Joint Training of a Predictor Network and a Generative Adversarial Network for Time Series Forecasting: A Case Study of Bearing Prognostics

Hao Lu[1,2], Vahid Barzegar[3], Venkat Pavan Nemani[1], Chao Hu[1,2,*], Simon Laflamme[2,3], and Andrew Todd Zimmerman[4,5]

[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]Department of Civil, Environmental and Construction Engineering, Iowa State University, Ames, IA 50011 USA

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

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

* Indicates corresponding author (chaohu@iastate.edu, huchaostu@gmail.com)

Authors' email addresses: hlu1@iastate.edu, barzegar@iastate.edu, vnemani@iastate.edu, chaohu@iastate.edu, laflamme@iastate.edu, andyz@gracetechnologies.com

Abstract

The lack of bearing run-to-failure data has been one of the challenges in developing and practically implementing robust bearing prognostics models. This paper proposes a new Generative Adversarial Network (GAN) based prognostics method for RUL prediction. We propose a novel joint training strategy to integrate the training process of a bearing health predictor within the GAN architecture. GAN uses available time series degradation data to generate synthetic degradation data that enhances the predictor's learning and forecast performance, thus improving the RUL prediction accuracy. We demonstrate the utility and performance of the proposed method through two examples. The first numerical toy case study of forecasting polynomial-like time series shows that the proposed Jointly Trained Health Predictor (HP-JT) method produces smaller one- and multi-step-ahead prediction errors than a traditional health predictor (HP). In the second case study, we design a cross-validation study utilizing an open-source bearing dataset to evaluate the model's performance in RUL prediction. Compared to HP, the proposed method decreases the bearing RUL prediction average error by 29.4% in a five-fold cross-validation study. We further compare the model with standard data augmentation techniques such as adding noise and using a variational autoencoder (VAE). The results from the case studies show that the proposed method can generate time series representing the real-data distribution.

Keywords: long short-term memory; generative adversarial network; time series prediction; bearing prognostics

1. Introduction

As one of the most common and critical components in rotating machines, the rolling element bearings play a crucial part in rotating machinery. The primary purpose of using bearing is to prevent direct metal-to-metal contact between rotating components, friction, heat generation, and the wear and tear of parts (Lei et al., 2018; Wang et al., 2017; Zhang et al., 2017). The unexpected failure of the bearing may severely affect the adjacent machine components, leading to abrupt plant shutdown, financial loss, and even catastrophic accidents (Hu et al., 2019; Liu et al., 2018; J. Wu et al., 2018). Therefore, accurate prediction of bearing remaining useful life (RUL) improves productivity and reduces maintenance costs. According to current literature, a general bearing failure prognostic methodology comprises four essential processes: data acquisition, health indicator construction, health stage division, and RUL prediction (Lei et al., 2018).

The process of data acquisition collects signals that reflect bearing health stages. There are many sensing techniques, such as vibration (Guo et al., 2017; Wang, 2012; Wu et al., 2017), acoustic emission (Aye & Heyns, 2017; Motahari-Nezhad & Jafari, 2021), and temperature (Ren et al., 2017), have been applied to the data collection for bearing failure prognostics. The vibration sensors are most commonly used for bearing health monitoring due to their sensitivity and widespread availability.

Health indicators, extracted from the acquired sensory data, are metrics that reflect the health states of the bearing. The construction of the health indicator is pivotal for failure prognostics. A well-defined health indicator could simplify the modeling of the degradation process and increase the RUL prediction accuracy. According to the construction strategies, the health indicators can be categorized into physics-based and virtual health indicators. Generally, physics-based health indicators are extracted from the raw signal using signal processing methods. Soualhi et al. (2014) used Hilbert-Huang transform to analyze vibration signals and constructed health indicators using amplitude values located at bearing characteristic frequencies. (Zhang et al., 2015) constructed a health indicator using the kurtosis values extracted from band-pass filtered vibration signals. The root mean square (RMS) is one of the most widely used physics-based health indicators. Malhi et al. (2011) analyzed signals using wavelet transform, the RMS and peak values of wavelet coefficients were used to predict the RUL. Lu et al. (2018) extracted RMS values from band-pass filtered signal to quantify the bearing damage severity. The virtual health indicators are constructed by fusing multiple physics-based health indicators or multi-sensor signals. Wang (2012) used principal component analysis to fuse multiple features and construct the health indicator for bearing degradation. Ren et al. (2018) extracted features from the time domain, frequency domain, and time-frequency domain, then adopted an autoencoder to construct the health indicator. One limitation of virtual health indicators is that virtual health indicators lack physical meaning and only present a virtual description of the degradation trends of the target bearings (Lei et al., 2018).

The constructed health indicators could help divide the health stages of the bearing by identifying when the bearing degradation starts. Typically, the bearing is healthy at the early stage of its life, where the health indicator values do not change significantly. After the formation of the bearing fault, the bearing starts to degrade, and the bearing state transforms from the healthy stage into the degradation stage. The failure prognostics approaches focus on the degradation stage where an obvious trend can be observed in the health indicator values. The prognostics model is trained using the available degradation data then used to predict the RUL of the bearing.

The RUL prediction approaches can be broadly classified into two categories based on the type of model: (a) model-based and (b) data-driven. The model-based approaches construct mathematical models by analyzing the bearing degradation mechanisms (Cubillo et al., 2016). Nowadays, model-based approaches such as the Paris-Erdogan model (Lei et al., 2016), particle filter (Jouin et al., 2016), the Eyring model (Saxena et al., 2008), and exponential model (Li et al., 2015) have been well applied to predict the general trend of degradation. However, the model-based approaches require accurate estimation of the model parameters. Since the rotating machinery has several different working settings, building a mathematical model that fits all the possible working conditions is challenging. Due to the poor adaptability of the model, if there is a change in the operating condition, the prediction results of model-based approaches tend to become less accurate and not reliable (Liu et al., 2021).

On the other hand, data-driven approaches typically employ machine learning techniques to extract and learn the patterns from the available observations without utilizing any knowledge of the degradation mechanisms (Wu et al., 2020). In this regard, several well-known machine learning algorithms, such as Gaussian process regression (Pan et al., 2016), support vector machine (Lei, 2012), and artificial neural networks (Xue et al., 2020), have been implemented.

In the past few years, deep learning techniques have attracted widespread attention. Yoo and Baek (2018) used wavelet transform analysis to extract the time-frequency features then the convolutional neural network (CNN) was employed to estimate the RUL. Guo et al. (2017) selected model input by looking at the correlation and monotonicity of extracted features, then developed a recurrent neural network (RNN) for RUL prediction. As a special type of recurrent neural network, long short-term memory (LSTM) has become a powerful tool in extracting temporal information for bearing failure prognostics. Y. Wu et al. (2018) adopted the LSTM model for bearing RUL prediction. A dynamic differential feature extraction method was utilized that helped the model capture the changes in features under different operating conditions. Other variants of deep learning models are also applied for bearing prognostics. For example, Chen et al. (2020) adopt the attention mechanism into the LSTM network to adaptively select features that are important for RUL prediction, resulting in accurate prediction results. Zhu et al. (2018) adopted a multiscale convolutional neural network, which keeps the global and local information synchronously to enhance the prediction performance.

Based on the output of the model, the data-driven approaches for bearing prognostics can be sorted into two types: 1) direct mapping approaches (Cheng et al., 2021; Zhu et al., 2018); and 2) forecasting approaches (He et al., 2022; Shi & Chehade, 2021). The direct mapping approaches take the raw signal or constructed health indicator values as input and produce an RUL estimate as output. The forecasting approaches take the historical health indicator values as the input, forecast the future degradation trajectory of the health indicator values until the failure threshold is reached, then calculate the RUL.

Although the data-driven approaches have shown promising results, they often face the following challenges:

Many data-driven approaches map the model input with the RUL directly. However, it has been previously shown that the degradation process of the bearing is nonlinear (Sadoughi et al., 2019; Wang et al., 2016). During the early stage of the run-to-failure tests, the bearings are considered healthy and do not show any significant change in the collected vibration data. After a certain period of operation, bearing-related faults can be detected, and a degradation trend can be observed. In addition to the variation in the time for the development of an incipient bearing fault, the degradation rate at which the bearing approaches failure is highly nonlinear with the health indicator, a measure of bearing health condition. Therefore, directly mapping the extracted features to the RUL can produce nonphysical results that do not ensure RUL convergence as the bearing approaches failure.

Besides, data-driven approaches heavily rely on a large amount of training data to acquire degradation information. However, it is time-consuming and costly to gather a large amount of bearing run-to-failure data. On the other hand, insufficient training data may lead to overfitting (Wen et al., 2020). Data augmentation is one way to alleviate this problem by generating synthetic data. Some commonly used approaches have been well applied, such as adding noise and extending or shrinking the data. The core of data augmentation is to ensure that the generated data is similar to the original data, not only in terms of magnitude but also in data distribution. In this regard, the generative adversarial network has attracted wide attention recently. GAN has been used in several fields to generate high-quality synthetic data for data augmentation, where traditional data augmentation methods do not yield good results. The implementation of GAN-based data augmentation has been applied to solve a variety of engineering problems, including but not limited to: (1) image classification(Abdelhalim et al., 2021; Frid-Adar et al., 2018; Shorten & Khoshgoftaar, 2019), (2) electroencephalography signal classification(Hatamian et al., 2020; Luo et al., 2020), and (3) time series anomaly detection (Li et al., 2019; Lim et al., 2018). Many of these papers compared GAN-based data augmentation with conventional data augmentation approaches and demonstrated the GAN-based approach delivers significant improvement in model performance, such as sensitivity and prediction accuracy. For example, Frid-Adar et al. (2018) showed that compared to affine augmentation, using GAN-generated synthetic data increases the classification accuracy from 78.6% to 85.7%.

This paper proposes a GAN-based LSTM predictor for bearing fault prognostics. A Jointly Trained Health Predictor (HP-JT) method is proposed to forecast a health indicator. A preliminary version of this work was presented at the 2021 IEEE International Conference on Prognostics and Health Management (Lu et al., 2021). This work is a significantly expanded version of our conference paper. In addition to using real bearing run-to-failure data, we also devise a toy problem to mimic a simplified bearing degradation behavior. We compare the proposed method with other data augmentation methods, such as adding noise and using a variational autoencoder (VAE). Our main contributions are summarized as follows:

1) Unlike traditional approaches where the maximum or mean vibration amplitude is used for RUL prediction, we define a bearing health indicator, which measures bearing health based on the root mean square values in the velocity domain. This definition complies with ISO 10816, which we also referred to in defining the threshold for bearing failure (Eshleman & Nagle-Eshleman, 1999; *ISO 10816-3:2009*, 2021). The proposed HP-JT method forecasts the health indicator (by marching in time) until the failure threshold is reached.
2) To deal with the challenge of insufficient training data, we develop a GAN-based data augmentation method by integrating HP-JT into the GAN architecture and propose a joint training strategy. The performance of HP-JT is boosted by acquiring knowledge from both training data and synthetic data.

The remainder of this paper is organized as follows. Section 2 introduces the proposed framework and the models used for comparison. Section 3 includes two case studies to evaluate the proposed method. Finally, the conclusions are summarized in section 4.

2. Proposed HP-JT prognostics method

Three different stages of the proposed method for bearing elements failure prognostics are summarized in Figure 1 (a)-(c), illustrating data preparation, offline training of HP-JT, and finally, online RUL prediction. Detailed discussions of the three stages in Figure 1 are presented in sections 2.1, 2.2, and 2.3. In section 2.4, we introduce benchmark models for comparison.

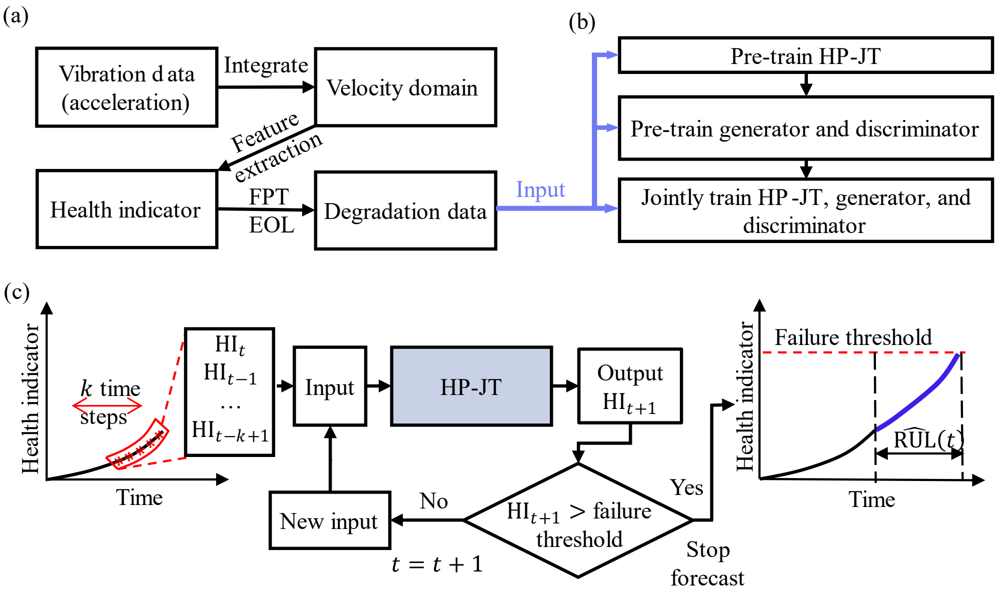


Figure 1: The main components and flowchart of (a) data preparation, (b) offline training of HP-JT, and (c) online RUL prediction

2.1 Data preparation

Data preparation is a process that converts the raw inputs into features that better represent the bearing health condition. In the proposed method, the RMS value of a sub-band filtered velocity signal is extracted and used as HI.

