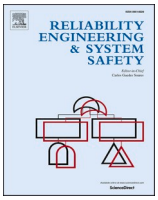
Reliability Engineering and System Safety 205 (2021) 107257

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress



A dual-LSTM framework combining change point detection and remaining useful life prediction

Zunya Shi, Abdallah Chehade *

Department of Industrial and Manufacturing Systems Engineering, University of Michigan-Dearborn, Dearborn, MI 48128, USA

ARTICLE INFO

Keywords:
Remaining useful life
Prognosis
Sensor fusion
Change point detection
Long short-term memory
Neural networks

ABSTRACT

Remaining Useful Life (RUL) prediction is a key task of Condition-based Maintenance (CBM). The massive data collected from multiple sensors enables monitoring the complex systems in near real-time. However, such multiple sensors data environments pose a challenging task of combining the sensor data to infer the quality and RUL of the system. To address this task, we propose a Dual-LSTM framework that leverages Long-Short Term Memory (LSTM) for degradation analysis and RUL prediction. The Dual-LSTM relaxes the strong assumption of the fixed change point and detects the uncertain change point unit by unit at first. Then, the Dual-LSTM predicts the health index beyond the change point which can be leveraged to calculate the RUL. The proposed Dual-LSTM (i) achieves real-time high-precision RUL prediction by connecting the change point detection and RUL prediction with the health index construction, (ii) introduces a novel one-dimension health index function, (iii) leverages historical information to achieve detection and prediction tasks by characterizing both long and short-term dependencies of sensor signals through LSTM network. The effectiveness of the proposed Dual-LSTM framework is validated and compared to state-of-art benchmark methods on two publicly available turbofan engine degradation datasets.

1. Introduction

In recent years, Condition-based Maintenance (CBM) has been growing rapidly [1-5], since it can avoid unnecessary and excessive costs generated by unexpected system failures. Remaining Useful Life (RUL) prediction is a key task in the field of CBM [6, 7]. A good RUL prediction provides accurate information about when the unit or system will come to the end of the useful life [8, 9]. With accurate RUL predictions, the decision-makers are able to raise early alarms of unexpected failures so as to improve production efficiency and reduce life-cycle costs [2, 10, 11].

Due to the rapid development of sensing technology, a huge variety of sensors are used to monitor the health condition of the units [5, 12-15]. With a large number of sensor signals on hand, researchers proposed many data-driven approaches for RUL prediction. The classical data-driven RUL approaches are almost statistics-based. For example, C. Ordóñez et al. [16] proposed a hybrid ARIMA-SVM model for RUL prediction. First, an ARIMA model is designed to predict the sensor signals in advance. Then, an SVM model is developed to predict the RUL with the previously predicted sensor signals. A. Chehade et al. [13] proposed a statistical hypothesis testing based sensor fusion framework. This framework can not only predict the RUL but also the remaining time to each degradation state of the unit. A. Chehade et al. [5] also developed a convex quadratic formulation to combine the weighted models of historical units, thus reconstructing the Bayesian updated degradation model of the operating units. Wang et al. [17] presented a Continuous Hidden Markov Model (CHMM) based approach for RUL prediction. A. Chehade and A. A. Hussein [18] proposed a multi-output convolved Gaussian Process Model to enhance the SOC and RUL prediction accuracy of the Lithium-ion batteries. Si et al. [19] established a Wiener-process-based degradation model with a recursive filter algorithm to predict the RUL. Overall, the statistical data-driven models are usually more accurate, quicker, and cheaper to deploy than the traditional physics-based models. What's more attractive is that they don't require a comprehensive physical understanding of the unit [20].

Compared with statistics-based models, the Recurrent Neural Network (RNN), a Deep Learning technique, shows the stronger capabilities of capturing the complex non-linear relationships between inputs and outputs, as well as the short-term dependencies in time series [21, 22]. Therefore, it became more and more prevalent in the application of

* Corresponding author.
E-mail address: achehade@umich.edu (A. Chehade).

https://doi.org/10.1016/j.ress.2020.107257
Received 4 March 2020; Received in revised form 12 September 2020; Accepted 30 September 2020
Available online 4 October 2020

RUL prediction [23, 24]. However, RNN usually underperforms when learning long-term dependencies of time series due to the vanishing and exploding gradient problems [25-28]. Therefore, the Long-Short Term Memory (LSTM) network, a variant version of RNN, is designed to prevent the vanishing and exploding gradient problems. Within the LSTM network, the inputs can be remembered through the cell states over time. And the memories in and out of the cell are regulated by the unique three-gate structure of the LSTM. By doing so, the vanishing and exploding problem can be solved since the truly relevant and important information can be kept for a long time [28–30].

There have been some efforts that leveraged LSTM networks for RUL prediction [31-33]. S. Zheng et al. [31] and S. Dong et al. [32] utilized the LSTM network to predict the RUL with sensor signals directly. And they tested the LSTM on different case-studies. In most cases, LSTM performed better than other benchmark models, like Multilayer Perceptron (MLP), RNN, and Gated Recurrent Units (GRU). J. Zhang et al. [33] used the LSTM network to predict the RUL in a different way. First, they derived a one-dimension Health Index (HI) with multiple sensor signals. Then, they continuously predicted one-cycle ahead HI by feeding the sequence of previous HIs into a Bi-directional LSTM. The RUL was finally determined as the length from the current time to the time when the predicted HI went beyond a predefined threshold. Wang et al. [34] proposed a hybrid model combining LSTM and Gradient Boosting Regression (GBR) for RUL prediction. LSTM was used to predict RUL with long sequences of sensor signals, and GBR was adopted to predict the RUL with short sequences. Finally, both RULs returned by LSTM and GBR were used to predict the final RUL with the Back-propagation Neural Network (BPNN).

Besides the standard LSTM network, some variants of the LSTM are further proposed for RUL prediction under more complex conditions and achieve higher precisions. Elsheikh et al. [35] established a new Bidirectional Handshaking LSTM (BHLSTM) to predict the RUL given short sequences of sensor signals with random initial wear. Miao et al. [21] designed a dual-task LSTM network to assess the degradation stages and predict the RUL of turbofan engines in a parallel way. Song et al. [36] proposed a hybrid prediction model integrating the autoencoder and Bidirectional LSTM (BLSTM) to improve the RUL prediction accuracy. Li et al. [20] developed a Directed Acyclic Graph (DAG) network which combines Convolutional Neural Network (CNN) and LSTM in a parallel manner for RUL prediction.

Although the LSTM has presented an outstanding ability in the application of RUL prediction, there still exists a drawback that needs to be overcome. In practice, it is common to see that the engine unit starts at the normal operation level and begins to degrade at an uncertain change point. It is unreasonable to predict the RUL before the change point because the degradation is negligible during this period [19, 31, 37]. Additionally, the sensor signals collected before the change point provide very limited information about the degradation process. Therefore, it is necessary to detect the change point and mainly focus on the degradation process beyond this change point. However, most existing LSTM networks developed for RUL prediction assume that there is no change point exists and RUL decreases linearly with time from the beginning to the end [34]. Other LSTMs simply assign a fixed change point for any unit regardless of the operational conditions [20, 22, 31, 33, 35, 36]. In this case, because of the lack of accuracy in change point detection, most of those approaches are not practically reliable for degradation modeling. There are only a few research efforts that apply change point detection prior to RUL prediction using LSTMs [38]. For example, Y. Wu et al. [38] used the support vector machine (SVM) to detect the change point but showed poor performance for complex systems. To overcome this major limitation in existing DL models, this paper proposes the Dual-LSTM Network for both change point detection and RUL prediction.

The main contributions of the paper are:

(i) Proposing a new Dual-LSTM framework that connects the change point detection and RUL prediction with a newly proposed HI construction function. Within the Dual-LSTM framework, the change point detection is a prerequisite which serves to filter out the unnecessary sensor signals irrelevant to the degradation. By detecting the uncertain change point in advance, the Dual-LSTM can achieve the high-precision real-time RUL prediction.
(ii) Introducing a novel one-dimension piecewise decreasing HI construction function. On the one hand, the HI constructed with the function can be used to capture the complex non-linear correlations across different sensors. On the other hand, the HI constructed accurately indicates the health condition of the unit.
(iii) Characterizing both long- and short-term dependencies within each sensor via the Dual-LSTM framework. Therefore, the historical information can be preserved as much as possible and fully utilized for change point detection and RUL prediction.

