but we choose a statistical value of 0.3 ips assuming a medium-sized motor [48]. In situations where vibration sensors are mounted both horizontally and vertically along the radial direction (see Figure *2*), we define the bearing to reach its end-of-life when the RMSs of both the horizontal and vertical velocities exceed 0.3 ips.

In addition to the velocity and acceleration domains, studying the jerk domain, which is the differential of the acceleration vibration signal, can be important to detect abnormal vibration signals, particularly at low rotational speeds [52], [53]. Although the case study which we present later employs a moderate operating speed, we nevertheless find and show later in section 2.2 that the features extracted from the jerk domain show good correlations with RUL for the bearings.

In practice, accelerometers are widely used due to their availability, small form factor, and low cost as opposed to velocity sensors which are expensive and bulky. Unless directly measured, the velocity vibration $v(t)$ can be obtained by numerical integration of the acceleration vibration signal, $v(t) = \int_0^t a(t)dt$, and the jerk signal can be obtained by differentiating the same, $j(t) = \frac{da(t)}{dt}$. After integration, the vibration signal will be modulated with a low-frequency signal as a numerical artifact stemming from the assumption that the initial condition for integration is

$v(t=0)=0$. To avoid this effect, we consider the frequency signal beyond 0.2ω. One can use a high-pass filter or just extract the RMS values from the frequency domain (for all three signals $a(t), v(t)$ and $j(t)$) within certain frequency ranges using Parseval's theorem [54] which is based on the principle of energy conservation. In particular, the RMS of a signal $x(t)$ can be calculated both in the time domain and based on a single-sided frequency spectrum $X(f)$ with frequency resolution of df as:

$$x^{\mathrm{RMS}} = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t} x(i)^2} = \sqrt{|X(0)| + \sum_{f=df}^{sf/2} \frac{|X(f)|^2}{2}} \qquad 1$$

where n_t is the total number of points in the time domain signal during the sampling period of t_s and $n_t = t_s \times sf$. The RMS value between two frequencies f_1 and f_2 can therefore be calculated as

$$x^{\mathrm{RMS}}_{f_1-f_2} = \sqrt{\sum_{f=f_1}^{f_2} \frac{|X(f)|^2}{2}} \qquad 2$$

Note that the summation is over the discrete $X(f)$ values between f_1 and f_2 and FFT hereafter refers to the single-sided FFT spectrum. In this study, we use two physics-based features extracted from the velocity domain: $V^{\mathrm{RMS}}_{BFF-sf/2}$ and $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$, where BFF refers to beginning of bearing fault frequencies $BFF = 0.9\min(BPFO, BPFI, BSF) - sf/2$. The pre-factor of 0.9 ensures a 10% frequency error margin for the onset of bearing degradation owing due to shaft speed variations. $V^{\mathrm{RMS}}_{BFF-sf/2}$ is used to determine the FPT for bearing prognostics (see Appendix B) and $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ is used to determine whether a bearing has failed or requires immediate maintenance based on ISO standards, which is approximately 0.3 ips for a medium-sized electric motor [48].

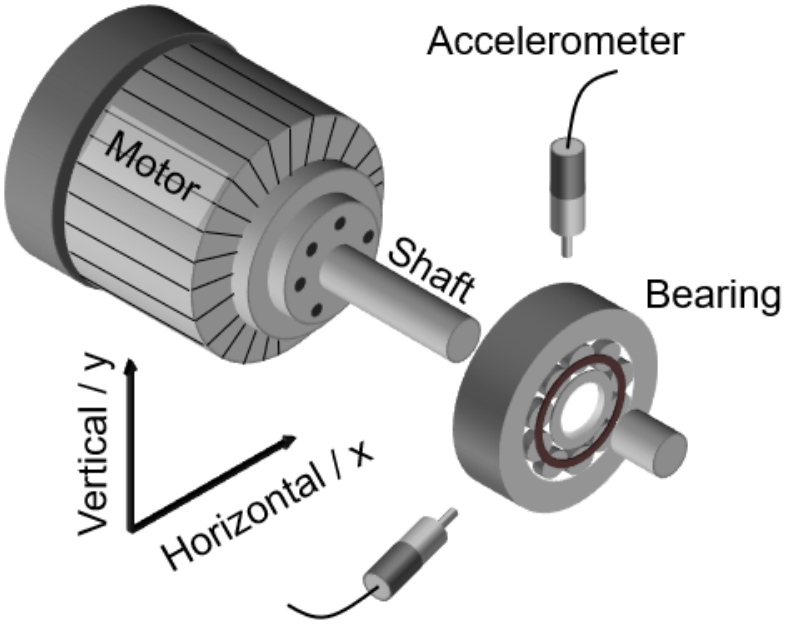

Figure 2: Sensor mounting in the radial direction

2.2 Proposed Model

2.2.1 Fundamental LSTM architecture

The proposed model utilizes LSTM networks for forecasting the bearing health condition. LSTMs utilize memory cells in addition to standard RNN units which help in retaining useful information for both long and short periods of time and do not face the issue of vanishing gradients common to RNNs. The basic architecture of the proposed model is shown in Figure 3. The structure of an LSTM memory cell is shown in Figure 3(b) where each cell contains three gates (1) forget gate, (2) input gate, and (3) output gate. The equations for the gates within the memory cell can be described as

Forget gate:

$$f_j = \sigma\left(w_f\left[h_{j-1}, X_j\right] + b_f\right) \quad (3)$$

where the sigmoid layer takes the input X_j and the output of the previous LSTM block h_{j-1} to determine which parts from the old output be removed and w_f is the weight of the forget gate with bias b_f.

Input gate:

$$i_j = \sigma\left(w_i\left[h_{j-1}, X_j\right] + b_i\right) \quad (4)$$

$$\widetilde{c_j} = \tanh\left(w_c\left[h_{j-1}, X_j\right] + b_c\right) \quad (5)$$

$$c_j = f_j \otimes c_{j-1} + i_j \otimes \widetilde{c_j} \quad (6)$$

where the sigmoid layer decides which of the new information be stored and $\tanh(\cdot)$ creates all possible values from the input X_j. These two are then multiplied to update the new cell state $\widetilde{c_j}$. This new memory is added to the previous cell state c_{j-1} after the forget gate. w_i and w_c are the respective weights of the input gate with corresponding biases b_i and b_c.

Output gate:

$$o_j = \sigma\left(w_o\left[h_{j-1}, X_j\right] + b_o\right) \quad (7)$$

$$h_j = o_j \otimes \tanh\left(c_j\right) \quad (8)$$

where the sigmoid layer determines the output of the cell. $\tanh(\cdot)$ generates all possible values which when multiplied to the output o_j becomes selective of the output. w_o and b_o are respectively the weight and bias of the output gate. One important thing to note is the use of $\tanh(\cdot)$ in the input and the output gates overcome the vanishing gradient problem where the second derivative of the internal state variables can sustain for a long range before becoming zero.

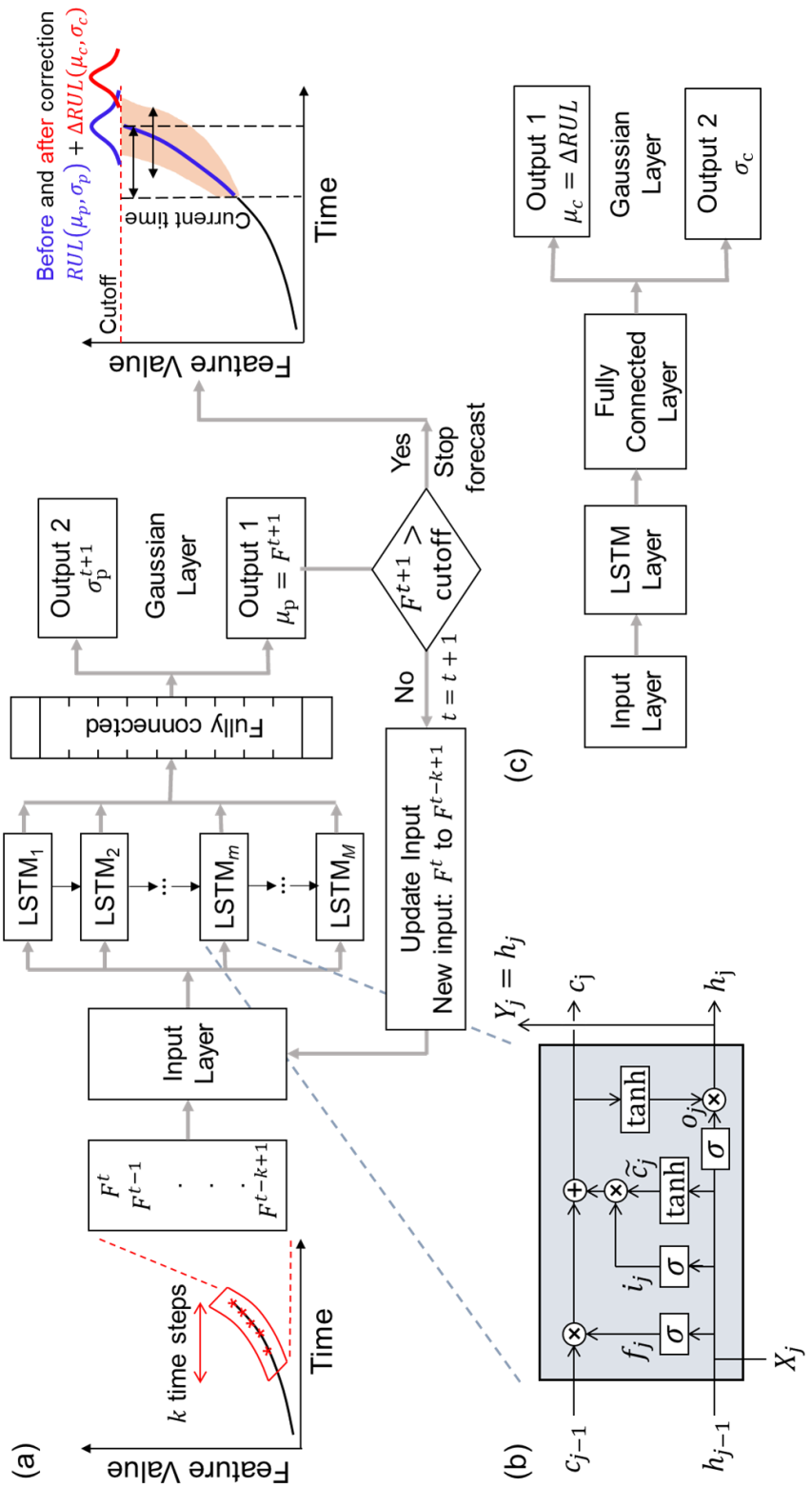

Figure 3: (a) Predictor LSTM architecture. (b) Fundamental LSTM unit. (c) Corrector LSTM schematic.

2.2.2 Gaussian Layer for Uncertainty Quantification

Traditional deep neural networks (DNNs) like LSTMs are designed for a single output prediction (or a point prediction), which can be viewed as an overconfident prediction. For practical applications like bearing failure, overconfident RUL predictions are dangerous and costly as they might either lead to premature maintenance requests (due to an early prediction) or catastrophic failure of the bearing and connected equipment (due to a late prediction). On the other hand, models that quantify the uncertainty of RUL prediction allow the user to make risk-based maintenance decisions that balance out maintenance resource requirements while avoiding early maintenance triggers stemming from low confidence prediction models. Probabilistic DNNs are often achieved through Bayesian formalism [55] where the parameters of the DNN are subjected to a prior distribution and after training, the posterior distribution over the parameters is computed which can then be used to quantify predictive uncertainty. To make the Bayesian implementation tractable for DNNs, a variety of approximations such as Markov chain Monte Carlo (MCMC) are used. However, Bayesian methods are computationally more expensive and model training takes more time when compared to non-Bayesian methods. To address this issue in DNNs, Monte Carlo dropout was proposed by Gal et al. [44]. Also, Lakshminarayanan et al. proposed a simple and scalable technique for predictive uncertainty estimation by using a proper scoring rule during training combined with model ensembles [56], which we use in our work for estimating uncertainty in bearing prognostics. For input features x, we use an LSTM network to model the prediction distribution $p_\theta(y|x)$ for real-valued output y and θ are the parameters of the LSTM network. We first state the methodology for a single LSTM model and then later combine them to generate an ensemble of LSTM models.

A scoring rule is used to measure the quality of the prediction $p_\theta(y|x)$ giving a higher numerical score to better-calibrated predictions. Let the scoring rule be $S\big(p_\theta, (y,x)\big)$ and the true distribution be $q(y,x)$. The expected scoring rule is

$$S(p_\theta, q) = \int q(y,x) S\big(p_\theta, (y,x)\big) dy dx \tag{9}$$

$$S(p_\theta, q) \leq S(q,q);\ S(p_\theta, q) = S(q,q) \ \text{ iff } \ p_\theta(y|x) = q(y|x) \tag{10}$$

Therefore, by minimizing the loss function $\mathcal{L}(\theta) = -S(p_\theta, q)$, $p_\theta(y|x)$ can approach $q(y|x)$. When maximizing the likelihood, the score function can be given as $S\big(p_\theta\ (y,x)\big) = \log p_\theta(y|x)$ which satisfies the Gibbs inequality. Commonly used loss functions like mean squared error stated as $MSE = \sum_{n=1}^{N} \big(y_n - \mu(x_n)\big)^2$ for a training dataset containing N datapoints of (x, y) do not capture predictive uncertainty. We, therefore, devise a Gaussian layer (see Figure 3(a)) which gives two outputs: the predicted mean $\mu(x)$ and variance $\sigma^2(x)$. By treating the sample values to obey the Gaussian distribution with the predicted mean and variance, we minimize the negative log-likelihood (NLL) criterion

$$-\log p_\theta(y_n|x_n) = \frac{\log \sigma_\theta^2(x)}{2} + \frac{\big(y - \mu_\theta(x)\big)^2}{2\sigma_\theta^2(x)} + \text{constant} \tag{11}$$

In other words, training the model using the scoring rule gives two outputs: mean $\mu(x)$ and variance $\sigma^2(x)$ accounting for the aleatoric uncertainty, which is a measure of the variation within each prediction model. On the other hand, the accuracy of a deep learning model depends on the amount of data available, leading to epistemic uncertainty which we capture through an ensemble of LSTM networks. With the availability of more data, the predictions of the LSTM networks in the ensemble tend to merge, thereby reducing the epistemic uncertainty. Each LSTM network in

the ensemble is trained independently through different weight initializations and shuffling the input data. To that end, we train $M = 5$ LSTM models on the same data that only differ through the learned parameters θ_m. One could also change the number of LSTM unit cells among different LSTM networks and still obtain good uncertainty estimations. We then treat the ensemble as a uniformly-weighted mixture model and combine the predictions as

$$p(y|x) = \frac{1}{M}\sum_{m=1}^{M} p_{\theta_m}(y|x) \quad 12$$

In our study, $p_{\theta_m}(y|x)$ refers to the Gaussian probability distribution of the forecast trajectory of $V_{0.2\omega-sf/2}^{RMS}$ of each of the M LSTM models. We can further derive that the ensemble of all the LSTM models to also be Gaussian with the mean and variance taking the following forms

$$\mu_*(x) = \frac{1}{M}\sum_{m=1}^{M} \mu_{\theta_m}(x) \quad 13$$

$$\sigma_*^2(x) = \frac{1}{M}\sum_{m=1}^{M} \left(\sigma_{\theta_m}^2(x) + \mu_{\theta_m}^2(x)\right) - \mu_*^2(x) \quad 14$$

2.2.3 Proposed Model Architecture

The proposed model is an ensemble of multiple simple LSTMs with a Gaussian layer for uncertainty quantification. The proposed method involves three steps

1) Predictor LSTM ensemble (EnPLSTM): where the feature ($V_{0.2\omega-sf/2}^{\mathrm{RMS}}$) is forecasted to a certain alarm threshold and hence predict the RUL.
2) Corrector LSTM ensemble (EnCLSTM): where the output of the EnPLSTM is used to determine the possible correction to the RUL.
3) Temporal fusion: where the predictions from the recent past are considered to provide a final RUL prediction.