Most run-to-failure datasets use vibration signals in the acceleration domain obtained from accelerometers. However, the industrial-relevant ISO standards define the end-of-life or the warning threshold values based on the feature values in the velocity domain (*ISO 10816-3:2009*, 2021). This is because the amplitude of acceleration changes dramatically under different shaft speeds. In contrast, the amplitude of the signal in the velocity domain provides a more stable representation. In the proposed method, firstly, the acceleration signal is converted into velocity domain $v(t)$ by performing numerical integration. To avoid interference from low-frequency noise and to obtain the frequency information that reflects the bearing damage severity as much as possible, the velocity RMS in the frequency range of $0.2\omega - f_s/2$ Hz is extracted, where ω denotes shaft frequency and f_s denotes the sampling frequency. The RMS values of the time series are obtained from its Fourier transform spectrum using Parseval's theorem (Nussbaumer, 1981), written as:

$$V^{\mathrm{RMS}}_{0.2\omega-f_s/2} = \sqrt{\sum_{f=0.2\omega}^{f_s/2} \frac{|V(f)|^2}{2}} \tag{1}$$

were $V(f)$ is the single-side frequency spectrum for $v(t)$. To improve the reliability of the extracted features, we apply a moving average method. In this study, the smoothed health indicator value is the average of the current observation with two previous observations from the recent past.

In the proposed method, the 2σ approach was used to locate the first prediction time (FPT). The data collected at the early stage of the experiment are considered as healthy data with a calculated feature mean (μ) and standard deviation (σ). Using these, a threshold of ($\mu + 2\sigma$) is set on the feature value. The FPT was obtained when two consecutive observations ($V^{RMS}_{0.2\omega-f_{s/2}}$) exceed the threshold. The End of Life (EOL) time of the bearing was obtained when $V^{RMS}_{0.2\omega-f_s/2}$ reaches a given threshold. In this study, following the ISO 10816 alarm threshold for medium-sized motors, we define the failure threshold value as 0.27 ips (*ISO 10816-3:2009*, 2021). The development of $V^{RMS}_{0.2\omega-f_s/2}$ for a typical bearing is illustrated in Figure 2. Before the FPT, the bearing is healthy, the $V^{RMS}_{0.2\omega-f_s/2}$ present random fluctuation. After FPT, the bearing is in the degradation stage, the $V^{RMS}_{0.2\omega-f_s/2}$ value increase with the deterioration of the bearing until it reaches the failure threshold.

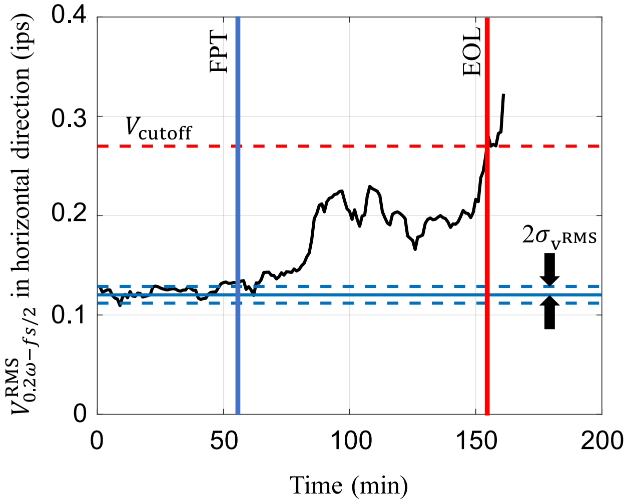


Figure 2: Evolution of $V^{\mathrm{RMS}}_{0.2\omega-\mathrm{f}_s/2}$ and identification of FPT, EOL for a typical bearing.

2.2 Offline training of HP-JT

The proposed HP-JT is formed by an LSTM layer followed by a dense layer. A detailed, mathematical description of LSTM can be found in Appendix A.1. The input of the HP-JT is the health indicator values starting from the previous $k-1$ time to the current time. And the output of the model is the health indicator value at the next time step. As shown in Figure 3, The training of the proposed HP-JT model consists of three steps: 1) Pre-train HP-JT, 2) Pre-train generator and discriminator, and 3) Jointly train all the components.

In the first step, the HP-JT was pre-trained using the raw data by utilizing the structure shown in Figure 3 (a). In this step, the mean squared error loss is adopted to optimize the parameters of the predictor.

After the pre-training of the HP-JT, GAN is utilized to generate synthetic data to boost the performance of the HP-JT model. A brief description of a standard GAN can be found in Appendix A.2. A traditional way of performing GAN-based data augmentation is to use the training data to train the GAN, then combine the synthetic data with the training data to form augmented training data for model training. In this paper, different from that traditional GAN-based data augmentation, a joint training approach was designed, integrating the HP-JT into the GAN architecture. The architecture of the proposed GAN-LSTM network is shown in Figure 3 (b). The network is optimized by back-propagating error and the standard gradient descent optimization method (Ruder, 2016).

The generator takes a random vector of length k and outputs a vector with the same length. The generated vector $\tilde{x}_{i,1:k}$ is then fed into the HP-JT to predict the value at the next time step. The predicted next-step value is then attached to the generator's output to get the synthetic data. The discriminator takes the synthetic and real data as the input and identifies the input as real or fake.

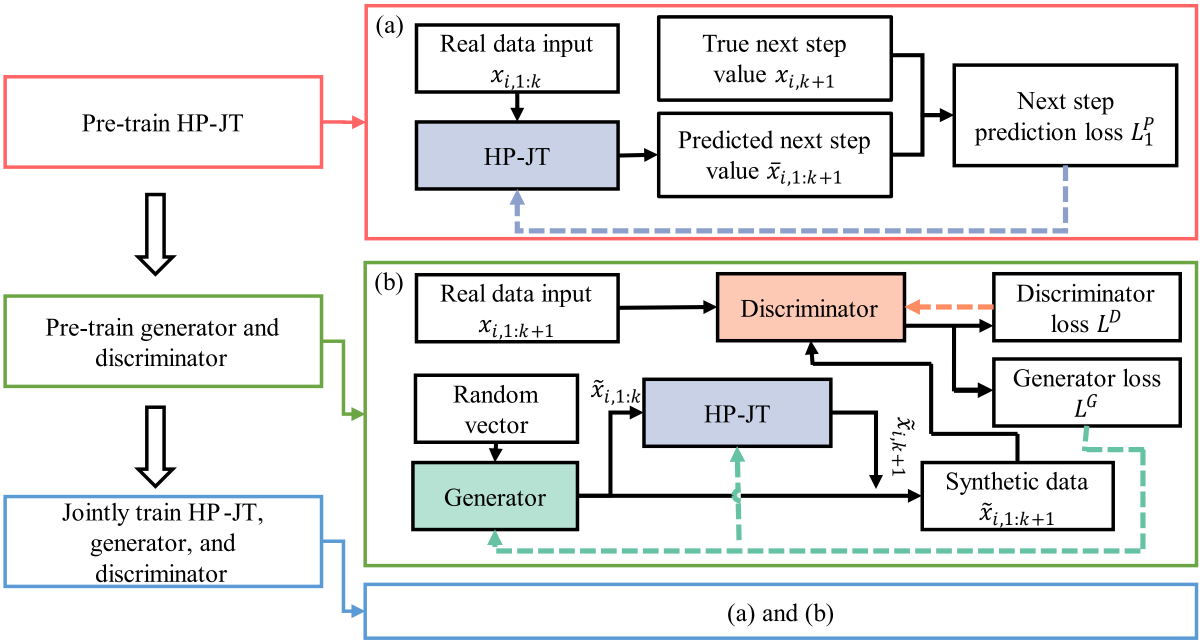

Pre-train HP-JT
Pre-train generator and discriminator
Jointly train HP-JT, generator, and discriminator
(a)
Real data input $x_{i,1:k}$
HP-JT
True next step value $x_{i,k+1}$
Predicted next step value $\tilde{x}_{i,1:k+1}$
Next step prediction loss L_1^P
(b)
Real data input $x_{i,1:k+1}$
Discriminator
Discriminator loss L^D
Generator loss L^G
Random vector
Generator
$\tilde{x}_{i,1:k}$
HP-JT
$\tilde{x}_{i,k+1}$
Synthetic data $\tilde{x}_{i,1:k+1}$
(a) and (b)

Figure 3: The training procedures of the proposed HP-JT model. The architecture used for each step is marked within a different color-coded box. (a) the architecture that trains the HP-JT model using the real data (b) the GAN-LSTM network that trains the HP-JT model, generator, and discriminator. (The dashed lines represent backpropagation of loss values)

In the proposed GAN-LSTM network, both the generator, HP-JT, and discriminator contribute to synthetic data generation. The generator's output is noise when it is initialized. To guarantee the HP-JT acquires degradation knowledge from synthetic data, we fix the parameters of the HP-JT, pre-train the generator and discriminator before all the GAN-LSTM components are jointly trained.

During the pre-training of GAN-LSTM, the discriminator and generator are optimized iteratively. For the training of discriminator, the loss function is composed of two pieces: 1) the real data is classified correctly, and 2) the synthetic data is classified as fake data, which is written as:

$$L^D = -\frac{1}{n}\sum_{i=1}^{n}[\log D(x_{i,1:k+1}) + \log(1 - D(\tilde{x}_{i,1:k+1}))] \tag{2}$$

where n represents the number of training samples, $x_{i,1:k+1}$ is the ith real data with length $=k$. $D(x_{i,1:k+1})$ denotes the output of the discriminator with input as $x_{i,1:k+1}$. $\tilde{x}_{i,1:k+1}$ is the ith synthetic data with length$=$ k, which is generated by concatenating the output of the generator ($\tilde{x}_{i,1:k}$) with the output of the predictor ($\tilde{x}_{i,k+1}$). The objective of the discriminator training is to minimize L^D so that the discriminator can correctly identify the input data as real or fake.

For each epoch during the pre-training of the generator and discriminator, after the discriminator is optimized, the training of the generator begins by fixing the parameters of the discriminator. The objective of the generator training is to make the discriminator classify synthetic data as real data. And the loss function is written as:

$$L^G = \frac{1}{n}\sum_{i=1}^{n}\log\left(1 - D(\tilde{x}_{i,1:k+1})\right) \tag{3}$$

The training of generator and discriminator can be interpreted as a two-player game in which the discriminator tries to identify the generated signal from all the inputs, and the generator tries to generate synthetic data that can fool the discriminator. Conceptually, the training of GAN corresponds to a minimax two-player game written as (Goodfellow et al., 2020):

$$\min_{G}\ \max_{D}\ L(D,G) = \frac{1}{n}\sum_{i=1}^{n}[\log D(x_{i,1:k+1}) + \log(1 - D(\tilde{x}_{i,1:k+1}))] \tag{4}$$

Overall, the training process for the proposed method consists of the following steps:

1) Pre-training HP-JT: The HP-JT model is pre-trained using the real data by utilizing the structure shown in Figure 3 (a).

2) Pre-training the generator and discriminator: The parameters of the generator and the discriminator are iteratively updated according to the structure illustrated in Figure 3 (b), keeping the parameters of HP-JT fixed.

3) Joint training of all the components: During every joint training epoch, the HP-JT, the generator, and the discriminator are simultaneously trained. The joint training epoch consists of two sub-steps. Firstly, the HP-JT learns from every iteration of synthetic data while providing better next-step prediction. And the next-step prediction is involved in the training of the generator and the discriminator. In other words, the three components now enhance the performance of each other. We note that joint training works only after pre-training the GAN components on real data, following the above two steps, without which the predictor focuses its learning on the random data provided by the generator. In the second sub-step, the predictor is fine-tuned using the real data. The pre-training and fine-tuning ensure a general direction of learning bearing degradation is achieved, which is further enhanced during joint training.

The architecture of the proposed generator, discriminator, and HP-JT is shown in Table 1. The generator consists of three fully connected layers; the ReLU activation function is adopted to prevent negative synthetic signal values. The discriminator consists of four fully connected layers. The discriminator needs to output classification probabilities; therefore, the Sigmoid activation function is adopted at the last fully connected layer. And the proposed HP-JT is formed by an LSTM layer followed by a fully connected layer.

Table 1: The structure of the proposed GAN-LSTM network.