The remainder of this paper is organized as follows. Section 2 provides an introduction about related theoretical methods, which include RNN and LSTM neural networks. Section 3 outlines the concepts of the proposed Dual-LSTM framework, which combines change point detection and RUL prediction. Section 4 illustrates the experiment and presents the results. Conclusions and discussions about future research directions are given in Section 5.

2. Theoretical methods

2.1. RNN

RNN is a neural network with a feedback loop as shown in Fig. 1. In Fig. 1, $\boldsymbol{x}_t$ and $\boldsymbol{h}_t$ are the input and hidden state at time step t respectively. At the hidden layer, $\boldsymbol{h}_t$ is an activation function of a linear combination of $\boldsymbol{h}_{t-1}$ and $\boldsymbol{x}_t$, as shown in Eq. (1). The activation function represented as σ here is usually a sigmoid function. At the output layer, the output $\boldsymbol{y}_t$ is calculated as Eq. (2) shows. The $\boldsymbol{W}_h$, $\boldsymbol{V}_h$, and $\boldsymbol{b}_h$ in Eq. (1) as well as the $\boldsymbol{W}_y$, $\boldsymbol{b}_y$ in Eq. (2) are parameters of RNN that will be optimized during the training process.

$$\boldsymbol{h}_t = \sigma(\boldsymbol{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{V}_h \boldsymbol{x}_t + \boldsymbol{b}_h) \tag{1}$$

$$\boldsymbol{y}_t = \sigma(\boldsymbol{W}_y \boldsymbol{h}_t + \boldsymbol{b}_y) \tag{2}$$

The recursive function of Eq. (1) is exactly the feedback loop of the RNN in Fig. 1. It allows the RNN to store the memories in the hidden state and pass the hidden state step by step to the next. With the feedback loop, the output is not only dependent on the input at the current time but also relies on the memories from previous time steps.

However, RNN proves to be problematic for characterizing long-term dependencies in practice [25-28], since it usually experiences vanishing and exploding gradient problems, as the MLP does. Specifically, during

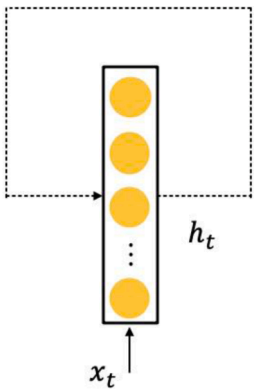


Fig. 1. RNN cell structure.

the backpropagation step in the training phase, the derivative of $\boldsymbol{h}_t$ is a product of all derivatives of hidden states at time steps after t based on the chain rule. If those derivatives are pretty small or large, then the derivative of $\boldsymbol{h}_t$ is highly possible to vanish or explode, and so are the derivatives $\boldsymbol{W}_h$ of and $\boldsymbol{V}_h$. This problem will stop the RNN from further training in a poorly trained network.

2.2. LSTM

To overcome the gradient vanishing and exploding limitation, researchers proposed the LSTM network. LSTM prevents gradients from vanishing or exploding by keeping both long- and short-term memories in the cell state. Besides, three special gates are designed to control the information flow in the LSTM network. They are forgot gate, input gate, and output gate. The cell structure of the LSTM is shown in Fig. 2 and the three gates are shaded in yellow.

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \boldsymbol{h}_{t-1} + \boldsymbol{V}_f \boldsymbol{x}_t + \boldsymbol{b}_f\right) \tag{3}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \boldsymbol{h}_{t-1} + \boldsymbol{V}_i \boldsymbol{x}_t + \boldsymbol{b}_i) \tag{4}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \boldsymbol{h}_{t-1} + \boldsymbol{V}_o \boldsymbol{x}_t + \boldsymbol{b}_o) \tag{5}$$

$$\widetilde{\boldsymbol{c}}_t = tanh(\boldsymbol{W}_a \boldsymbol{h}_{t-1} + \boldsymbol{V}_a \boldsymbol{x}_t + \boldsymbol{b}_a) \tag{6}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \widetilde{\boldsymbol{c}}_t \tag{7}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ tanh(\boldsymbol{c}_t) \tag{8}$$

$$\boldsymbol{y}_t = \sigma\left(\boldsymbol{W}_y \boldsymbol{h}_t + \boldsymbol{b}_y\right) \tag{9}$$

Eq. (3)-9 explains how LSTM works mathematically. Here, $\boldsymbol{f}_t$, $\boldsymbol{i}_t$, and $\boldsymbol{o}_t$ are the forget, input, and output gate's activation vectors respectively. Similarly, σ in Eq. (3)-5 and 9 is the activation function (sigmoid function usually) and *tanh* in Eq. (6) represents the hyperbolic tangent function. Both σ and *tanh* introduce non-linearity into the LSTM network. The $\boldsymbol{W}_{f,\ i,\ o,\ a,\ y}$, $\boldsymbol{V}_{f,\ i,\ o,\ a,\ y}$, and $\boldsymbol{b}_{f,\ i,\ o,\ a,\ y}$ in Eq. (3)-9 are the model parameters of LSTM that need to be updated during the training process. Each value of the vector ranges from 0 to 1 to control the information flow within the network. The cell state at current time t is noted as $\boldsymbol{c}_t$, which is used to store the memories from prior inputs. From Eq. (7), we can see that $\boldsymbol{c}_t$ is updated based on the previous cell state $\boldsymbol{c}_{t-1}$ and a candidate cell state $\widetilde{\boldsymbol{c}}_t$. In RNN, all values in the hidden state will be updated at each time step, as shown in Eq. (1). However, the situation is different in LSTM. Not all values of $\boldsymbol{c}_t$ will be updated, which is determined by the forget gate and input gate. More specifically, in Eq. (7), $\boldsymbol{f}_t$ decides which value of $\boldsymbol{c}_{t-1}$ should be added into $\boldsymbol{c}_t$, and $\boldsymbol{i}_t$ decides which value of $\widetilde{\boldsymbol{c}}_t$ should be added into $\boldsymbol{c}_t$. Next, $\boldsymbol{o}_t$ decides which value of $\boldsymbol{c}_t$ should be outputted through the hidden state $\boldsymbol{h}_t$. Finally, the output $\boldsymbol{y}_t$ is calculated based on the hidden state $\boldsymbol{h}_t$, as shown in Eq. 9.

With the cell state and three special gates, the memories can be carried for a long period without changes, which avoids the gradient from vanishing or exploding.

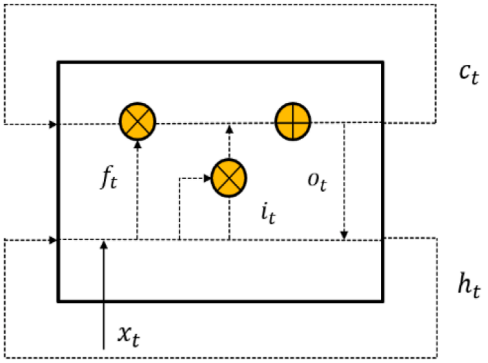


Fig. 2. LSTM cell structure.

3. Proposed dual-LSTM framework

To achieve a high-precision RUL prediction, a Dual-LSTM framework combining change point detection and RUL prediction is proposed. With the Dual-LSTM framework, we can characterize the unit-to-unit heterogeneity of the change points to mainly focus on the degradation process beyond the uncertain change point. Fig. 3 explains how the proposed Dual-LSTM framework works.

As Fig. 3 shows, the Dual-LSTM framework consists of three stages, data preparation stage, training stage, and testing stage. The data preparation aims to make the sensor signals more appropriate for training and testing. In the training stage, the LSTM1 designed for change point detection is learned on the training dataset at first. Then, the HI values of the training units at different time points are constructed based on the change points detected by LSTM1. Then, the constructed HI becomes the target of the LSTM2 which is designed for HI prediction. Finally, the two LSTM networks learned in the training stage are used in the testing stage to predict the RUL of the testing units in real-time.

