Soft Computing (2023) 27:3641–3654
https://doi.org/10.1007/s00500-022-07625-4

DATA ANALYTICS AND MACHINE LEARNING

A deep learning-based conditional system health index method to reduce the uncertainty of remaining useful life prediction

Jaeyeon Jang[1]

Accepted: 1 November 2022 / Published online: 9 November 2022

Abstract
Many recent data-driven studies have used sensor profile data for prognostics and health management (PHM). However, existing data-driven PHM techniques are vulnerable to three types of uncertainty: sensor noise inherent to the sensor profile data, uncertainty regarding the current health status diagnosis caused by monitoring a single health index (HI), and uncertainty in predicting the remaining useful life (RUL), which is affected by unpredictable changes in system operating conditions and the future external environment. This study proposes a deep conditional health index extraction network (DCHIEN) for PHM to effectively manage these three types of uncertainty. DCHIEN is a model that combines a stacked denoising autoencoder that extracts high-level features robust to sensor noise with a feed-forward neural network that produces an HI based on user-defined monitoring conditions. This approach supports system health monitoring using the conditional HI, as well as prognostics using RUL interval predictions. Extensive experiments were conducted using NASA's turbofan engine degradation dataset. The results show that the proposed method achieves a superior RUL prediction performance compared to state-of-the-art methods and that uncertainties can be effectively managed.

Keywords Health index · Prognostics and health management · Remaining useful life · Stacked denoising autoencoder · Uncertainty management

1 Introduction

Prognostics and health management (PHM) is a systematic approach for diagnosing a system's current health status and predicting the failure time (Zhao et al. 2014; Zhang et al. 2019) based on information from sensors, domain knowledge, and external environmental factors. Effective PHM can significantly reduce system maintenance and operational costs and ensure long-term system stability (Huynh et al. 2014; Zhu et al. 2019). Recently, PHM has played a key role in the stable and cost-efficient operation of ship and aircraft engines (Sun et al. 2015; Wang et al. 2019; Diez-Olivan et al. 2018), lithium-ion batteries (Rezvanizaniani et al. 2014; Waag et al. 2014; Liu et al. 2015), and equipment for various manufacturing processes (Yang and Lee 2012; Yang et al. 2016; Zhao et al. 2017).

In the past few years, advances in statistical learning and machine learning have enabled engineers to analyze the vast amounts of sensor profile data provided by multiple sensors in engineering systems (Wijayasekara et al. 2014). The information extracted from these learning techniques helps engineers determine not only the current system's health status but also the expected failure time.

Data-driven PHM approaches are broadly divided into Bayesian and machine learning approaches (Zhang et al. 2018). In general, the Bayesian approaches transform the sensor profile data into a health index (HI) that represents the system's health status and models HI transitions over time based on certain probability distributions. Because HI transition models estimate the HI at the next time point based on previous HI change patterns, recursive HI estimation can predict the future performance degradation patterns of a system and ultimately predict its remaining useful life (RUL) (Wang and Gao 2015; Lim et al. 2017; Hong et al. 2015; Si et al. 2017).

Machine learning approaches can be broadly categorized into two types. The first type predicts a system's future degradation patterns by extracting the HI at certain time points

✉ Jaeyeon Jang
jaeyeon.jang@catholic.ac.kr

[1] Department of Data Science, The Catholic University of Korea, 43 Jibong-ro, Bucheon 14662, Republic of Korea

from the sensor profile data and then learning the change patterns of the extracted HI using machine learning models (Yang et al. 2016). Various models, including autoregression (Qian et al. 2013), support vector regression (SVR) (García Nieto et al. 2015; Tran et al. 2012), and long short-term memory (LSTM) (Zhang et al. 2018), have been used to learn the change patterns. The second approach directly predicts the RUL without learning the health degradation pattern of a system. Traditionally, logistic regression (Liao et al. 2006), multilayer perceptron (Tian et al. 2010; Tian 2012), and SVR (Benkedjouh et al. 2013) have been used as the RUL prediction models in this approach.

In the past few years, efforts to employ deep learning models have increased. For example, Jiang and Kuo (2017) and Li et al. (2018) used a convolutional neural network (CNN) to predict the RUL of aircraft engines. They showed that CNNs are better than traditional machine learning models in terms of prediction performance. Similarly, another deep learning model, LSTM, was successfully applied to predict the RUL of computer numerical control milling machine cutters (Zhao et al. 2017) and aircraft engines (Wu et al. 2018). Recently, Zhang et al. (2022) proposed a bidirectional gated recurrent unit with a temporal self-attention mechanism to consider the reverse flow of profile data and to reflect the difference in RUL prediction results at different time instances. They suggested that the bidirectional gated recurrent unit, as an improved LSTM, can improve the prediction performance by simplifying the network structure. Deutsch and He (2018) used a deep belief network to develop an RUL prediction model for rotating components such as gears. They showed that when a vast amount of sensor profile data is used in PHM, deep learning models can yield better performance than shallow machine learning models that require prior knowledge of the model structure and signal processing technology. Recently, advanced learning techniques have been introduced to cope with limited failure data in real-world PHM cases. For example, Listou Ellefsen et al. (2019) applied a semi-supervised learning technique to train a new deep neural network (DNN) with insufficient high-quality labeled data. Similarly, Jang and Kim (2021) proposed a Siamese network-based RUL prediction method that utilizes training samples as references to cope with situations in which limited historical data are available.

However, the aforementioned existing data-driven PHM methods are vulnerable to the influence of uncertainties. In this paper, we consider the following three types of uncertainty, which are redefined based on Sankararaman et al. (2013):

1. Noise occurs when sensors measure the system parameters and adds variability to the system health status diagnosis and RUL prediction (Daigle et al. 2014; Javed et al. 2015). Thus, noise-reduced data should be input into the PHM model.
2. System degradation patterns occur for a wide variety of reasons according to the system characteristics. Even the degradation patterns of a single system can vary with its health status (Eker et al. 2012). Thus, the PHM model often does not effectively retrieve information about the system health status (as represented by the HI) from the sensor profile data. Therefore, the PHM model should be able to handle the uncertainty in diagnosing the current health status by providing a variety of HI monitoring results.
3. The task of predicting the RUL is highly uncertain. When predicting the RUL, it is difficult to fully consider the system operating conditions and external environment changes that may occur in future, leading to uncertainty in the prediction results (Sankararaman 2015). Therefore, it is not reliable to model the PHM system assuming that accurate RUL prediction is possible. Rather, the PHM system should be responsive to prediction errors by providing supplemental information.

This paper proposes a PHM method that focuses on managing the three types of uncertainty mentioned above. The proposed PHM method is based on the deep conditional HI extraction network (DCHIEN) presented here. DCHIEN is a model that combines a stacked denoising autoencoder (SDAE) (Vincent et al. 2010), which extracts high-level features that are robust in regard to the noise inherent in the input data, with a feed-forward neural network. The network predicts the HI after considering the user-defined monitoring conditions and the features extracted by the SDAE. When the actual health status of the system is the same as or similar to the user-defined monitoring conditions, DCHIEN properly learns the degradation patterns for those conditions. Therefore, the proposed model allows engineers to set multiple monitoring conditions and monitor the changes in conditional HIs (CHIs) for each set condition. In addition, DCHIEN can be used to estimate the RUL prediction intervals. Therefore, even when the RUL point predictions are poor, engineers can establish conservative maintenance plans based on the RUL prediction intervals.

The main contributions of the paper are as follows:

- In this paper, the novel model DCHIEN is proposed to address the three major types of uncertainty. In DCHIEN, a denoising autoencoder is first pretrained to minimize the risk of sensor noise and is used as a feature extractor. Then, we apply a new training method to learn CHIs.
- DCHIEN can help engineers manage systems effectively by providing a diverse set of information, including multiple CHIs, RUL point estimates, and RUL prediction intervals.