Predictor LSTM ensemble (EnPLSTM):

The EnPLSTM consists of individual predictor LSTM (PLSTM) models for which the input at any time t consists of the feature values of the previous k timesteps, $F^{t-k+1}, F^{t-k+2}, \ldots, F^t$ (k is also called the lookback time step). The input has the form ($\#\text{samples} \times k \times n_{\text{features}}$) with the output being the next-step feature prediction F^{t+1} (here $n_{\text{features}} = 1$ as we only forecast $F = V_{0.2\omega-sf/2}^{\mathrm{RMS}}$). We then march forward in time until the cutoff is reached at T_{cutoff} and determine the mean value of RUL as $\mu_m^{RUL}(t) = T_{cutoff} - t$. The use of a Gaussian layer for each PLSTM model provides information about the uncertainty in the forecast feature which can then be used to determine the uncertainty in the RUL prediction at every time instant $\sigma_m^{RUL}(t)$. After performing the ensemble of all the PLSTMs using eqns. 13 and 14, we obtain $RUL\left(\mu_{*\mathrm{p}}, \sigma_{*\mathrm{p}}\right)$ as the final output of the EnPLSTM. The schematic in Figure 3(a) refers to just one PLSTM network and Table 1 lists the various layers in each PLSTM model with $k = 20$. The Gaussian layer in Table 1 has two outputs – the mean and standard deviation of the next step prediction. Each PLSTM model consists of 16,142 parameters which is at least two orders of magnitude smaller than some of the other contemporary deep learning models that quantify uncertainty [36].

Table 1: Architecture of a PLSTM network with Gaussian layer

Layer	Output shape	# Parameters
Input layer	(Samples, 20, 1)	0
LSTM	(Samples, 60)	14,880

Dense	(Samples, 20)	1,220
Gaussian layer	[(Samples, 1), (Samples, 1)]	42
Total:		16,142

Corrector LSTM ensemble (EnCLSTM):
The input and output of the PLSTM model are respectively the features from the previous k time steps and the next step prediction. We observe the RUL prediction of a trained EnPLSTM shows deviation from RUL^{true} even for the training dataset. We note that our approach of forecasting is different from the commonly used bearing prognostic approach of directly mapping features to RUL, in which case we can expect a good RUL fit at least for the training dataset. In other words, RUL becomes a secondary outcome of the EnPLSTM method unlike a primary output when developing feature-RUL mapping models. Therefore, the EnPLSTM is used to evaluate the error in RUL prediction on the training dataset. The error in forecasting for each bearing can be quantified as $\Delta RUL(t) = RUL^{\text{true}}(t) - RUL(t)$.

A shown in Figure 3(c), the architecture of the CLSTM model is similar to that of the PLSTM model with two differences: (1) the input now includes $RUL(\mu_{*\text{p}})$ from the EnPLSTM model, in addition to the input to the predictor step, and (2) the output is now $\Delta RUL(t)$, rather than the next-step feature prediction. Unlike the EnPLSTM (which is a one-step-ahead prediction), the EnCLSTM attempts to map the RUL prediction error. The architecture of a single CLSTM model is shown in Table 2 with the LSTM layer having 80 hidden units. The shape of the input layer is $(\text{samples}, 20, 2)$ with a lookback of 20-time steps with two features: $RUL(\mu_{*\text{p}})$ and $V^{\text{RMS}}_{0.2\omega-sf/2}$. The final output from the Gaussian layer is the mean and standard deviation of the error correction $\Delta RUL(\mu_{\text{c}}, \sigma_{\text{c}})$. After training, the CLSTM model gives an estimate of the mean and standard deviation of the error $\Delta RUL(\mu_{\text{c}}, \sigma_{\text{c}})$, which after ensemble (following the same logic as in eqns. $13 - 14$) becomes $\Delta RUL(\mu_{*\text{c}}, \sigma_{*\text{c}})$.

Table 2: Architecture of the CLSTM model with Gaussian layer

Layer	**Output shape**	**# Parameters**
Input layer	(Samples, 20, 2)	0
LSTM	(Samples, 80)	26,560
Dense	(Samples, 20)	1,620
Gaussian layer	[(Samples, 1), (Samples, 1)]	42
Total:		28,222

The correction term $\Delta RUL(\mu_{*\text{c}}, \sigma_{*\text{c}})$ can be positive or negative, however, in our experience, we find that the PLSTM model, generally, underpredicts the RUL during the early stage of bearing degradation but converges to actual RUL in the second half of the bearing life. In other words, at the beginning of bearing degradation, the CLSTM model plays a significant role but loses importance as the bearing approaches EOL. To this end, we combine $RUL(\mu_{*\text{p}}, \sigma_{*\text{p}})$ and $\Delta RUL(\mu_{*\text{c}}, \sigma_{*\text{c}})$ through a weight which is a function of the feature value $W(F = V^{\text{RMS}}_{0.2\omega-sf/2})$. The net RUL prediction $RUL(\mu_{\text{final}}, \sigma_{\text{final}})$ can be stated as

$$RUL(\mu_{\text{net}}, \sigma_{\text{net}}) = RUL(\mu_{*\text{p}}, \sigma_{*\text{p}}) + W(F = V^{\text{RMS}}_{0.2\omega-sf/2}) \times \Delta RUL(\mu_{*\text{c}}, \sigma_{*\text{c}}) \quad 15$$

A logistic sigmoidal function is used as the weight function $W(F = V^{\text{RMS}}_{0.2\omega-sf/2})$ with the sigmoid midpoint F_0 pinned at 1.25 times the feature value at FPT.

$$W(F) = 1 - \frac{1}{1 + e^{-\alpha(F - F_0)}} \quad (16)$$

where α determines the growth rate/ steepness of the sigmoidal curve and $F_0 = 1.25 \times F(t = t_{\text{FPT}}) = 1.25 \times V^{\text{RMS}}_{0.2\omega - sf/2}(t = t_{\text{FPT}})$.

Temporal Fusion:

Rapid changes in the vibration measurements can often lead to highly time-varying RUL predictions, especially when using data mapping models like the CLSTM. Sudden changes in the RUL predictions are not physically meaningful from a maintenance perspective. We, therefore, devise a simple technique where the RUL predictions in the recent past are weighed in to make a final prediction. A simple half-normal weighting function is used to determine the importance of the RUL predictions where the predictions closest to the current time get more weight than those in the distant past. At a time t, the RUL prediction after temporal fusion $RUL\left(\mu_{tf}(t)\right)$ can be stated as

$$RUL\left(\mu_f(t)\right) = \sum_{i=0}^{L} \overline{w_{tf,i}} \times \left(RUL\left(\mu_{\text{net}}(t - i)\right) - i\right) \quad (17)$$

$$w_{tf,i} = \frac{1}{\sigma_{tf}} e^{\left(-\left(\frac{i\Delta t}{2\sigma_{tf}}\right)^2\right)} \quad (18)$$

$$\overline{w_{tf,i}} = \frac{w_{tf,i}}{\sum_{i=0}^{L} w_{tf,i}} \quad (19)$$

where L is the number of discrete past RUL predictions the user wants to consider, Δt is the time interval between two consecutive RUL predictions and σ_{tf} is a user-defined parameter that accounts for the spread of the half-normal curve. A larger value of σ_{tf} would give more similar weights to recent RUL predictions whereas a smaller value of σ_{tf} gives more importance to the current RUL prediction at time t. The weights across the $(L + 1)$ RUL predictions are normalized in eqn. 19. We observe that performing temporal fusion provides smoother RUL prediction curves while also reducing the RMSE error. The entire algorithm is presented in Table 3.

Table 3: Algorithm for the proposed predictor-corrector LSTM model for bearing prognostics

Algorithm: Probabilistic RUL prediction for bearing prognostics (test dataset)	
Inputs:	Accelerometer vibration signal $a(t)$ over the past k time steps Lookback time k: 20 time steps Cutoff: 0.3 ips $M = 5$ trained PLSTMs and CLSTMs
Output:	Probabilistic remaining useful life $RUL(\mu_{\text{final}}, \sigma_{\text{final}})$ at time t
1	**Calculate** the velocity vibration $v(t) = \int_0^t a(t)dt$ and the corresponding FFT in frequency domain $V(f)$. Calculate $V^{\text{RMS}}_{BFF - sf/2}(t)$ and $V^{\text{RMS}}_{0.2\omega - sf/2}(t)$.
	Predictor LSTM model
2	**Reshape** input feature $F = V^{\text{RMS}}_{0.2\omega - sf/2}(t - k + 1 \to t)$ into shape $X_{\text{p}} = (1, \text{lookback}, 1)$. The input sample at a given time t is of the form $F(t - k + 1 \to t)$ with the corresponding output will be $F(t + 1)$.
3	**Use** $V^{\text{RMS}}_{BFF - sf/2}(t)$ in eqn. $B.1$ to determine if FPT is reached. Proceed iff $t \geq t_{\text{FPT}}$.
4	**for** each PLSTM $m = 1{:}M$

5 **Initialize**: forecast time $t^f = 1, f = F(t)$
6 **while** ($f \leq 0.3$ ips):
7 Next step prediction: $(f, \sigma_f) = PLSTM(X_\mathrm{p})$
8 Modify $X = concatenate(X, f)$
9 Update $X_\mathrm{p} = X_\mathrm{p}(t + t^f - k + 1 \rightarrow t + t^f)$
10 $t^f = t^f + 1$
11 **end while**
12 $\mu_m^{RUL}(t) = t^f$. Calculate aleatoric uncertainty $\sigma_m^{RUL}(t)$ using σ_f.
13 **end for** (line 7)
14 **Calculate** ensemble mean and variance $RUL(\mu_{*\mathrm{p}}, \sigma_{*\mathrm{p}})$ as $\mu_{*\mathrm{p}}(t) = \frac{1}{M}\sum_{m=1}^{M} \mu_m^{RUL}(t)$ and $\sigma_{*\mathrm{p}}^2(t) = \frac{1}{M}\sum_{m=1}^{M}\left(\sigma_m^{RUL^2}(t) + \mu_m^{RUL^2}(t)\right) - \mu_{*\mathrm{p}}^2(t)$. This is the final output of EnPLSTM.

Corrector LSTM model

15 **Reshape** input features $V_{0.2\omega-sf/2}^{\mathrm{RMS}}(t - k + 1 \rightarrow t)$ and $RUL\left(\mu_{*\mathrm{p}}(t - k + 1 \rightarrow t)\right)$ into shape $X_\mathrm{c} = (1, \mathrm{lookback}, 2)$.
16 **Determine** the error correction $\Delta RUL(\mu_\mathrm{c}, \sigma_\mathrm{c}) = \mathrm{CLSTM}(X_\mathrm{c})$ for each CLTM and calculate ensemble correction $\Delta RUL(\mu_{*\mathrm{c}}, \sigma_{*\mathrm{c}})$ similar to line 14.
17 **Calculate** the final RUL prediction using eqn. 15.
18 **Temporal fusion**: Use eqn. 17 for smoothing the RUL prediction.

2.3 Models for Comparison

In this section, we briefly present three data-driven approaches, which are (1) CNN-based feature-RUL mapping, (2) similarity-based interpolation, (3) Monte Carlo (MC) Dropout (see Appendix C), and two model-based approaches, (1) optimized particle filter and (2) regression fitting (see Appendix D for quadratic and double exponential regression fitting). In a later section, we compare the performance of the proposed model against these four benchmark models typically employed in prognostic literature.

2.3.1 CNN

Traditionally, CNN was used for image processing to capture spatial and temporal dependencies of image features by application of several filters [57]–[59]. Many bearing prognostic models were built upon a CNN framework [31]–[33] and we, therefore, adopt a basic CNN architecture in our study to compare against our proposed method. Each input sample at a given time t of the CNN model is the set of 24 features (see Appendix A) for the previous 20-time steps and the output is the corresponding RUL of the bearing.

The CNN model consists of six convolution blocks, a dropout layer, and two fully connected layers (see Appendix C for the model architecture). The convolution blocks contain three layers, namely, 1-D convolution, 1-D batch normalization, and a Leaky ReLU non-linear activation function. The dropout layer serves to prevent overfitting of the training data. The two fully connected layers further reduce the features generated by the convolution blocks to a single output, the estimated RUL. The CNN model was implemented using PyTorch in a Python environment configured to run on a single Nvidia RTX-2070 video card with 8 Gb of onboard graphics memory. The model was trained for 100 epochs using AdamW optimizer with beta 1 of 0.5, beta 2 of 0.999, weight decay of 0.01, and initial learning rate of 0.001. The training was performed with mean squared

error as the loss function. The learning curve of CNN and PLSTM model are shown in Appendix G.

2.3.2 Similarity-based Interpolation

Similarity-based interpolation is a data-driven prognostic approach where a portion of the bearing health data, such as the feature development F_{test} from a test bearing is compared against similar feature(s) from the training dataset F_{train}. The hypothesis of this method is that the partial F_{test} is similar to an equal-sized portion from F_{train}, the time-scale of which is determined by optimizing the difference between the two data [60]–[63]. To predict the RUL of a test bearing at time t, the test feature F_{test} in our study will be the $V_{0.2\omega-sf/2}^{\text{RMS}}(t-k+1 \rightarrow t)$ with a lookback of k time steps. To determine the optimal fit with respect to each training bearing, F_{test} is displaced along the time axis and the time instant T_0 at which the sum of squared differences (SSD) between F_{test} and F_{train} is minimum is determined. Figure 4 depicts the procedure for determining T_0. Mathematically, this can be stated as

$$\min SSD = \sum_{j=1}^{k} \left(F_{\text{test}}(t-j+1) - F_{\text{train}}(T_0+k-j)\right)^2 \quad 20$$

subject to $T_0 \in [0, L-k]$ where L is the total life of the training bearing dataset. T_0 determined from eqn. 20 is then used to calculate RUL based on the training dataset given as

$$RUL = L - k - T_0 \quad 21$$

In many cases, the training dataset consists of run-to-failure vibration data from multiple bearings (say n_{train} in number) and RUL determined from eqn. 21 for each of the bearings in the training dataset can be added using a simple weight function which is the inverse of SSD. In other words, a smaller value of SSD indicates greater similarity, and the appropriate RUL is given greater importance. This can be stated as

$$RUL_{\text{net}} = \frac{1}{W} \sum_{i=1}^{n_{\text{train}}} W_i \times RUL_i \quad 22$$

$$W = \sum_{i=1}^{n_{\text{train}}} W_i \quad 23$$

$$W_i = \frac{1}{SSD_i} \quad 24$$

A major advantage of this method is the non-requirement of defining failure. However, this method cannot guarantee that the RUL prediction converges to true RUL as the bearing is close to EOL.

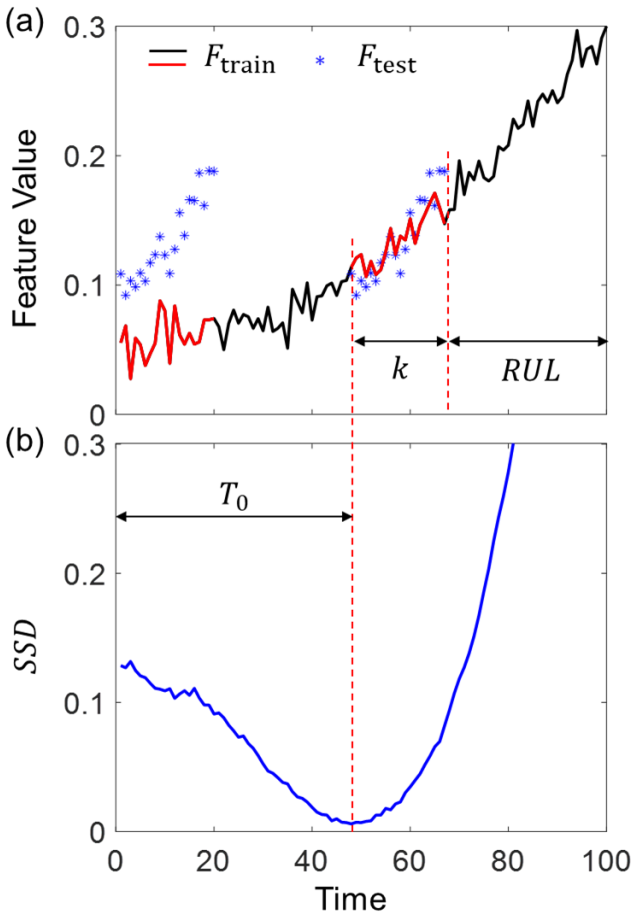

Figure 4: (a) Depiction of similarity-based interpolation for RUL prediction and (b) variation of SSD and determination of T_0.

2.3.3 Optimized Particle Filter

Particle filter (PF) is based on the concepts of Bayesian inference and the sequential Monte Carlo method and excels in modeling dynamic non-linear systems [64]. PF has been found to be successful in other bearing prognostics studies [65]–[67]. A set of random particles approximately satisfying the model equations are used for estimating the potential RUL with uncertainty. However, this method is very sensitive to the initial guess of the system state and resampling strategies and improper selection of the same often leads to degeneracy or leading to loss of particle diversity [68]. The fundamentals of PF are described in Appendix D and in this section, we briefly describe our implementation of PF with optimized initial states utilizing Latin-hypercube sampling.