The overall structure of the Dual-LSTM framework is shown in Fig. 4, which clearly illustrates the relationships between the LSTM1 and LSTM2. As Fig. 4 shows, the Dual-LSTM framework is an innovative combination of the change point detection and RUL prediction connected by the HI constructed which is an accurate indication of the unit's health condition. The value of the HI is that it allows the Dual-LSTM framework to capture the complex non-linear relationships between the sensor signals and the real health condition of the unit, thereby inferring the RUL, which changes the traditional method of directly predicting the RUL according to the sensor signals with DL models. By characterizing the relationship between the sensor signals and the health condition of the unit, the Dual-LSTM framework can achieve more robust and accurate RUL prediction results.

3.1. Change point detection

As the flowchart in Fig. 3 shows, once the data is fully prepared, the first step is to detect the uncertain change point for each unit based on the multi-sensor data with LSTM1. The network structure of LSTM1 is shown in Fig. 5.

In Fig. 5, the input $\boldsymbol{x}_t = [x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(n)}]$ represents the multi-sensor data at time t, where n is the number of sensors, and m is the number of time steps looked back.$\boldsymbol{X}_t = [\boldsymbol{x}_{t-m}, \boldsymbol{x}_{t-m+1}, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t]$ represents the multi-dimensional input sequence from time $t-m$ to time t. y_t indicates if the unit is in the normal operation phase or the degradation process phase, which is labeled as 0 before the change point (normal operation) or 1 (degradation process) otherwise. LSTM1 predicts $\hat{y}_t$ as a non-linear autoregressive function of the multi-sensor data $\boldsymbol{X}_t$ as shown in Eq. (10) to capture the complex relationship between the multi-sensor data and the change point. Here, $\boldsymbol{\theta}^1$ is the set of parameters of LSTM1, which includes all $\boldsymbol{W}$, $\boldsymbol{V}$, and $\boldsymbol{b}$ in Eq. (3) - 9. The LSTM1 designed for change point detection is a two-class classifier with two categories and the optimal parameters $\boldsymbol{\theta}^1$ are determined to minimize the binary cross-entropy in Eq. (11). T here is the number of time points in the training dataset. Finally, the change point, denoted as T_{cp}, is defined as the first time when the output $\hat{y}_t$ goes beyond the threshold, as Eq. (12) shows.

$$\hat{y}_t = F\left(\boldsymbol{X}_t, \boldsymbol{\theta}^1\right) \tag{10}$$

$$\mathcal{L}1 = -\frac{1}{T}\sum_{t}^{T} y_t \log\left(\hat{y}_t\right) + (1-y_t)\log\left(1-\hat{y}_t\right) \tag{11}$$

$$T_{cp}(t) = \inf\left(t^* \le t \middle| \hat{y}_{t^*} = F\left(\boldsymbol{X}_{t^*}, \boldsymbol{\theta}^1\right) \right\rangle threshold\right) \tag{12}$$

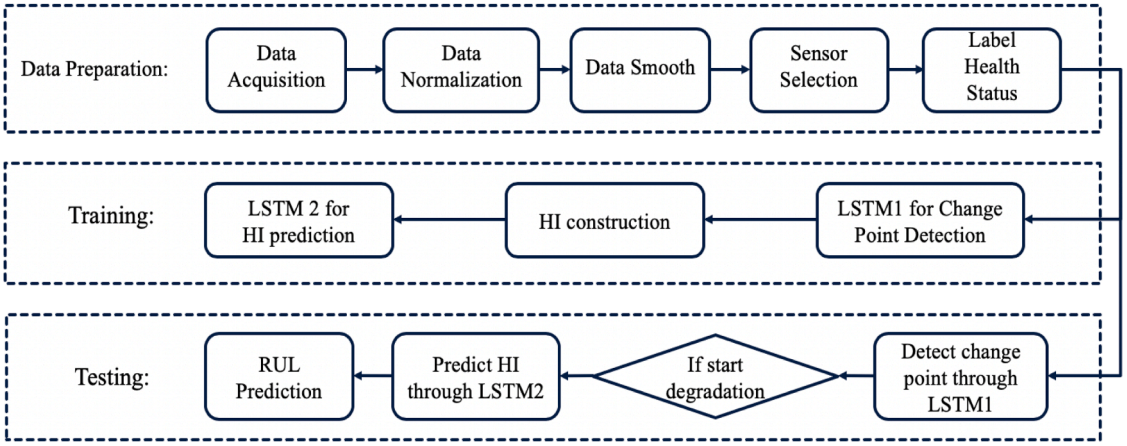


Fig. 3. Flowchart of the proposed Dual-LSTM framework.

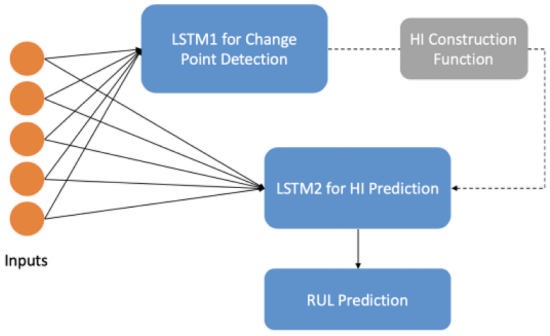


Fig. 4. Illustration of the Dual-LSTM framework.

In the experiment of this paper, the threshold is set to be 0.5, because the dataset we have is almost balanced. In other words, the number of negative (before the change point) points is similar to the number of positive (after the change point) points. If the dataset is severely imbalanced, the threshold value should be determined by precision or other evaluation metrics for binary classification problems (like recall, F1 score, ROC, etc.).

3.2. Health index prediction

Most existing LSTM-based networks designed for RUL prediction are one-step models, which feed multi-sensor data into neural networks to predict the RUL directly, without any intermediates. Usually, the true RUL is labeled according to the piecewise linear RUL target function shown in Fig. 6. The red dash line suggests the position of the change point. As Fig. 6 shows, the RUL value before the change point is defined as a constant number, implying that the unit has not started to degrade, and the degradation is negligible. After the change point, the unit starts to degrade, and the RUL linearly decreases until it reaches zero.

However, the position of the change point (red dash line) is assumed to be fixed for different units in most existing models. First, it is unrealistic in most cases, which has been argued in the first section of this paper. As Fig. 7 shows, different units may have different change points and different length of life span. Second, the same health condition level may have different RUL values under this RUL target function. In this case, a unified HI construction function should be introduced to accurately indicate the health condition of the unit.

Fig. 8 shows the HI curves of two units in Fig. 7 based on the unified HI construction function created in this paper, as Eq. (13) shows. In Eq. (13), T_f is the total life span of the unit and $T_{cp}(t)$ is the detected change point from the previous step. According to Fig. 8 and Eq. (13), the unified HI is a piecewise decreasing function. Before the change point, the HI always equals to 1, representing the unit is under the normal

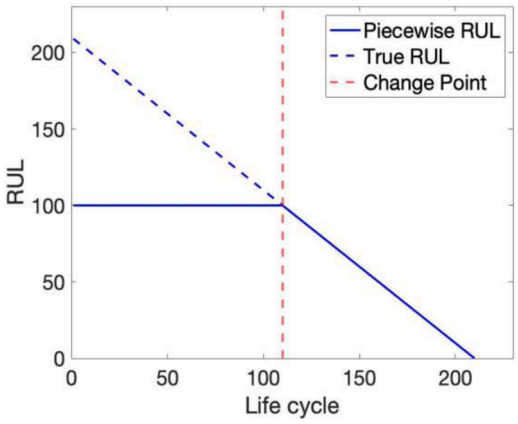


Fig. 6. Piecewise linear RUL target function.