- We compare the proposed RUL prediction method with state-of-the-art methods, and the comparison results show that the proposed method achieves the best performance. In addition, it is shown that the prediction performance can be significantly improved by considering the points in the very small RUL prediction interval as possible points of failure.

The remainder of this paper is organized as follows: In Sect. 2, related works are reviewed. Section 3 describes the structure and learning method of the DCHIEN model. Section 4 presents the PHM method based on the DCHIEN model. In Sect. 5, the proposed PHM method is evaluated and compared with existing RUL prediction methods. Finally, Sect. 6 describes the conclusions drawn from this study and briefly discusses the limitations of this paper and future research directions.

2 Related works

2.1 Uncertainty management in prognostics and health management

There are several sources of uncertainty that affect the current operations and future behaviors of systems (Sankararaman et al. 2013; Sankararaman 2015), so uncertainty management is important for PHM. Therefore, often, it is not meaningful to perform diagnosis and prognosis without considering the presence of uncertainty inherent in the system of interest. There are three major uncertainties described in Sankararaman et al. (2013): state uncertainty, future loading uncertainty, and process noise. First, in most cases, the true state at any instant is not known precisely and is impossible to estimate with certainty due to some uncertainty sources, including sensor measurement noise and model uncertainty. Let s_t be the state of a system and η_t be the sensor measurement noise at t. Then, the state uncertainty can be defined as $|f(s_t+\eta_t)-\hat{f}(s_t)|$, where f reveals the current health status of the system of interest and $\hat{f}$ is a model that approximates f. Second, we cannot predict future loading and environmental conditions that are necessary information for estimating the RUL. This is called future loading uncertainty. In addition, process noise in future cannot be estimated in advance. Let $s_{t:t+\tau} + \eta_{t:t+\tau}$ be the set of true states considering process noise, and let $\hat{s}_{t:t+\tau}$ be the set of states estimated by a model from t to $t+\tau$. Then, the uncertainty in RUL prediction can be defined as $|q(s_{t:t+\tau}+\eta_{t:t+\tau})-q(\hat{s}_{t:t+\tau})|$, where q outputs the health status at the last instant in the input.

Not many studies have taken these uncertainties into account in PHM modeling. Only a few studies have proposed estimating the RUL prediction interval and using it to mitigate the effect of the uncertainty in RUL prediction. The main idea is that we can formulate a more robust maintenance strategy by considering all points in the prediction interval as possible failure points. For example, Bressel et al. (2016) estimated probability bounds by repeating extended Kalman filter-based RUL estimation. Similarly, Liao et al. (2018) estimated the prediction interval of the RUL using LSTM and showed that more effective maintenance decisions can be obtained with prediction intervals. Liu et al. (2018) showed that the RUL prediction error can be reduced by considering the uncertainty of random degradation.

2.2 Stacked denoising autoencoder

An autoencoder is a simple feed-forward neural network that consists of an input layer, a single hidden layer, and an output layer. The autoencoder is trained to reconstruct the input data in the output layer. Here, the hidden layer learns the nonlinear features to effectively reconstruct the data. A denoising autoencoder (DAE) is an extended autoencoder version, the goal of which is to reconstruct the original input from an input corrupted by random noise, and it is trained to minimize the reconstruction loss. A well-trained DAE can extract features that are robust to the noise inherent in the input data (Vincent et al. 2008; Jang et al. 2019).

An SDAE has a structure that stacks several layers of a DAE; it uses a layer-by-layer training method in which the hidden layers are trained sequentially and individually to solve the gradient vanishing problem (Hochreiter 1998). An SDAE trained using the layer-by-layer training method can extract high-level features that are robust to noise. The SDAE training method is discussed in detail in Vincent et al. (2010).

An SDAE is a high-level feature extractor that learns through unsupervised learning. A DNN is formed by connecting an SDAE to a feed-forward neural network that predicts an output based on the higher-level features extracted by the SDAE. Therefore, the SDAE training process can be considered a pretraining process that initializes the weights of the entire network. Then, the pretrained DNN is trained again through a back-propagation algorithm to find the relationship between the input and output, a process called fine-tuning. The fine-tuning process for DCHIEN training is described in detail in Sect. 3.

3 Deep conditional health index extraction network

DCHIEN's structure combines an SDAE and a single hidden-layer CHI extractor, which receives the high-level features extracted by the SDAE and the monitoring conditions as input and produces the CHI as output, as shown in Fig. 1. Let σ be an activation function, $\boldsymbol{Z}_L$ be a feature vector pro-

duced by the SDAE with N_L elements, and C be a monitoring condition. Then, the hidden representation of the CHI extractor $\boldsymbol{Z}^h$ is computed as shown in (1), where $\boldsymbol{W}_j^h$ and $\boldsymbol{b}_j^h$ are the jth weight matrix and bias vector of the CHI extractor, respectively, and $[C/\boldsymbol{Z}_L] = [C, z_{L1}, \ldots, z_{LN_L}]^T$. When $\boldsymbol{Z}^h$ is mapped to CHI, h, the sigmoid function is used as the activation of the DCHIEN output node so that the CHI is assigned a value between zero and one, as shown in (2). For simplicity, we define f, the function that produces h through the SDAE and CHI extractors, as shown in (3).

$$\boldsymbol{Z}^h = \sigma\left(\boldsymbol{W}_1^h \begin{bmatrix} C \\ \boldsymbol{Z}_L \end{bmatrix} + \boldsymbol{b}_1^h\right). \tag{1}$$

$$h = \text{Sigmoid}\left(\boldsymbol{W}_2^h \boldsymbol{Z}^h + \boldsymbol{b}_2^h\right) \tag{2}$$

$$h = f(\boldsymbol{X}, C) \tag{3}$$

In this paper, the CHI is intended to show the relative value of a given condition C compared to the current health status, where C is the RUL value expected by an engineer. Thus, DCHIEN is trained to produce a CHI of 1 if C is to large compared to the true RUL. On the other hand, a CHI of 0 is produced if the expected RUL (C) is too small. Thus, we set 0.5 as the middle point, at which the true RUL and expected RUL are equal.

The DCHIEN model extracts the CHI, which is information that cannot be provided beforehand. Thus, it requires a fine-tuning method that differs from that of a supervised DNN, in which both the input and output are clearly provided. Let us assume that we are provided with a training dataset that includes the sensor profile data and the RUL values that correspond to each data sample. Specifically, 1 is assigned when C is greater than the RUL, 0 is assigned when C is less than the RUL, and 0.5 is assigned if C and the RUL are the same. By repeatedly learning a randomly assigned C and the resulting changes in the CHI for each iteration, the method automatically learns the degradation pattern for all Cs, where the CHI value approaches 1 or 0 from 0.5 as the true RUL moves farther from C. After training, DCHIEN can be utilized in two ways. First, because this model is trained to output a CHI of 0.5 when the given condition C and the true RUL are equal, we can monitor several degradation patterns according to C. In general, as C increases, degradation occurs earlier because CHI reaches 0.5 at earlier time instances when the true RUL is high. Second, given the same input profile data, DCHIEN produces a higher CHI for a higher C because DCHIEN is more likely to output 1 with a larger C by the training method. This characteristic is used to suggest an RUL prediction interval. The specific fine-tuning method is as follows:

1. The number of training iterations TR, the batch size used in one weight update B, and ρ, which is the ratio of data that have their RUL values as C values, are assigned. ρ is needed because the probability that C and the RUL values are identical is extremely low when C is assigned randomly.
2. B samples that have not been used for training in the current iteration are randomly selected as the training batch.
3. DS_{higher}, DS_{lower}, and DS_{same} are initialized as empty sets ϕ.
4. Samples are randomly selected from the training batch according to ρ. For each selected sample, the RUL is assigned as its C value and the sample is included in DS_{same}. For the other samples in the batch, $C \sim Uniform(1, RUL^{\max})$ is assigned. Here, $RUL^{\max}$ is the maximum RUL of the system, which is defined by the engineer. Additional details on $RUL^{\max}$ are given in Sect. 4. The samples in which the assigned C is larger than the RUL are included in DS_{higher}, those in which C is less than the RUL are included in DS_{lower}, and the remaining samples are included in DS_{same}.
5. If the assigned monitoring condition of sample $\boldsymbol{X}_i$ is C_i, the batch error E is defined as shown in (4). The weights of DCHIEN are updated through a back-propagation algorithm so that E is minimized.

$$E = \frac{1}{B}\Bigg(\sum_{i \in DS_{higher}} (f(\boldsymbol{X}_i, C_i) - 1)^2 + \sum_{i \in DS_{\text{same}}} (f(\boldsymbol{X}_i, C_i) - 0.5)^2 + \sum_{i \in DS_{\text{lower}}} f(\boldsymbol{X}_i, C_i)^2\Bigg). \tag{4}$$

6. Repeat (2)–(5) until all training data have been used in the current iteration.
7. Repeat (2)–(6) until TR is satisfied.

4 Proposed PHM method

In this section, the procedure for the proposed PHM method is provided, and Fig. 2 shows an outline. The first step is data preprocessing, which includes min–max normalization, time window processing, and the application of a piecewise linear degradation assumption, which assumes that performance degradation rarely occurs at the beginning of system operation. Subsequently, the preprocessed training dataset is used to train DCHIEN. The trained DCHIEN extracts the CHI from the incoming data and uses it to perform real-time health monitoring and prognostics to predict the RUL point estimate and prediction interval. Detailed descriptions of the data preprocessing method, CHI-based real-time health monitoring, and prognostics are provided in the next subsections.

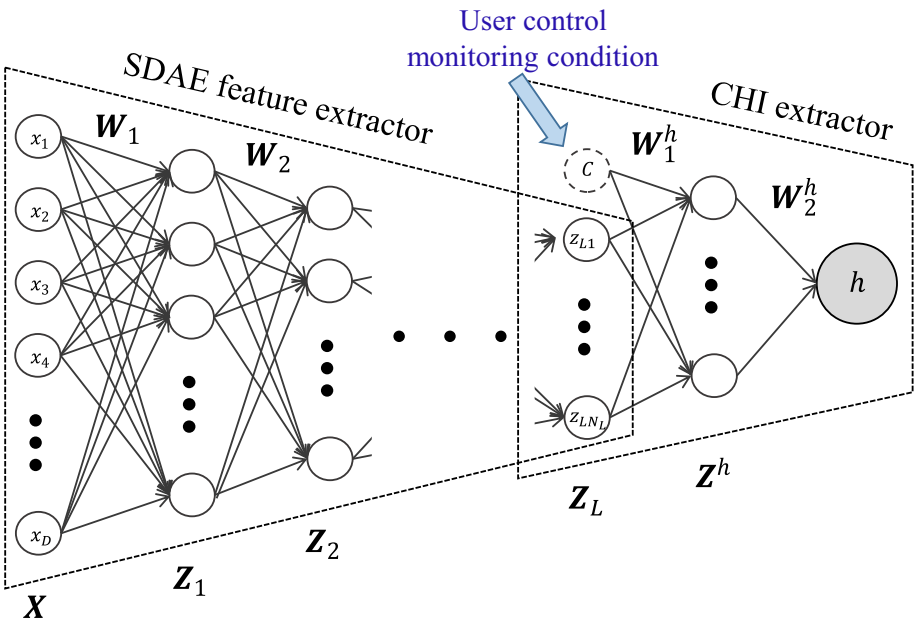

Fig. 1 The structure of DCHIEN

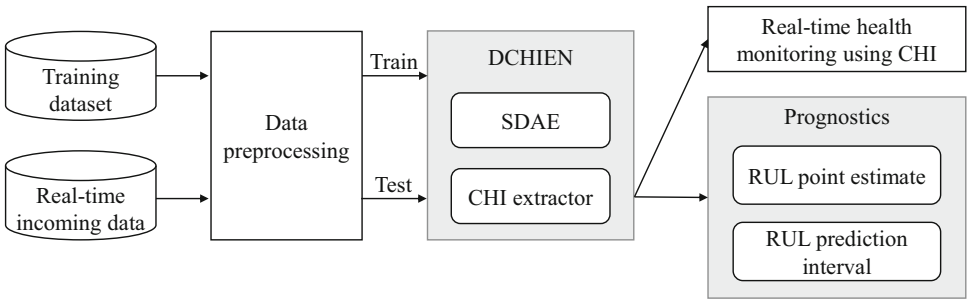

Fig. 2 The procedure of the proposed PHM method

4.1 Data preprocessing

We apply min–max normalization, allowing the system parameter measurement values for each sensor to have the same scale. If the tth measurement value of the system parameter k is called x_{kt}, its normalized value is defined as $x_{kt}^{\text{norm}} = (x_{kt} - x_k^{\min})/(x_k^{\max} - x_k^{\min})$. Here, $x_k^{\max}$ and $x_k^{\min}$ are the maximum and minimum values of the measurements of parameter k in the training dataset, respectively.

When using the sensor profile data to diagnose the system status and predict the RUL, more information can be obtained by using temporal sequence data than by using the system parameter values measured at a single point in time because temporal sequence data include momentary system parameter information and information on the parameter change patterns over time (Ramasso et al. 2013). Therefore, this study introduces time window processing and uses multivariate temporal sequence data as a single input sample. For example, if the time window size is TW and the number of system parameters is K, the normalized data $[x_{1t}^{\text{norm}}, \ldots, x_{1(t+TW-1)}^{\text{norm}}, x_{2t}^{\text{norm}}, \ldots, x_{K(t+TW-1)}^{\text{norm}}]$ are treated as a single sample, and the time-to-failure at the last time point is the RUL of the sample.

Normally, when an engineering system is newly introduced or reintroduced directly after receiving maintenance, almost no performance degradation occurs during system operation; instead, degradation progresses gradually after a certain operational period (Heimes 2008). Therefore, the model is constructed so that during the initial system operation, the RUL remains constant with no degradation. Then, after a certain time, it is assumed that linear degradation occurs, causing a linear reduction in performance (Jiang and Kuo 2017; Li et al. 2018; Zhao et al. 2017; Wu et al. 2018). To adopt this assumption, we set a maximum RUL value $RUL_{\max}$. Samples that have an RUL above $RUL_{\max}$ are assigned $RUL_{\max}$.

4.2 Health monitoring using CHIs

Even for a single system, the system performance degradation patterns can vary significantly based on system health (Eker et al. 2012). Considering this, it is proposed that the engineer set several monitoring conditions and monitor the

CHIs produced by DCHIEN in response to each monitoring condition. DCHIEN's training process results in an extracted CHI that properly reflects the degradation patterns when the system's state of health is poor under low monitoring conditions C; conversely, it also properly reflects the degradation patterns when the system is healthy under a high C. Therefore, when the engineer monitors multiple CHIs, even if the CHI based on a certain C does not properly represent the system degradation pattern, a CHI that is based on another C may be able to represent that pattern. In this way, the engineer can consider a variety of possibilities for the health status, minimizing the uncertainty in diagnosing the current health status.