Module name	Layer	Output shape, Activation
Generator	Input	(Samples, 20)
	Fully Connected	(Samples, 64), Linear
	Fully Connected	(Samples, 32), Linear
	Fully Connected	(Samples, 20), ReLU
Discriminator	Input layer	(Samples, 21)
	Fully Connected	(Samples, 64), Linear
	Fully Connected	(Samples, 128), ReLU
	Fully Connected	(Samples, 64), ReLU
	Fully Connected	(Samples, 1), Sigmoid
HP-JT	Input	(Samples, 20, 1)
	LSTM	(Samples, 60), Tanh
	Fully Connected	(Samples, 1), Linear

2.3 Online RUL prediction

The trained health predictor is used to forecast the health indicator values until a failure threshold is reached. With the current time step as t_{current}, the health indicator values from the previous $k-1$ time steps to the current time constitute the model input, and the model predicts the health indicator value at the next time step. The model output is then concatenated to the original input, and the model is reevaluated by marching in time to forecast the feature value until the model output exceeds the predefined threshold (V_{cutoff}) at time T_{EOL}. The predicted RUL can be determined by:

$$\text{RUL}(t) = T_{\text{EOL}} - t_{\text{current}} \tag{5}$$

2.4 Benchmark models

Four commonly used methods are briefly introduced as benchmark models. The models are a) Health Predictor (HP), b) HP-Noise, c) HP-VAE, and d) quadratic regression model. In section 3, we compare the performance of the proposed HP-JT against the four benchmark methods.

a) HP

We use the predictor without being integrated into the GAN as a baseline model, which we name HP. It has the same architecture as the HP-JT and is only trained on real data.

b) HP-Noise

We also want to compare the proposed method against other data augmentation approaches. To do that, we include the predictor trained with a simple data augmentation method, which we refer to as HP-Noise. The synthetic data is generated by adding a certain Gaussian noise to the real data. The training dataset of HP-Noise is composed of synthetic and real data. The HP-Noise model also has the same architecture as the HP-JT model.

c) HP-VAE

Besides GAN, VAE is another powerful deep generative model for data augmentation (Huang et al., 2021). The HP-VAE method is composed of two steps. First, the VAE is trained to provide synthetic data following a similar pattern to the training data. Then, the data generated by VAE are combined with the real data to train the predictor. The detailed configuration of HP-VAE is included in Table 2.

Table 2: The specific configuration of HP-VAE

Module name	Layer	Output shape, Activation
Encoder	Input	(Samples, 21)
	Fully connected	(Samples, 16), Linear
	Fully connected	(Samples, 12), ReLU
Decoder	Input	(Samples, 6)
	Fully connected	(Samples, 16), Linear
	Fully connected	(Samples, 21), ReLU
HP-VAE	Input	(Samples, 20, 1)
	LSTM	(Samples, 60), Tanh
	Fully connected	(Samples, 1), Linear

d) Quadratic regression model

The quadratic regression model is a simple mathematical model that captures the degradation trend by fitting a quadratic model to the feature values. The model is defined as:

$$V^{\mathrm{RMS}}(t) = m_1 t^2 + m_2 t + m_3 \tag{6}$$

where $V^{\mathrm{RMS}}(t)$ represents the health indicator value at time t, and m_1, m_2 and m_3 are the model parameters that are optimized during regression using real data. The ordinary least square method fits the model in eqn. (6) using the current and previous $k-1$ measurements. After the model parameters, m_1,m_2 and m_3, are determined, followed by the prediction of the future health indicator values, RUL is calculated by measuring the time when the predicted health indicator reaches a predefined threshold. A certain RUL prediction will be deemed unreliable in the quadratic regression model if the bearing forecast values are monotonically decreasing and thus do not reach the threshold. In such cases, the model takes the nearest reliable RUL result minus the time difference between that prediction and current times as the predicted RUL. One other difference is that the regression model does not learn from the run-to-failure data of other bearings and relies only on the target bearing measurements.

3. Case studies

Two case studies are employed to demonstrate the effectiveness of the proposed method. Case study 1 is a numerical toy problem that evaluates the model's performance in predicting future values. Case study 2 is a practical example, using publicly available Xi'an Jiaotong University and Changxing Sumyong Technology Co., Ltd. (XJTU-SY) bearing dataset to verify the performance of the proposed method by considering the RUL prediction accuracy through a 5-fold cross-validation. The performance of the proposed method in uncertainty estimation is analyzed in section 3.3. The computational efficiency is discussed in Appendix D, focusing on the training and prediction time.

3.1 Case study 1: time series prediction of a toy problem

3.1.1 Experimental setting

A numerical toy problem is defined to mimic a simplified behavior of bearing degradation. In this case study, two types of trend functions are defined based on the bearing degradation patterns (Lei et al., 2018):

Quadratic degradation trend:

$$S_{\text{Quadratic}}(t) = a_1 t^3 + b_1 t^2 + w \quad 0 \le t \le T \tag{7}$$

Three-stage degradation trend:

$$S_{\text{Three-stage}}(t) = \begin{cases} a_2 t^3 + b_2 t^2 + c_2 + w & 0 \le t < t_1 \\ a_3 t^3 + b_3 t^2 + c_3 + w & t_1 \le t < t_2 \\ a_4 t^3 + b_4 t^2 + c_4 + w & t_2 \le t < T \end{cases} \tag{8}$$

where a_i, b_i and c_i $(i = 1,2,3)$ are the coefficients of the degradation trends, and w is Gaussian noise. The signal generated by the quadratic degradation function represents a monotonically increasing trend. The three-stage degradation function generates the signal where the rate of degradation is significant during the early stages (due to the formation of the defect) followed by its decrease (due to smoothening effect) and an increase close to EOL. This behavior is similar to the degradation processes with multiple stages summarized in (Lei et al., 2018). Eight simulated signals are generated by following eqns. (7) and (8) are illustrated in Figure 4. The parameter settings of each signal are included in Appendix B.

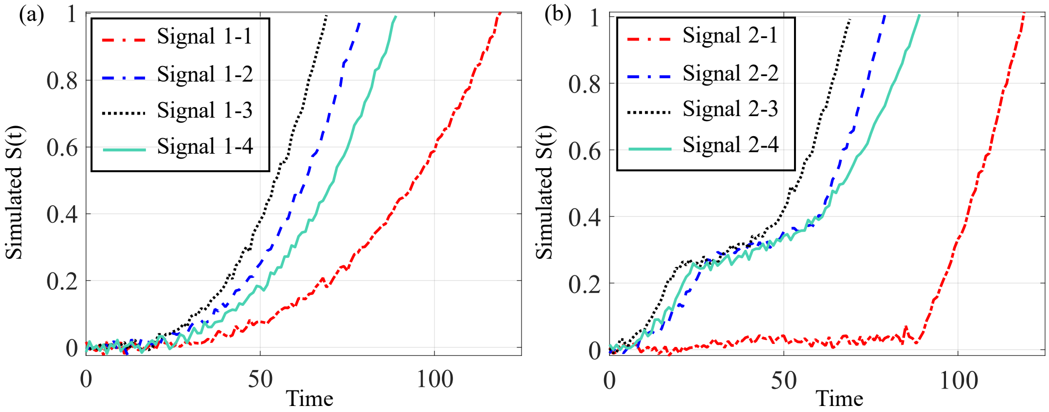


Figure 4: Summary of the generated signals with (a) quadratic and (b) three-stage degradation behavior.

A cross-validation study was conducted using the simulated signals. For each cross-validation experiment, one signal was selected as the test data, and the other seven were used to train the model. In this case study, the predictor takes the signal values of current and previous $k - 1$ time steps to forecast the value for the next N_s steps. The HP-JT model was compared with HP, HP-Noise, and HP-VAE. After a preliminary optimization study, the learning rates for the HP, HP-Noise, and HP-VAE models were set as 0.00015. The learning rate for the HP-JT model is also set to 0.00015, with the learning rates of both the generator and discriminator fixed at 0.0001.

Given a test signal with a total length of T_{signal}, we look at all the forecasts beginning from $t = k + 1$ to the last possible forecast of length N_s at $t = T_{\text{signal}} - N_s$. The RMSEs of all the forecasts are combined into a single evaluation metric, written as:

$$\text{RMSE}_{\text{signal}} = \sqrt{\frac{1}{T_{\text{signal}} - N_s - k + 1} \frac{1}{N_s} \sum_{t=k}^{T_{\text{signal}} - N_s} \sum_{i=1}^{N_s} \left(S_t^P(i) - S^T(t+i)\right)^2} \tag{9}$$

where $S_t^P(i)$ represents the i^{th} predicted value from the prediction time t, $S^T(t+i)$ represents the true value at time $t + i$.

3.1.2 Results

An eight-fold cross-validation test was conducted for the eight signals, and the RMSE over all the prediction results was calculated (RMSE_{All}). Figure 5 (a) and (b) show the variation of RMSE_{All} with the number of predicted steps N_s with 50 and 200 training epochs, respectively. For both numbers of epochs, the HP-JT model produced the least $\mathrm{RMSE}_{\mathrm{All}}$ value when forecasting $N_s = 1$ to $N_s = 10$ steps.

All the data augmentation methods enhanced the RSME of the predictions compared to the HP model, with HP-VAE outperforming the HP-Noise. Note that HP-JT yields the best prediction RMSE for all forecast steps.

For the remainder of this case study, we attempt to explain how the proposed method outperforms other data augmentation techniques. For comparison, all the models were trained with a total training epoch of 200. HP-JT is pre-trained for 30 epochs, followed by joint training with 170 epochs.

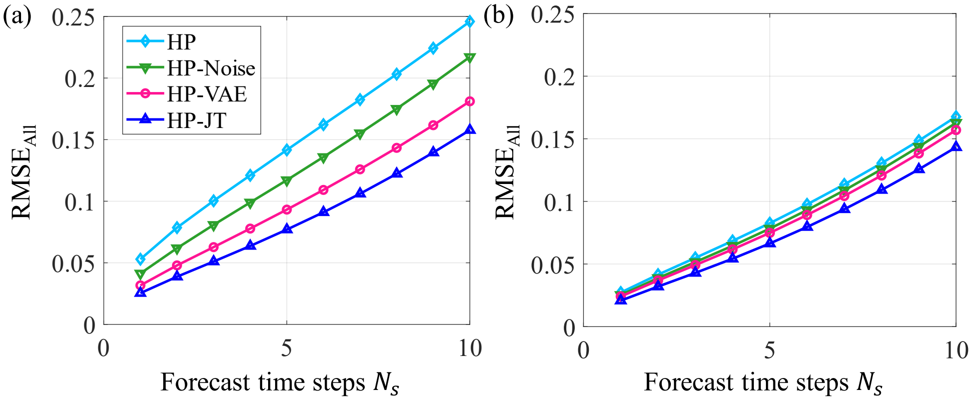


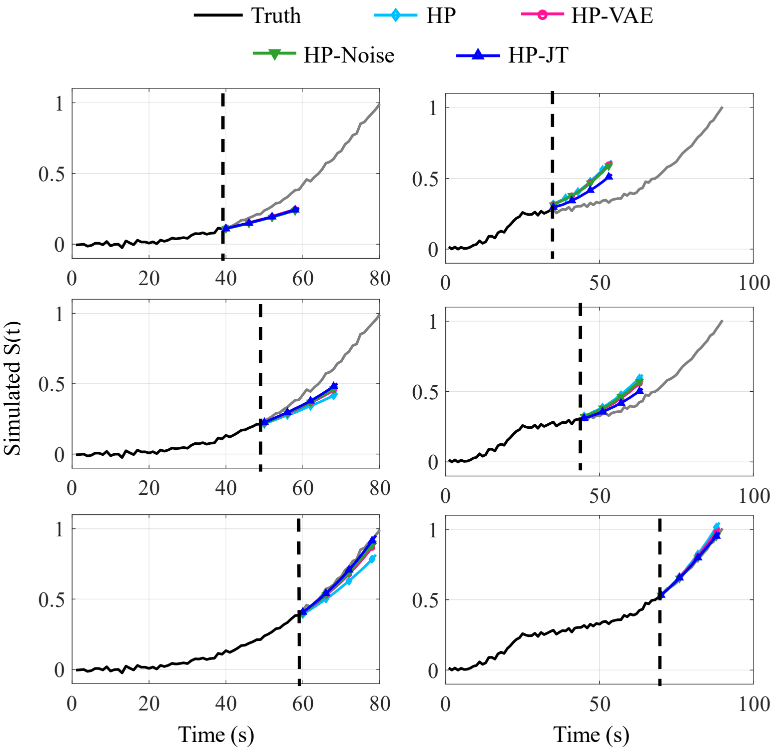
Figure 5: $\mathrm{RMSE}_{\mathrm{All}}$ results by multiple methods under different training settings; (a) the total training epoch = 50, the HP-JT was pre-trained with 30 epochs, and the joint training took 20 epochs (b) the total training epoch = 200, the HP-JT was pre-trained with 30 epochs and the joint training of 170 epochs

To show the forecasting capability of all the methods, we show the RMSE values for the one-step-ahead ($N_s = 1$) and five-step-ahead ($N_s = 5$) predictions in Table 3. Among eight cross-validation experiments, the HP-JT produced the least $\mathrm{RMSE}_{\mathrm{All}}$ value. On average, HP-JT outperformed 23.5%, 17.6%, and 13.7% for $N_s = 1$, and 19.6%, 15.2%, and 11.2% for $N_s = 5$, compared to HP, HP-Noise, and HP-VAE, respectively. Note that, compared with the quadratic degradation signals, there is a noticeable increase in the $\mathrm{RMSE}_{\mathrm{signal}}$ value for the three-stage degradation signals given the more complicated health degradation structure, especially for signal 2-1.