Modeling the state and measurement equations for bearings can be quite complex as the failure modes are quite diverse and we, therefore, use a combination of exponential and linear terms in describing the development of bearing features over time. Mathematically, we use the following equations:

State transition equation:

$$a_t = a_{t-1} + u_{1,t}, \quad b_t = b_{t-1} + u_{2,t}, \quad c_t = c_{t-1} + u_{3,t} \tag{25}$$

Measurement equation:

$$y_t = a_t e^{b_t(t-\text{FPT})} + c_t(t - t_{\text{FPT}}) + \nu_t \quad (26)$$

where y_t is the feature measurement (obtained from vibration data) at time t, u_1, u_2, u_3, ν are the Gaussian noise variables with a certain standard deviation (and zero mean). Proper execution of PF involves the following steps (1) particle initialization, (2) state update, (3) particle weight update, (4) resampling, and (5) state estimation (which we describe in Table D.1).

As measurements are collected in real-time, the system parameters of the particles are trained to start from the initial guess, and the updated state of the particles is used to forecast the features until a threshold is reached and hence obtain the RUL^j for the j^{th} particle. The effective RUL is obtained by a weighted sum of RUL^j. This can be mathematically expressed as

$$RUL^j(t) = \underset{t^*}{\text{Solve}}\left(a_t^j e^{b_t^j t^*} + c_t^j t^* = \text{cutoff}\right) - (t - t_{\text{FPT}}) \quad (27)$$

$$RUL(t) = \sum_{j=1}^{N_p} w_t^j \times RUL^j(t) \quad (28)$$

Often the selection of the initial state values (which can be considered as hyperparameters) is heuristic and can change from bearing to bearing which defeats the purpose of a generalized PF model. To this end, we develop the PF algorithm by optimizing the initial state parameters $\{a_0, b_0, c_0\}$ on the training bearing dataset by minimizing an RUL prediction error metric and using the same initial state for the test bearings.

3. Case Study Using the XJTU-SY Dataset

In this section, we demonstrate the advantage of our proposed prognostic method utilizing the run-to-failure vibration data provided by Ref. [29]. We also compare our proposed method against the methods described in section 2.3.

3.1 Dataset

The XJTU-SY bearing dataset consists of run-to-failure vibration data of 15 roller element bearings (LDK UER204). The failure of these bearings is accelerated by applying a radial load. The 15 bearings are divided into three groups of 5 bearings and each group is subject to a certain radial load and rotational speed (see Table 4). Two PCB 352C33 accelerometers are mounted perpendicularly along the radial direction, which the authors of Ref. [29] refer to as horizontal and vertical directions. We refer to the same as vibrations in the x and y directions consistent with the schematic shown in Figure *2*. Data is collected for 1.28 sec every minute at a sampling frequency of 25.6 kHz. For further details regarding the experimental setup, we refer the readers to Ref. [29].

Table 4: Summary of bearings from XJTU-SY dataset [29]

Operating condition	Bearing ID	Rotating speed (rpm)	Radial force (kN)
Condition 1	Bearing 1_1 Bearing 1_2 Bearing 1_3 Bearing 1_4 Bearing 1_5	2100	12
Condition 2	Bearing 2_1 Bearing 2_2	2250	11

	Bearing 2_3 Bearing 2_4 Bearing 2_5		
Condition 3	Bearing 3_1 Bearing 3_2 Bearing 3_3 Bearing 3_4 Bearing 3_5	2400	10

Figure 5(a) shows the run-to-failure vibration data obtained from the accelerometer mounted in the x direction for Bearing 1_1. The reported total life of the bearing is 123 min with vibration measurements taken at every minute. For purposes of illustration, we highlight the vibration data obtained at $t = 100$ min in Figure 5(a) and also show the corresponding FFT of this signal in Figure 5(c). Since the provided data is obtained from accelerometers whereas our proposed method is primarily aimed at bearing prognostics using ISO standards, we first convert the acceleration signal into the velocity domain by integration (see section 2). The result of integration is shown in Figure 5(b) and the corresponding FFT of $v(t = 100)$ is presented in Figure 5(d). Numerical integration of the acceleration signal introduces low-frequency component as can be seen by a wavy nature of $v(t)$. This can also be seen in the FFT of $v(t)$ in Figure 5(d) where we can observe large amplitudes in the very low-frequency domain of $< 0.2\omega$. This numerical artifact is taken care of by considering the RMS value calculated from $f \geq 0.2\omega$. The fault frequencies for this bearing are determined to be $BPFO = 3.08\omega$ and $BPFI = 4.92\omega$. In Figure 5(c) and (d), we also show $1 \times, 2 \times$ and $3 \times BPFO \pm 5\%$ Hz bands, and $1 \times$ and $2 \times BPFI \pm 5\%$ Hz bands (as defined in section 2.1). One can observe peaks in BPFO bands indicating an outer race fault which is also confirmed in Ref. [29]. Also, the process of integration into the velocity domain preserves the peaks at characteristic fault frequencies.

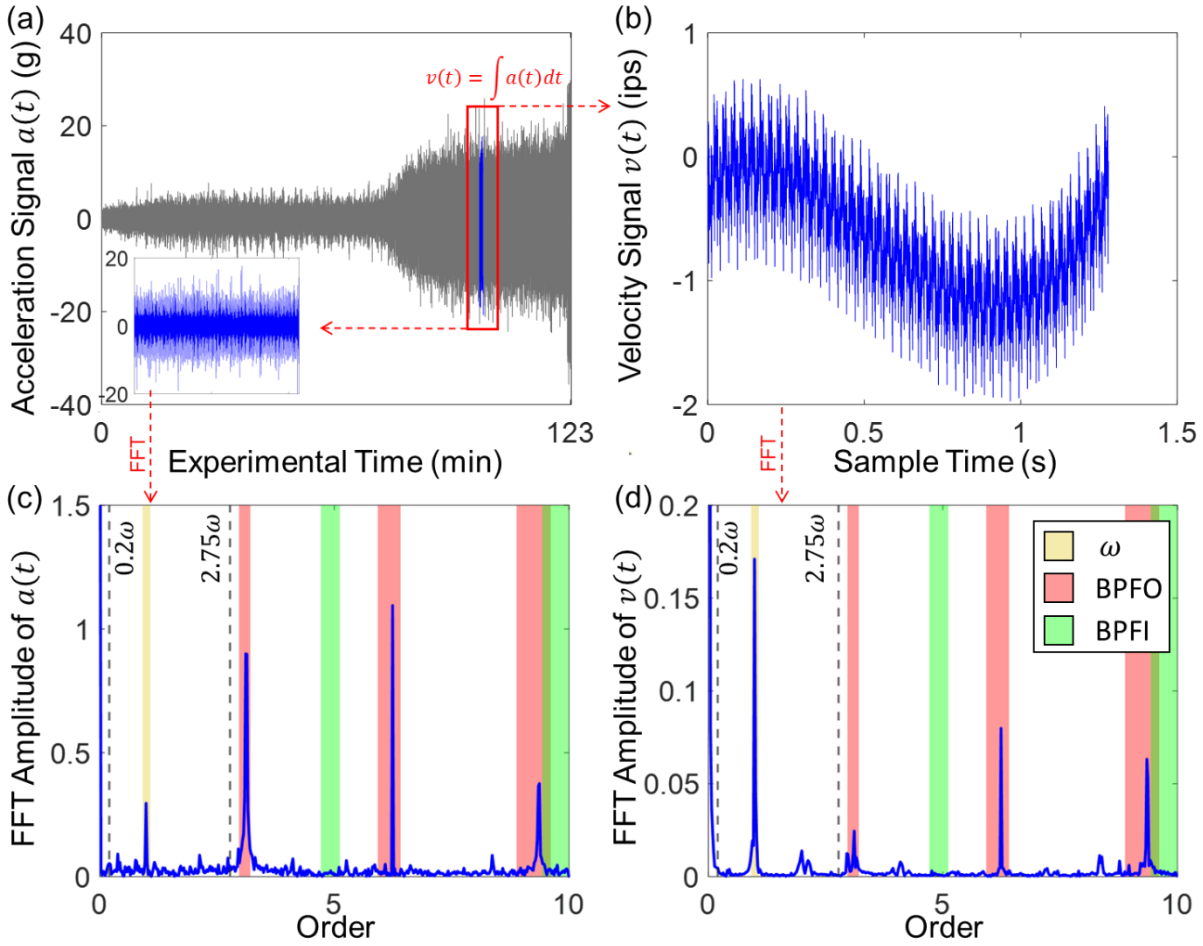

Figure 5: (a) Run-to-failure vibration data for Bearing 1_1 with a snapshot of the vibration signal collected at $t = 100$ min. (b) Corresponding velocity signal at $t = 100$ min. The FFT spectra of the acceleration and velocity signals along with BPFO and BPFI are shown in (c) and (d) respectively.

3.2 FPT Determination

The bearing prognostic algorithm is triggered at FPT as this marks the beginning of bearing degradation. Before we present the results of FPT on this dataset, we first show a waterfall plot revealing the development of a bearing fault in the frequency domain. Figure 6 shows the FFT waterfall plot of Bearing 1_1 within the first ten orders of shaft frequency. One can observe the advent of an outer race defect at around 80 min which is accompanied by an increase in FFT amplitudes in the BPFO characteristic frequency range (and its harmonics). We have suppressed the DC component ($f = 0$ Hz) of the FFT for presentation purposes.

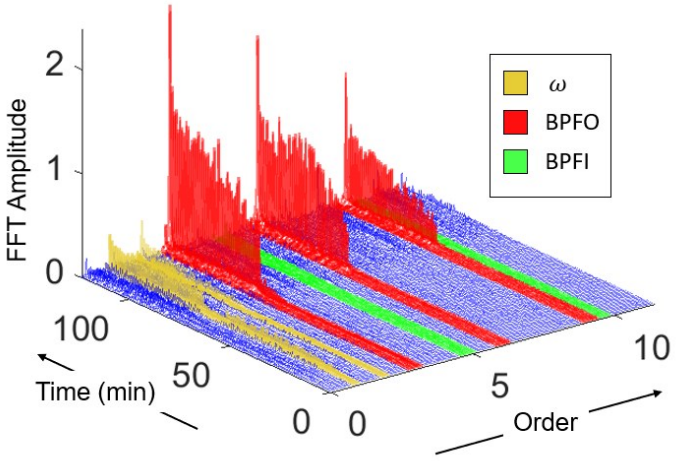

Figure 6: Waterfall plot of the FFT for Bearing 1_1 with characteristic fault frequency bands

As stated in section 2.3, FPT is determined by the 2σ method [35] applied on $V^{\mathrm{RMS}}_{BFF-sf/2}$ where BFF refers to the bearing fault frequencies $BFF = 0.9\min(BPFO, BPFI, BSF)$. In this study, we neglect BSF, and hence we get $BFF \cong 2.75\omega$. We mark this frequency in Figure 5(c) and (d). In Figure 7 we show the variation of $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ and $V^{\mathrm{RMS}}_{2.75\omega-sf/2}$ for two candidate bearings, Bearing 1_1 and Bearing 2_3, in both the x and y directions. Several observations can be made from Figure 7. First, $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ which is a measure of the overall health of the bearing assembly is always greater than $V^{\mathrm{RMS}}_{2.75\omega-sf/2}$ which primarily measures the bearing health condition. This stems from the fact that the energy within the frequency range of $0.2\omega - sf/2$ already contains the energy associated with $2.75\omega - sf/2$. As a corollary, a large difference between $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ and $V^{\mathrm{RMS}}_{2.75\omega-sf/2}$ is indicative of synchronous defects such as shaft unbalance, misalignment and mechanical looseness. On the contrary, a smaller difference between the two RMS values indicates a good fit/assembly. As can be seen in Figure 7, Bearing 1_1 experiences a relatively larger degree of synchronous faults when compared to Bearing 2_3. Second, $V^{\mathrm{RMS}}_{2.75\omega-sf/2}$ is much more stable than $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ and is therefore a good metric to determine the FPT using the 2σ method. On the other hand, $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ is used to determine the EOL, based on the cutoff of 0.3 ips, as it reflects the overall vibration energy levels within the system. Third, the FPT and EOL vary in both directions for both bearings. We, therefore, determine the effective FPT conservatively by choosing the earlier occurrence of t_{FPT_x} and t_{FPT_y}.

$$t_{\mathrm{FPT}} = \min\left(t_{\mathrm{FPT}_x}, t_{\mathrm{FPT}_y}\right) \tag{29}$$

The effective EOL is determined when the overall RMS reaches the threshold value in both x and y directions to ensure good utility of the bearing and avoiding early maintenance.

$$t_{\mathrm{EOL}} = \max\left(t_{\mathrm{EOL}_x}, t_{\mathrm{EOL}_y}\right) \tag{30}$$

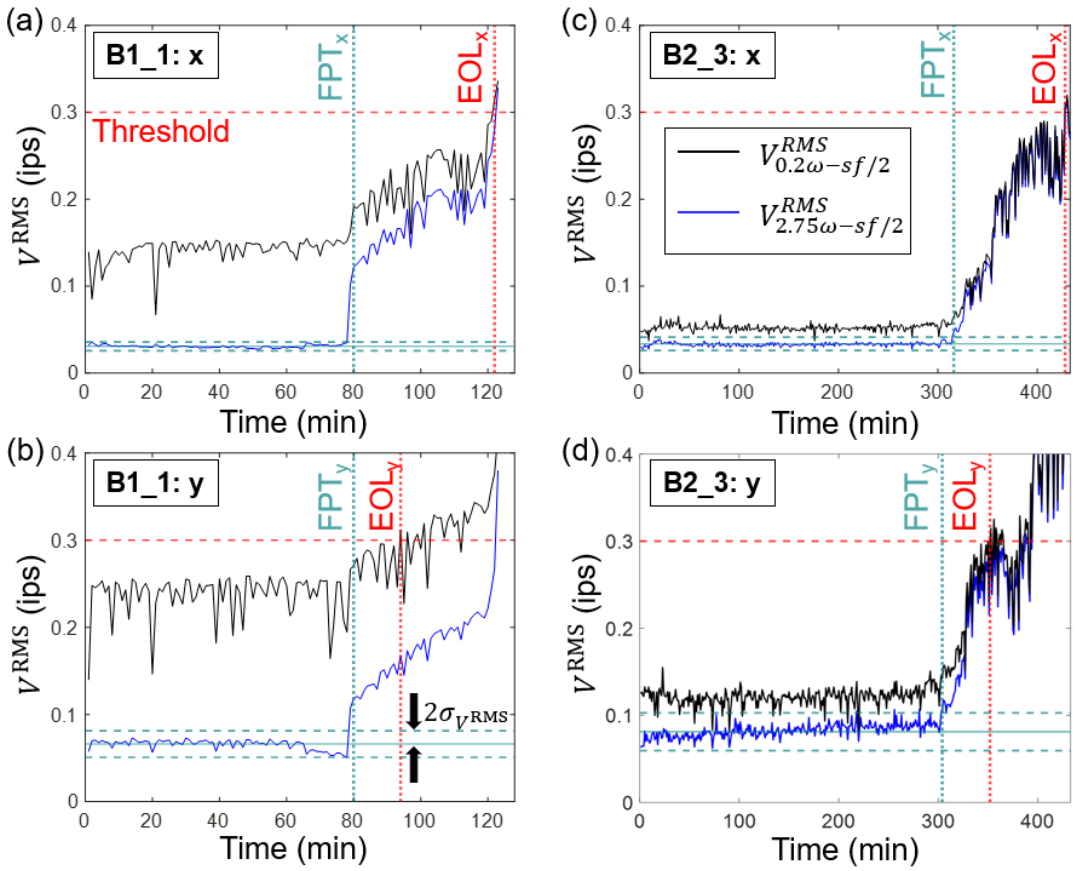

Figure 7: Development of features $V^{RMS}_{0.2\omega-sf/2}$ and $V^{RMS}_{2.75\omega-sf/2}$ for Bearing 1_1 in (a) x-direction and (b) y-direction, and for Bearing 2_3 in (c) x-direction and (d) y-direction. FPT, EOL and $2\sigma_{V^{RMS}}$ are also plotted in each case.

3.3 Development of the Proposed Model

In this section, we first describe the test-train data for cross-validation followed by a parametric study, focused on the PLSTM model. We then depict the advantage of the proposed model when compared to other models discussed in section 2.6.

3.3.1 Cross-Validation

A 5-fold cross-validation study is conducted on the set of 15 bearings. The five folds are as follows:
Fold-1: Bearing 1_1, Bearing 2_1, Bearing 3_1
Fold-2: Bearing 1_2, Bearing 2_2, Bearing 3_2
Fold-3: Bearing 1_3, Bearing 2_3, Bearing 3_3
Fold-4: Bearing 1_4, Bearing 2_4, Bearing 3_4
Fold-5: Bearing 1_5, Bearing 2_5, Bearing 3_5
While performing the cross-validation study, one fold is chosen to be the test set while the other four folds are used for training the models. For example, for the first cross-validation trial, Fold-1 serves as the test set whereas Folds 2, 3, 4, and 5 are used for training the model. Cross-validation ensures the generality of the model and any result of a bearing presented hereafter is obtained when the bearing is a part of the test set during the cross-validation study.