4.3 RUL and interval prediction

DCHIEN is trained to output 0.5 when the monitoring condition C equals the true RUL. Therefore, we can use a C value that produces a CHI of 0.5 because it is considered the most probable estimate of the RUL. $\widehat{RUL}_i$, the predicted value of the RUL for the normalized sample $\boldsymbol{X}_i^{\text{norm}}$, is defined as follows:

$$\widehat{RUL}_i = \underset{C}{\text{argmin}} \, |f(\boldsymbol{X}_i^{\text{norm}}, C) - 0.5|. \tag{5}$$

In addition, this paper proposes a method for predicting the interval of the RUL. As a result of the proposed training scheme, ideally, DCHIEN produces a CHI that increases from 0.5 as the input C increases from the true RUL. Conversely, CHI decreases from 0.5 as C decreases from the true RUL. Thus, considering the uncertainty in predicting the RUL, if we find that a C produces a CHI that is higher or lower than 0.5, then Cs can be used as upper or lower bounds of the RUL interval. Let us assume that a deviation of δ with respect to 0.5 is allowed in the CHI. Then, the RUL interval CI_δ according to δ is as follows:

$$\begin{aligned} CI_\delta = [&\underset{C}{\text{argmin}} \, |f(\boldsymbol{X}_i^{\text{norm}}, C) - (0.5 - \delta)|, \\ &\underset{C}{\text{argmin}} \, |f(\boldsymbol{X}_i^{\text{norm}}, C) - (0.5 + \delta)|]. \end{aligned} \tag{6}$$

In practice, 0.01–0.2 is suggested for δ.

5 Case study

5.1 Experimental setup

In this study, extensive experiments are conducted using NASA's C-MAPSS turbofan engine degradation dataset (Saxena et al. 2008). The dataset was obtained by simulating an engine that has a random initial wear level. In each simulation, a fault degrades the performance of the engine, which was initially considered healthy. In the engine, 21 sensors are equipped to measure system parameters such as temperature, pressure, and fan speed during each operation cycle.

The C-MAPSS dataset includes four subdatasets collected under different simulation settings: FD001, FD002, FD003, and FD004. They are categorized based on whether the setting has multiple operational conditions and which kinds of faults cause the degradation. A summary of the four subdatasets is given in Table 1. Specifically, FD001 and FD002 are affected by high-pressure compressor faults, while FD003 and FD004 are subject to high-pressure compressor and fan faults. For the training profiles, sensor measurement values from the start of engine operation to the failure point are included. In contrast, the test profiles include sensor measurement values only up to a certain time point before engine failure occurs. Min–max normalization, time window processing, and a piecewise linear degradation assumption, under which $RUL_{\max}$ is 125 based on Li et al. (2018), were applied to the datasets. The time window size is set to 30 for FD001 and FD003, 20 for FD002, and 15 for FD004 based on the subdataset information and sensitivity analysis results provided in Li et al. (2018).

To evaluate the proposed RUL prediction method, five performance metrics were used. The first metric is Score, which has commonly been used to measure RUL prediction performance (Listou Ellefsen et al. 2019). Score is defined as follows:

$$\text{Score} = \sum_{i=1}^{N} s_i, \; s_i = \begin{cases} e^{-\frac{d_i}{13}} - 1 & \text{for } d_i < 0 \\ e^{\frac{d_i}{10}} - 1 & \text{for } d_i \geq 0 \end{cases}, \tag{7}$$

where N is the total number of test samples and $d_i = \widehat{RUL}_i - RUL_i$ (predicted RUL–true RUL). The second performance measure is Accuracy, which was proposed in Saxena et al. (2008) to evaluate the percentage of correct predictions. Here, a prediction is considered correct if $-13 \leq d_i \leq 10$. We also used common prediction performance measures, including the root mean squared error (RMSE), mean absolute error (MAE), and mean percentage absolute error (MAPE), defined in (8), (9), and (10), respectively.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d_i^2}. \tag{8}$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |d_i|. \tag{9}$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^{N} \frac{|d_i|}{RUL_i}. \tag{10}$$

Several model parameters should be set in advance to train the proposed DCHIEN. Thus, to find the optimal model

Table 1 Summary of the Four C-MAPSS Subdatasets

Dataset	FD001	FD002	FD003	FD004
Number of operating conditions	1	6	1	6
Number of fault modes	1	1	2	2
Number of training profiles	100	260	100	249
Number of testing profiles	100	259	100	248

parameters that minimize the RMSE, we used Bayesian optimization, which is an optimal parameter search method based on a Gaussian process (Snoek et al. 2012; Jang et al. 2020). The training profiles of FD001 were used for Bayesian optimization. The model parameters that were found through Bayesian optimization included parameters related to the model structure and training; these model parameters and set values are listed in Table 2. In addition, some of the model parameters were preset to reduce the Bayesian optimization search space. First, the SDAE has two hidden layers, and the CHI extractor has one hidden layer. The training batch size was set to 100, and a tangent hyperbolic activation was used as the activation function of the hidden nodes.

5.2 CHI monitoring and RUL prediction results

In general, the RUL is the best abstraction that can show the current health status in a real-world PHM scenario. Thus, the more accurately the HI reveals the RUL change pattern, the better it can be used for monitoring purposes. Figure 3 shows the CHI change patterns according to C for the 24th and 38th test profiles of FD003, which have almost all the data samples until just before system failure occurs. Figure 3 includes the CHI values, the true RUL values for each cycle, and the CHI trend line. The changes in the CHI values were similar to the shape of a sigmoid curve. Thus, the SciPy Python package was used to find the optimal sigmoid trend line that best fit the pattern of CHI change.

Figure 3 shows that a single CHI cannot represent the health status for all cycles. However, the CHI can precisely reflect the RUL change pattern in certain intervals. For example, when C was 50, the CHI change pattern was almost the same as the RUL change pattern in cycles in which the true RUL was less than 40. In contrast, when C was 80 and the true RUL values of the cycles were above 80, as shown in Fig. 3(b) and (d), the CHI change patterns were almost the same as the changes in the true RUL. These observations show that engineers can mitigate the uncertainty in diagnosing the current health status by monitoring the CHI changes based on low C values after the engine has been running for a long time and based on high C values when the engine is healthy.

Figure 4 shows the RUL point estimate and its prediction interval results for the 24th and 38th test profiles of FD003. At first, the predicted RUL estimate follows the true RUL better than the CHI. However, in some intervals, the predicted RUL does not fit well with the true RUL. For example, when the true RUL is larger than 80 and less than 150, there is a large difference between the true and predicted values for the two profiles in Fig. 4. In this case, engineers can apply the prediction interval, which can cover the true RULs for most cycles, as supplemental information to support robust maintenance decisions. Furthermore, considering that the CHI produced with $C = 80$ can reflect the RUL change pattern where the true RUL is above 80 in Fig. 3, the CHI can be used as auxiliary information to compensate for errors in RUL prediction.

5.3 Ablation study

This section provides an ablation study to analyze the contribution of each component to the RUL prediction performance. FD001 and FD003 were used for the ablation study, and all available samples in the testing profiles were used. In other words, not only the sample just before engine failure but also the other samples were used for testing. For this analysis, we varied the time window size TW to confirm its effect. Specifically, we compared the following baselines:

1. NN: The first baseline model is a neural network that adopts the same network configuration as DCHIEN without a condition node. This model is trained to directly predict the RUL.
2. NN with stacked autoencoder (SAE) pretraining (NN-SAE): In this baseline, the first two hidden layers of the neural network are pretrained using the SAE's layer-by-layer training method.
3. NN with SDAE pretraining (NN-SDAE): Random noise corruption for denoising is added in the layer-by-layer pretraining of NN-SAE.
4. CHI modeling with SAE pretraining (CHI-SAE): Here, the denoising component for sensor noise reduction is excluded from the proposed model.
5. Proposed method: The proposed method can be seen as adding the denoising component to CHI-SAE, adding the CHI modeling component to NN-SDAE, and adding both components to the simple neural network.