Table 3: Prediction results by HP-JT and benchmark models

Signal ID	Degradation type	RMSE ($\times 10^{-2}$)							
		$N_s = 1$				$N_s = 5$			
		HP	HP-Noise	HP-VAE	HP-JT	HP	HP-Noise	HP-VAE	HP-JT
1-1	Quadratic degradation	2.16	1.20	1.31	**1.08**	6.15	3.42	3.73	**2.84**
1-2		1.53	1.32	1.20	**1.19**	4.25	3.77	**3.19**	3.30
1-3		2.02	1.80	1.56	**1.51**	6.42	5.78	4.84	**4.65**
1-4		1.36	1.38	**1.25**	1.30	3.68	3.79	**3.18**	3.55
2-1	Three stage degradation	3.89	3.85	3.48	**2.68**	11.93	11.86	10.81	**8.84**
2-2		3.23	3.06	3.01	**2.88**	10.96	10.53	10.61	**10.06**
2-3		2.97	2.85	2.98	**2.66**	9.83	10.11	10.17	**9.54**
2-4		2.34	2.31	2.31	**1.98**	7.49	7.29	7.44	**6.43**
$\mathrm{RMSE}_{\mathrm{All}}$		2.64	2.45	2.34	**2.02**	8.26	7.83	7.48	**6.64**

To further investigate the superior performance of HP-JT, the 20-step-ahead forecast results of the four models for two representative signals 1-2 and 2-4 at three different prediction times are shown in Figure 6. For the simple quadratic signal 1-2, all four predictor models achieved satisfactory predicting accuracy, yet the augmented data models slightly outperformed HP. However, for the more complicated signal 2-4, the HP-JT outperforms the benchmark models in the forecast, especially for time steps right after the second stage (from $t = 25$ to 60 min). As we get further away from the second stage, HP-JT starts to perform similarly to the benchmark models as the signal starts to follow a simpler trend.



(a) (b)

Figure 6: The predicted results at different prediction times; (a) signal 1-2 at prediction times $t = 20, 50, \text{and } 60$ s, and (b) signal 2-4 at prediction times $t = 35, 45, \text{and } 70$ s

To investigate the reduction in average RMSE of the RUL, the t-Distributed Stochastic Neighbor Embedding (t-SNE) analysis (Van der Maaten & Hinton, 2008) was performed to visualize the similarity between real and synthetic data generated with adding noise, VAE and HP-JT. The t-SNE results for each data augmentation method are shown in Figure 7, with the total training epochs of 200 and signal 2-4 as the test data.

The HP-Noise method augments the data by adding Gaussian noise to the training data; thus, the data points only shift slightly and maintain the original distribution. Therefore, this slightly shifted distribution does not add enough new information to generate novel data for training in the case of limited available data. HP-VAE was partially successful in mimicking the global trend of the real-data distribution; however, its restriction to Gaussian distribution modeling limited the generated data distribution to only partially represent the real data distribution resulting in little improvement. On the other hand, the data generated by HP-JT follows the global distribution and captures the local variations of the real-data distribution. Unlike HP-Noise, HP-JT is not limited to individual data points' immediate neighborhood, thus creating novel yet representative synthetic data.

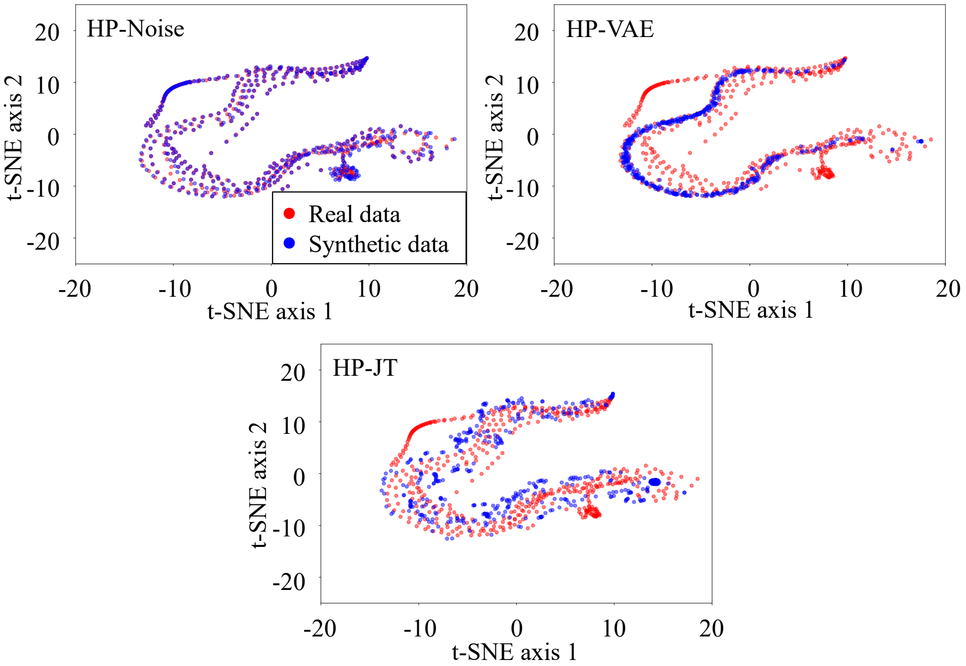


Figure 7: t-SNE results of real and synthetic data generated by multiple methods

In the case of HP-JT, the synthetic data generated by the GAN-LSTM network changes during every epoch of pre-training and joint training to encompass the entire distribution of the training dataset. The evolution of the synthetic data distribution during the pre-training and joint training, which yields superior performance to the benchmark methods, is shown in Appendix B.

To further explore the benefits of joint training strategy over static data augmentation, we study another model called HP-noJT that uses fixed synthetic data without joint training. The HP-noJT was initialized using the same parameters as that of the pre-trained HP-JT predictor, with the difference being that there is no joint training in the training procedure of HP-noJT. The synthetic data was combined with the original training data to train the model. In other words, the HP-noJT predictor only learns the original training data and the static synthetic data generated after the pre-training of the generator. On the other hand, during joint training in HP-JT, the generator is also simultaneously trained with the HP-JT predictor. As an effect of training the generator, slightly different synthetic data is generated at every epoch, which the HP-JT model also sees. In other words, the HP-JT model learns from different synthetic data at each joint train epoch, whereas the HP-noJT predictor was trained using fixed synthetic data.

The prediction results of HP-noJT are summarized in Table 4. HP-noJT performs better than LSTM (Table 3) by 9.1% $\mathrm{RMSE}_{\mathrm{All}}$ for one-step-ahead prediction. This observation proves that using synthetic data enhanced the accuracy of the next-step prediction. The HP-JT, however, outperforms HP-noJT by 15.8% in one-step-ahead ($N_s = 1$) prediction and by 13.1% in five-step-ahead ($N_s = 5$) prediction, respectively. The data generated at each epoch may contain some unreliable samples that are different from the real data. The HP-noJT may be forced to learn these unreliable samples during the training process. For HP-JT, the generated sample changed at each epoch, preventing the predictor from memorizing unreliable samples. The integration of the predictor and GAN architecture helps improve the generality of the HP-JT model, which leads to the least average RMSE error among all the tests. The discussion of change of the next-step prediction errors is included in Appendix B.3.

Table 4: Prediction results by HP-noJT

Signal ID	Degradation type	$N_s = 1$	$N_s = 5$
1-1	Quadratic degradation	1.18	3.21
1-2		1.39	3.68
1-3		1.72	5.49
1-4		1.38	3.76
2-1	Three-stage degradation	3.56	11.15
2-2		2.92	10.19
2-3		3.14	10.59
2-4		2.42	7.64
RMSE$_{\text{All}}$		2.40	7.64

3.2 Case study 2: bearing RUL prediction

3.2.1 Experimental setting

We now evaluate the performance of the proposed method aimed at RUL prediction using the publicly available XJTU-SY dataset. The XJTU-SY dataset provides run-to-failure data collected from 15 rolling element bearings (Wang et al., 2018). The vibration data can be divided into three groups based on the operating condition, shown in Table 5.

Table 5: XJTU-SY bearing dataset.

	Operating condition		
	Condition 1	Condition 2	Condition 3
Radial load	12 kN	11 kN	10 kN
Speed	35 Hz	37.5 Hz	40 Hz
Bearing ID	1-1	2-1	3-1
	1-2	2-2	3-2
	1-3	2-3	3-3
	1-4	2-4	3-4
	1-5	2-5	3-5

The bearings were affected by the radial load; therefore, the data collected from the x-axis (horizontal direction) is more obvious (Kundu et al., 2019). In this case study, the x-axis vibration data were used to extract $V_{0.2\omega-fs/2}^{\text{RMS}}$ values for RUL prediction. The extracted $V_{0.2\omega-fs/2}^{\text{RMS}}$ features (from FPT to EOL) of each bearing are presented in Appendix C. The feature value of the most recent $k = 20$ measurements was used in forecasting $V_{0.2\omega-fs/2}^{\text{RMS}}$ to a failure threshold of 0.27 and thus determine the RUL.

We conducted a five-fold cross-validation study on the XJTU-SY dataset where the 15 bearings were divided into five folds, with each fold containing data collected from three different working conditions:

Fold-1: Bearings 1-1, 2-1, and 3-1

Fold-2: Bearings 1-2, 2-2, and 3-2

Fold-3: Bearings 1-3, 2-3, and 3-3

Fold-4: Bearings 1-4, 2-4, and 3-4

Fold-5: Bearings 1-5, 2-5, and 3-5

The proposed HP-JT model was compared with benchmark models presented in section III. The input length of the generator, HP-JT, and discriminator were set at 20, 20, and 21, respectively. The learning rate of the HP-JT was set as 0.001, and the learning rate of both generator and discriminator was 0.0001. The HP-JT was pre-trained for 60 epochs. The generator and discriminator were pre-trained for 1000 epochs. Finally, the joint training of all the GAN-LSTM components was performed for 60 epochs.

Similar to case study 1, the HP and HP-Noise models had the same architecture as the HP-JT. The learning rates and the training epochs of those two models were set to 0.001 and 120, respectively. Note that HP-JT only gets optimized during the pre-training and joint training. Therefore, the total training epoch of the HP-JT is equal to the predictor trained by other methods (HP, HP-Noise, and HP-VAE).

The RMSE of RUL prediction results (from t_{FPT} to t_{EOL}) was used as an evaluation metric that measures the prediction error, written as:

$$\mathrm{RMSE}_{\mathrm{RUL}} = \sqrt{\frac{1}{(t_{\mathrm{EOL}} - t_{\mathrm{FPT}} + 1)} \sum_{t=t_{\mathrm{FPT}}}^{t_{\mathrm{EOL}}} \left(\mathrm{RUL}_{\mathrm{pred}}(t) - \mathrm{RUL}_{\mathrm{true}}(t)\right)^2} \qquad (10)$$

where t_{FPT} is the time when prognostics starts, and $\mathrm{RUL}_{\mathrm{pred}}(t)$ and $\mathrm{RUL}_{\mathrm{true}}(t)$ are the predicted and true RUL at time step t, respectively.

3.2.2 Results

The RUL prediction results for all the test bearings, as a result of the five-fold cross-validation, are summarized in Table 6. The bearings are sorted in ascending order of the total prognostic duration ΔT, defined as $\Delta T = t_{\mathrm{EOL}} - t_{\mathrm{FPT}} + 1$. For each bearing in Table 6, the model with the least prediction error is highlighted in bold. The cumulative $\mathrm{RMSE}_{\mathrm{RUL}}$ is calculated by doing a weighted average of the individual bearing $\mathrm{RMSE}_{\mathrm{RUL}}$ scaled by ΔT. Overall, the proposed HP-JT produces better RUL prediction accuracies with 40.3%, 29.4%, 26.8%, and 20.4% improvement in RMSE error compared to the quadratic regression, HP, HP-Noise, and HP-VAE models. Note that for bearings 2-4, 3-5, 3-3, 1-5, and 1-4, the number of time steps in the prognostic time period is smaller than the selected input length of the LSTM predictor (k=20 min). For these bearings, data points before t_{FPT} were used as input, and these data do not provide enough prognostic information for making accurate predictions. Also, for most tests, the proposed HP-JT model outperformed other methods with bearings that have a longer prognostic time.

Table 6: RUL prediction results by HP-JT and benchmark models

Bearing ID	ΔT (min)	RMSE				
		Quadratic regression	HP	HP-Noise	HP-VAE	HP-JT
2-4	4	13.18	13.26	**4.74**	25.46	24.39
3-5	6	6	13.47	**4.18**	15.73	10.89
3-3	10	8.93	14.91	**7.35**	9.58	21.99
1-5	16	60.06	22.97	**13.49**	37.72	30.93
1-4	17	36.04	42.66	**22.05**	33.38	48.43
2-1	34	9.70	**5.34**	10.14	18.61	42.59
1-2	42	16.17	13.91	13.21	**7.97**	12.83
1-1	43	13.70	15.71	14.4	27.36	**9.04**
3-2	46	33.07	12.2	10.9	9.27	**4.89**
3-4	60	16.37	**8.23**	12.99	11.98	12.6
2-5	77	38.10	**11.72**	13.36	15.64	17.41
2-3	83	20.20	15.74	25.12	28.17	**9.93**
1-3	91	45.73	25.59	31.12	32.38	**20.37**
2-2	105	58.77	43.99	43.56	33.97	**29.69**
3-1	124	35.16	53.84	47.29	36.52	**21.25**
Cumulative[#]		36.70	31.00	29.91	27.50	**21.90**

[#] RMSE among all the bearings weighted by the prognostic time duration ΔT.