3.3.2 Evaluation Criteria

Several evaluation criteria are used to evaluate and compare the performance of all the models in terms of prediction error as well as uncertainty quantification. First, the root mean squared error ($RMSE$) is calculated as

$$RMSE = \sqrt{\frac{1}{(T - t_{\text{FPT}} + 1)} \sum_{t=t_{\text{FPT}}}^{T} \left(RUL^{\text{true}}(t) - RUL(t)\right)^2} \quad 31$$

where $RUL^{\text{true}}(t)$ and $RUL(t)$ are respectively the true RUL and predicted RUL at time t and T is the total duration of RUL prediction. $RMSE$ is a measure of the error in RUL prediction from FPT to EOL. Another important feature of a good prediction model is the convergence to the true RUL as bearing approaches EOL. To assess this, we use a weighted $RMSE$ which can be defined as

$$wtRMSE = \sqrt{\frac{1}{(T - t_{\text{FPT}} + 1)} \sum_{t=t_{\text{FPT}}}^{T} \overline{w(t)} \left(RUL^{\text{true}}(t) - RUL(t)\right)^2} \quad 32$$

where $\overline{w(t)}$ is the weight assigned to the squared prediction error at time t and this weight increases as the bearing approaches its EOL. To obtain $\overline{w(t)}$, we first defined weight $w(t)$ as $w(t) = t - t_{\text{FPT}}$ and then normalize this weight as $\overline{w(t)} = w(t)/\sum_{t=t_{\text{FPT}}}^{T} w(t)$.

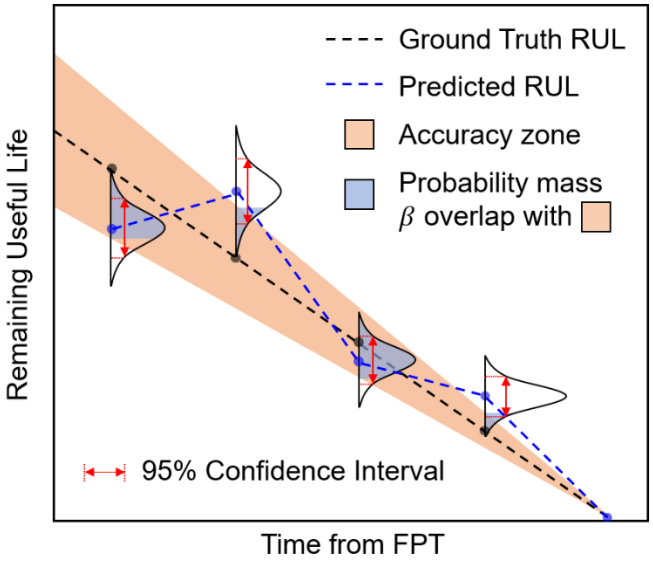

Figure 8: Uncertainty quantification metrics

Uncertainty quantification metrics are adapted from Refs. [69], [70] with a schematic shown in Figure 8. A good prognostic model would have decreasing uncertainty when approaching EOL to provide more confident RUL predictions. To quantify this, an accuracy zone (see Figure 8), bounded by $RUL^{\text{true}}(t)(1 \pm \alpha\%)$, is used to determine several metrics: (1) α-accuracy, which is defined as the number of predicted RUL points within the accuracy region with respect to the total number of predictions, (2) β-probability, which is the average of the probability mass of the RUL(t) PDF within the accuracy region and (3) percentage of early predictions (PEP) which measures the number of RUL(t) prediction below $RUL^{\text{true}}(t)$. It is preferred that α-accuracy approaches 100% where most RUL prediction points are within the accuracy zone. Ideally, β probability should be equal to 1 indicating a model to have a compact confidence interval which

also decreases as the bearing approaches EOL. The PEP metric provides insight into how conservative a given prognostic model is.

3.3.3 LSTM Parametric Study

A parametric study is important to optimize the model hyperparameters, such as the number of hidden units in the LSTM models, lookback k, the number of epochs (Appendix G), etc. For brevity, we only present the parametric study related to the number of hidden units in PLSTM. Figure 9(a) shows both the $RMSE$ and $wtRMSE$ of the PLSTM model on the training dataset for six different numbers of hidden units within the LSTM layer. By using a fewer number of hidden units (and hence fewer parameters), the deep learning model is too simple and becomes less sensitive to variation in the input data. On the other hand, using too many hidden units makes the model overly complex for the amount of data available tending towards overfitting. For the XJTU-SY dataset, we find that using 60 hidden units provides minimum $RMSE$ and $wtRMSE$.

Like any other prognostic model, LSTM-based architecture also has its limitations. Particularly in the bearing prognostic scenario, we find the following challenges: (1) very noisy feature data, (2) limited training data, and (3) most of the training data is in the domain pertaining to a healthy bearing suppressing learning from the bearing degradation domain. Although the third scenario can be tackled by considering only the bearing degradation data for training the LSTM network, this further accentuates the second problem of limited data. The use of data augmentation is particularly useful to address this aspect for a stable forecast. To demonstrate this, we use a simple toy example of linear degradation with noise to train and test an LSTM network as shown in Figure 9(b). When very little data is available and is noisy, the LSTM forecast can almost be flat especially near the onset of bearing degradation. By using data augmentation of duplicating the training data with added Gaussian noise, we observe the forecast to be much more intuitive and stable. To this end, for the XJTU-SY bearing dataset, we add Gaussian noise to $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ as a simple data augmentation technique similar to Refs. [71], [72].

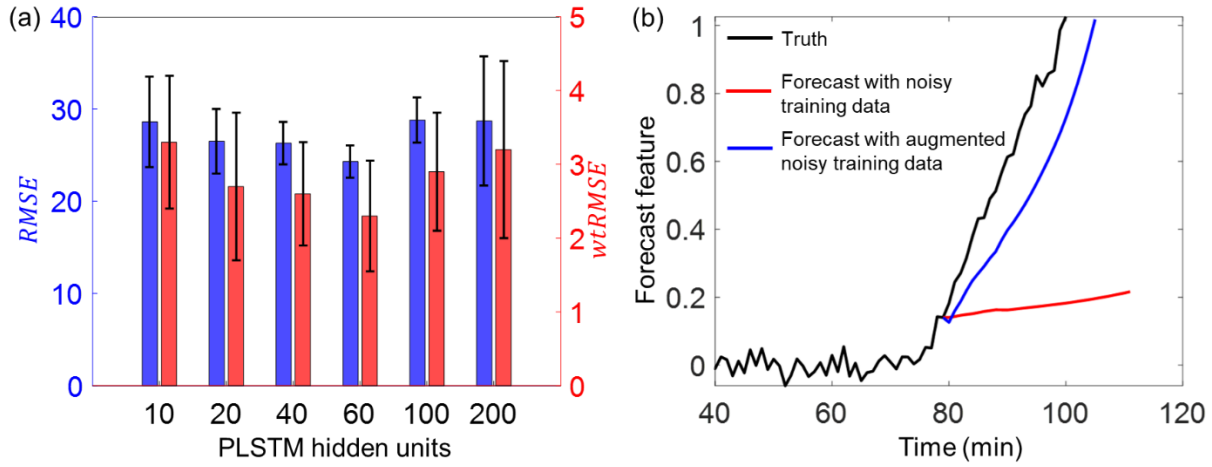

Figure 9: (a) Performance of various PLSTM on the training dataset. (b) Demonstration of the effect of data augmentation on noisy feature forecasting using a toy problem.

3.3.4 RUL Prediction Results

In this section, we first demonstrate the working of the EnP/CLSTM ensemble followed by depicting the RUL prediction results of certain bearings. Finally, we evaluate the various models in terms of accuracy and uncertainty quantification based on the metrics defined in section 2.4.

The PLSTM model forecasts $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ at a given instant in time till a cutoff of 0.3 ips is reached with uncertainty. Figure 10(a) shows the $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ forecast of five PLSTM models at $t = 2470$ mins for Bearing 3_2 (cross-validation Fold-2). The use of the Gaussian layer provides information regarding the uncertainty of the forecast which translates to the uncertainty in RUL prediction for each PLSTM model in the form of $RUL(\mu_m, \sigma_m)|_{m=1:5}$. The mean RUL prediction by each PLSTM model, $RUL(\mu_m)|_{m=1:5}$, is shown in Figure 10(b) (we suppress showing the uncertainty for clarity). An effective RUL, $RUL(\mu_{*P}, \sigma_{*P})$, is calculated using $\mu_{*\mathrm{p}}(t) = \frac{1}{5}\sum_{m=1}^{5} \mu_m^{RUL}(t)$ and $\sigma^2_{*\mathrm{p}}(t) = \frac{1}{5}\sum_{m=1}^{5}\left(\sigma_m^{RUL^2}(t) + \mu_m^{RUL^2}(t)\right) - \mu^2_{*\mathrm{p}}(t)$. We observe from Figure 10(b) that the ensemble of the five PLSTM models underpredicts the RUL in the first half of the prediction period and approaches the true RUL in the second half. After implementing the EnCLSTM, the RUL prediction in the first half is increased closer to the true RUL as shown by the green line in Figure 10(b). However, the prediction sequences change drastically when there are sudden changes in the measurements. After implementing the temporal fusion step (section 2.5.3), the RUL prediction is smoothened. The 95% confidence interval around the RUL prediction accommodates most parts of the true RUL. Therefore, maintenance decisions can be confidently made according to the uncertainty in RUL prediction.

In Figure 11, we compare the RUL prediction results from PF, similarity-based interpolation, CNN-RUL correlation, quadratic regression fitting, MC Dropout, and the proposed method for three representative bearings, each from a unique operating condition, viz. Bearing 1_3 (cross-validation Fold-3), Bearing 2_1 (cross-validation Fold-1), Bearing 3_4 (cross-validation Fold-4). Figure 11 (a) and (b) show the RUL prediction of the different models and the corresponding $\mathrm{V}^{\mathrm{RMS}}_{0.2\omega-sf/2}$ for Bearing 1_3 respectively. Here, we can observe that the noisy feature data right from the start of FPT distracts the PF learning, similarity-based approach, and quadratic regression, thus drastically affecting the RUL prediction accuracy. In all the three bearings shown in Figure 11, the proposed EnP/CLSTM model shows superior prognostic capability. Also, the similarity-based approach is often observed to overpredict the RUL in the provided bearing dataset. This is because the similarity of the feature development in the test bearing is mapped to an early stage of the training bearings, which leads to overpredicting the RUL. Data mapping methods such as the CNN-RUL, which are not built on physics, have a good chance of predicting highly varying RUL depending on the input.

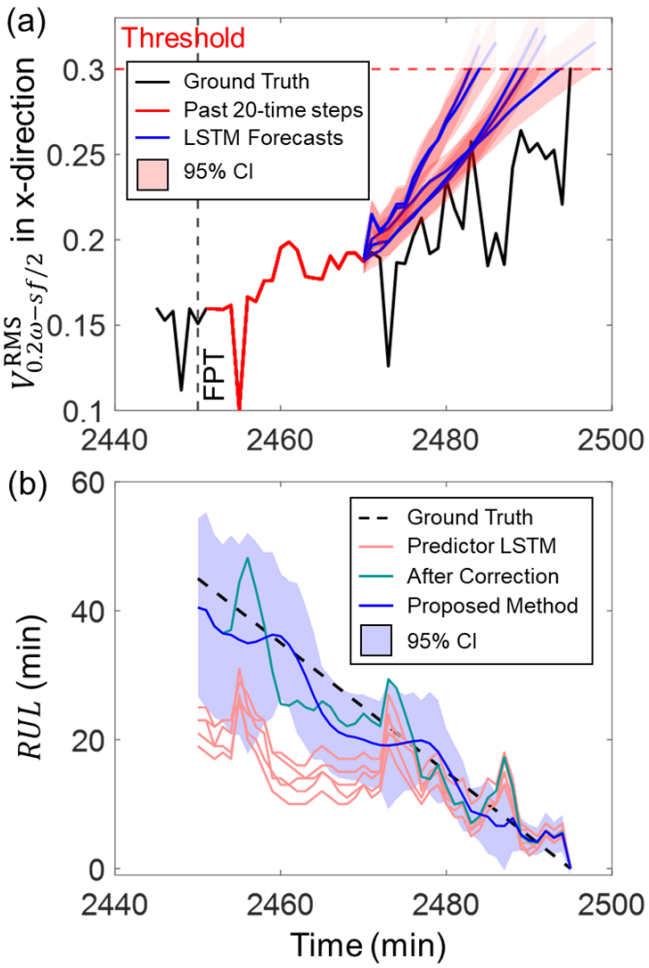

Figure 10: (a) Forecast of the feature $V^{RMS}_{0.2\omega-sf/2}$ by the PLSTM and (b) RUL prediction results for Bearing 3_2.

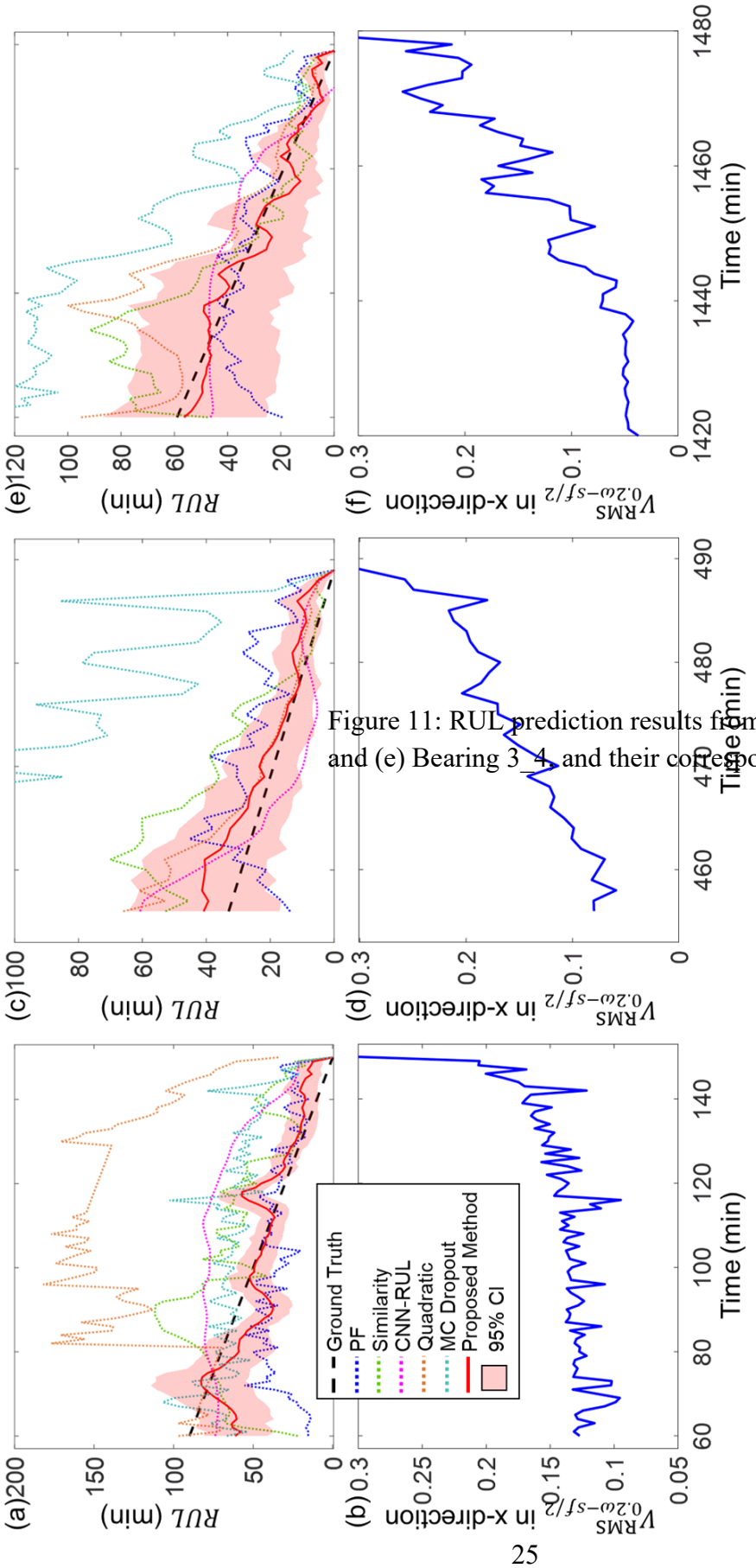

Figure 11: RUL prediction results from various models for (a) Bearing 1_3, (c) Be
and (e) Bearing 3_4, and their corresponding $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ in (b), (d) and (f) respecti

In Table 5, we compare the proposed model to several probabilistic RUL prediction models, namely optimized particle filter (section 2.3.3) and Bayesian-like Monte-Carlo (MC) Dropout [44]. The models are evaluated using the metrics defined in section 3.3.2 with $\alpha = 30\%$ in addition to NLL (eqn. 11). Each entry of Table 5 is the t-distributed 95% confidence interval of all the test bearings. Each model is run independently for five times to ensure consistency. First, the non-Bayesian EnPLSTM model performs at least as good if not better when compared to MC Dropout as also concluded by Refs. [56], [73]. Moreover, execution of MC Dropout for prognostics takes considerably longer time than EnP/CLSTM. For example, the execution of trained MC Dropout models on an Intel Core i5 processor with 16GB RAM, computing the entire prognostic curve for Bearing 3_2 (Figure 10), takes about 5 minutes whereas the EnP/CLSTM takes less than 30 seconds. Also, MC Dropout is observed to over-predict the RUL and hence has a low PEP value (see Figure 11). On the other hand, both PLSTM and EnPLSTM models provide more conservative RUL estimates and hence have high PEP. Low $wtRMSE$ values of both PLSTM and EnPLSTM models indicate that these models have better accuracy in predicting RUL close to EOL. However, the NLL of the PLSTM is larger as this model only accounts for the aleatoric uncertainty and fails to provide good RUL predictions especially at the onset of bearing degradation.