Table 2 Optimized model parameter set

Model parameters	Values
ρ	43
Number of nodes in the 1st hidden layer of the SDAE	43
Number of nodes in the 2nd hidden layer of the SDAE	72
Number of hidden nodes in the hidden layer of the CHI extractor	37
Learning rate for the 1st hidden layer of the SDAE	9×10^{-5}
Learning rate for the 2nd hidden layer of the SDAE	4×10^{-3}
Fine-tuning learning rate	2×10^{-4}

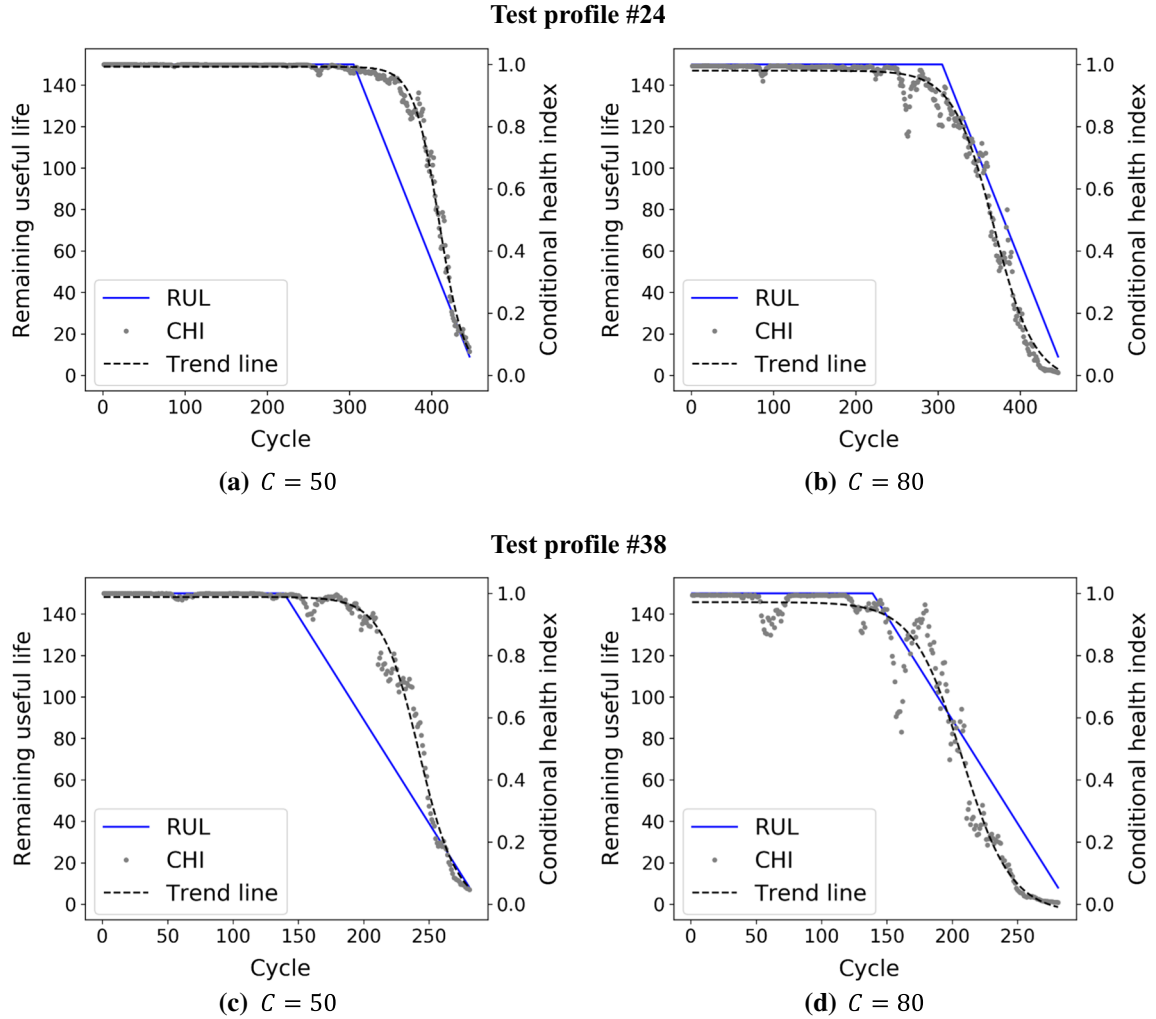

Fig. 3 The CHI change patterns for the 24th test profile of FD003 when **a** $C = 50$ and **b** $C = 80$, and those for the 38th test profile of FD003 when **c** $C = 50$ and **d** $C = 80$

The experimental results are given in Tables 3 and 4. The best and second-best results are highlighted in bold and underlined, respectively. First, for all metrics, a large TW contributes to an improvement in performance. However, in practice, there is a limitation on TW, and the contribution becomes marginal as TW increases (Li et al. 2018). Because each evaluation metric has a different characteristic, the proposed method could not provide the best performance for all metrics in all experimental settings. However, the two tables show that the proposed method achieved the best or the second-best performance for each combination of TW and dataset type, except in only one case. In addition, CHI-SAE showed the second-best overall performance. The results