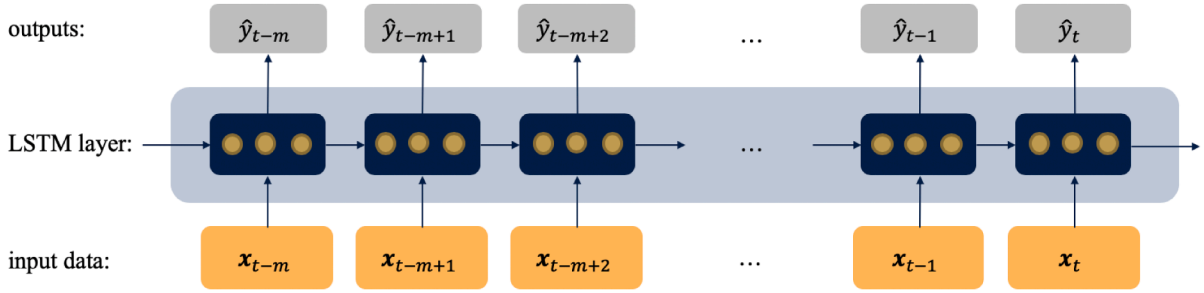


Fig. 5. Structure of LSTM1 for change point detection.

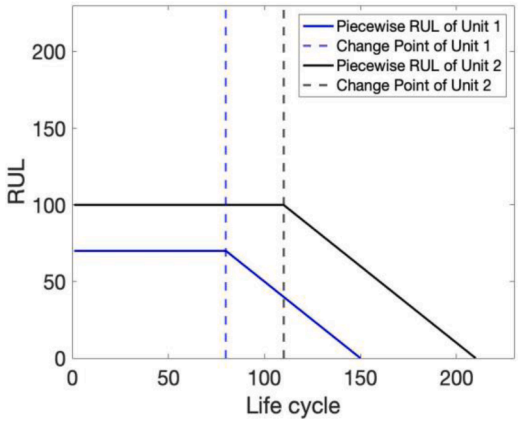


Fig. 7. Piecewise decreasing RUL function on two units with different change points and life spans.

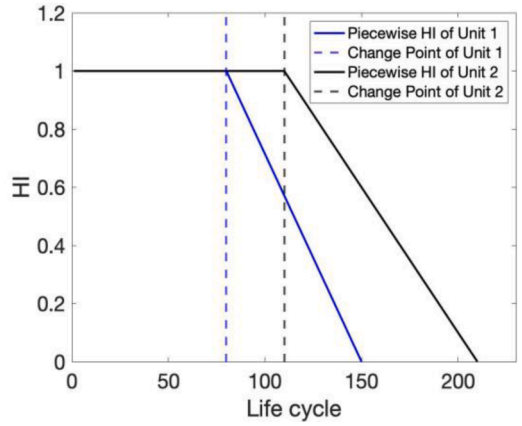


Fig. 8. Piecewise decreasing HI function on two units with different change points and life spans.

operation. Once the unit starts to degrade (after the change point), the HI will linearly decrease from 1 to 0 until the unit comes to the failure. With this unified HI function, we can make sure that the same health condition shares the same HI value.

$$HI(t) = 1 - \frac{\max(t - T_{cp}, 0)}{T_f - T_{cp}} \tag{13}$$

To predict the $HI(t)$ based on sensor signals collected, we design an LSTM2 to characterize the non-linear relationship between inputs $\boldsymbol{X}_t$ and $HI(t)$, expressed by Eq. (14). Here, $\boldsymbol{\theta}^2$ is the set of parameters for LSTM2 and it is optimized by minimizing the customized loss function in Eq. (15). The loss function is the squared difference between the predicted health index $\widehat{HI}(t)$ and true health index $HI(t)$ we labeled based on Eq. 13.

$$\widehat{HI}(t) = H(\boldsymbol{X}_t, \boldsymbol{\theta}^2) \tag{14}$$

$$\mathcal{L}2 = \sqrt{\frac{\sum_t^T \left(\widehat{HI}(t) - HI(t)\right)^2}{T}} \tag{15}$$

The overall structure of LSTM2 for HI prediction is shown in Fig. 9.

3.3. RUL prediction

The last step is to predict the RUL based on $\widehat{HI}(t)$. The RUL is defined as the length from the current time to the failure time, thus it should be calculated as the life span T_f minus the current time t. According to the unified HI function in Eq. (13), the HI is a one-to-one function of T_f. Therefore, T_f can be predicted given the predicted $\widehat{HI}(t)$, as Eq. (16) shows. $\widehat{T_f}(t)$ here is the predicted T_f.

$$\widehat{T}_f(t) = \frac{t - T_{cp}(t)}{1 - \widehat{HI}(t)} + T_{cp}(t) \tag{16}$$

Finally, the predicted $\widehat{RUL}(t)$ can be obtained by subtracting $\widehat{T_f}(t)$ from t.

$$\widehat{RUL}(t) = \widehat{T}_f(t) - t = \frac{\widehat{HI}(t)\left(t - T_{cp}(t)\right)}{1 - \widehat{HI}(t)} \tag{17}$$

The detailed algorithm of the Dual-LSTM framework for RUL prediction from end to end is summarized in Table 1. Adam algorithm is used to train the LSTM network.

4. Experiments and results

This section demonstrates the RUL prediction performance of the proposed Dual-LSTM framework, compared with benchmark DL models. Performance evaluation is carried out on the aircraft turbofan engine degradation dataset. To assess the consistency of the proposed Dual-LSTM framework, we developed 10 Dual-LSTM models with a random selection of the training/testing units and random initial weights of LSTM networks. In the training stage, the LSTM1 and LSTM2 networks of the Dual-LSTM framework are learned on the training data set. Then, the trained Dual-LSTM framework is applied to the testing data set to evaluate the performances. Three metrics, which are the scoring function (SF), RMSE, and relative prediction error, are used for evaluation.

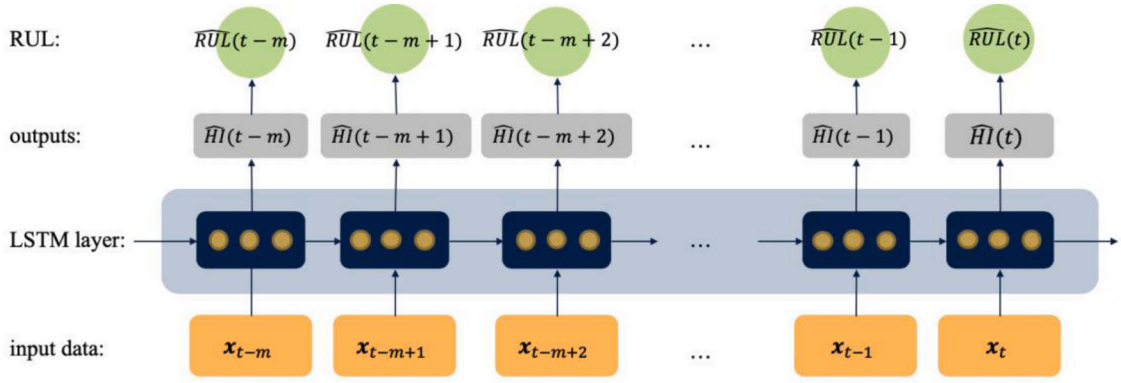


Fig. 9. Structure of LSTM2 for HI prediction.

Table 1
Dual-LSTM framework for RUL prediction.