To better compare the prediction results, we analyze the mean absolute error (MAE) that quantifies the magnitude of the prediction error, and also include the mean error that quantifies the overall direction of the prediction error (overestimation or underestimation). At the same level of prediction accuracy, underestimating the bearing RUL is often more desirable than overestimating it in industry settings because overestimation brings misleading confidence to the end user and may cause unexpected machine failure. Figure 8 (a) summarizes the MAE and mean error of RUL prediction by various models. The MAE of quadratic regression, HP, HP-Noise, and HP-VAE are 23.84, 21.77, 21.62, and 21.68 min, respectively. HP-JT yields the least MAE, 16.70 min. Compared to quadratic regression, the four health predictor models (i.e., HP, HP-Noise, HP-VAE, and HP-JT) predict RUL with smaller mean errors that are all less than zero (i.e., underestimating the RUL on average).

At an early stage of degradation, $V^{\text{RMS}}_{0.2\omega-fs/2}$ of a bearing tends not to change significantly. As a result, the RUL predictions at this stage may contain larger errors than those when the bearing is close to failure. Suppose we only consider the samples from the time when $V^{\text{RMS}}_{0.2\omega-fs/2}$ first exceeds 0.17 ips to EOL and we label these samples as the late-stage degradation samples. The prediction errors on these samples are shown in Figure 8 (b). The MAE and error spread both decrease for all the five methods. Excluding the HP-JT model, the HP-Noise model produces the lowest MAE. A paired *t*-test is conducted to analyze the mean difference between the prediction errors of HP-JT and HP-Noise. The null hypothesis in the paired *t*-test is the mean difference between the prediction errors by HP-JT and HP-Noise is zero. The *p*-value is $2.26 \times 10^{-16} \ll 0.001$, which provides strong evidence against the null hypothesis. Thus, HP-JT yields a significantly different mean error compared to HP-Noise. As the mean error of HP-JT is closer to zero and its MAE is smaller, HP-JT on average archives higher accuracy than HP-Noise as well as the other three models.

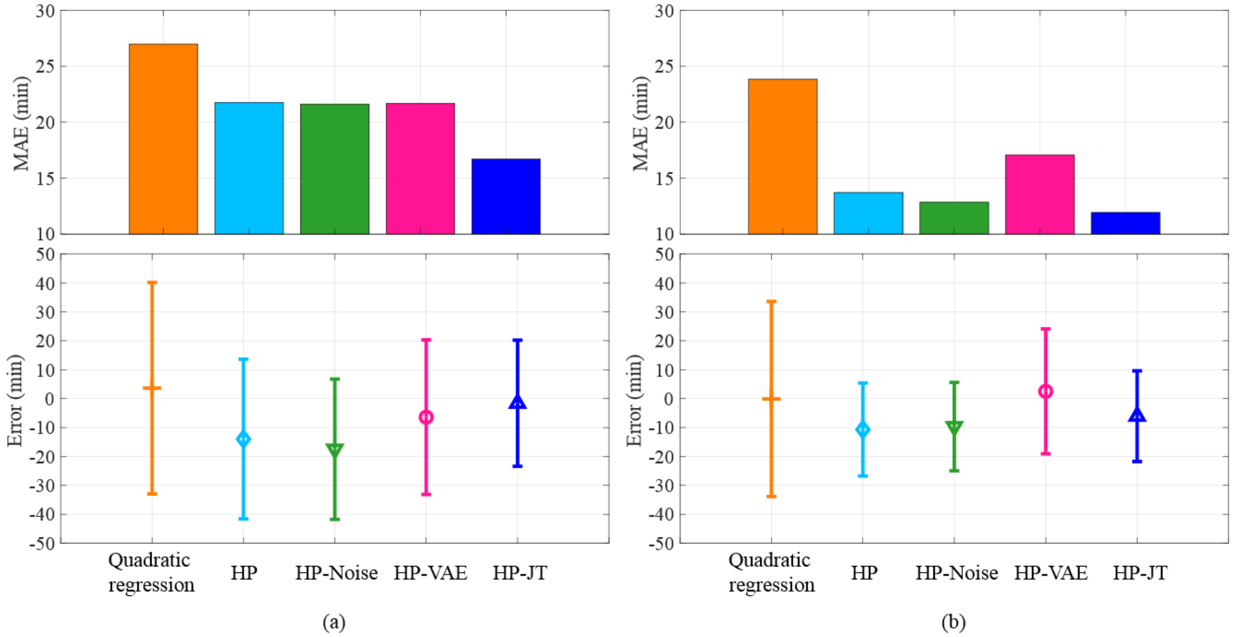


Figure 8: The MAE, mean error, and error spread for five different methods: (a) the prediction errors from FPT to EOL and (b) the prediction errors from the first time $V^{\text{RMS}}_{0.2\omega-fs/2} > 0.17$ ips to EOL. The error bars indicate mean $\pm$ one standard deviation.

Figure 8 shows a typical predicted RUL and the corresponding $V^{\text{RMS}}_{0.2\omega-fs/2}$ for test bearing 3-2. Note that in the early stages of the bearing degradation, the HP-JT provided the most accurate results compared to the benchmark models. The quadratic regression model yielded the least accurate. As the bearing degradation progressed with time, the extracted feature became closer to the failure threshold and made RUL prediction easier with cumulative multi-step-ahead prediction error. This led to similar RUL prediction results across all the approaches.

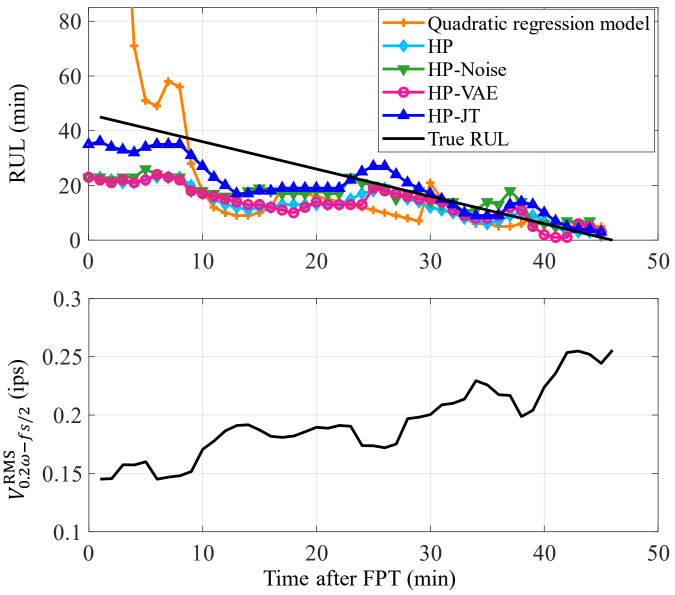


Figure 8: The RUL prediction result and the degradation curve for bearing 3-2

To further investigate the results, the predicted feature values generated by all the comparative models at three different prediction times are shown in Figure 9. The HP-JT model predicted the degradation trend most accurately, especially at the onset of bearing degradation. Note that, at time $t = 20$ min there is almost no change in the amplitude of input features, yet HP-JT successfully provides the most accurate result. The accuracy of the proposed method can be attributable to the quality of the synthetic training data where both global and local novel features are generated (see Appendix C). Note that HP, HP-Noise, and HP-VAE tend to underestimate the RUL, which indicates not being able to distinguish between local and global trends. The quadratic model is the most sensitive to the local trends as it only relies on information provided by recent local observations. If the local trends follow the global trend, it can provide accurate results as in the top plot. Otherwise, the results are unreliable, as in the middle plot of Figure 9.

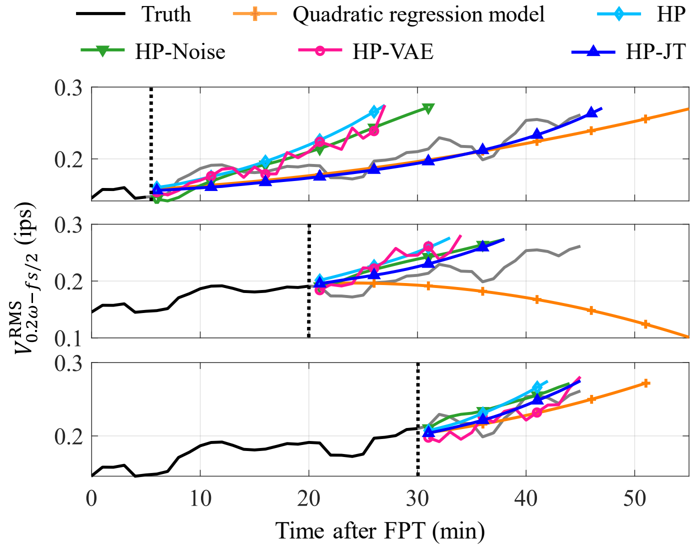


Figure 9: The predicted feature values at selected prediction times for bearing 3-2

3.3 Performance of the proposed method in uncertainty estimation

The models presented thus far are deterministic. Now, we explore the forecasting performance of HP-JT when considering uncertainty. The two major types of uncertainty are aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty is irreducible uncertainty in the training data, which can be estimated by treating the model output as a distribution. Epistemic uncertainty is the uncertainty that occurs due to inadequate knowledge and data. Epistemic uncertainty can be reduced by having more training data. In the case of bearing prognostics, building a probabilistic model could help capture the aleatoric uncertainty of the data. And the use of data augmentation techniques such as HP-JT should theoretically provide a more reliable measure of epistemic uncertainty.

One way to build a probabilistic model is to treat the model output to obey a Gaussian distribution by adding a Gaussian layer as the model's last layer (Nemani et al., 2021). This added layer estimates both the mean $\mu(x)$ and variance $\sigma^2(x)$ of the Gaussian output. For a perfectly trained model, the output $\mu(x)$ is close to the true value y, and $\sigma^2(x)$ accounts for the uncertainty of the output. The negative log-likelihood (NLL) criterion is used to train the model with the Gaussian layer:

$$-\log p(y_n|x_n) = \frac{\log\sigma^2(x)}{2} + \frac{\left(y-\mu(x)\right)^2}{2\sigma^2(x)} + \text{constant} \quad (11)$$

For the bearing prognostic implementation in case study 2, the performance of HP-JT was compared against the HP model. To construct a probabilistic model, we replaced the HP-JT's last layer (dense layer) with a Gaussian layer. The architecture of the proposed HP-JT probabilistic network is shown in Appendix E. During the joint training of the HP-JT, the $\mu\left(\tilde{x}_{i,k+1}\right)$ output was concatenated with $\tilde{x}_{i,1:k}$ to form synthetic data ($\tilde{x}_{i,1:k+1}$).

After the model was trained, the predictor estimates the bearing RUL following a similar procedure described in section 2. The time when next-step prediction $\mu\left(\text{HI}_\text{Input} = V^\text{RMS}_{0.2\omega-fs/2}\right)$ reaches the threshold (V_cutoff) is marked as μ_RUL and the time when $\mu\left(\text{HI}_\text{Input}\right) + \sigma^2(\text{HI}_\text{Input})$ reaches the threshold defined equal to $\mu_\text{RUL} - \sigma^2_\text{RUL}$.

The reliability curve is used to evaluate the model's performance in uncertainty estimation. The reliability curve displays the predicted fraction of points in each confidence interval relative to the expected fraction of points in that interval (Roman et al., 2021). Given a dataset $\left\{x_{T_p,1:k+1}, \text{RUL}_{T_p}\right\}, T_p = 1, \dots, T_\text{total}$. At each prediction time T_p, the probabilistic model provides a Gaussian distribution $\mathcal{N}(\mu_\text{RUL}, \ \sigma^2_\text{RUL})$. We choose m confidence levels $0 \le p_1 < p_2 < \cdots < p_m \le 100$; for each threshold p_j, we compute the observed confidence level:

$$\hat{p}_j = \frac{\sum_{T_p=1}^{T_\text{total}} \mathrm{F}_{T_p}(\mu_\text{RUL}, \sigma^2_\text{RUL}, p_j)}{T_\text{total}} \times 100 \quad (12)$$

where F_{T_p} is a function that classifies whether the true $\mathrm{RUL}_{\mathrm{T}_p}$ lies within a predefined interval. If the $\mathrm{RUL}_{\mathrm{T}_p}$ lies below the p_j-th quantile of the produced Gaussian distribution, $\mathcal{N}(\mu_\text{RUL}, \ \sigma^2_\text{RUL})$, then we have $F_{T_p}\left(\mu_\text{RUL}, \sigma^2_\text{RUL}, p_j\right) = 1$; otherwise, $F_{T_p}\left(\mu_\text{RUL}, \sigma^2_\text{RUL}, p_j\right) = 0$. The set $\left\{\left(p_j, \hat{p}_j\right)\right\}_{j=1}^{M}$ forms a reliability curve.