Table 5: Evaluation metrics for various probabilistic models

	$RMSE$ (min)	$wtRMSE$ (min)	α-accuracy %	β-probability -	PEP %	NLL -
Particle Filter	34.0 ± 8.9	3.1 ± 0.6	15.2 ± 3.4	0.13 ± 0.04	$\mathbf{60.8 \pm 14.5}$	22.0 ± 10.5
MC Dropout	31.8 ± 10.5	3.6 ± 2.3	23.9 ± 10.1	0.20 ± 0.11	35.5 ± 16.4	$\mathbf{4.8 \pm 1.4}$
PLSTM	23.7 ± 9.2	1.8 ± 0.6	22.4 ± 15.1	0.24 ± 0.11	$\mathbf{68.3 \pm 11.6}$	7.5 ± 2.1
EnPLSTM	21.2 ± 7.8	$\mathbf{1.6 \pm 0.5}$	26.6 ± 8.1	0.27 ± 0.09	$\mathbf{68.4 \pm 14.4}$	6.3 ± 1.9
EnP/CLSTM	$\mathbf{15.9 \pm 5.2}$	$\mathbf{1.4 \pm 1.2}$	$\mathbf{46.1 \pm 14.0}$	$\mathbf{0.35 \pm 0.10}$	$\mathbf{62.4 \pm 10.7}$	$\mathbf{3.8 \pm 1.3}$

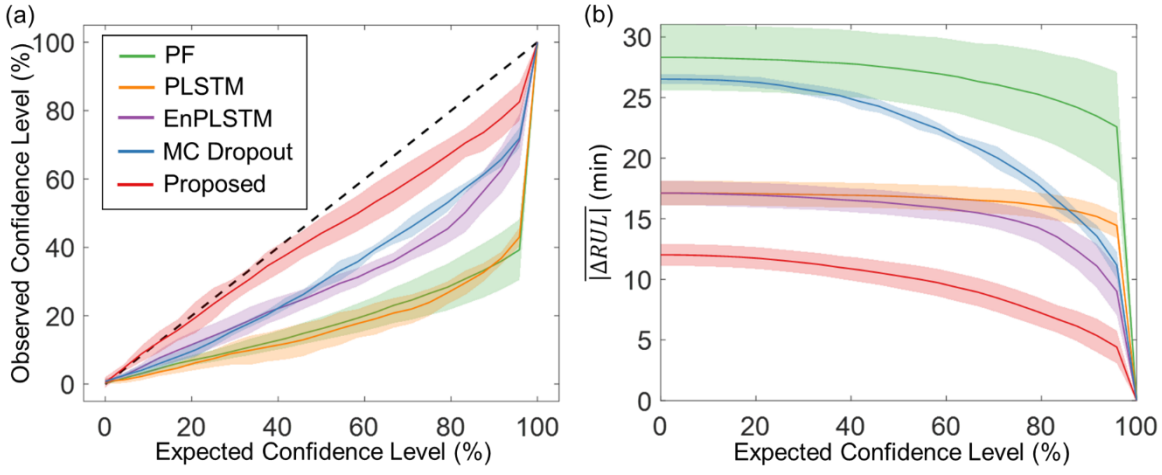

Figure 12: Uncertainty quantification by each of the probabilistic prediction methods. (a) Reliability plot showing the variation of the observed confidence level against the expected confidence level (the black dashed line is the ideal case) and (b) variation of average RUL prediction error for points outside the confidence intervals against the expected confidence level (a consistent inverse relationship is desired).

In practice, it is desired to have accurate uncertainty estimates from a model, particularly in safety-critical applications where the model is used in a decision-making framework. The reliability curve of a perfectly calibrated model will fall on the black dashed line in Figure 12(a), indicating that the observed confidence exactly matches the expected confidence. The PLSTM and PF models in Figure 12(a) exhibit an extreme level of overconfidence in their RUL predictions, i.e. for most of the reliability curve, the model is observed to provide much lower confidence than is asked, or "expected" of it. The low observed confidence of PF stems from large prediction errors whereas for PLSTM, the overconfidence is primarily due to low aleatoric uncertainty in the forecasts albeit lower prediction errors (see Figure 12(b)). The inclusion of the epistemic uncertainty in EnPLSTM leads to a better reliability curve closer to the ideal line. However, after correction, the proposed method is shown to have the best reliability curve of all the probabilistic models, with the least overall deviation from the ideal line. The average absolute prediction error $\overline{|\Delta RUL|}$ in Figure 12(b) is calculated based on the RUL predictions outside the expected confidence intervals for all the bearings. The EnPLSTM and MC Dropout models also exhibit a high level of overconfidence with EnPLSTM having a lower $\overline{|\Delta RUL|}$. In the limit of low confidence level, both PLSTM and EnLSTM have similar $\overline{|\Delta RUL|}$. However, as EnPLSTM also accounts for epistemic uncertainty, $\overline{|\Delta RUL|}$ of EnPLSTM decreases significantly with an increase in the expected confidence level diverging from the $\overline{|\Delta RUL|}$ of PLSTM. The proposed method exhibits the lowest prediction error, indicating that the uncertainty estimates from the proposed model are better calibrated.

Table 6 lists the FPT and EOL for all bearings while also listing the $RMSE$ and $wtRMSE$ values for all the methods used for comparison. The $RMSE$ and $wtRMSE$ entries are color-coded to clearly distinguish the prognostic methods that perform the best for each bearing. The greener the color, the higher the model's prognostic accuracy. The proposed method gives minimum $RMSE$ and $wtRMSE$ values for most of the bearings. The cumulative $RMSE$ and $wtRMSE$ values shown in Table 6 for the different models is calculated similar to eqn. $D.5$ where more importance is given to bearings that have larger prognostic durations. To further compare the performance of all the models across all bearings (when treated as test bearings during cross-validation), we plot the predicted RUL and true RUL for all 741 test samples in Appendix Figure G.2.

Table 6: Comparing the various prognostic methods for all bearings. The prognostic models that are more accurate are shaded Green.

B ID	FPT (min)	EOL (min)	ΔT (min)	RMSE							
				Quadratic	Similarity	CNN-RUL	PF	MC Dropout	PLSTM	En PLSTM	EnP/ CLSTM
1_1	79	121	43	13.5	13.8	7.5	20.1	29.2	13.0	13.0	10.9
1_2	55	96	42	17.9	8.7	8.5	17.1	10.3	12.0	12.0	12.4
1_3	60	150	91	91.6	26.9	23.3	39.4	25.8	23.5	23.5	12.2
1_5	26	41	16	128.8	62.3	40.2	14.0	27.8	14.1	14.1	29.4
2_1	456	489	34	12.1	19.7	16.5	16.2	90.0	6.0	5.9	6.9
2_2	50	154	105	53.8	28.0	33.4	47.3	36.2	34.0	34.0	33.0
2_3	316	398	83	15.2	19.0	20.8	30.1	24.7	14.1	14.1	11.1
2_4	32	35	4	10.7	18.5	45.9	4.6	29.9	7.7	7.7	8.1
2_5	123	199	77	42.7	20.5	33.6	38.4	15.8	16.5	16.4	11.3

3_1	2404	2527	124	62.4	32.7	36.9	56.4	38.0	39.8	39.8	26.0
3_2	2450	2495	46	22.5	16.7	7.8	21.5	8.8	12.1	12.1	3.7
3_3	343	352	10	31.8	25.6	41.4	4.9	35.2	8.4	8.4	7.9
3_4	1420	1479	60	19.9	17.3	23.3	17.2	48.4	4.6	4.0	5.2
3_5[#]	8(20)	25	6	3.3	21.0	1.2	7.9	46.9	5.5	5.5	5.8
Net			741	44.1	23.6	25.3	34.4	31.6	21.0	20.9	16.1

B ID	FPT (min)	EOL (min)	ΔT (min)	wtRMSE							
				Quadratic	Similarity	CNN-RUL	PF	MC Dropout	PLSTM	En PLSTM	EnP/ CLSTM
1_1	79	121	43	1.6	1.7	1.1	2.6	4.4	1.4	1.4	1.5
1_2	55	96	42	1.9	1.4	1.2	2.1	1.5	1.5	1.5	1.7
1_3	60	150	91	10.7	2.7	2.9	3.3	3.1	1.6	1.6	1.0
1_5	26	41	16	19.2	14.2	10.0	3.1	5.5	3.6	3.6	6.2
2_1	456	489	34	1.0	2.4	2.7	2.7	11.4	0.8	0.8	0.9
2_2	50	154	105	6.3	1.9	2.6	3.2	2.5	2.3	2.3	2.5
2_3	316	398	83	1.4	1.6	1.5	2.5	2.0	1.2	1.2	1.0
2_4	32	35	4	3.2	5.8	13.7	1.7	8.4	2.2	2.2	2.1
2_5	123	199	77	4.8	2.6	4.0	4.1	1.0	1.1	1.1	0.8
3_1	2404	2527	124	5.2	2.1	2.0	4.4	3.8	2.3	2.3	1.4
3_2	2450	2495	46	1.6	1.4	0.8	2.9	1.0	1.2	1.2	0.4
3_3	343	352	10	3.2	4.0	8.3	1.5	10.4	2.1	2.1	1.8
3_4	1420	1479	60	2.0	1.6	2.7	1.8	4.8	0.5	0.5	0.6
3_5[#]	8(20)	25	6[#]	2.4	7.4	0.6	1.4	15.2	2.6	2.6	2.6
Net			741	4.7	2.3	2.6	3.1	3.5	1.6	1.6	1.4

[*]Bearing 1_4 undergoes a sudden catastrophic failure and is therefore not shown. [#]Bearing 3_5: Although FPT is at 8 min mark, at least 20 data points are needed for prediction using the deep learning models

3.3.5 Discussion on the Advantages of EnPLSTM

While Bayesian-like techniques tend to provide uncertainty around a single-mode, deep ensemble models explore diverse modes within the same function space [73]. Typically, deep ensemble models are generated with random initializations which when trained on the same training dataset, take different optimization trajectories in trying to describe the function space. In this paper, the function space corresponds to feature forecasting for bearing prognostics. The PLSTMs trained with different initializations have vastly dissimilar weights, as shown by the cosine similarity plot in Figure 13(a), even though the NLL loss (eqn. 11) of each of these models is similar. Here, the cosine similarity of a pair of trained models with parameters θ_i and θ_j is defined as $(\theta_i \cdot \theta_j)/\left(||\theta_i|| \, ||\theta_j||\right)$. Each individual PLSTM model can therefore be hypothesized to have obtained different but related optimum modes within the function space which is also the reason for obtaining different forecast trajectories in Figure 10(a). To show this, we plot the t-Distributed Stochastic Neighbor Embedding (t-SNE) [74] of the $V^{\text{RMS}}_{0.2\omega-sf/2}$ forecasts on Bearing 2_1 (cross-validation Fold-1) for three representative PLSTM models from Figure 13(a). Each datapoint on

the t-SNE plot in Figure 13(b) corresponds to forecasting forty time steps ahead of the current measurement. At the beginning of the training process (epoch = 0), all three PLSTM models have distinct weights, a result of the random weight initialization process. After training for 80 epochs, each of the PLSTM model forecasts is observed to approach the true forecast distribution through different optimization routes (see Figure 13(b)). The size of the squares in Figure 13(b) is proportional to the aleatoric uncertainty in the forecast of each of the PLSTM models. The origin of epistemic uncertainty is precisely what is observed in Figure 13(c). The different model weight initializations lead to different trained models which lead to slightly different RUL predictions. The model-to-model variation in model weights and hence forecasts/RUL predictions directly quantifies the epistemic uncertainty. For samples that are outside the distribution of the training data, each PLSTM model predicts high aleatoric uncertainty which, when combined into an ensemble, provides an even larger epistemic uncertainty (Figure 13(c)). When determining the RUL of bearings, if the time series describing the test bearing health condition is not seen during the training process, the proposed model would predict large uncertainties (both aleatoric and epistemic) indicating the model's lack of confidence in such an RUL prediction. In the case of a single model, there is no way to determine whether or not it has obtained a best forecast/RUL prediction, and therefore no way to quantify the epistemic uncertainty in its prediction. This is why a single data-driven model for prognostics should not be trusted.

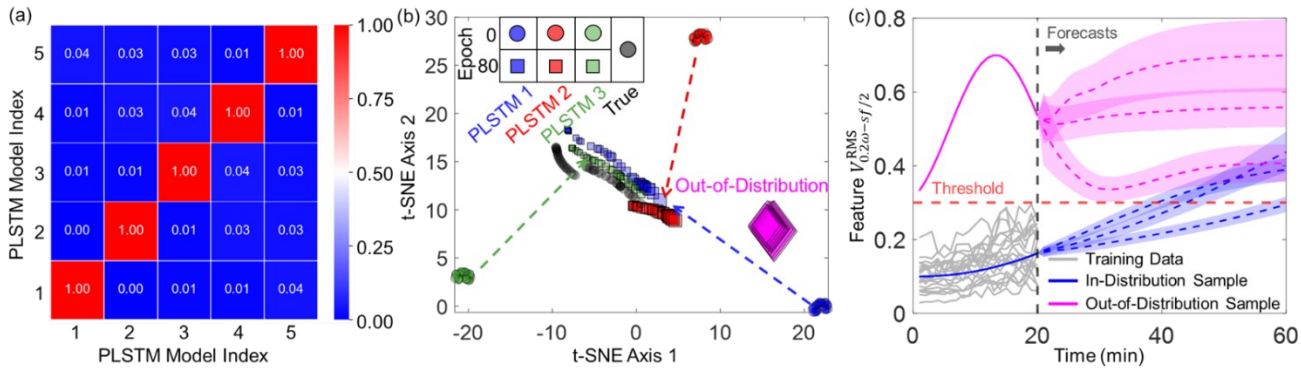

Figure 13: (a) Cosine similarity of the weights of five PLSTM models trained for 80 epochs with different model weight initializations on the same training dataset. (b) t-SNE plot of forecasting 40 time-steps of $V_{0.2\omega-sf/2}^{RMS}$ by three representative PLSTM models at epochs 0 and 80 for Bearing 2_1. The ground truth of $V_{0.2\omega-sf/2}^{RMS}$ is also shown along with a sinusoidal out-of-distribution sample. The sizes of the squares and diamonds are proportional to the aleatoric uncertainty in forecast. (c) Forecasting of the three PLSTM models for a sample within the training data distribution and the sinusoidal out-of-distribution sample.