Training Stage:
Initialization:
$iter = 0$, $\boldsymbol{m}_{iter} = 0$, $\boldsymbol{v}_{iter} = 0$
Assign values of α, β_1, β_2
Randomly initialize $\boldsymbol{\theta}^1_{iter}$, $\boldsymbol{\theta}^2_{iter}$
While $\boldsymbol{\theta}^{1/2}_{iter}$ **not converged:**
if change point detection:
gradient $\boldsymbol{g}^1_{iter} = \frac{\partial \mathcal{L}1}{\partial \boldsymbol{\theta}^1_{iter}}$
else if HI prediction:
gradient $\boldsymbol{g}^2_{iter} = \frac{\partial \mathcal{L}2}{\partial \boldsymbol{\theta}^2_{iter}}$
$\boldsymbol{m}_{iter+1} = \beta_1 \boldsymbol{m}_{iter} + (1-\beta_1)\, \boldsymbol{g}^{1/2}_{iter}$
$\boldsymbol{v}_{iter+1} = \beta_1 \boldsymbol{v}_{iter} + (1-\beta_1)\, \boldsymbol{g}^{1/2}_{iter}$
$\widehat{\boldsymbol{m}}_{iter+1} = \frac{\boldsymbol{m}_{iter+1}}{1-\beta_1^{iter+1}}$
$\widehat{\boldsymbol{v}}_{iter+1} = \frac{\boldsymbol{v}_{iter+1}}{1-\beta_2^{iter+1}}$
$\boldsymbol{\theta}^{1/2}_{iter+1} = \boldsymbol{\theta}^{1/2}_{iter} - \frac{\alpha \widehat{\boldsymbol{m}}_{iter+1}}{\sqrt{\widehat{\boldsymbol{v}}_{iter+1}} + \varepsilon}$
$iter = iter + 1$
Return $\boldsymbol{\theta}^{1/2}_{iter}$
Testing Stage:
$\widehat{y}_t = F(\boldsymbol{X}_t, \boldsymbol{\theta}^1_{iter})$
if $\widehat{y}_t < 0.5$:
The unit has not started degradation.
else:
$T_{cp}(t) = \inf(t^* \le t | \widehat{y}_{t^*} = F(\boldsymbol{X}_{t^*}, \boldsymbol{\theta}^1)0.5)$
$\widehat{HI}(t) = H(\boldsymbol{X}_t, \boldsymbol{\theta}^2_{iter})$
Return: $\widehat{RUL}(t) = \frac{\widehat{HI}(t)(t - T_{cp}(t))}{1 - \widehat{HI}(t)}$

4.1. Dataset overview

The turbofan engine degradation dataset is simulated by the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) tool developed by NASA, which is widely used in the field of PHM [39,40]. In the simulation, the engine model runs under a variety of operating conditions with different altitude levels, Mach numbers, and throttle resolver angles (TRA), as well as two fault modes, high-pressure compressor (HPC) degradation, and fan degradation. In our experiment, both settings of FD001 and FD003 with the single operation condition are investigated. FD001 has one failure mode and FD003 has two failure modes. In the dataset, 21 sensor signals are collected to monitor the health conditions of the turbofan engines, which randomly start from a normal operation level to the failure. Table 2 gives a detailed description of the 21 sensors.

Both FD001 and FD003 contain a training set and a testing set, and each set consists of 100 engine units. In the training set, the engines run until failure, whereas the engines in the testing set stop at some random time point before failure. For performance evaluation purposes, only the training set data is used. Among the training set data, 90 engines are randomly chosen for the training purpose, whereas the remaining 10 engines are used for testing.

4.2. Data preprocessing

Fig. 10 shows the normalized measurements from three sensors for a random engine. It is worth noting that different sensor signals show different trends. For example, sensor 2 and sensor 7 reveal increasing and decreasing trends respectively, while sensor 1 doesn't show significant variations on values. The trend information of all 21 sensors is summarized in Table 3. According to previous publications [34, 38], only the sensor signals with monotonic trends provide valuable information about the degradation and so should be kept for RUL prediction. Therefore, we follow the existing publications [34, 38] and finally keep

Table 2
Detailed description of sensors.

#	symbol	description	units
1	T2	Total temperature at fan inlet	°R
2	T24	Total temperature at LPC inlet	°R
3	T30	Total temperature at HPC inlet	°R
4	T50	Total temperature at LPT inlet	°R
5	P2	Pressure at fan inlet	psia
6	P15	Total pressure in bypass-duct	psia
7	P30	Total pressure at HPC outlet	psia
8	Nf	Physical fan speed	rpm
9	Nc	Physical core speed	rpm
10	Epr	Engine pressure ratio	–
11	Ps30	Static pressure at HPC outlet	psia
12	Phi	Ratio of fuel flow to Ps30	pps
13	NRf	Corrected fan speed	rpm
14	NRc	Corrected core speed	rpm
15	BPR	Bypass ratio	–
16	farB	Burner fuel-air ratio	–
17	htBleed	Bleed enthalpy	–
18	NF_dmd	Demanded fan speed	rpm
19	PCNR_dmd	Demanded corrected fan speed	rpm
20	W31	HPT coolant bleed	lbm/s
21	W32	LPT coolant bleed	lbm/s

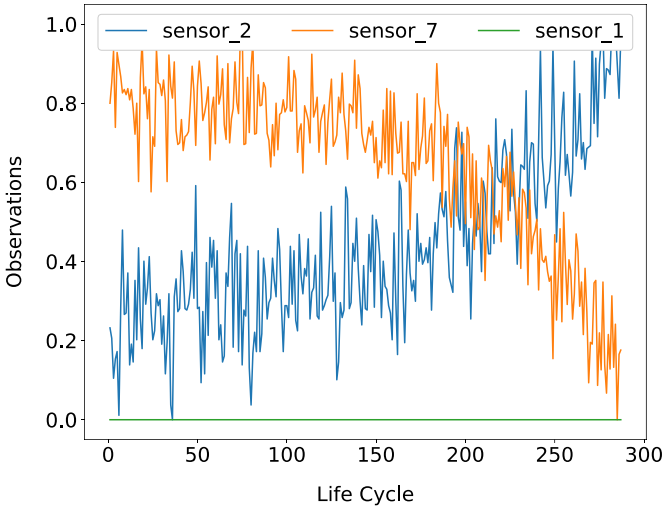


Fig. 10. Normalized sensor measurements for a random engine.

Table 3
Sensor trends summary.

Trend type	Sensor number
Increasing	2, 3, 4, 8, 9, 11, 13, 14, 15, 17
Decreasing	7, 12, 20, 21
Irregular/ unchanged	1, 5, 6, 10, 14, 16, 18, 19

14 sensors with monotonic increasing/decreasing trends, as Table 3 shows. The Min-Max rescaling method is used to normalize the data to be in the range of 0 to 1. Besides, to eliminate the effect of noise, the Savitzky-Golay filter with frame length 51 and polynomial order 3 is used to smooth the sensor signals.

As discussed in the proposed framework, LSTM1 is trained to predict the change point of degradation, which requires pre-labeled true change points for training. However, such labeled true change points are not available for this case-study. Therefore, to label the true change points of the training engines, we utilize the sensor fusion framework proposed by [13]. This sensor fusion framework is shown to provide better prior insights on the degradation process than each sensor. The framework proposes a weighted average of the sensor signals (composite index) that

maximizes the separability between the degradation states. The true change point is labeled as the first point for which the composite index passes a predefined threshold. To determine the threshold value, we run a ten-repeat five-fold cross-validation study on the LSTM1 for change point detection with different threshold values, which are 0.1, 0.15, 0.2, 0.25, and 0.3. Finally, threshold 0.2 shows the highest classification accuracy, thus being defined as the threshold for change point labeling.

4.3. Training stage

There are two steps in the training stage. The first step is to learn the parameters of the LSTM1 designed for change point detection. The 14 sensor signals selected as inputs are fed into the LSTM1 network. The parameters of LSTM1 are learned to minimize the cross-entropy loss function in Eq. (11). Finally, the determined LSTM1 is a three-layer neural network: one input layer (multi-sensor data), one LSTM layer with 32 hidden neurons and 20 look-back time-steps, and one output layer (0 or 1). For training, Adam optimizer is used with a learning rate of 0.001 and a batch size of 100 for 100 epochs.

The second step is to train the LSTM2 for modeling the HI as a function of the 14 sensor signals and the HI is given as the output. The determined LSTM2 consists of four layers: one input layer (multi-sensor data), two hidden LSTM layers with 32 neurons and 20 look-back time-steps, and one output layer (HI). For training, Adam optimizer is used with a learning rate of 0.0001 and a batch size of 100 for 80 epochs.

4.4. Performance evaluation

Three metrics are used to evaluate the RUL prediction performance of the proposed Dual-LSTM framework. They are scoring function (SF), RMSE, and relative prediction error. The formulas of these three metrics are given in Eq. (18) - 20. N is the number of testing units.