In Figure 10, we compare the uncertainty estimation performance of the probabilistic HP-JT method and a simple probabilistic HP method for the case study 2 dataset. A total of five models are trained for each method to show run-to-run variation. The reliability of an ideal model falls on the black dashed line where the model is neither underconfident nor overconfident. Both HP and HP-JT are shown to be overconfident in their RUL predictions. However, the reliability curves produced by the HP-JT model are closer to the dashed line (the ideal case), meaning that the observed confidence level is overall closer to the expected confidence level. This means that the HP-JT provides more reliable uncertainty estimations of RUL.

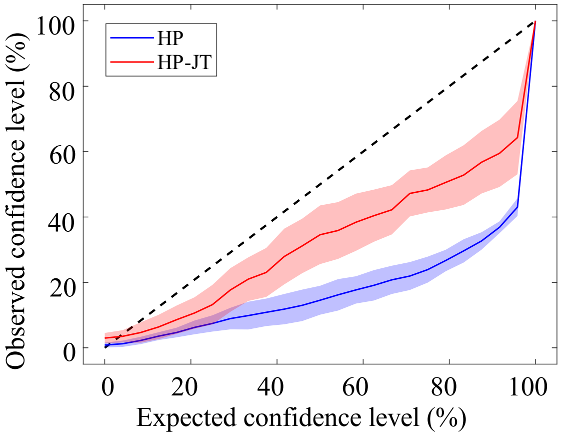


Figure 10: Reliability plot showing the variation of the observed confidence level against the expected confidence level

4. Conclusions

In this paper, we propose a novel HP-JT method for forecasting the bearing health condition and predicting the bearing remaining useful life. We establish the superior performance of the proposed method by performing experiments on a toy problem mimicking simplified bearing failure behavior and using a publicly available XJTU-SY bearing dataset. We find that the GAN-LSTM architecture adds significant diversity to the training data while maintaining the original training data distribution instead of other data augmentation techniques such as adding noise and using VAE, which tend to mimic a local distribution of the training data. This leads to better learning of the long-term dependencies by the HP-JT model, leading to the lowest average RMSE in forecasting the time series for the toy problem. For the XJTU dataset, the HP-JT method achieves a 29.4% reduction in RMSE and a 25% reduction in MAE compared to the HP method. Also, the prediction error distribution indicates that the proposed method provides more accurate and conservative RUL prediction than the other methods used for comparison. The training of the proposed method requires more computational time relative to the benchmark methods; however, since in the industrial implementation, machine health assessments are carried out periodically, the process of bearing RUL prediction is not time-constrained, and thus, the model accuracy is more important than the training time. As long as the model provides higher accuracy, the added training complexity is not as important.

The proposed HP-JT method can be applied to solve other engineering problems where time series prediction is required and the amount of available training data is limited. These problems include, for example, cutting tool health forecasting, battery capacity forecasting and life prediction, and sales forecasting. In this work, we assume bearing degradation is slow and gradual and does not involve extreme, short-term damage leading to sudden failure. As a result, the applicability of the method is limited to slow, gradual degradation trajectories. In this study, bearings 2-4, 3-3, 1-5, and 1-4 have fast ($\Delta T < 20$ min) and dramatically changing degradation trajectories. The RMSEs of RUL prediction on these fast degrading bearings are larger than 20 min (i.e., larger than the maximum true RUL among the four bearings), which signifies this limitation. Prognostics on fast degrading bearings that fail almost instantaneously upon the start of degradation is a topic for future research.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation under Grant IIP-1919265, the Regents Innovation Fund that is part of the Proof-of-Concept Initiative at Iowa State University, and Grace Technologies. Any opinions, findings, or conclusions in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

References

Abdelhalim, I. S. A., Mohamed, M. F., & Mahdy, Y. B. (2021). Data augmentation for skin lesion using self-attention based progressive generative adversarial network. *Expert Systems with Applications*, *165*, 113922.

Aye, S. A., & Heyns, P. (2017). An integrated Gaussian process regression for prediction of remaining useful life of slow speed bearings based on acoustic emission. *Mechanical Systems and Signal Processing*, *84*, 485-498.

Barzegar, V., Laflamme, S., Hu, C., & Dodson, J. (2021). Multi-time resolution ensemble lstms for enhanced feature extraction in high-rate time series. *Sensors*, *21*(6), 1954.

Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., & Li, X. (2020). Machine remaining useful life prediction via an attention-based deep learning approach. *IEEE Transactions on Industrial Electronics*, *68*(3), 2521-2531.

Cheng, H., Kong, X., Chen, G., Wang, Q., & Wang, R. (2021). Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors. *Measurement*, *168*, 108286.

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

Eshleman, R. L., & Nagle-Eshleman, J. (1999). *Basic machinery vibrations: An introduction to machine testing, analysis, and monitoring*. VIPress.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, *321*, 321-331.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139-144.

Guo, L., Li, N., Jia, F., Lei, Y., & Lin, J. (2017). A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, *240*, 98-109.

Hatamian, F. N., Ravikumar, N., Vesal, S., Kemeth, F. P., Struck, M., & Maier, A. (2020). The effect of data augmentation on classification of atrial fibrillation in short single-lead ECG signals using deep neural networks. ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),

He, R., Tian, Z., & Zuo, M. J. (2022). A semi-supervised GAN method for RUL prediction using failure and suspension histories. *Mechanical Systems and Signal Processing*, *168*, 108657.

Hu, C.-H., Pei, H., Si, X.-S., Du, D.-B., Pang, Z.-N., & Wang, X. (2019). A prognostic model based on DBN and diffusion process for degrading bearing. *IEEE Transactions on Industrial Electronics*, *67*(10), 8767-8777.

Huang, Y., Tang, Y., & Vanzwieten, J. (2021). Prognostics with Variational Autoencoder by Generative Adversarial Learning. *IEEE Transactions on Industrial Electronics*.

ISO 10816-3:2009. (2021). @isostandards. https://www.iso.org/standard/50528.html

Jouin, M., Gouriveau, R., Hissel, D., Péra, M.-C., & Zerhouni, N. (2016). Particle filter-based prognostics: Review, discussion and perspectives. *Mechanical Systems and Signal Processing*, *72*, 2-31.

Kim, S., Kim, N. H., & Choi, J.-H. (2020). Prediction of remaining useful life by data augmentation technique based on dynamic time warping. *Mechanical Systems and Signal Processing*, *136*, 106486.

Kundu, P., Darpe, A. K., & Kulkarni, M. S. (2019). Weibull accelerated failure time regression model for remaining useful life prediction of bearing working under multiple operating conditions. *Mechanical Systems and Signal Processing*, *134*, 106302.

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

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

Lei, Z. (2012). Fault prognostic algorithm based on multivariate relevance vector machine and time series iterative prediction. *Procedia engineering*, *29*, 678-686.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J., & Ng, S.-K. (2019). MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. International Conference on Artificial Neural Networks,

Li, N., Lei, Y., Lin, J., & Ding, S. X. (2015). An improved exponential model for predicting remaining useful life of rolling element bearings. *IEEE Transactions on Industrial Electronics*, *62*(12), 7762-7773.

Lim, S. K., Loo, Y., Tran, N.-T., Cheung, N.-M., Roig, G., & Elovici, Y. (2018). Doping: Generative data augmentation for unsupervised anomaly detection with gan. 2018 IEEE International Conference on Data Mining (ICDM),

Liu, H., Mo, Z., Zhang, H., Zeng, X., Wang, J., & Miao, Q. (2018). Investigation on rolling bearing remaining useful life prediction: A review. 2018 Prognostics and System Health Management Conference (PHM-Chongqing),

Liu, L., Song, X., Chen, K., Hou, B., Chai, X., & Ning, H. (2021). An enhanced encoder–decoder framework for bearing remaining useful life prediction. *Measurement*, *170*, 108753.

Lu, H., Barzegar, V., Nemani, V. P., Hu, C., Laflamme, S., & Zimmerman, A. T. (2021). GAN-LSTM predictor for failure prognostics of rolling element bearings. 2021 IEEE International Conference on Prognostics and Health Management (ICPHM),

Lu, Y., Li, Q., Pan, Z., & Liang, S. Y. (2018). Prognosis of bearing degradation using gradient variable forgetting factor RLS combined with time series model. *IEEE Access*, *6*, 10986-10995.

Luo, Y., Zhu, L.-Z., Wan, Z.-Y., & Lu, B.-L. (2020). Data augmentation for enhancing EEG-based emotion recognition with deep generative models. *Journal of Neural Engineering*, *17*(5), 056021.

Malhi, A., Yan, R., & Gao, R. X. (2011). Prognosis of defect propagation based on recurrent neural networks. *IEEE Transactions on Instrumentation and Measurement*, *60*(3), 703-711.

Motahari-Nezhad, M., & Jafari, S. M. (2021). Bearing remaining useful life prediction under starved lubricating condition using time domain acoustic emission signal processing. *Expert Systems With Applications*, *168*, 114391.

Nemani, V. P., Lu, H., Thelen, A., Hu, C., & Zimmerman, A. T. (2021). Ensembles of Probabilistic LSTM Predictors and Correctors for Bearing Prognostics Using Industrial Standards. *Neurocomputing*.

Nussbaumer, H. J. (1981). The fast Fourier transform. In *Fast Fourier Transform and Convolution Algorithms* (pp. 80-111). Springer.

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

Ren, L., Cui, J., Sun, Y., & Cheng, X. (2017). Multi-bearing remaining useful life collaborative prediction: A deep learning approach. *Journal of Manufacturing Systems*, *43*, 248-256.

Ren, L., Sun, Y., Cui, J., & Zhang, L. (2018). Bearing remaining useful life prediction based on deep autoencoder and deep neural networks. *Journal of Manufacturing Systems*, *48*, 71-77.

Roman, D., Saxena, S., Robu, V., Pecht, M., & Flynn, D. (2021). Machine learning pipeline for battery state-of-health estimation. *Nature Machine Intelligence*, *3*(5), 447-456.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Sadoughi, M., Lu, H., & Hu, C. (2019). A Deep Learning Approach for Failure Prognostics of Rolling Element Bearings. 2019 IEEE International Conference on Prognostics and Health Management (ICPHM),

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. 2008 international conference on prognostics and health management, Shi, Z., & Chehade, A. (2021). A dual-LSTM framework combining change point detection and remaining useful life prediction. *Reliability Engineering & System Safety*, *205*, 107257.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, *6*(1), 1-48.

Soualhi, A., Medjaher, K., & Zerhouni, N. (2014). Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression. *IEEE Transactions on Instrumentation and Measurement*, *64*(1), 52-62.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(11).

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

Wang, D., Tsui, K.-L., & Miao, Q. (2017). Prognostics and health management: A review of vibration based bearing and gear health indicators. *IEEE Access*, *6*, 665-676.

Wang, T. (2012). Bearing life prediction based on vibration signals: A case study and lessons learned. 2012 IEEE Conference on Prognostics and Health Management,

Wang, Y., Xiang, J., Markert, R., & Liang, M. (2016). Spectral kurtosis for fault detection, diagnosis and prognostics of rotating machines: A review with applications. *Mechanical Systems and Signal Processing*, *66*, 679-698.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., & Xu, H. (2020). Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*.

Wu, B., Li, W., & Qiu, M.-q. (2017). Remaining useful life prediction of bearing with vibration signals based on a novel indicator. *Shock and Vibration*, *2017*.

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

Wu, J., Wu, C., Cao, S., Or, S. W., Deng, C., & Shao, X. (2018). Degradation data-driven time-to-failure prognostics approach for rolling element bearings in electrical machines. *IEEE Transactions on Industrial Electronics*, *66*(1), 529-539.

Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, *275*, 167-179.

Xue, Y., Dou, D., & Yang, J. (2020). Multi-fault diagnosis of rotating machinery based on deep convolution neural network and support vector machine. *Measurement*, *156*, 107571.

Yoo, Y., & Baek, J.-G. (2018). A novel image feature for the remaining useful lifetime prediction of bearings based on continuous wavelet transform and convolutional neural network. *Applied Sciences*, *8*(7), 1102.

Zhang, W., Jia, M.-P., Zhu, L., & Yan, X.-A. (2017). Comprehensive overview on computational intelligence techniques for machinery condition monitoring and fault diagnosis. *Chinese Journal of Mechanical Engineering*, *30*(4), 782-795.

Zhang, Z.-X., Si, X.-S., & Hu, C.-H. (2015). An age-and state-dependent nonlinear prognostic model for degrading systems. *IEEE Transactions on Reliability*, *64*(4), 1214-1228.

Zhu, J., Chen, N., & Peng, W. (2018). Estimation of bearing remaining useful life based on multiscale convolutional neural network. *IEEE Transactions on Industrial Electronics*, *66*(4), 3208-3216.

Appendix A: background supplementary materials

A.1 Fundamental LSTM architecture

The LSTM uses memory cells to retain useful information in the long and short term to help with the vanishing gradient issues of RNNs. As shown in Fig A.1, each LSTM unit uses three internal gates to control the information flow for each time step, named forget gate, input gate, and output gate (Barzegar et al., 2021).