3.3.6 Discussion on the Advantages of EnCLSTM

The EnPLSTM model often underpredicts the true RUL, especially at the beginning of bearing failure. This is true even for the bearings used to train the PLSTM as described in section 2.5.3. The main purpose of the EnCLSTM model is to correct this error and provide a more accurate RUL estimate. Figure 14(a), shows the variation of the normalized RUL error prediction obtained from one PLSTM against the feature $V_{0.2\omega-sf/2}^{\mathrm{RMS}}$ for both the training and testing datasets of a representative cross-validation fold, Fold-3. The circle symbol size in Figure 14(a) is proportional to the uncertainty in prediction. At low $V_{0.2\omega-sf/2}^{\mathrm{RMS}}$ values, indicative of the onset of bearing

degradation, the predicted error and uncertainty are large. Ideally, the model error should not vary with time. However, in the case of RUL prediction, almost any model may exhibit high errors close to the FPT and then gradually increase in accuracy as the model is able to process more data over time. This is particularly true for LSTMs as they store relevant temporal information in their network architecture which is used at a later time to improve prediction accuracy. The errors in Figure 14(a) exhibit a relatively clear decreasing trend with $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$, and for this reason, the error can be learned by another model. Error correction, delta-learning, and residual learning [75], [76] are all names for these types of models which have been proposed for the same task of correcting model predictions using learned errors. Therefore, a data mapping based correction model CLSTM would help reduce the prediction error ΔRUL especially when combined with a weighting function $W\left(F = V^{\mathrm{RMS}}_{0.2\omega-sf/2}\right)$ as mentioned in eqn. 15. However, this approach would only work if the training dataset and the testing dataset have similar input/output distributions. As shown in Figure 14(a), the training dataset (black) and the test dataset (blue) are found to have similar error distributions (output of CLSTM). The error in RUL from EnPLSTM is due to the accumulated uncertainty when forecasting $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$. A t-SNE plot in Figure 14(b) reveals that the $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ feature distributions of the training and testing datasets are also similar and the symbol size, which is proportional to the uncertainty of the next step prediction ($\sigma_{\mathrm{p}}^{k+1}$ from Figure 4), also indicate that the magnitude of aleatoric uncertainties at $(k+1)$ time step are similar across training and testing datasets. However, for samples that are out of distribution, like the artificially generated sinusoidal-like time series shown as red circles in Figure 14(b), the uncertainty is large at the $(k+1)$ time step even from a single PLSTM model (aleatoric uncertainty). When considering an ensemble, several PLSTM model disagreements in the forecast lead to an even larger epistemic uncertainty proving the effectiveness of the ensemble method in determining non-confident predictions.

The t-SNE plot in Figure 14(c) compares the train and test distributions of the EnCLSTM input which consists of $k = 20$ lookback time steps of $V^{\mathrm{RMS}}_{0.2\omega-sf/2}$ and RUL predictions from EnPLSTM (see Table 3). The symbol size in Figure 14(c) is proportional to the EnPLSTM RUL prediction error $RUL^{\mathrm{true}} - RUL\left(\mu_{*\mathrm{p}}, \sigma_{*\mathrm{p}}\right)$ for the training dataset and predicted error correction $\Delta RUL(\mu_{*\mathrm{c}}, \sigma_{*\mathrm{c}})$ for the testing dataset. Figure 14(c) indicates that the EnCLSTM inputs as well the magnitude of RUL corrections of the testing dataset are similar to the training dataset. The overlapping of the two datasets in the t-SNE space is a good indication of their distribution similarity which makes the predictions from the EnCLSTM model trustworthy. We further compare the predicted $\Delta RUL(\mu_{*\mathrm{c}}, \sigma_{*\mathrm{c}})$ to that of true RUL errors of EnPLSTM in Figure 14(d), where, the horizontal and vertical error bars correspond to variation in RUL prediction errors from the EnPLSTM and EnCLSTM models for five independent runs, respectively. Ideally, EnCLSTM would predict the exact RUL error of EnPLSTM leading to a perfect RUL prediction model. However, the predictions from EnCLSTM deviate from the ideal line, indicating the model was not able to perfectly predict the RUL error. Regardless, when compared to the EnPLSTM model (i.e. without the correction term), the EnCLSTM model provides largely improved predictions of RUL error as evidenced by a significant improvement in the overall RUL evaluation metrics for EnP/CLSTM in Table 5. Even though the EnCLSTM model provides accurate predictions of RUL error, it is still susceptible to making errant predictions because of noise in the data. The implementation of weighted correction (eqn. 15) and temporal fusion (eqn. 17) restrict the influence of sudden noise spikes in the error correction predictions which are sometimes observed

for data-mapping models like CLSTM. Although the analysis pertaining to Figure 14 is described for Fold-3, we find similar observations for all the cross-validation folds giving confidence in the effectiveness of reducing the prediction error through the implementation of EnCLSTM.

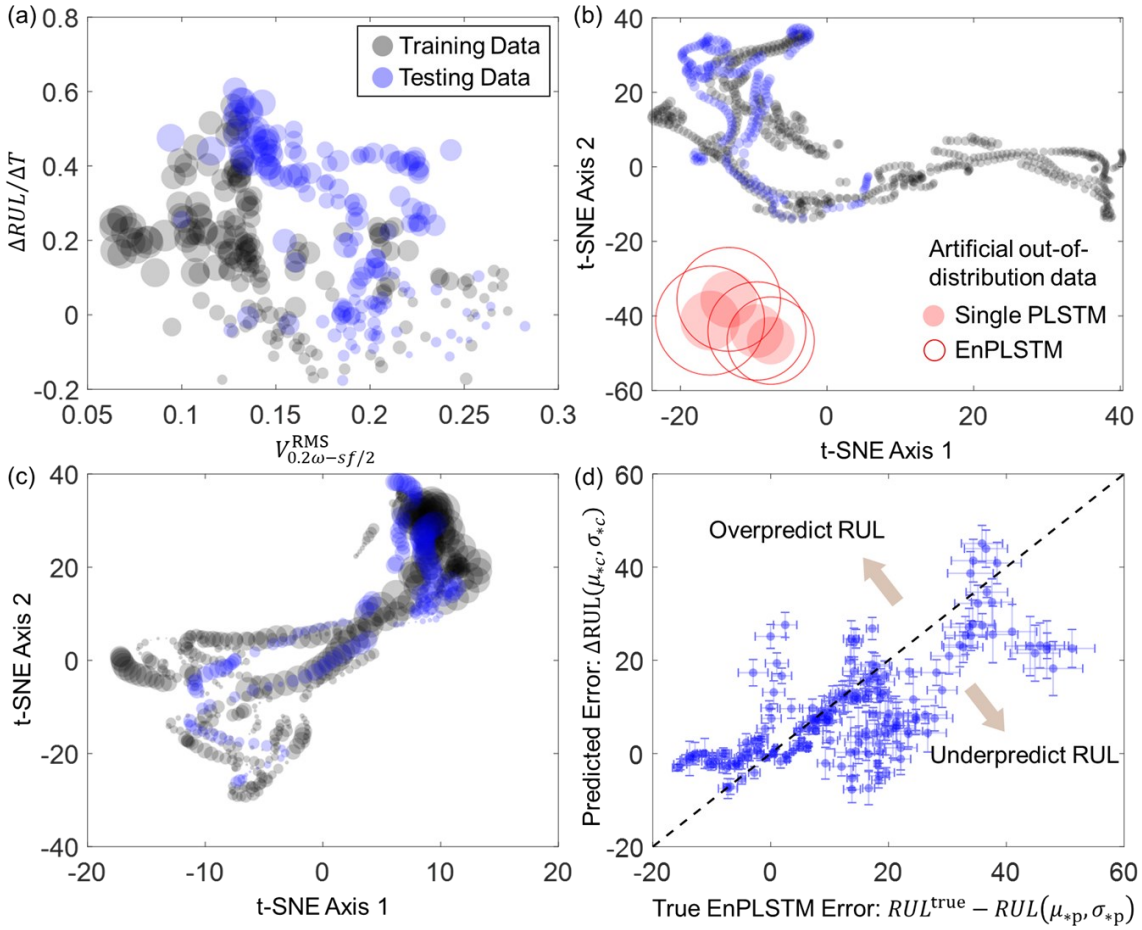

Figure 14: (a) Error in RUL prediction for representative training and testing bearings from Fold-3 with the size of a circle proportional to the standard deviation of the RUL prediction, σ_m, by one PLSTM model. (b) t-SNE plot of training and testing data for Fold-3, where each point corresponds to k-time steps of $V^{RMS}_{0.2\omega-sf/2}$. The symbol size is proportional to the standard deviation of the next-step feature prediction σ_p^{k+1} from the single PLSTM model used in (a). For the out-of-distribution samples, standard deviations of both single PLSTM and EnPLSTM are shown. (c) t-SNE plot of the input to the CLSTM model where the circle size is proportional to the RUL error. (d) Comparing EnCLSTM RUL prediction error to the true prediction error of EnPLSTM. The horizontal and vertical error bars represent the variation in RUL prediction error from EnPLSTM and EnCLSTM for five runs respectively.

4. Conclusion

High productivity demands on modern-day machinery require intelligent solutions to avoid machine downtime and prevent catastrophic failures. In this paper, we present an ensemble approach to bearing prognostics that not only provides probabilistic RUL predictions but is also lightweight, making it suitable for embedding on IIoT platforms. To make our work more

industrially relevant, we adopt the ISO standards for defining bearing failure, which is established in the velocity domain. We also incorporate physics by capturing energy-based features in the velocity domain (in the form of RMS) that reflect both characteristic bearing fault frequencies and overall bearing health. Unlike purely data-driven algorithms, the inclusion of bearing failure physics has the potential to generalize our approach to other bearings in different working conditions.

The proposed algorithm is built upon a vanilla LSTM model with an added Gaussian layer to forecast an RMS feature (obtained from the velocity domain) while also obtaining the aleatoric uncertainty of such a forecast. The proposed algorithm consists of three major steps: (1) a predictor step PLSTM, where the feature is forecasted to a certain threshold by doing a one-step-ahead prediction and marching in time, (2) a corrector step CLSTM, which offsets the RUL prediction obtained from the predictor step and (3) temporal fusion, which effectively smoothens the RUL prediction based on the recent history of predictions. The proposed algorithm also uses an ensemble approach EnP/CLSTM because the limited amount of available bearing run-to-failure data causes deep learning models to train differently every time. By combining RUL predictions from models with a similar architecture that have been trained on the same dataset but with different initial conditions, we can capture the epistemic uncertainty in our predictions.

Using a publicly available dataset, we show the superiority of our proposed model, in terms of accuracy as well as uncertainty quantification, when compared to other traditional models such as particle filter, similarity-based approaches, CNN-RUL correlation, Bayesian-like MC Dropout, and simple regression techniques. The proposed EnP/CLSTM model reduces the $RMSE$ and $wtRMSE$ by at least 50% when compared to Bayesian-like MC Dropout. To compare the uncertainty capability of models we introduce α-accuracy, β probability, and percentage of early prediction (PEP) metrics. The proposed model ensures around 50% of the RUL prediction points lie within the 30% α-accuracy region which is superior to all other models. In general, the LSTM-based models make conservative RUL predictions with high PEP. The proposed method has one order of magnitude faster execution time when compared to MC Dropout making it feasible for IIoT applications.

The proposed predictive approach was developed in collaboration with Grace Technologies with an IIoT deployment in mind, and the authors are in the process of implementing it for commercial use inside the GraceSense™ Vibration & Temperature Node. The main benefit of this embedded deployment is to reduce the need to wirelessly transmit raw acceleration data – in exchange for a small amount of additional computational capability and time. In a GraceSense™ deployment, this results in a greater than 10,000X reduction in transmission requirements, which eliminates problems stemming from overcrowding of the 2.4GHz band in industrial facilities and can allow a vibration node to predict the remaining useful life of a bearing once per hour for up to five years without needing a change of battery. This represents at least a 50X improvement in battery life for this node.

Acknowledgments
The authors would like to thank Carey Novak, Sheng Shen, and Bryce Brewer for helpful discussions. This work was supported in part by the Regents Innovation Fund that is part of the Proof-of-Concept Initiative at Iowa State University, Grace Technologies, and the U.S. National

Science Foundation under Grant IIP-1919265. Any opinions, findings, or conclusions in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

Appendix A: Feature Extraction and Selection

The various models (such as CNN) use different features in addition to the velocity RMS values within certain frequency ranges. We, therefore, extract the following data/physics-based features from both the time and frequency domains

- Time-domain features: max amplitude, RMS, kurtosis
- Frequency domain features:
    - BPFO fault frequency max amplitude and RMS: $1\times, 2\times$ or $3\times$ $BPFO \pm 5\%$ Hz
    - BPFI fault frequency max amplitude and RMS: $1\times, 2\times$ or $3\times$ $BPFI \pm 5\%$ Hz
    - BSF fault frequency max amplitude and RMS: $1\times, 2\times$ or $3\times$ $BSF \pm 5\%$ Hz
    - RMS within the frequency ranges to capture FTF $0.2\omega - 0.8\omega$, shaft frequency $0.8\omega - 1.2\omega$, two harmonics of shaft frequency $1.2\omega - 3.2\omega$, entire frequency range $0.2\omega - sf/2$, frequency range after shaft frequency $1.2\omega - sf/2$, bearing fault frequencies $\text{BFF} = 0.9\min(\text{BPFO}, \text{BPFI}, \text{BSF}) - sf/2$.

The above-listed 27 features are calculated in the radial direction for both directions in the velocity, acceleration, and jerk domains making a total of 162 ($= 27 \times 2 \times 3$) features. The code for feature extraction has been provided at https://github.com/VNemani14/Bearing_LSTMPrognostics.

In bearing prognostics, the true RUL of the bearing is defined to decrease linearly with time from the FPT to the EOL. The goal of feature selection is to identify features that contain strong information regarding the bearing health condition while discarding other features. Selecting features that have a strong linear behavior correlates well to true RUL, thus enabling an accurate RUL estimate. To this end, we use two criteria to determine a score for each feature and select features with the best scores. The two criteria used are (1) Monotonicity and (2) Pearson correlation coefficient for testing the linearity. Monotonicity is defined in terms of the feature to have either a continuously increasing or decreasing characteristic, given in terms of counting the differential of each feature F_i with a total of T observations.

$$\text{Mon}_\text{i} = \left| \frac{Num(dF_i > 0)}{T-1} - \frac{Num(dF_i < 0)}{T-1} \right| \qquad A.1$$

In the literature, the RUL of a bearing is always treated as a straight line between the onset of bearing degradation and its EOL. Therefore, a Pearson correlation coefficient is used to determine the linear correlation between each feature F_i and RUL. This can be defined as

$$\text{Cor}_\text{i} = \frac{\left|\sum_{t=1}^{T}(F_i^t - F_i^1)\,(RUL^t - RUL^1)\right|}{\sqrt{\sum_{t=1}^{T}(F_i^t - F_i^1)^2 \sum_{t=1}^{T}(RUL^t - RUL^1)}} \qquad A.2$$

The final score is an average of the above two selection criteria, expressed as

$$\text{score}_\text{i} = \frac{\text{Mon}_\text{i} + \text{Cor}_\text{i}}{2} \qquad A.3$$

Top 24 features are selected from the total of 162 features based on the score. Each selected feature is subjected to moving average smoothing with a lookback window size of three. In other words, the smoothened feature value after moving averaging is the average of the current measurement with two measurements from the recent past. Each of the features is then normalized and averaged to determine the health index $\text{HI}(t) \in (0,1)$ where a health index of one refers to a perfectly healthy bearing.

Appendix B: First Prediction Time (FPT) Determination

At the beginning of operation, machines are often healthy considering a good initial setup. After a certain duration of operation, the bearings will start to degrade, and the signature of the degradation process can be determined from the vibration signals. The FPT is the time at which the beginning of bearing degradation is evident and is also the time at which prognostics is triggered. Predicting the RUL prior to the onset of degradation is unrealistic and not practical as there is little to no fault signature in the observed data. In this study, we use the RMS in the velocity domain pertaining to the beginning of characteristic bearing fault frequencies $V^{\mathrm{RMS}}_{\mathrm{BFF}-sf/2}$ to determine the FPT. We use the 2σ method also used in previous bearing prognostics literature [35], with the difference being we use $V^{\mathrm{RMS}}_{\mathrm{BFF}-sf/2}$ to employ the 2σ criterion instead of kurtosis. During the early machine life, the mean $\mu_{V^{\mathrm{RMS}}}$ and standard deviation $\sigma_{V^{\mathrm{RMS}}}$ are determined, and then the FPT is obtained whenever $V^{\mathrm{RMS}}_{\mathrm{BFF}-sf/2}$ crosses the threshold of $\mu_{V^{\mathrm{RMS}}} + 2\sigma_{V^{\mathrm{RMS}}}$ for two consecutive observations. In other words, the FPT is the time t_{FPT} at which

$$\left|V^{\mathrm{RMS}}_{\mathrm{BFF}-sf/2}(t_{\mathrm{FPT}} - j) - \mu_{V^{\mathrm{RMS}}}\right| > 2\sigma_{V^{\mathrm{RMS}}}, \qquad j = 0 \text{ and } 1 \qquad B.1$$

Appendix C: CNN Model Architecture

Table C.1 and C.2 show the architecture of the CNN model described in section 2.3.1.