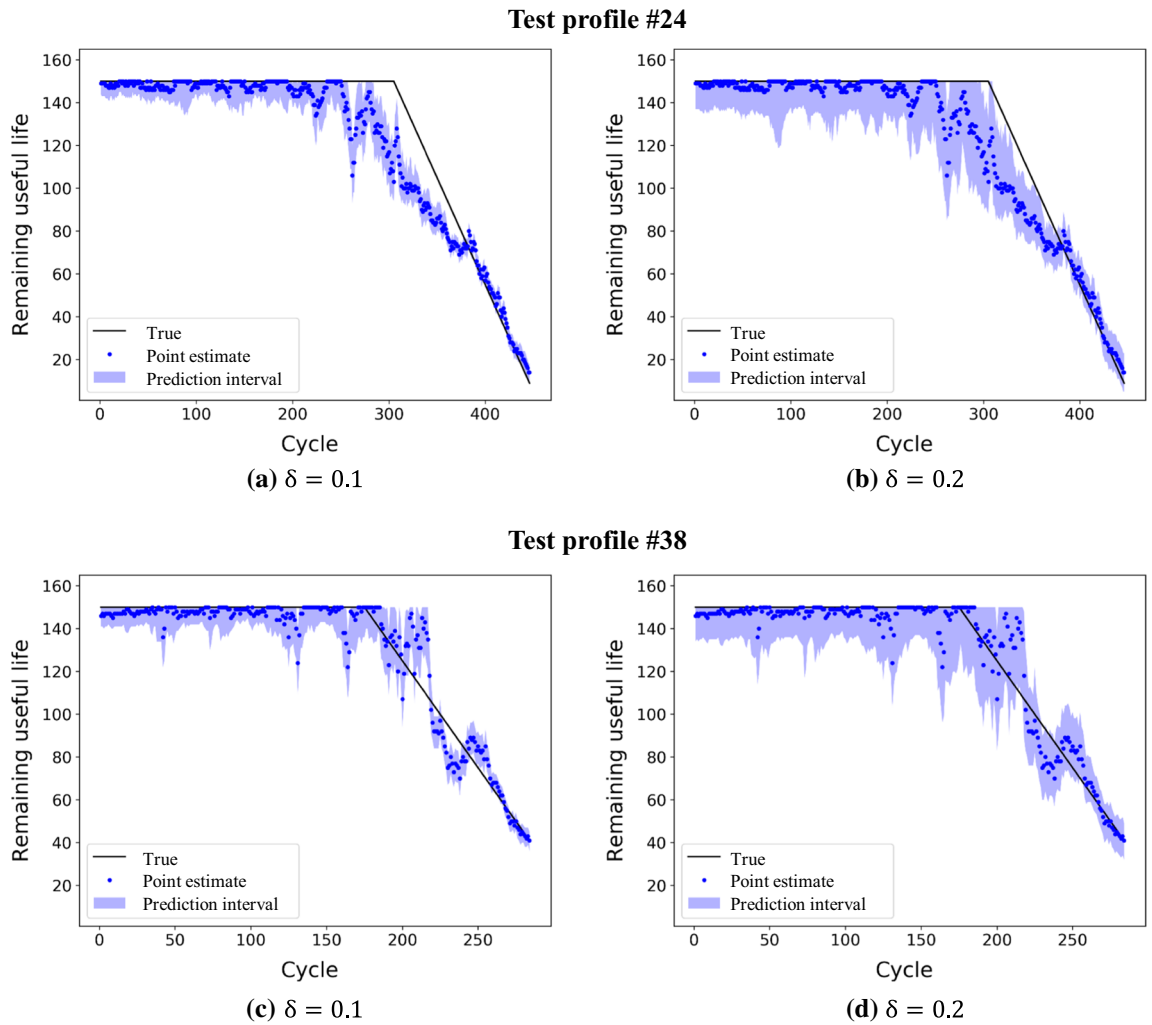

Fig. 4 Predicted RUL and its intervals for the 24th test profile of FD003 when **a** $\delta = 0.1$ and **b** $\delta = 0.2$ and for the 38th test profile of FD003 when **c** $\delta = 0.1$ and **d** $\delta = 0.2$

Table 3 Comparison with different baselines for FD001

TW	Metric	NN	NN-SAE	NN-SDAE	CHI-SAE	Proposed method
5	RMSE	24.4	23.4	**23.3**	23.4	**23.3**
	MAE	18.9	17.9	18.1	**15.9**	16.4
	MAPE	20.2	19.0	18.8	18.1	**17.8**
10	RMSE	23.7	**23.3**	**23.3**	23.7	**23.3**
	MAE	18.3	18.2	17.8	**16.1**	16.4
	MAPE	19.0	19.1	19.5	**18.5**	18.7
30	RMSE	20.8	21.4	19.5	21.9	**19.2**
	MAE	16.0	16.8	15.3	15.9	**14.0**
	MAPE	16.8	15.6	15.4	14.7	**14.2**

Table 4 Comparison with different baselines for FD003

TW	Metric	NN	NN-SAE	NN-SDAE	CHI-SAE	Proposed method
5	RMSE	21.1	21.2	**20.8**	21.5	21.1
	MAE	15.7	16.6	15.6	**12.4**	**12.4**
	MAPE	15.9	15.8	15.4	13.9	**13.8**
10	RMSE	20.5	20.6	**20.3**	21.0	20.9
	MAE	14.7	15.1	14.3	12.9	**12.5**
	MAPE	15.1	15.3	15.2	**13.9**	13.9
30	RMSE	17.9	17.4	16.9	16.8	**16.7**
	MAE	12.2	11.8	11.4	**11.0**	**11.0**
	MAPE	11.3	11.9	12.1	10.6	**10.4**

reveal that the CHI modeling component and the denoising component contribute to improvements in performance.

To analyze the contribution of each component of DCHIEN in more detail, we calculated the error reduction ratio by comparing the proposed method with CHI-SAE, NN-SDAE, and NN. Table 5 shows the average error reduction ratios of the MSE, MAE, and MAPE. The ablation analysis reveals that each individual component contributes to the improvement in RUL prediction performance, and the best performance is produced when the two components are combined. In particular, CHI modeling not only supports engineers by providing information on RUL prediction intervals and CHI change patterns but also improves the RUL prediction performance by 6.17% on average.

5.4 Comparison with state-of-the-art methods

In this section, the proposed method is compared with state-of-the-art RUL prediction methods that harness advanced machine learning techniques to learn degradation patterns or to directly predict the RUL. For a fair comparison, we followed a general evaluation protocol that predicts the RUL at the end point of each testing profile. Tables 6, 7, and 8 show the comparison results in terms of Score, RMSE, and Accuracy, respectively. Accuracy was reported for only two methods in the literature. The method with the best performance for each evaluation criterion is highlighted in bold.

The proposed method achieved the lowest Score and RMSE for two subdatasets, showing the best overall results. Specifically, in terms of Score, RULCLIPPER also achieved the best performance for two subdatasets. However, RULCLIPPER was not able to show good results with the other metrics. Similarly, CNN and CNN+LSTM, which obtained the lowest RMSE on one subdataset, performed poorly in terms of Score. In addition, the proposed method achieved the highest Accuracy for all subdatasets. The experimental results reveal that the proposed RUL prediction method can provide the best overall performance, although it may perform slightly worse in a few cases.

Finally, we evaluated the degree to which Accuracy could be increased by considering the RUL prediction interval. If we can effectively increase Accuracy with a small prediction interval, robustness in managing a system can be achieved. That is, fatal system failure can be more effectively prevented by considering all time points in the RUL prediction interval as possible failure points. Table 9 shows the change in Accuracy (%) and the length of the prediction interval according to δ. First, Accuracy significantly increases as the length of the RUL prediction interval increases. More interestingly, it is found that very small prediction intervals can even help prevent system failures. For example, when $\delta = 0.01$, the average Accuracy increased by 2.5% compared to the average Accuracy of the proposed method in Table 8 by considering 0.69 more points as possible failure points. These results show that uncertainty in RUL prediction can be managed well with the proposed method. In terms of Accuracy improvement by length, the table shows that up to a 1.98 Accuracy improvement can be obtained by considering an RUL interval of 1. On average, we can achieve a 1.35 Accuracy improvement only with an RUL interval of 1 if we apply $\delta = 0.01$.

6 Conclusion and discussion

In this paper, we propose a PHM method that can manage the uncertainty introduced by sensor noise, uncertainty in health status diagnoses, and uncertainty in the RUL predictions. The proposed PHM method is based on the CHI information extracted by the DCHIEN model. The experimental results show that the proposed method can properly monitor the degradation patterns for all intervals, from a healthy engine state to a system failure, by monitoring multiple CHIs. The proposed RUL prediction method shows the best performance under the largest number of test scenarios. In addition, it is shown that poor RUL predictions can be compensated with a small RUL prediction interval.

We show that the uncertainties in PHM can be effectively managed using DCHIEN. The proposed PHM method can

Table 5 Contributions of the components to error reduction

	TW	Denoising (%)	CHI (%)	Denoising +CHI (%)
FD001	5	−0.26	5.06	10.08
	10	−0.43	3.91	4.58
	30	9.28	5.84	11.80
FD003	5	1.00	10.01	11.36
	10	1.07	5.98	7.00
	30	0.55	6.20	8.05
Average		1.87	6.17	8.81

Table 6 Comparison with state-of-the-art methods in terms of Score

Dataset	FD001	FD002	FD003	FD004
CNN (Li et al. 2018)	360	3352	618	5861
LSTM (Zheng et al. 2017)	338	4450	852	5550
MODBNE (Zhang et al. 2017)	334	5585	422	6558
CNN+LSTM (Kong et al. 2019)	303	3440	1420	4630
BiLSTM-ED (Yu et al. 2019)	273	3099	574	3202
RULCLIPPER (Ramasso 2020)	**216**	2796	317	**3132**
Ours	324	**2528**	**294**	4860

Table 7 Comparison with state-of-the-art methods in terms of RMSE

Dataset	FD001	FD002	FD003	FD004
CNN (Li et al. 2018)	13.86	**18.91**	13.55	23.32
LSTM (Zheng et al. 2017)	16.14	24.49	16.18	28.17
MODBNE (Zhang et al. 2017)	15.04	25.05	12.51	28.66
CNN+LSTM (Kong et al. 2019)	16.13	20.46	17.12	**23.26**
BiLSTM-ED (Yu et al. 2019)	14.74	22.07	17.48	23.49
RULCLIPPER (Ramasso 2020)	13.27	22.89	16.00	24.33
Ours	**12.8**	20.92	**12.46**	23.67

address many practical issues related to the uncertainties in PHM at the operational level. However, this paper still has the limitation that the uncertainties are considered qualitatively without measuring quantitative uncertainty reduction, despite the definitions of uncertainties suggested in Sect. 2.1. This is because current definitions of uncertainties require the true value of the current health status, which is difficult to know in advance. If uncertainty can be precisely measured, engineers can use this information for more systematic PHM. Thus, this issue should be addressed.