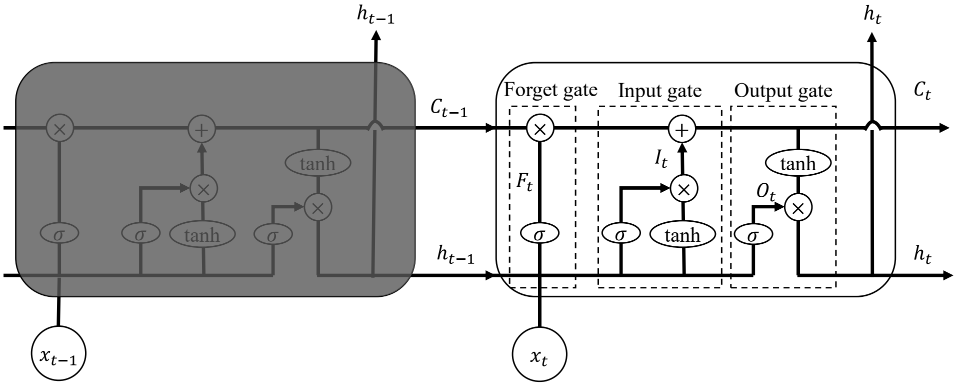


Figure A.1: LSTM unit architecture

The equations for each gate can be described as:

Forget gate:

$$F_t = \sigma(W_F[h_{t-1}, x_t] + b_F) \tag{13}$$

where the sigmoid layer (σ) takes the previous output of the LSTM unit h_{t-1} and input x_t, and decides which parts of the past information to forget by outputting a value closer to 0 and what to retain by outputting a value closer to 1. W_F and b_F are the weights and biases of the forget gate, respectively.

Input gate:

$$I_t = \sigma(W_I[h_{t-1}, x_t] + b_I) \otimes \tanh\left(W_C[h_{t-1}, x_t] + b_C\right) \tag{14}$$

$$C_t = F_t \otimes C_{t-1} + I_t \tag{15}$$

where the sigmoid layer decides which of the new information to be stored into the cell state C_t, W_I and b_I are the weights and biases of the input gate, respectively, $\tanh(\cdot)$ creates the new candidate values for the new cell state, W_C and b_C are the weights and biases related to cell state calculation, respectively, and the previous cell state C_{t-1} is multiplied with F_t, then added with I_t to get the new current cell state C_t.

Output gate:

$$O_t = \sigma(W_O[h_{t-1}, x_t] + b_O) \tag{16}$$

$$h_t = O_t \otimes \tanh\left(C_t\right) \tag{17}$$

where the sigmoid layer determines the output of the cell, W_O and b_O are the weights and biases of the output gate, respectively, and $\tanh(\cdot)$ creates all possible values, which become the output after being multiplied with O_t.

A.2 GAN

GAN belongs to the class of generative models that aims to produce synthetic samples with a similar distribution as that of the input data. The GAN comprises two networks: generator and discriminator, as shown in Fig. Those two networks are trained with opposing goals. The goal of the generator is to produce synthetic data that has a similar distribution to that of the real data; the goal of the discriminator is to take the synthetic and real data and try to identify which input samples are real or fake (Goodfellow et al., 2020). The training of the GAN network aims to make the generator compete with the discriminator. A successfully trained generator converts the random noise into synthetic data with a similar distribution to the real data, which the discriminator fails to identify as fake.

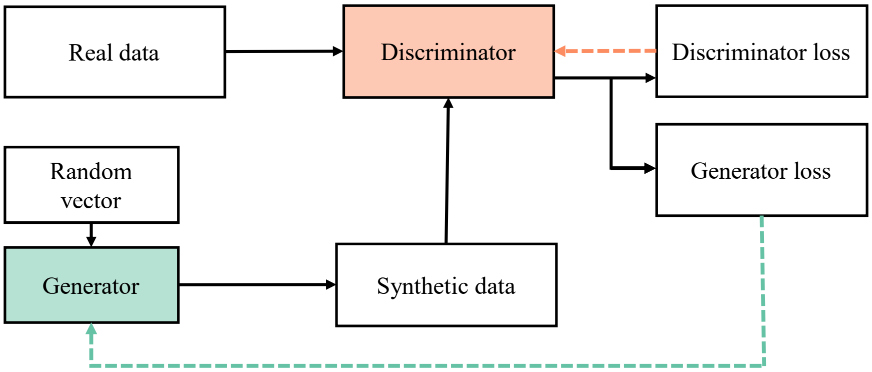


Figure A.2: GAN architecture (the dashed lines represent backpropagation)

Appendix B: case study 1 supplementary materials

B.1 toy problem parameter setting

The detailed parameters for each signal are listed in Table B.1.

Table B.1: Generated signals

Signal ID	Signal Function
1-1	$2t^3 - t^2 \quad 0 \le t < 120$
1-2	$2t^3 - t^2 \quad 0 \le t < 80$
1-3	$2t^3 - t^2 \quad 0 \le t < 70$
1-4	$2t^3 - t^2 \quad 0 \le t < 90$
2-1	$\begin{cases} 5t^2 - 0.5\mathrm{t} & 0 \le t < 40 \\ 0.5b_3t^2 - 15\mathrm{t} + 7780 & 40 \le t < 90 \\ 0.3t^3 + 10480 & 90 \le t < 120 \end{cases}$
2-2	$\begin{cases} 8t^2 - 5t & 0 \le t < 30 \\ 1.5t^2 - 55t + 7350 & 30 \le t < 60 \\ 0.05t^3 - 1350 & 60 \le t < 80 \end{cases}$
2-3	$\begin{cases} 12t^2 - \mathrm{t} & 0 \le t < 20 \\ 2t^2 - 55\mathrm{t} + 5080 & 20 \le t < 50 \\ 0.05t^3 + 1080 & 50 \le t < 70 \end{cases}$
2-4	$\begin{cases} 35t^2 - \mathrm{t} & 0 \le t < 25 \\ 5t^2 - 105\mathrm{t} + 21350 & 25 \le t < 60 \\ 0.1t^3 + 11450 & 60 \le t < 90 \end{cases}$

*In case study 1, each signal is normalized by dividing by its maximum value to rescale to [0,1], then the normalized signal is added with $w(\mu, \sigma)$, where $\mu = 0$ and $\sigma = 0.01$

B.2 Evolution of data distribution

This section shows the evolution of the synthetic data distribution to explain the superior outcome of HP-JT compared to HP-Noise and HP-VAE. Figure B.1 shows the evolution of the distribution with training epoch during pre-training and joint training. To begin with (epoch 0 of pre-training), the generated synthetic data is random noise depicted as blue dots in the t-SNE plot. As the generator gets trained with the input data, the synthetic data gradually approaches the distribution of the training dataset. During joint training, the distribution is further refined around the training distribution, but primarily, the HP-JT predictor learns the distribution of the synthetic data. Compared to the HP-Noise and HP-VAE models, the HP-JT model is trained with more diverse data that has a similar distribution to the real data.

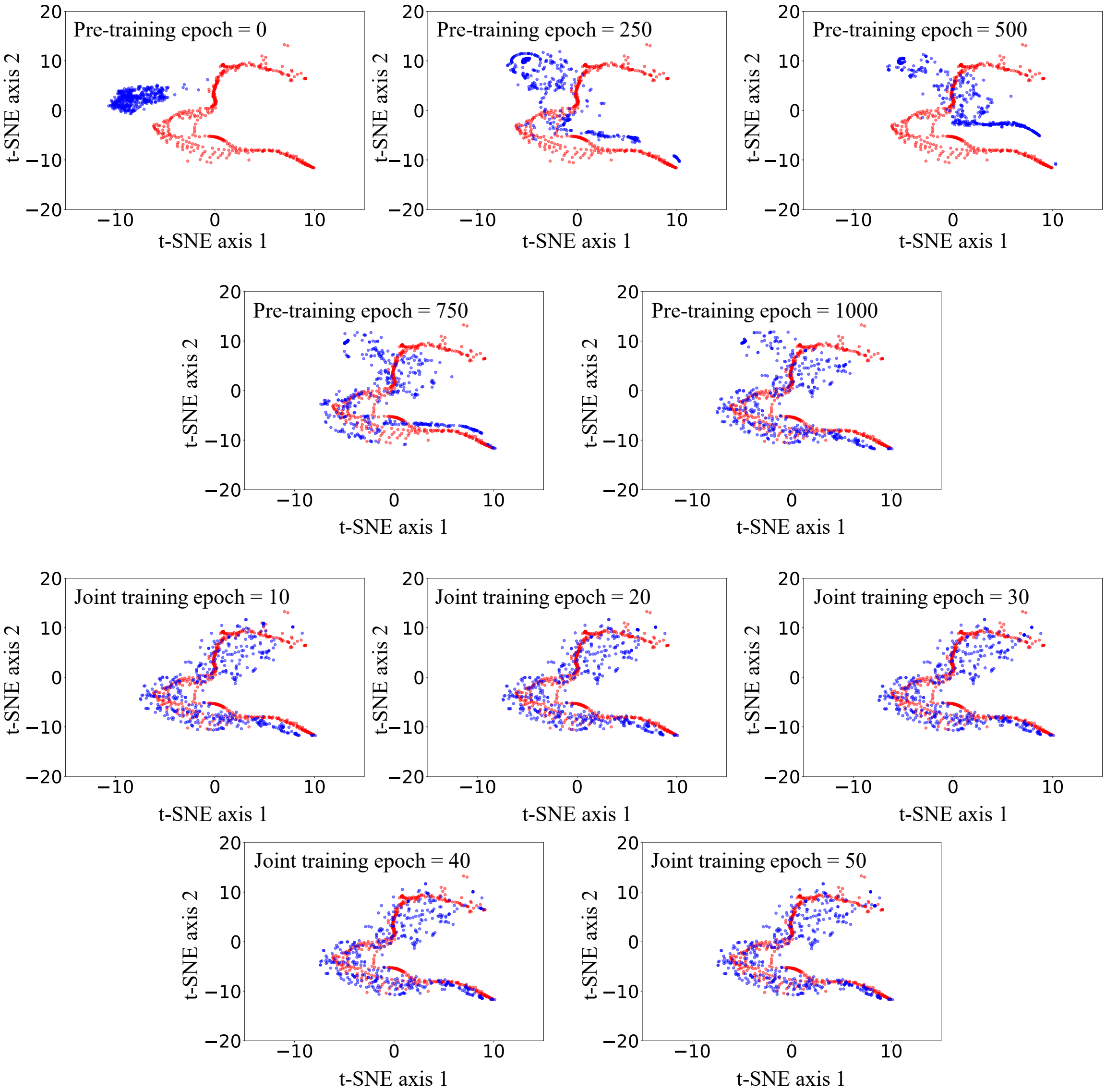


Figure B.1: t-SNE results of real data and synthetic data by GAN-LSTM at different training epochs

B.3 LSTM-noJT

As mentioned in section 3.1, the HP-noJT was initialized using the parameters provided by the pre-trained HP-JT model. In Figure B.2, we include an example where signal 2-4 is selected as test data to show the evolution of next-step prediction RMSE during the training. HP-JT converged faster than HP- noJT. After 100 training epochs, there is no significant change of training error for both HP-JT and HP- noJT. For the test error results, at the beginning of the training, HP-JT and HP-noJT have similar test errors, as HP-noJT was initialized using HP-JT's parameters. HP-JT converged faster than HP- noJT, and HP-JT provided less test error at the end of training.

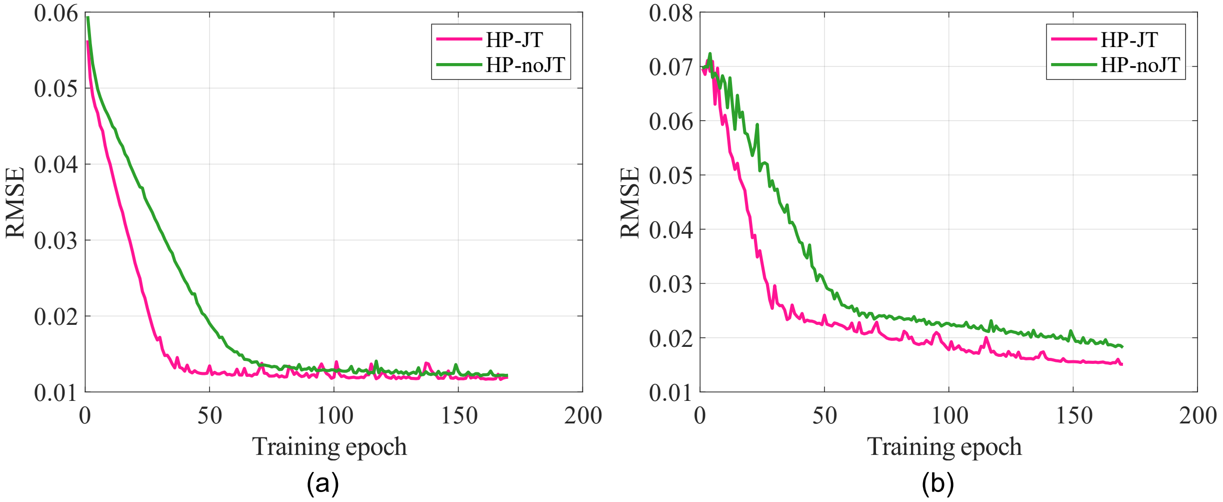


Figure B.2: Next-step prediction error at different training epoch (a) training error (b) test error