Table C.1: Convolution Block Architecture

Layer
Convolution-1D
Batch Normalization-1D
Leaky ReLU

Table C.2: Convolution Network Architecture

Layer	Output Shape	# Parameters
Convolution Block 1	(Samples, 32, 19)	4,704
Convolution Block 2	(Samples, 32, 17)	3,168
Convolution Block 3	(Samples, 32, 15)	3,168
Convolution Block 4	(Samples, 64, 13)	6,336
Convolution Block 5	(Samples, 64, 11)	12,480
Convolution Block 6	(Samples, 64, 9)	12,480
Dropout	Probability = 0.10	
Dense	(Samples, 64)	36,928
Dense – Output	(Samples, 1)	65
Total:		79,329

Appendix D: Particle Filter:

A nonlinear state-space model can be defined in terms of the system state vector $x(t)$, system model parameters $\theta(t)$ and noisy observations $y(t)$ given as

State transition equation:

$$x_t = f(x_{t-1}, \theta_{t-1}) + u_i, \qquad \theta_t = \theta_{t-1} + r_t \qquad D.1$$

Measurement equation:

$$y_t = g(x_i, \theta_i) + v_t \qquad D.2$$

where $f(\cdot,\cdot)$ is the state transition function, $g(\cdot,\cdot)$ is the measurement function, u_t is the process noise for the system states, r_t is the process noise for model parameters and v_t is the measurement noise with the subscript indicating the time at which the system equations are evaluated.

The posterior probability distribution functions (PDFs) of the states given the past observation $p(x_t|y_{1:t})$ can be posed as a Bayesian interference problem. In PF, the posterior PDFs are determined based on the Monte Carlo method by utilizing a set of particles and associated weights that are updated with every measurement. Following the theoretical background presented in [67], [77]–[79], the posterior PDF can be stated as

$$p(x_t|y_{1:t}) \approx \sum_{j=1}^{N_p} w_t^j \delta\left(x_t - x_t^j\right) \qquad D.3$$

where x_t^j and w_t^j are the j^{th} particle state and weights at the time t and N_p is the number of particles. The weights w_t^j are determined by using the importance density function which is often chosen to be equal to prior pdf. Based on this assumption, the weight update equation can be stated as

$$w_t^j \propto w_{t-1}^j p\left(y_t|x_t^j\right) \qquad D.4$$

Following the discussion from section 2.2.3, to optimize the initial states $\{a_0, b_0, c_0\}$, we use the Latin hypercube sampling (LHS) technique [80] to generate a set of random initial state parameters within certain bounds. For each set of initial state parameters, $wtRMSE$ is calculated for the RUL prediction for each of the bearing in training dataset. A final score is calculated by combining the $wtRMSE$ of all the training bearings using the equation

$$S_{RMSE} = \sum_{i=1}^{n_{\text{train}}} \left(\text{EOL}_i - t_{\text{FPT}_i} + 1\right) \times wtRMSE_i \qquad D.5$$

where n_{train} is the number of bearings in the training dataset and the pre-factor $(\text{EOL} - t_{\text{FPT}} + 1)$ is a measure of the time duration between the FPT and EOL. Bearings that trigger prognostics for a longer duration are given importance. The overall algorithm regarding LHS-based optimization and PF methodology is presented in Table D.1.

Table D.1: Algorithm for LHS optimized PF that determines RUL of a test bearing after optimizing the initial parameters using the training bearing dataset.

Algorithm: LHS optimized PF for bearing prognostics	
Inputs:	$y(t)$ – measured feature for bearings k – lookback time p_0 – initial probability distribution of states $\{a, b, c\}$ N_p – number of particles
Output:	RUL(t) for test bearing
1	**Latin hypercube sampling**: Generate N_{LHS} samples of $\{a_0, b_0, c_0\}$
2	**for** $n = 1$ to N_{LHS}
3	**for** $i = 1$ to n_{train}

4 **Initialize** N_p particles around $\{a_0, b_0, c_0\}_{j=1:N_p}^{j}$ and assign equal weights $w_0^j = 1/N_p$.
5 **for** $t = \text{FPT}_i$ to EOL_i
6 **for** $j = 1$ to N_p
7 **Evaluate** state transition eqn. 25
8 **Update** weights of the particles using eqn. $D.4$
9 **Calculate** $RUL_i^j(t)$ using eqn. 27
10 **end for** (from line 6)
11 **for** $j = 1$ to N_p
12 Normalize weights $w_t^j = w_t^j / \sum_{j=1}^{N_p} w_t^j$
13 **end for** (from line 11)
14 Calculate $RUL_i(t)$ using eqn. 28
15 **Multinomial resampling** Ref. [81]
16 **Assign** equal weights to the resampled particles
17 **end for**
18 **end for**
19 **Calculate** score of this LHS sample S_{RMSE}^n using eqn. $D.5$
20 **end for**
21 **Identify** optimally $\{a_0^*, b_0^*, c_0^*\}$ by $\min\{S_{RMSE}^n\}_{n=1:N_{LHS}}$
22 **for** each test bearing
23 **Initialize** N_p particles similar to line 4 but with $\{a_0^*, b_0^*, c_0^*\}$
24 **Determine** RUL(t) by modifying lines 5–17 with while loop instead to determine EOL
25 **end for**

Appendix E: Monte Carlo (MC) Dropout

MC Dropout is reported to have Bayesian-like behavior [44]. The basic model for MC Dropout is similar to that of the PLSTM model and is shown in Table E1 (Table 1 without the Gaussian layer). The dropout value is set at 0.1. A single model is trained but is run multiple times with the 10% dropout to achieve an uncertainty estimate. The code for implementation of a single MC Dropout is provided at https://github.com/VNemani14/Bearing_LSTMPrognostics where RUL of a test bearing is determined by model training followed by forecasting by marching in time till $V_{0.2\omega-sf/2}^{\text{RMS}}$ reaches the cutoff.

Table E1: Architecture of the MC Dropout model

Layer	Output shape	# Parameters
Input layer	(Samples, 20, 1)	0
LSTM	(Samples, 60)	14,880
Dense	(Samples, 20)	61
Total:		14,941

Appendix F: Regression Fitting (Quadratic and Exponential)

At every instant in time t, regression fitting is performed by considering the past $k = 30$ time steps of feature data $F = V_{0.2\omega-sf/2}^{\text{RMS}}(t-k+1 \rightarrow t)$ consistent with the rest of the models. The quadratic model used to model the degradation trend can be stated as [82]:

$$F(t) = m_1 t^2 + m_2 t + m_3 \qquad F.1$$

where $F(t)$ represents feature value at a time t. Unknown parameters $m1, m2, m3$ are determined by the ordinary least squares method.

A double exponential model [83] is also used for regression. The mathematical formula of the exponential model can be written as:

$$F(t) = ae^{bt} + ce^{dt} \qquad F.2$$

a, b, c, and d are four unknown parameters identified by the nonlinear least square curve fitting method.

To predict the bearing RUL, the fitted degradation curve is extrapolated up to the predefined failure threshold of 0.3 ips. The bearing RUL at the current inspection time t is given as:

$$\text{RUL}(t) = T_{\text{EOL}} - t \qquad F.3$$

where T_{EOL} is the time when the extrapolated degradation curve first reaches 0.3ips. We would like to note that for the particular selected feature derived from the XJTU-SY bearing dataset, both the double exponential and quadratic regression fitting do not provide satisfactory results. Thus, we only show the $RMSE$ and $wtRMSE$ of the quadratic regression fitting in Table 5.

Appendix G: Comparing Model Training and RUL Prediction Results

Figure G.1 shows the learning curves of (a) PLSTM and (b) CNN model for Fold-4. Among the 12 training bearings, 2 bearings are used for validation. Both the learning curves indicate model convergence with no overfitting.

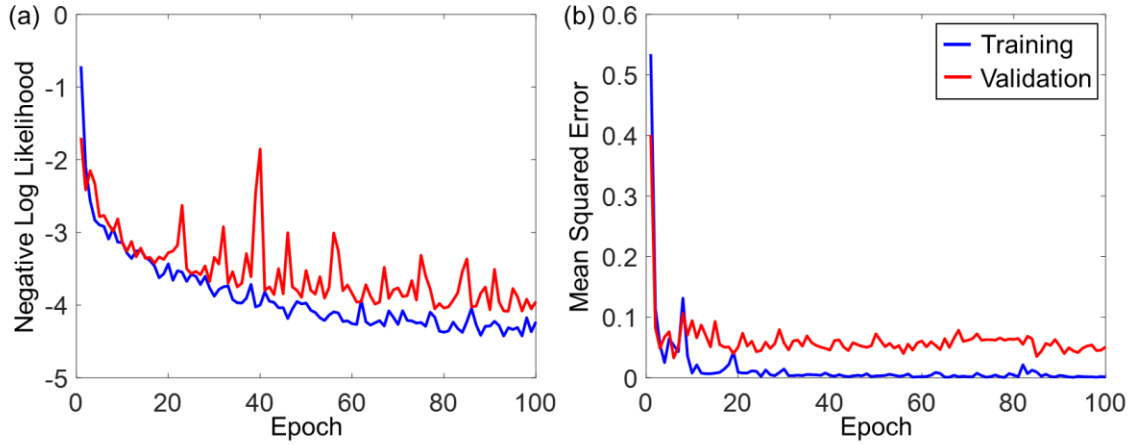

Figure G.1: Learning curve of (a) PLSTM with NLL loss function and (b) CNN-RUL with mean squared error loss function.

Figure G.2 compares the RUL predictions of all the models across all bearings (when treated as test bearings during cross-validation). Note that the test samples are sorted in the ascending order of the true RUL. We observe that (1) the proposed model provides a more conservative prediction and (2) the prediction is centered around the true RUL especially when the RUL has a low value, indicating the convergence of the model towards the true RUL when the bearing is approaching failure. Having a conservative prediction is critical from a maintenance perspective to avoid false negatives.

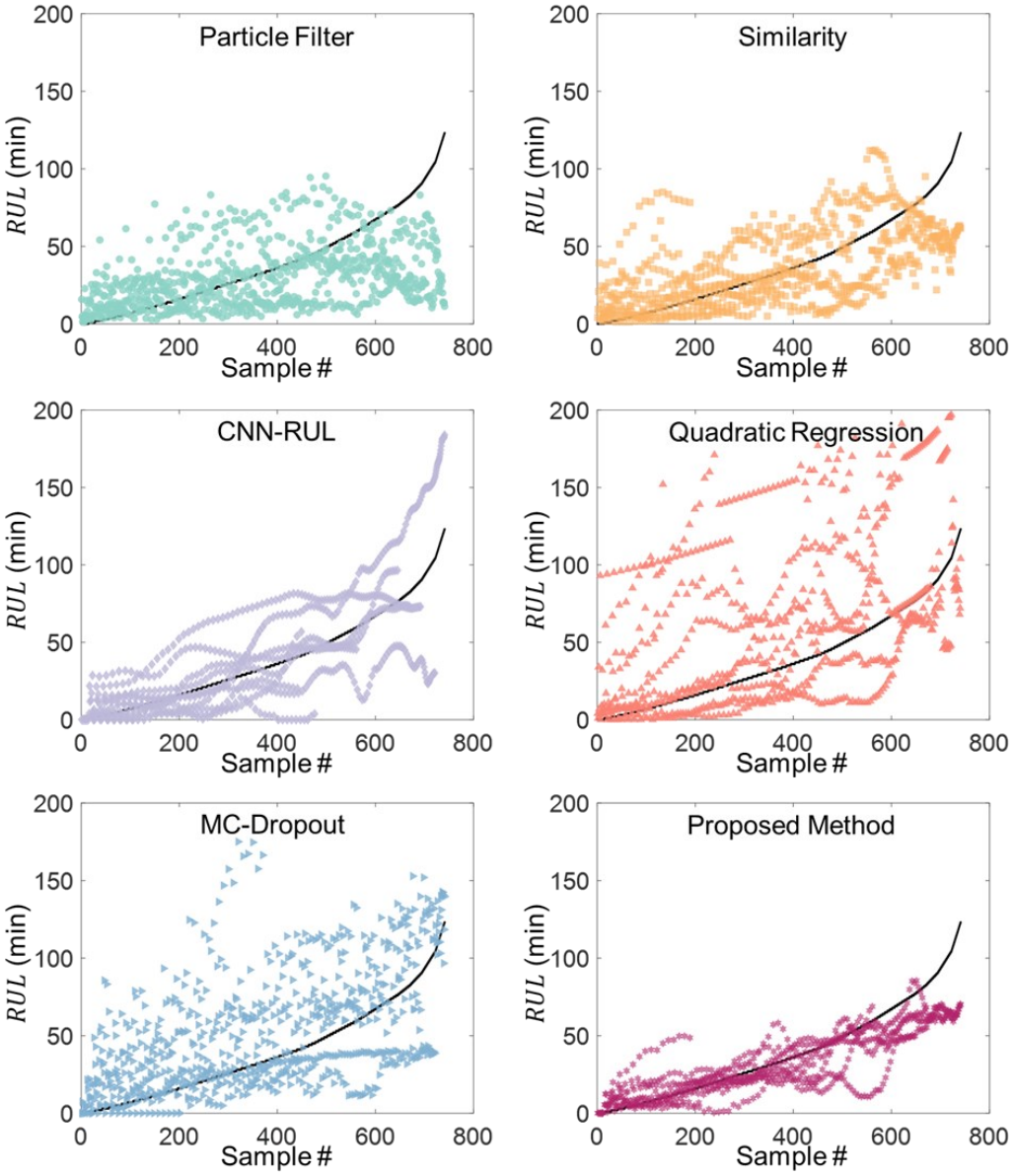

Figure G.2: Performance of all the models with all 15 bearings as test bearings during the 5-fold cross-validation study.

References

[1] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech. Syst. Signal Process.*, vol. 104, pp. 799–834, May 2018, doi: 10.1016/j.ymssp.2017.11.016.

[2] J. Z. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mech. Syst. Signal Process.*, vol. 25, no. 5, pp. 1803–1836, Jul. 2011, doi: 10.1016/j.ymssp.2010.11.018.

[3] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far, and K. Khorasani, "Failure Prognosis and Applications—A Survey of Recent Literature," *IEEE Trans. Reliab.*, pp. 1–21, 2019, doi: 10.1109/TR.2019.2930195.

[4] M. Behzad, H. A. Arghan, A. R. Bastami, and M. J. Zuo, "Prognostics of rolling element bearings with the combination of paris law and reliability method," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, Jul. 2017, pp. 1–6. doi: 10.1109/PHM.2017.8079187.

[5] J. Wu, C. Wu, S. Cao, S. W. Or, C. Deng, and X. Shao, "Degradation Data-Driven Time-To-Failure Prognostics Approach for Rolling Element Bearings in Electrical Machines," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 529–539, Jan. 2019, doi: 10.1109/TIE.2018.2811366.

[6] D. Wang and K. Tsui, "Statistical Modeling of Bearing Degradation Signals," *IEEE Trans. Reliab.*, vol. 66, no. 4, pp. 1331–1344, Dec. 2017, doi: 10.1109/TR.2017.2739126.

[7] M. Sadoughi and C. Hu, "Physics-Based Convolutional Neural Network for Fault Diagnosis of Rolling Element Bearings," *IEEE Sens. J.*, vol. 19, no. 11, pp. 4181–4192, Jun. 2019, doi: 10.1109/JSEN.2019.2898634.

[8] S. Shen *et al.*, "A physics-informed deep learning approach for bearing fault detection," *Eng. Appl. Artif. Intell.*, vol. 103, p. 104295, Aug. 2021, doi: 10.1016/j.engappai.2021.104295.

[9] W. K. Yu and T. A. Harris, "A New Stress-Based Fatigue Life Model for Ball Bearings," *Tribol. Trans.*, vol. 44, no. 1, pp. 11–18, Jan. 2001, doi: 10.1080/10402000108982420.

[10] A. Muetze and E. G. Strangas, "The Useful Life of Inverter-Based Drive Bearings: Methods and Research Directions from Localized Maintenace to Prognosis," *IEEE Ind. Appl. Mag.*, vol. 22, no. 4, pp. 63–73, Jul. 2016, doi: 10.1109/MIAS.2015.2459117.

[11] R. K. Singleton, E. G. Strangas, and S. Aviyente, "Extended Kalman Filtering for Remaining-Useful-Life Estimation of Bearings," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1781–1790, Mar. 2015, doi: 10.1109/TIE.2014.2336616.

[12] Y. Wang, Y. Peng, Y. Zi, X. Jin, and K. Tsui, "A Two-Stage Data-Driven-Based Prognostic Approach for Bearing Degradation Problem," *IEEE Trans. Ind. Inform.*, vol. 12, no. 3, pp. 924–932, Jun. 2016, doi: 10.1109/TII.2016.2535368.

[13] Y. Qian, R. Yan, and S. Hu, "Bearing Degradation Evaluation Using Recurrence Quantification Analysis and Kalman Filter," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 11, pp. 2599–2610, Nov. 2014, doi: 10.1109/TIM.2014.2313034.