SF is defined by the data creator [41] and commonly used to evaluate the performance of RUL prediction in many existing papers [20, 22, 34]–[36], [38]. It is an asymmetric function, which prefers early prediction (predicted RUL is less than true RUL) rather than a late prediction error (predicted RUL is larger than true RUL).

$$SF_t = \begin{cases} \dfrac{\sum_i^N e^{\frac{d_{it}}{13}} - 1}{N}, \ where\ d_{it} = \widehat{RUL}_{it} - RUL_{it} < 0 \\ \dfrac{\sum_i^N e^{\frac{d_{it}}{10}} - 1}{N}, \ otherwise \end{cases} \tag{18}$$

RMSE is also a common evaluation metric for RUL prediction performance. However, different from SF, RMSE gives equal penalties to early and late prediction errors.

$$RMSE_t = \sqrt{\frac{1}{N}\sum_i^N {d_{it}}^2} \tag{19}$$

The relative prediction error is defined as the absolute difference between true and predicted RUL values divided by the total lifespan.

$$err_t = \frac{1}{N}\sum_i^N \frac{|d_{it}|}{T_{f_i}} \tag{20}$$

According to the definitions of three metrics shown above, the best model for RUL prediction is the one that results in the smallest SF, RMSE, and relative prediction error on the testing units.

4.5. Performance evaluation

Figs. 11 and 12 visualize the experiment results on testing units 97 from FD001 and unit 33 from FD003 respectively. The green dash line suggests the true change point and the blue dash line represents the predicted change point. The green shadow reflects a time buffer of +/- 10 cycles, which allows other acceptable change points. Because it is possible to mislabel the true change point by the sensor fusion framework. Figs. 11 and 12 clearly show that both predicted change points are inside the green shadow, which means the trained LSTM1 successfully detects the change point.

Besides, the blue solid line is the HI constructed based on the change point detected. The red dash line is the predicted HI (the predicted HI before the change point is automatically assigned as 1). From the figures, we can see that the predicted HI values are pretty close to the constructed HI beyond the change point, which proves that the trained LSTM2 performs well in HI prediction. Finally, the predicted HI is utilized to calculate the RUL using the function in Eq. 17.

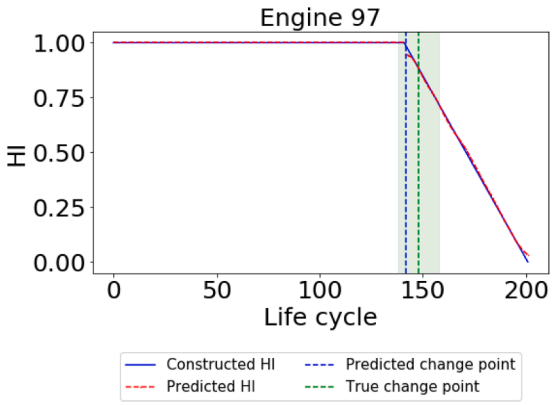


Fig. 11. Prediction results on Engine 97 from FD001.

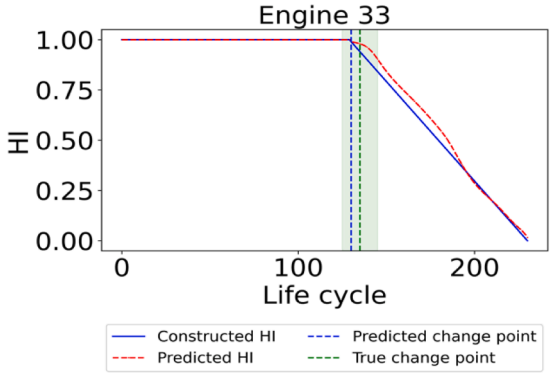


Fig. 12. Prediction results on Engine 33 from FD003.

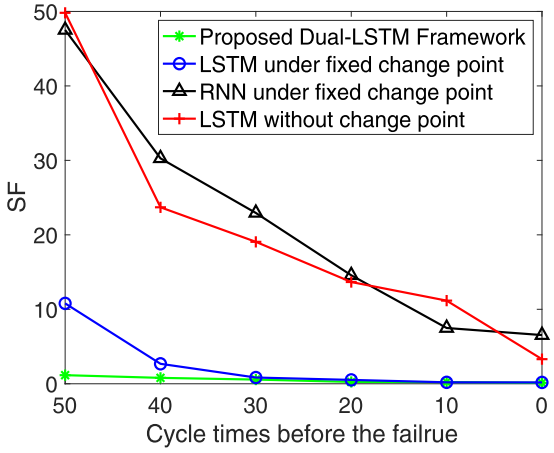


Fig. 14. RMSE comparison results on FD001.

To further explain the effectiveness of the proposed Dual-LSTM framework for RUL prediction, we compare it with three benchmark models. The first is a one-layer LSTM network (32 hidden neurons, 20 look-back time-steps) under the strong assumption that there is no change point. The second is a one-layer RNN (32 neurons, 20 look-back time-steps) under the situation that there is a fixed change point (we follow the reference [29] and set the fixed change point at the 130th cycle). The last one is the same LSTM network as the first benchmark model where the fixed change point is set at the 130th cycle. All models are trained by Adam optimizer with a batch size of 100 for 100 epochs.

Figs. 13-15 present the comparison results on FD001 dataset based on three evaluation metrics mentioned above at different monitoring points with different remaining cycles to failure. The comparison results on FD003 dataset are summarized in Table 4. The results both in Figs. 13-15 and Table 4 are the average of the 10 Dual-LSTM networks. The comparison results show that:

i) The proposed Dual-LSTM framework achieved the best RUL prediction performances (lowest SF, RMSE, and relative prediction errors) at all monitoring points. It means that the proposed Dual-LSTM framework can effectively improve the RUL prediction performance. The accurate RUL predictions of the Dual-LSTM at the earliest monitoring time (50 cycle times before the failure) provide confidence in raising early failure alarm and scheduling maintenance.
ii) Particularly, the SF from the Dual-LSTM framework is an order magnitude smaller than that of the benchmark models on FD001. Unlike the RMSE relative prediction errors, the Dual-LSTM improvement in SF is much more evident. This may be caused by the change point detection within the framework, which helps to correct the RUL labeling errors (especially the late errors) induced by the fixed change point assumption.
iii) Both the proposed Dual-LSTM and the benchmark networks show a descending trend for three evaluation metrics. Therefore, more accurate results are expected as the unit approaches failure. This is a natural consequence because more data about the unit is collected as it approaches failure and that supports more accurate predictions.

In summary, the first comparison study shows that the Dual-LSTM outperformed the standard RNN and LSTM neural networks at all monitoring times using three common evaluation metrics.

Fig. 16 shows the boxplot of relative prediction errors on different monitoring points over 10 experiments on FD001. For each Dual-LSTM model, the training/testing units and the initial parameters were randomly selected. The boxplot shows small variations in the relative prediction errors between the 10 Dual-LSTM models, which validates the consistency of the proposed Dual-LSTM framework for RUL predictions. Furthermore, the boxplot shows that relative predictions errors

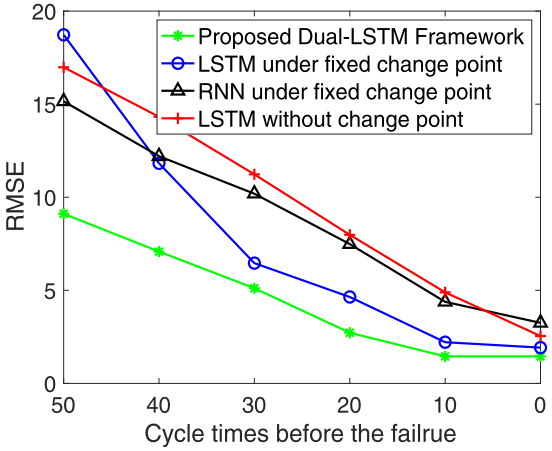


Fig. 13. SF comparison results on FD001.