Specifically, the following two topics are considered for follow-up work: First, the proposed model should be extended so that the RUL prediction intervals are based on a well-grounded probabilistic foundation, which will allow more systematic management of uncertainty in RUL predictions. Second, the model presented in this paper cannot quantify uncertainties. Thus, further studies will be conducted to define and quantify uncertainties with more precise

Table 8 Comparison with state-of-the-art methods in terms of Accuracy (%)

Dataset	FD001	FD002	FD003	FD004
BiLSTM-ED (Yu et al. 2019)	57	49	42	40
RULCLIPPER (Ramasso 2020)	67	46	59	45
Ours	**70**	**57**	**78**	**49**

Table 9 Accuracy improvement by the length of the prediction interval

δ		0.01	0.05	0.10
FD001	Length	1.48	7.33	13.93
	Accuracy	71	82	90
	Accuracy improvement by length	0.68	1.64	1.44
FD002	Length	2.15	10.72	20.55
	Accuracy	61	69	80
	Accuracy improvement by length	1.86	1.12	1.12
FD003	Length	1.11	5.72	11.36
	Accuracy	79	82	86
	Accuracy improvement by length	0.90	0.70	0.70
FD004	Length	2.02	10.20	20.38
	Accuracy	53	65	74
	Accuracy improvement by length	1.98	1.57	1.23
Average	Length	1.69	8.49	16.56
	Accuracy	66	75	82
	Accuracy improvement by length	1.35	1.26	1.12

mathematical expressions for more systematic uncertainty management.

Author contributions JJ is the only author of this paper. Therefore, all contributions are made by him.

Funding This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) under Grant NRF-2022R1F1A1069969 and in part by the Research Fund, 2022 of The Catholic University of Korea.

Data availability The datasets generated during and/or analyzed during the current study are available at https://www.kaggle.com/code/phamvanvung/cmapss.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest to this work.

Ethical approval This paper does not deal with any ethical problems.

Informed consent We declare that all authors have informed consent.

References

Benkedjouh T, Medjaher K, Zerhouni N, Rechak S (2013) Remaining useful life estimation based on nonlinear feature reduction and support vector regression. Eng Appl Artif Intell 26(7):1751–1760. https://doi.org/10.1016/j.engappai.2013.02.006

Bressel M, Hilairet M, Hissel D, Bouamama BO (2016) Fuel cell remaining useful life prediction and uncertainty quantification under an automotive profile. In: Proceedings of conference of the IEEE industrial electronics society, pp 5477–5482. https://doi.org/10.1109/IECON.2016.7793300

Daigle MJ, Bregon A, Roychoudhury I (2014) Distributed prognostics based on structural model decomposition. IEEE Trans Reliab 63(2):495–510. https://doi.org/10.1109/TR.2014.2313791

Deutsch J, He D (2018) Using deep learning-based approach to predict remaining useful life of rotating components. IEEE Trans Syst Man Cybern Syst 48(1):11–20. https://doi.org/10.1109/TSMC.2017.2697842

Diez-Olivan A, Pagan JA, Khoa NLD, Sanz R, Sierra B (2018) Kernel-based support vector machines for automated health status assessment in monitoring sensor data. Int J Adv Manuf Technol 95:327–340. https://doi.org/10.1007/s00170-017-1204-2

Eker OF, Camci F, Jennions IK (2012) Major challenges in prognostics: Study on benchmarking prognostics datasets. In: Proceedings of the European conference of the prognostics and health management society, pp 148–155

García Nieto PJ, García-Gonzalo E, Sánchez Lasheras F, de Cos Juez FJ (2015) Hybrid PSO-SVM-based method for forecasting of the remaining useful life for aircraft engines and evaluation of its reliability. Reliab Eng Syst Saf 138(11):219–231. https://doi.org/10.1016/j.ress.2015.02.001

Heimes FO (2008) Recurrent neural networks for remaining useful life estimation. In: Proceedings of the international conference on prognostics and health management, pp 1–6. https://doi.org/10.1109/PHM.2008.4711422

Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int J Uncertain Fuzziness Knowl Based Syst 6(2):107–116. https://doi.org/10.1142/S0218488598000094

Huynh KT, Castro IT, Barros A, Berenguer C (2014) On the use of mean residual life as a condition index for condition-based maintenance decision-making. IEEE Trans Syst Man Cybern Syst 44(7):877–893

Jang J, Kim CO (2021) Siamese network-based health representation learning and robust reference-based remaining useful life prediction. IEEE Trans Ind Inform. https://doi.org/10.1109/TII.2021.3126309

Jang J, Min BW, Kim CO (2019) Denoised residual trace analysis for monitoring semiconductor process faults. IEEE Trans Semicond Manuf 32(3):293–301. https://doi.org/10.1109/TSM.2019.2916374

Jang J, Seo M, Kim CO (2020) Support weighted ensemble model for open set recognition of wafer map defects. IEEE Trans Semicond Manuf 33(4):635–643. https://doi.org/10.1109/TSM.2020.3012183

Javed K, Gouriveau R, Zerhouni N (2015) A new multivariate approach for prognostics based on extreme learning machine and fuzzy clustering. IEEE Trans Cybern 45(12):2626–2639. https://doi.org/10.1109/TCYB.2014.2378056

Jiang J-R, Kuo C-K (2017) Enhancing convolutional neural network deep learning for remaining useful life estimation in smart factory applications. In: International conference on information, communication, engineering and technology, pp 120–123. https://doi.org/10.1109/ICICE.2017.8478928

Kong Z, Cui Y, Xia Z, Lv H (2019) Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. Appl Sci 9(19):4156. https://doi.org/10.3390/app9194156

Li Hong, Pan Donghui, Chen CLP (2015) Reliability modeling and life estimation using an expectation maximization based wiener degradation model for momentum wheels. IEEE Trans Cybern 45(5):969–977. https://doi.org/10.1109/TCYB.2014.2341113

Li X, Ding Q, Sun J-Q (2018) Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf 172:1–11. https://doi.org/10.1016/j.ress.2017.11.021

Liao Y, Zhang L, Liu C.(2018) Uncertainty prediction of remaining useful life using long short-term memory network based on bootstrap method. In: Proceedings of the international conference on prognostics and health management, pp 1–8. https://doi.org/10.1109/ICPHM.2018.8448804

Liao H, Zhao W, Guo H (2006) Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model. In: Proceedings of the annual reliability and maintainability symposium, pp 127–132. https://doi.org/10.1109/RAMS.2006.1677362

Lim P, Goh CK, Tan KC, Dutta P (2017) Multimodal degradation prognostics based on switching Kalman filter ensemble. IEEE Trans Neural Netw Learn Syst 28(1):136–148. https://doi.org/10.1109/TNNLS.2015.2504389