[14] X. Jin, Y. Sun, Z. Que, Y. Wang, and T. W. S. Chow, "Anomaly Detection and Fault Prognosis for Bearings," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 9, pp. 2046–2054, Sep. 2016, doi: 10.1109/TIM.2016.2570398.

[15] C. Anger, R. Schrader, and U. Klingauf, "Unscented Kalman filter with gaussian process degradation model for bearing fault prognosis," 2012.

[16] L. Cui, X. Wang, Y. Xu, H. Jiang, and J. Zhou, "A novel Switching Unscented Kalman Filter method for remaining useful life prediction of rolling bearing," *Measurement*, vol. 135, pp. 678–684, Mar. 2019, doi: 10.1016/j.measurement.2018.12.028.
[17] X. Jin, Z. Que, Y. Sun, Y. Guo, and W. Qiao, "A Data-Driven Approach for Bearing Fault Prognostics," *IEEE Trans. Ind. Appl.*, vol. 55, no. 4, pp. 3394–3401, Jul. 2019, doi: 10.1109/TIA.2019.2907666.
[18] L. Liao, "Discovering Prognostic Features Using Genetic Programming in Remaining Useful Life Prediction," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2464–2472, May 2014, doi: 10.1109/TIE.2013.2270212.
[19] N. Li, Y. Lei, J. Lin, and S. X. Ding, "An Improved Exponential Model for Predicting Remaining Useful Life of Rolling Element Bearings," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7762–7773, Dec. 2015, doi: 10.1109/TIE.2015.2455055.
[20] Y. Qian, R. Yan, and R. X. Gao, "A multi-time scale approach to remaining useful life prediction in rolling bearing," *Mech. Syst. Signal Process.*, vol. 83, pp. 549–567, Jan. 2017, doi: 10.1016/j.ymssp.2016.06.031.
[21] N. Gebraeel, M. Lawley, R. Liu, and V. Parmeshwaran, "Residual life predictions from vibration-based degradation signals: a neural network approach," *IEEE Trans. Ind. Electron.*, vol. 51, no. 3, pp. 694–700, Jun. 2004, doi: 10.1109/TIE.2004.824875.
[22] R. Huang, L. Xi, X. Li, C. Richard Liu, H. Qiu, and J. Lee, "Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods," *Mech. Syst. Signal Process.*, vol. 21, no. 1, pp. 193–207, Jan. 2007, doi: 10.1016/j.ymssp.2005.11.008.
[23] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *2008 International Conference on Prognostics and Health Management*, Oct. 2008, pp. 1–6. doi: 10.1109/PHM.2008.4711422.
[24] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, May 2017, doi: 10.1016/j.neucom.2017.02.045.
[25] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, "Remaining useful life estimation based on nonlinear feature reduction and support vector regression," *Eng. Appl. Artif. Intell.*, vol. 26, no. 7, pp. 1751–1760, Aug. 2013, doi: 10.1016/j.engappai.2013.02.006.
[26] T. H. Loutas, D. Roulias, and G. Georgoulas, "Remaining Useful Life Estimation in Rolling Bearings Utilizing Data-Driven Probabilistic E-Support Vectors Regression," *IEEE Trans. Reliab.*, vol. 62, no. 4, pp. 821–832, Dec. 2013, doi: 10.1109/TR.2013.2285318.
[27] A. Soualhi, K. Medjaher, and N. Zerhouni, "Bearing Health Monitoring Based on Hilbert–Huang Transform, Support Vector Machine, and Regression," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 1, pp. 52–62, Jan. 2015, doi: 10.1109/TIM.2014.2330494.
[28] F. Di Maio, K. L. Tsui, and E. Zio, "Combining Relevance Vector Machines and exponential regression for bearing residual life estimation," *Mech. Syst. Signal Process.*, vol. 31, pp. 405–427, Aug. 2012, doi: 10.1016/j.ymssp.2012.03.011.
[29] B. Wang, Y. Lei, N. Li, and N. Li, "A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings," *IEEE Trans. Reliab.*, vol. 69, no. 1, pp. 401–412, Mar. 2020, doi: 10.1109/TR.2018.2882682.
[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, Art. no. 7553, May 2015, doi: 10.1038/nature14539.

[31] B. Wang, Y. Lei, N. Li, and T. Yan, "Deep separable convolutional network for remaining useful life prediction of machinery," *Mech. Syst. Signal Process.*, vol. 134, p. 106330, Dec. 2019, doi: 10.1016/j.ymssp.2019.106330.
[32] A. Z. Hinchi and M. Tkiouat, "Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network," *Procedia Comput. Sci.*, vol. 127, pp. 123–132, Jan. 2018, doi: 10.1016/j.procs.2018.01.106.
[33] Y. Yoo and J.-G. Baek, "A Novel Image Feature for the Remaining Useful Lifetime Prediction of Bearings Based on Continuous Wavelet Transform and Convolutional Neural Network," *Appl. Sci.*, vol. 8, no. 7, Art. no. 7, Jul. 2018, doi: 10.3390/app8071102.
[34] L. Ren, Y. Sun, H. Wang, and L. Zhang, "Prediction of Bearing Remaining Useful Life With Deep Convolution Neural Network," *IEEE Access*, vol. 6, pp. 13041–13049, 2018, doi: 10.1109/ACCESS.2018.2804930.
[35] X. Li, W. Zhang, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Reliab. Eng. Syst. Saf.*, vol. 182, pp. 208–218, Feb. 2019, doi: 10.1016/j.ress.2018.11.011.
[36] B. Wang, Y. Lei, T. Yan, N. Li, and L. Guo, "Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery," *Neurocomputing*, vol. 379, pp. 117–129, Feb. 2020, doi: 10.1016/j.neucom.2019.10.064.
[37] W. Peng, Z.-S. Ye, and N. Chen, "Bayesian Deep-Learning-Based Health Prognostics Toward Prognostics Uncertainty," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2283–2293, Mar. 2020, doi: 10.1109/TIE.2019.2907440.
[38] M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network," in *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*, Oct. 2016, pp. 135–140. doi: 10.1109/AUS.2016.7748035.
[39] P. Malhotra *et al.*, "Multi-Sensor Prognostics using an Unsupervised Health Index based on LSTM Encoder-Decoder," *ArXiv160806154 Cs*, Aug. 2016, Accessed: Feb. 01, 2021. [Online]. Available: http://arxiv.org/abs/1608.06154
[40] C. Huang, H. Huang, and Y. Li, "A Bidirectional LSTM Prognostics Method Under Multiple Operational Conditions," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8792–8802, Nov. 2019, doi: 10.1109/TIE.2019.2891463.
[41] S. Wu, Y. Jiang, H. Luo, and S. Yin, "Remaining useful life prediction for ion etching machine cooling system using deep recurrent neural network-based approaches," *Control Eng. Pract.*, vol. 109, p. 104748, Apr. 2021, doi: 10.1016/j.conengprac.2021.104748.
[42] W. Mao, J. He, J. Tang, and Y. Li, "Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network," *Adv. Mech. Eng.*, vol. 10, no. 12, p. 1687814018817184, Dec. 2018, doi: 10.1177/1687814018817184.
[43] Y. Jiang, S. Yin, J. Dong, and O. Kaynak, "A Review on Soft Sensors for Monitoring, Control, and Optimization of Industrial Processes," *IEEE Sens. J.*, vol. 21, no. 11, pp. 12868–12881, Jun. 2021, doi: 10.1109/JSEN.2020.3033153.
[44] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *International Conference on Machine Learning*, Jun. 2016, pp. 1050–1059. Accessed: Nov. 05, 2020. [Online]. Available: http://proceedings.mlr.press/v48/gal16.html

[45] J. R. Stack, T. G. Habetler, and R. G. Harley, “Fault classification and fault signature production for rolling element bearings in electric machines,” *IEEE Trans. Ind. Appl.*, vol. 40, no. 3, pp. 735–739, May 2004, doi: 10.1109/TIA.2004.827454.
[46] J. I. Taylor, *The Vibration Analysis Handbook: A Practical Guide for Solving Rotating Machinery Problems*. VCI, 2003.
[47] S. A. McInerny and Y. Dai, “Basic vibration signal processing for bearing fault detection,” *IEEE Trans. Educ.*, vol. 46, no. 1, pp. 149–156, Feb. 2003, doi: 10.1109/TE.2002.808234.
[48] “Literature Library,” *Rockwell Automation*. https://www.rockwellautomation.com/en-us/support/documentation/literature-library.html (accessed Oct. 28, 2020).
[49] R. L. Eshleman, *Basic Machinery Vibrations: An Introduction to Machine Testing, Analysis, and Monitoring*. VIPress, 1999.
[50] 14:00-17:00, “ISO 10816-3:2009,” *ISO*. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/05/50528.html (accessed Oct. 28, 2020).
[51] “Sensor Selection Guide.” Wilcoxon Research. [Online]. Available: https://wilcoxon.com/wp-content/uploads/2018/11/TN16_Sensor-selection-guide_2018.pdf
[52] N. Henmi and S. Takeuchi, “New Method Using Piezoelectric Jerk Sensor to Detect Roller Bearing Failure,” *Int. J. Autom. Technol.*, vol. 7, no. 5, pp. 550–557, Sep. 2013, doi: 10.20965/ijat.2013.p0550.
[53] D. Eager, A.-M. Pendrill, and N. Reistad, “Beyond velocity and acceleration: jerk, snap and higher derivatives,” *Eur. J. Phys.*, vol. 37, no. 6, p. 065008, Oct. 2016, doi: 10.1088/0143-0807/37/6/065008.
[54] H. J. Nussbaumer, “The Fast Fourier Transform,” in *Fast Fourier Transform and Convolution Algorithms*, H. J. Nussbaumer, Ed. Berlin, Heidelberg: Springer, 1981, pp. 80–111. doi: 10.1007/978-3-662-00551-4_4.
[55] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. John Wiley & Sons, 2009.
[56] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles,” *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 6402–6413, 2017.
[57] J. Gu *et al.*, “Recent Advances in Convolutional Neural Networks,” *ArXiv151207108 Cs*, Oct. 2017, Accessed: Dec. 31, 2020. [Online]. Available: http://arxiv.org/abs/1512.07108
[58] N. Q. K. Le, Q.-T. Ho, E. K. Y. Yapp, Y.-Y. Ou, and H.-Y. Yeh, “DeepETC: A deep convolutional neural network architecture for investigating and classifying electron transport chain’s complexes,” *Neurocomputing*, vol. 375, pp. 71–79, Jan. 2020, doi: 10.1016/j.neucom.2019.09.070.
[59] J. N. Sua *et al.*, “Incorporating convolutional neural networks and sequence graph transform for identifying multilabel protein Lysine PTM sites,” *Chemom. Intell. Lab. Syst.*, vol. 206, p. 104171, Nov. 2020, doi: 10.1016/j.chemolab.2020.104171.
[60] T. Wang, Jianbo Yu, D. Siegel, and J. Lee, “A similarity-based prognostics approach for Remaining Useful Life estimation of engineered systems,” in *2008 International Conference on Prognostics and Health Management*, Oct. 2008, pp. 1–6. doi: 10.1109/PHM.2008.4711421.
[61] P. Wang, B. D. Youn, and C. Hu, “A generic probabilistic framework for structural health prognostics and uncertainty management,” *Mech. Syst. Signal Process.*, vol. 28, pp. 622–637, Apr. 2012, doi: 10.1016/j.ymssp.2011.10.019.

[62] C. Hu, B. D. Youn, P. Wang, and J. Taek Yoon, “Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life,” *Reliab. Eng. Syst. Saf.*, vol. 103, pp. 120–135, Jul. 2012, doi: 10.1016/j.ress.2012.03.008.
[63] C. Hu, B. D. Youn, and P. Wang, “Time-Dependent Reliability Analysis in Operation: Prognostics and Health Management,” in *Engineering Design under Uncertainty and Health Prognostics*, C. Hu, B. D. Youn, and P. Wang, Eds. Cham: Springer International Publishing, 2019, pp. 233–301. doi: 10.1007/978-3-319-92574-5_8.
[64] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proc. F Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993, doi: 10.1049/ip-f-2.1993.0015.
[65] J. Deutsch, M. He, and D. He, “Remaining Useful Life Prediction of Hybrid Ceramic Bearings Using an Integrated Deep Learning and Particle Filter Approach,” *Appl. Sci.*, vol. 7, no. 7, Art. no. 7, Jul. 2017, doi: 10.3390/app7070649.
[66] Y. Chang and H. Fang, “A hybrid prognostic method for system degradation based on particle filter and relevance vector machine,” *Reliab. Eng. Syst. Saf.*, vol. 186, pp. 51–63, Jun. 2019, doi: 10.1016/j.ress.2019.02.011.
[67] Y. Qian and R. Yan, “Remaining Useful Life Prediction of Rolling Bearings Using an Enhanced Particle Filter,” *IEEE Trans. Instrum. Meas.*, vol. 64, no. 10, pp. 2696–2707, Oct. 2015, doi: 10.1109/TIM.2015.2427891.
[68] G. G. Rigatos, “Particle Filtering for State Estimation in Nonlinear Industrial Systems,” *IEEE Trans. Instrum. Meas.*, vol. 58, no. 11, pp. 3885–3900, Nov. 2009, doi: 10.1109/TIM.2009.2021212.
[69] A. Saxena *et al.*, “Metrics for evaluating performance of prognostic techniques,” in *2008 International Conference on Prognostics and Health Management*, Oct. 2008, pp. 1–17. doi: 10.1109/PHM.2008.4711436.
[70] D. Roman, S. Saxena, V. Robu, M. Pecht, and D. Flynn, “Machine learning pipeline for battery state-of-health estimation,” *Nat. Mach. Intell.*, vol. 3, no. 5, Art. no. 5, May 2021, doi: 10.1038/s42256-021-00312-3.
[71] W. Qian, S. Li, P. Yi, and K. Zhang, “A novel transfer learning method for robust fault diagnosis of rotating machines under variable working conditions,” *Measurement*, vol. 138, pp. 514–525, May 2019, doi: 10.1016/j.measurement.2019.02.073.
[72] T. Han, C. Liu, R. Wu, and D. Jiang, “Deep transfer learning with limited data for machinery fault diagnosis,” *Appl. Soft Comput.*, vol. 103, p. 107150, May 2021, doi: 10.1016/j.asoc.2021.107150.
[73] S. Fort, H. Hu, and B. Lakshminarayanan, “Deep Ensembles: A Loss Landscape Perspective,” *ArXiv191202757 Cs Stat*, Jun. 2020, Accessed: Jun. 17, 2021. [Online]. Available: http://arxiv.org/abs/1912.02757
[74] L. van der Maaten and G. Hinton, “Visualizing Data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
[75] Y. Chang, H. Fang, and Y. Zhang, “A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery,” *Appl. Energy*, vol. 206, pp. 1564–1578, Nov. 2017, doi: 10.1016/j.apenergy.2017.09.106.
[76] K. Liu, Y. Shang, Q. Ouyang, and W. D. Widanage, “A Data-Driven Approach With Uncertainty Quantification for Predicting Future Capacities and Remaining Useful Life of Lithium-ion Battery,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3170–3180, Apr. 2021, doi: 10.1109/TIE.2020.2973876.

[77] C. Hu, G. Jain, P. Tamirisa, and T. Gorka, “Method for estimating capacity and predicting remaining useful life of lithium-ion battery,” *Appl. Energy*, vol. 126, pp. 182–189, Aug. 2014, doi: 10.1016/j.apenergy.2014.03.086.
[78] C. Hu, H. Ye, G. Jain, and C. Schmidt, “Remaining useful life assessment of lithium-ion batteries in implantable medical devices,” *J. Power Sources*, vol. 375, pp. 118–130, Jan. 2018, doi: 10.1016/j.jpowsour.2017.11.056.
[79] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002, doi: 10.1109/78.978374.
[80] M. D. McKay, R. J. Beckman, and W. J. Conover, “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code,” *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979, doi: 10.2307/1268522.
[81] T. Li, M. Bolic, and P. M. Djuric, “Resampling Methods for Particle Filtering: Classification, implementation, and strategies,” *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015, doi: 10.1109/MSP.2014.2330626.
[82] W. Ahmad, S. A. Khan, M. M. M. Islam, and J.-M. Kim, “A reliable technique for remaining useful life estimation of rolling element bearings using dynamic regression models,” *Reliab. Eng. Syst. Saf.*, vol. 184, pp. 67–76, Apr. 2019, doi: 10.1016/j.ress.2018.02.003.
[83] B. Wang, Y. Lei, N. Li, and J. Lin, “An improved fusion prognostics method for remaining useful life prediction of bearings,” in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, Jun. 2017, pp. 18–24. doi: 10.1109/ICPHM.2017.7998300.