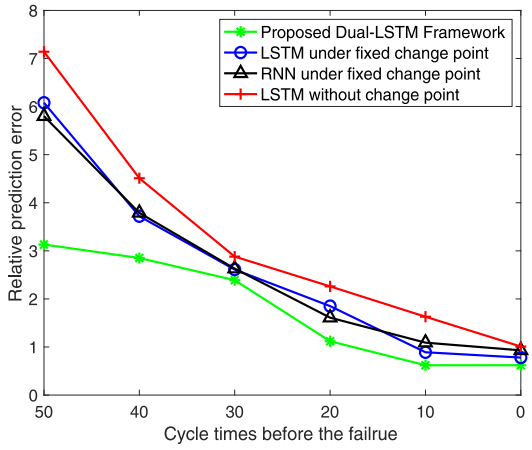


Fig. 15. Relative prediction error comparison results on FD001.

Table 4
Comparison results on FD003.

Cycles before failure		0	10	20	30	40	50
SF	1*	0.48	0.43	0.62	0.91	1.01	1.65
	2*	0.51	0.23	0.52	0.71	0.98	1.86
	3*	0.4	0.49	0.61	0.86	1.88	1.25
	4*	0.1	0.2	0.49	0.62	0.95	1.1
RMSE	1*	4.01	3.29	5.09	7.69	12.04	8.97
	2*	4.16	2.61	4.92	6.56	7.61	7.85
	3*	3.63	4.35	5.28	8.96	11.55	7.05
	4*	1.38	2.53	5.43	5.27	7.09	7.11
Relative Prediction Error	1*	1.76	1.98	2.06	3.01	3.75	4.03
	2*	1.9	0.93	1.86	2.66	3.1	3.14
	3*	1.49	1.81	2.12	2.9	3.82	3.55
	4*	0.44	0.87	1.88	2.03	2.54	3.01

1*: LSTM without change point.
2*: RNN under fixed change point.
3*: LSTM under fixed change point.
4*: Proposed Dual-LSTM framework.

The Dual-LSTM framework is also compared to two recently published LSTM-based models for RUL prediction that are applied to the same degradation dataset. The first benchmark model is the vanilla LSTM network introduced in [38]. The authors in [38] designed the vanilla LSTM network to predict the RUL of turbofan engines and also supported it by the SVM as the change point detector.

The comparison results both on FD001 and FD003 are summarized in Table 5. The standard RNN and GRU are additional benchmark models discussed in [38]. $\boldsymbol{P}_5$, $\boldsymbol{P}_{10}$, and $\boldsymbol{P}_{20}$ in Table 5 are percentages of samples in the testing set with relative prediction errors less than or equal to 5%, 10%, and 20% respectively. In this case, the larger the $\boldsymbol{P}_5$, $\boldsymbol{P}_{10}$, and $\boldsymbol{P}_{20}$, the better the prediction performance. From Table 5, we can see that the Dual-LSTM framework has the largest $\boldsymbol{P}_5$, $\boldsymbol{P}_{10}$, and $\boldsymbol{P}_{20}$, as well as the smallest SF at all monitoring points both FD001 and FD003. This suggests that the Dual-LSTM framework outperforms the vanilla LSTM network and the other two benchmark models in [38], even based on their predefined evaluation metrics.

The Dual-LSTM framework is also compared to the dual-task LSTM networks proposed by [21]. The dual-task LSTM uses LSTM to predict the RUL and classify the degradation states simultaneously. The comparison results are summarized in Table 6. The authors of [21] compared the results of three different cell structures LSTM, GRU, and peephole. GRU and peephole are variants of LSTM with different gate structures and activation functions. The evaluation metric *RUL_n* in Table 6 is the RMSE of the last n cycles before the failure.

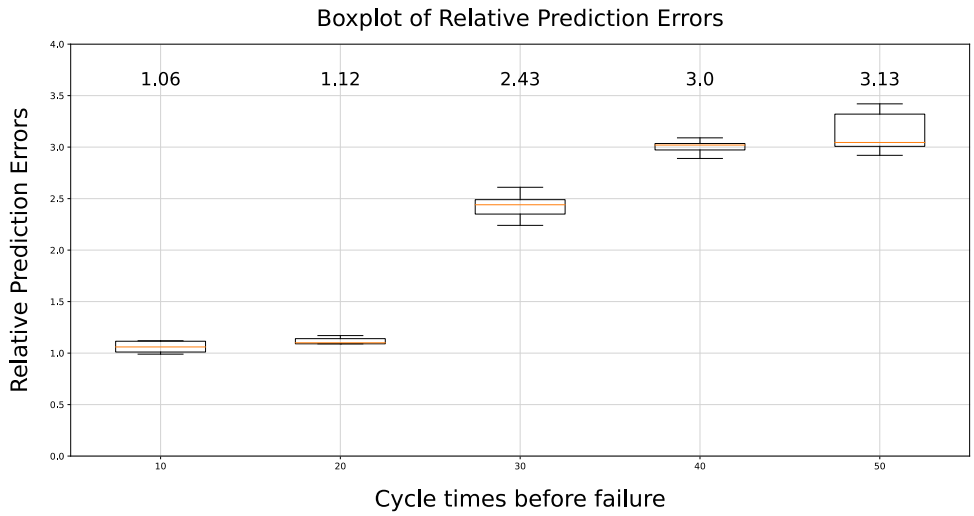


Fig. 16. Boxplot of relative prediction errors on different monitoring points over 10 experiments on FD001.

Table 5
Comparison results with LSTM-based models for RUL prediction in [38].

Dataset setting	Models	50 cycles left				10 cycles left				5 cycles left			
		P_5(%)	P_{10}(%)	P_{20}(%)	SF	P_5(%)	P_{10}(%)	P_{20}(%)	SF	P_5(%)	P_{10}(%)	P_{20}(%)	SF
FD001	Standard RNN	10	35	60	5581.41	75	90	95	256.34	85	95	95	118.52
	GRU	45	85	85	17.96	100	100	100	0.35	100	100	100	0.53
	Vanilla LSTM	30	55	70	37.51	100	100	100	0.43	100	100	100	0.30
	Dual-LSTM	40	70	100	1.15	100	100	100	0.22	100	100	100	0.12
FD003	Standard RNN	35	60	85	48.96	100	100	100	0.36	95	100	100	0.43
	GRU	40	65	95	14.29	90	100	100	1.45	85	100	100	1.74
	Vanilla LSTM	50	70	90	30.58	100	100	100	0.23	100	100	100	0.21
	Dual-LSTM	60	90	100	1.10	100	100	100	0.20	100	100	100	0.12

Table 6
RMSE comparison with three benchmark models in [21].

Cell	FD001				FD003	
	LSTM	GRU	peephole	Dual-LSTM	peephole	Dual-LSTM
*RUL*_50	7.03	6.37	3.58	1.11	5.23	3.30
*RUL*_20	5.01	6.09	3.16	0.86	5.87	1.01
*RUL*_10	4.17	4.86	3.37	1.06	5.02	0.76
*RUL*_5	2.32	2.45	2.38	1.16	2.45	0.93

$$RUL\_n = \sqrt{\frac{1}{n}\sum_{t=T_f-n+1}^{T_f}\left(\widehat{RUL}_t - RUL_t\right)^2} \tag{21}$$

Table 6 clearly shows that the peephole outperforms the other two models with lower RMSE at different levels of RUL on FD001; however, it still underperforms the proposed Dual-LSTM at all the tested levels of RUL. Specifically, the proposed Dual-LSTM framework showed the lowest RMSE of 1.11 before 50 cycles of the failure, which manifests its effectiveness for raising early failure alarms and optimizing condition-based maintenance strategies. Besides, for FD003, the proposed Dual-LSTM framework also performs better than the peephole model at all monitoring points according to the results provided by [21]. One major reason behind the outperforming performance of the Dual-LSTM is the robustness of its change point detection.

In summary, the proposed Dual-LSTM is compared to five benchmark methods using multiple evaluation metrics. All the comparative results show that the Dual-LSTM outperforms existing benchmark models for RUL predictions.

5. Conclusion and discussion

In this paper, a Dual-LSTM framework combining change point detection and RUL prediction is proposed to achieve the high-precision RUL prediction. With this proposed Dual-LSTM framework, we can detect the uncertain change point of each unit specifically and mainly focus on the degradation process beyond the change point to achieve more accurate RUL predictions.