Listou Ellefsen A, Bjørlykhaug E, Æsøy V, Ushakov S, Zhang H (2019) Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliab Eng Syst Saf 183:240–251. https://doi.org/10.1016/j.ress.2018.11.027

Liu D, Zhou J, Liao H, Peng Y, Peng X (2015) A health indicator extraction and optimization framework for lithium-ion battery degradation modeling and prognostics. IEEE Trans Syst Man Cybern Syst 45(6):915–928. https://doi.org/10.1109/TSMC.2015.2389757

Liu Z, Cheng Y, Wang P, Yu Y, Long Y (2018) A method for remaining useful life prediction of crystal oscillators using the Bayesian approach and extreme learning machine under uncertainty. Neurocomputing 305:27–38. https://doi.org/10.1016/j.neucom.2018.04.043

Qian Y, Hu S, Yan R (2013) Bearing performance degradation evaluation using recurrence quantification analysis and auto-regression model. In: Proceedings of the IEEE international instrumentation and measurement technology conference, pp 1713–1716. https://doi.org/10.1109/I2MTC.2013.6555707

Ramasso E (2020) Investigating computational geometry for failure prognostics. Int J Prognostics Health Manag 5(1):2205. https://doi.org/10.36001/ijphm.2014.v5i1.2205

Ramasso E, Rombaut M, Zerhouni N (2013) Joint prediction of continuous and discrete states in time-series based on belief functions. IEEE Trans Cybern 43(1):37–50. https://doi.org/10.1109/TSMCB.2012.2198882

Rezvanizaniani SM, Liu Z, Chen Y, Lee J (2014) Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle (EV) safety and mobility. J Power Sources 256:110–124. https://doi.org/10.1016/j.jpowsour.2014.01.085

Sankararaman S (2015) Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. Mech Syst Signal Process 52–53(1):228–247. https://doi.org/10.1016/j.ymssp.2014.05.029

Sankararaman S, Daigle M, Saxena A, Goebel K (2013) Analytical algorithms to quantify the uncertainty in remaining useful life prediction. In: Proceedings of the IEEE aerospace conference, pp 1–11. https://doi.org/10.1109/AERO.2013.6496971

Saxena A, Celaya J, Balaban E, Goebel K, Saha B, Saha S, Schwabacher M (2008) Metrics for evaluating performance of prognostic techniques. In: Proceedings of the international conference on prognostics and health management, pp 1–17. https://doi.org/10.1109/PHM.2008.4711436

Saxena A, Goebel K, Simon D, Eklund N (2008) Damage propagation modeling for aircraft engine run-to-failure simulation. In: Proceedings of the international conference on prognostics and health management, pp 1–9. https://doi.org/10.1109/PHM.2008.4711414

Si X-S, Hu C-H, Zhang Q, Li T (2017) An integrated reliability estimation approach with stochastic filtering and degradation modeling for phased-mission systems. IEEE Trans Cybern 47(1):67–80. https://doi.org/10.1109/TCYB.2015.2507370

Snoek J, Lrochelle H, Adams R.P.(2012) Practical Bayesian optimization of machine learning algorithms. In: Proceedings of the advances in neural information processing systems, pp 2951–2959

Sun C, He Z, Cao H, Zhang Z, Chen X, Zuo MJ (2015) A non-probabilistic metric derived from condition information for operational reliability assessment of aero-engines. IEEE Trans Reliab 64(1):167–181. https://doi.org/10.1109/TR.2014.2336032

Tian Z (2012) An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. J Intell Manuf 23(2):227–237. https://doi.org/10.1007/s10845-009-0356-9

Tian Z, Wong L, Safaei N (2010) A neural network approach for remaining useful life prediction utilizing both failure and suspension histories. Mech Syst Signal Process 24(5):1542–1555. https://doi.org/10.1016/j.ymssp.2009.11.005

Tran VT, Thom Pham H, Yang B-S, Tien Nguyen T (2012) Machine performance degradation assessment and remaining useful life prediction using proportional hazard model and support vector machine. Mech Syst Signal Process 32:320–330. https://doi.org/10.1016/j.ymssp.2012.02.015

Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408

Vincent P, Larochelle H, Bengio Y, Manzagol P-A (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the international conference on machine learning, pp 1096–1103. https://doi.org/10.1145/1390156.1390294

Waag W, Fleischer C, Sauer DU (2014) Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. J Power Sources 258:321–339. https://doi.org/10.1016/j.jpowsour.2014.02.064

Wang P, Gao RX (2015) Adaptive resampling-based particle filtering for tool life prediction. J Manuf Syst 37:528–534. https://doi.org/10.1016/j.jmsy.2015.04.006

Wang C, Lu N, Cheng Y, Jiang B (2019) A data-driven aero-engine degradation prognostic strategy. IEEE Trans Cybern 51(3):1531–1541. https://doi.org/10.1109/TCYB.2019.2938244

Wijayasekara D, Linda O, Manic M, Rieger C (2014) FN-DFE: fuzzy-neural data fusion engine for enhanced resilient state-awareness of hybrid energy systems. IEEE Trans Cybern 44(11):2065–2075. https://doi.org/10.1109/TCYB.2014.2323891

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. https://doi.org/10.1016/j.neucom.2017.05.063

Yang L, Lee J (2012) Bayesian belief network-based approach for diagnostics and prognostics of semiconductor manufacturing systems. Robot Comput Integr Manuf 28(1):66–74. https://doi.org/10.1016/j.rcim.2011.06.007

Yang F, Habibullah MS, Zhang T, Xu Z, Lim P, Nadarajan S (2016) Health index-based prognostics for remaining useful life predictions in electrical machines. IEEE Trans Ind Electron 63(4):2633–2644. https://doi.org/10.1109/TIE.2016.2515054

Yu W, Kim IY, Mechefske C (2019) Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. Mech Syst Signal Process 129:764–780. https://doi.org/10.1016/j.ymssp.2019.05.005

Zhang C, Lim P, Qin AK, Tan KC (2017) Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. IEEE Trans Neural Netw Learn Syst 28(10):2306–2318. https://doi.org/10.1109/TNNLS.2016.2582798

Zhang J, Wang P, Yan R, Gao RX (2018) Long short-term memory for machine remaining life prediction. J Manuf Syst 48:78–86. https://doi.org/10.1016/j.jmsy.2018.05.011

Zhang C, Wang C, Lu N, Jiang B (2019) An RBMs-BN method to RUL prediction of traction converter of CRH2 trains. Eng Appl Artif Intell 85:46–56. https://doi.org/10.1016/j.engappai.2019.06.001

Zhang J, Jiang Y, Wu S, Li X, Luo H, Yin S (2022) Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. Reliab Eng Syst Saf 221:108297. https://doi.org/10.1016/j.ress.2021.108297

Zhao Z, Quan Q, Cai K-Y (2014) A profust reliability based approach to prognostics and health management. IEEE Trans Reliab 63(1):26–41. https://doi.org/10.1109/TR.2014.2299111

Zhao R, Yan R, Wang J, Mao K (2017) Learning to monitor machine health with convolutional bi-directional LSTM networks. Sensors 17(2):273. https://doi.org/10.3390/s17020273

Zheng S, Ristovski K, Farahat A, Gupta C (2017) Long short-term memory network for remaining useful life estimation. In: Proceedings of the international conference on prognostics health management, pp 88–95. https://doi.org/10.1109/ICPHM.2017.7998311

Zhu J, Chen N, Peng W (2019) Estimation of bearing remaining useful life based on multiscale convolutional neural network. IEEE Trans Ind Electron 66(4):3208–3216. https://doi.org/10.1109/TIE.2018.2844856