Appendix C: case study 2 supplementary materials

C.1 Extracted features

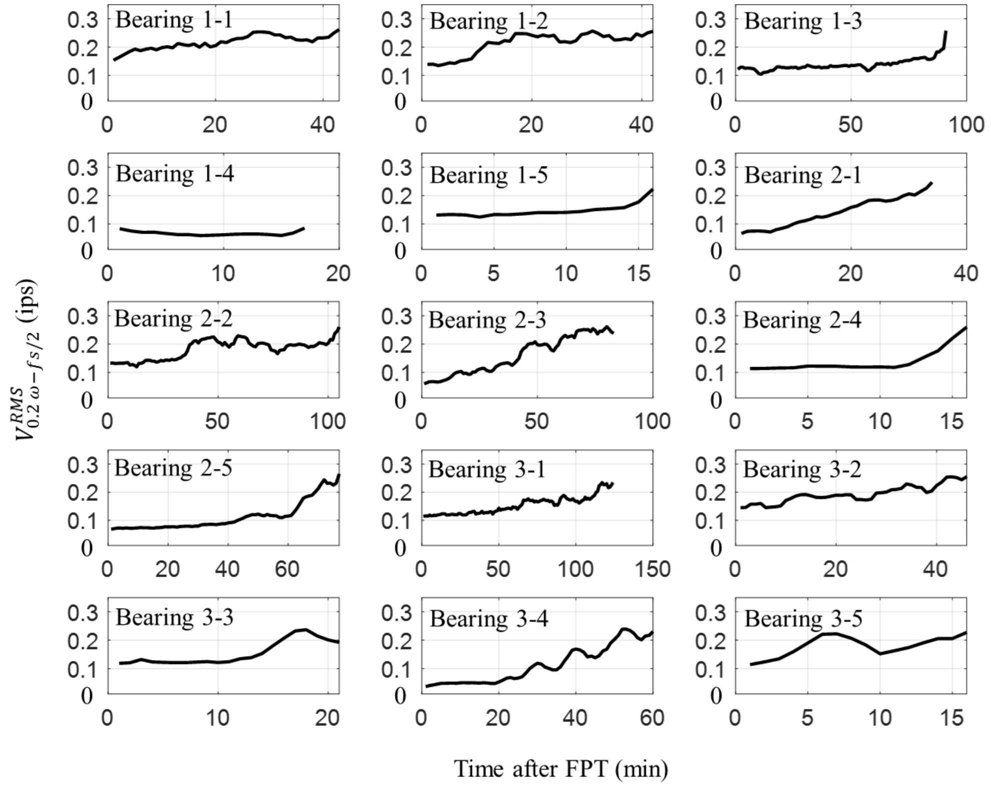


Figure C.1: The extracted $V^{\mathrm{RMS}}_{0.2\omega-f_s/2}$ values (from FPT to EOL) of each bearing

C.2 The distribution of synthetic data analysis

For the GAN-LSTM network, the evolution of the synthetic data generated distribution with respect to real data is shown in Figure C.2 at different training epochs before the joint training for cross-validation fold 2. Each point is a time series of length $k + 1$. At training epoch 0, when the generator and discriminator were just initialized, the generator's output was random noise, and the distribution of synthetic data was significantly different compared to the real data. As the training progressed, the generator learned the trend of real data through adversarial training, and the synthetic data became similar to the real data. Note that at training epoch 1000, like in the toy example, the distribution of synthetic data follows a similar local and global structure to the real data. The small variations between the distributions of the datasets help improve the HP-JT model's generality. This graphical comparison shows the generator network's ability to understand the distribution of the real data and produce high-quality synthetic data, which helps to deal with the challenge of limited training data.

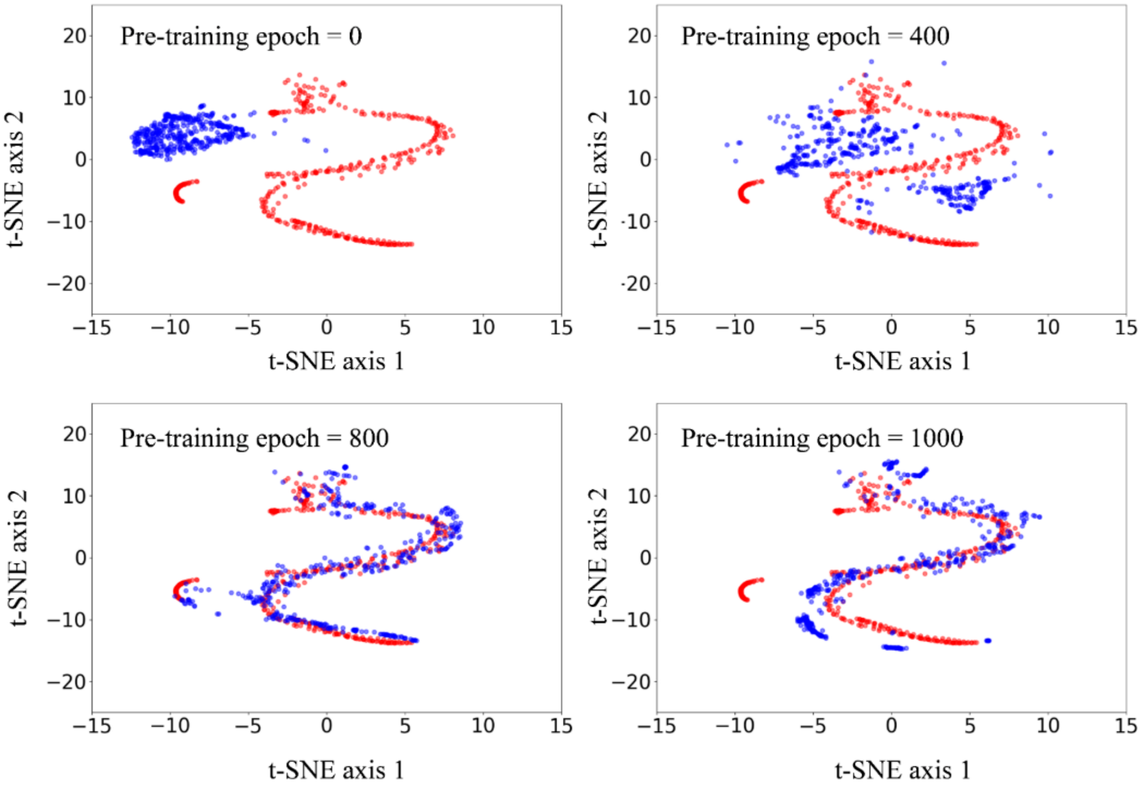


Figure C.2: t-SNE results of real and synthetic data at different training epochs

Next, we attempt to study the similarity between the real data of bearing degradation and the synthetic data generated from GAN-LSTM. When training the HP-JT model, the entire bearing degradation curve is split into time series of length k, which would also be the length of the synthetic data. Moreover, it is very difficult to identify which sample leads to which synthetic time series. Therefore, we use the dynamic time wrapping method (Kim et al., 2020) to select the synthetic data that is the most similar to a raw data segment. Finally, we concatenate the selected synthetic data and generate the synthetic bearing degradation curves.

Figure C.3 shows the raw and synthetic data for bearings 1-1, 2-1, 2-5, and 3-1. The synthetic bearing data, constructed from synthetic samples, have similar trends to real data. While following the overall trends of the training data, the synthetic data ignored some local fluctuations and showed smoother trends. Taking bearing 2-5 as an example, the artificial bearing data from 40 to 60 min is smoother than the real data. Looking at both Figure C.2 and Figure C. 3, we can see the data generated by the generator and HP-JT follows the overall degradation trends of the training data, which helps improve the HP-JT model's performance.

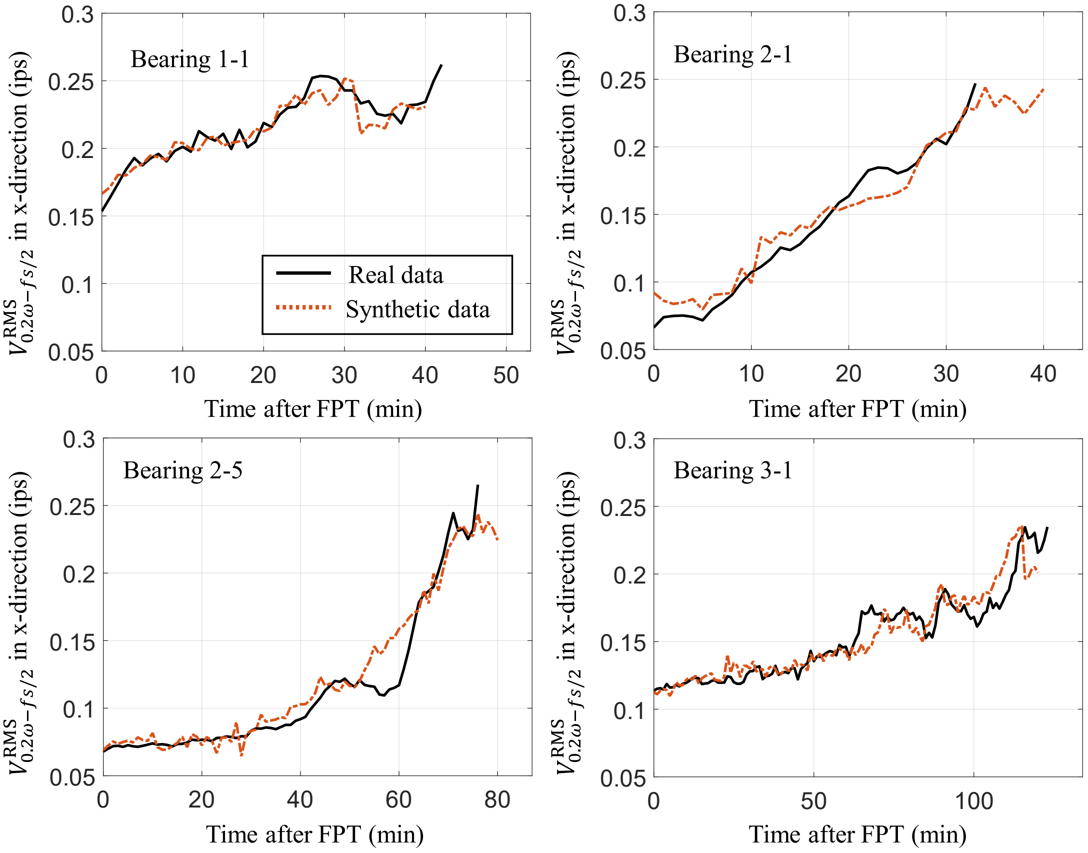


Figure C.3: Constructed artificial bearing data and the real data

Appendix D: Computational Efficiency

We compared the proposed HP-JT method with other deep learning methods in terms of the training time. The Fold 1 data of case study 2 were selected as test data, and the remainder data were used to train the model. Four models were trained using an Intel Core i7-10870H CPU @ 2.20GHz equipped with an NVIDIA RTX 3060 GPU with 6 GB dedicated GPU memory and 16 GB of system RAM. The scripts for constructing and optimizing all models were Python codes (Python Version 3.9.1).

To minimize the effects of randomness during the measurements, here we summarize the mean computational times over the ten runs. The training time of HP + VAE is composed of two parts: the training time of the VAE and the training time of the predictor. The training time of HP-JT is composed of three parts: (1) the pre-training time of HP-JT, (2) the pre-training time of the generator and discriminator, and (3) the joint training time of the generator, discriminator, and HP-JT.

Table D.1: Comparison of computational time of different approaches

	HP	HP-Noise	HP-VAE	HP-JT
Computational time (s) of each step	3.34	6.58	Train VAE: 6.55 Train predictor: 6.66	Pre-training of HP-JT: 1.49 Pre-training GAN: 14.56 Joint training: 2.40
Total training time (s)	3.34	6.58	13.21	18.45
Evaluation time (s)	0.023	0.021	0.018	0.019

The training time of the proposed HP-JT is 18.45 s, which is more than five times compared to the training time of HP. The pre-training of generator and discriminator takes 14.56 s, accounting for a large portion of the training time.

Though HP-JT requires a longer training time, only the predictor is involved in the RUL prediction process. As mentioned in section 2.4, the predictors used in HP, HP-Noise, HP-VAE, and HP-JT have the same architecture (one LSTM layer with the number of units = 60 followed by a fully connected layer). For each model, the time of performing 100 next-step predictions is listed in Table D.1. The evaluation time of each model is around 0.02 s. These numbers show that, regarding the RUL prediction process, the complexity of each model is similar.

Appendix E: Configuration of the probabilistic network

Table E.1: The specific configuration of the probabilistic network.

Module name	Layer	Output shape, Activation
Generator	Input	(Samples, 20)
	Fully connected	(Samples, 64), Linear
	Fully connected	(Samples, 32), Linear
	Fully connected	(Samples, 20), ReLU
Discriminator	Input	(Samples, 21)
	Fully connected	(Samples, 64), Linear
	Fully connected	(Samples, 128), ReLU
	Fully connected	(Samples, 64), ReLU
	Fully connected	(Samples, 1), Sigmoid
HP-JT	Input	(Samples, 20, 1)
	LSTM	(Samples, 60), Tanh
	Gaussian	(Samples, 2), Linear