The performance of the proposed Dual-LSTM framework is compared with the RNN and LSTM networks with fixed change points. The RUL performance comparisons demonstrate that the proposed Dual-LSTM framework outperforms the benchmark models. Additionally, the proposed Dual-LSTM framework also shows outperforming performances compared with LSTM-based models in [38] and [21] that do not assume fixed change points. Besides, the outstanding RUL performance achieved at the early monitoring points demonstrates its ability to raise early failure alarms and provides strong support for maintenance scheduling and work in progress regulation [42].

Future research studies must be directed to investigate advanced physics-based loss functions and develop other variants of LSTM networks for simultaneous change point detection and RUL prediction.

Author statement

Zunya Shi: Conceptualization, Methodology, Data Curation, Software, Investigation, Validation, Visualization, Writing- Original Draft.

Abdallah Chehade: Conceptualization, Methodology, Data Curation, Formal Analysis, Writing-Reviewing & Editing, Supervision.

Declaration of Competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Alaswad S, Xiang Y. A review on condition-based maintenance optimization models for stochastically deteriorating system. Reliab Eng Syst Saf 2017;157:54–63.
[2] Ghamlouch H, Fouladirad M, Grall A. The use of real option in condition-based maintenance scheduling for wind turbines with production and deterioration uncertainties. Reliab Eng Syst Saf 2019;188:614–23.
[3] Sakib N, Wuest T, Alain W, Stief P, Siadat A. Challenges and opportunities of condition-based predictive maintenance: a review. Procedia CIRP 2018;78:267–72.
[4] Wang L, Lu Z. A predictive production planning with condition-based maintenance in a deteriorating production system. Int. Conferen. Robot. Autom. Eng. (ICRAE) 2016;(1):35–8.
[5] Chehade A, Bonk S, Liu K. Sensory-based failure threshold estimation for remaining useful life prediction. IEEE Trans Rel 2017;66(3):939–49.
[6] Y. C Zhang B, Xu L, Li A. Remaining useful life based maintenance for deteriorating systems subject to continuous degradation and shock. Procedia CIRP 2018;72:1311–5.
[7] Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics : a systematic review from data acquisition to RUL prediction. Mech Syst Signal Process 2018;104:799–834.
[8] Si XS, Wang W, Hu CH, Zhou DH. Remaining useful life estimation - A review on the statistical data-driven approaches. Eur J Oper Res 2011;213(1):1–14.
[9] Arnold A, Nallapati R, Cohen WW. A comparative study of methods for transductive transfer learning. In: Seventh IEEE Int. Conf. Data Min. Work; 2007. p. 77–82.
[10] Schwabacher M, Goebel K. A survey of artificial intelligence for prognostics. In: AAAI Fall Symposium; 2006. p. 107–14.
[11] Okoh C, Roy R, Mehnen J, Redding L. Overview of remaining useful life prediction techniques in through-life engineering services. Procedia CIRP 2014;16:158–63.
[12] Liu K, Huang S. Integration of data fusion methodology and degradation modeling process to improve prognostics. IEEE Trans Autom Sci Eng 2014;13(1):344–54.
[13] Chehade A, Shi Z. Sensor fusion via statistical hypothesis testing for prognosis and degradation analysis. IEEE Trans Autom Sci Eng 2019;16(4):1774–87.
[14] Liu K, Chehade A, Song C. Optimize the signal quality of the composite health index via data fusion for degradation modeling and prognostic analysis. IEEE Trans Autom Sci Eng 2017;14(3):1504–14.
[15] Chehade A, Song C, Liu K, Saxena A, Zhang X. A data-level fusion approach for degradation modeling and prognostic analysis under multiple failure modes. J Qual Technol 2018;50(2):150–65. Apr.
[16] Ordóñez C, Sánchez F, Roca-pardiñas J, Javier F, Juez DC. A hybrid ARIMA – SVM model for the study of the remaining useful life of aircraft engines. J Comput Appl Math 2019;346:184–91.
[17] Wang M, Wang J. CHMM for tool condition monitoring and remaining useful life prediction. Int J Adv Manuf Tech 2012:463–71.
[18] Chehade A, Hussein AA. A multi-output convolved gaussian process model for capacity estimation of electric vehicle Li-ion battery cells. In: 2019 IEEE Transp. Electrif. Conf. Expo.; 2019. p. 1–4.
[19] Si X, Wang W, Hu C, Chen M, Zhou D. A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. Mech Syst Signal Process 2013;35(1–2):219–37.
[20] Li J, Li X, He D. A directed acyclic graph network combined With CNN and LSTM for remaining useful life prediction. IEEE Access 2019;7:75464–75.
[21] Miao H, Li B, Sun C, Liu J. Joint learning of degradation assessment and RUL prediction for aero-engines via dual-task deep LSTM networks. IEEE Trans Ind Informatics 2019;15(9):5023–32.
[22] Zheng C, Liu W, Chen B, Gao D, Cheng Y, Yang Y, Zhang X, Li S, Huang Z, Peng J. A data-driven approach for remaining useful life prediction of aircraft engines. In: IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC; 2018. p. 184–9.
[23] Savargaonkar M, Chehade A, Shi Z, Hussein AA. A cycle-based recurrent neural network for state- of-charge estimation of Li-ion battery cells. In: 2020 IEEE Transp. Electrif. Conf. Expo; 2020. p. 584–7.
[24] Savargaonkar M, Chehade A. An adaptive deep neural network with transfer learning for state-of-charge estimations of battery cells. In: 2020 IEEE Transp. Electrif. Conf. Expo; 2020. p. 598–602.
[25] Pascanu R, Mikolov T, Bengio Y. Understanding the exploding gradient problem. CoRR 2012. vol. abs/1211.5063[Online]. Available, http://arxiv.org/abs/1211.5063.
[26] Hochreiter S, Munchen TU. The vanishing gradient problem during learning recurrent nets and problem solutions. Int J Uncertainty, Fuzziness Knowl-Based Syst 1998;6(02):107–16.
[27] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. ICML; 2013. p. 1310–8. in.
[28] Gers FA, Schraudolph NN. Learning precise timing with LSTM recurrent networks. JMLR 2002;3:115–43.
[29] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1–32.
[30] Graves A, Liwicki M, Fern S, Bertolami R, Bunke H. A novel connectionist system for unconstrained handwriting recognition. IEEE T Pattern Anal 2009;31(5):855–68.
[31] Zheng S, Ristovski K, Farahat A, Gupta C. Long short-term memory network for remaining useful life estimation. In: Int. Conf. Progn. Heal. Manag; 2017. p. 88–95.
[32] Dong S, Liu Y, Yuan M, Lin L, Wu Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing 2017;275:167–79.
[33] Zhang J, Wang P, Yan R, Gao RX. Long short-term memory for machine remaining life prediction. J Manuf Syst 2018;48:78–86. June.
[34] Wang S, Zhang X, Gao D, Chen B, Cheng Y, Yang Y, Yu W, Huang Z, Peng J. A remaining useful life prediction model based on hybrid long-short sequences for engines. In: Int. Conf. Intell. Transp. Syst.; 2018. p. 1757–62.
[35] Elsheikh A, Yacout S, Ouali M. Bidirectional handshaking LSTM for remaining useful life prediction. Neurocomputing 2019;323:148–56.
[36] Song Y, Shi G, Chen L, Huang X, Xia T. Remaining useful life prediction of turbofan engine using hybrid model based on autoencoder and bidirectional long short-term memory. J Shanghai Jiaotong Univ 2018;23:85–94.
[37] Yuan M, Wu Y, Lin L. Fault diagnosis and remaining useful estimation of aero engine using LSTM neural network," in Int. In: Conf. Aircraft Utility Systems; 2016. p. 135–40.
[38] Wu Y, Yuan M, Dong S, Lin L, Liu Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing 2018;275:167–79.
[39] Ramasso E, Saxena A. Performance benchmarking and analysis of prognostic methods for CMAPSS datasets. In: Int. Conf. Progn. Heal. Manag.; 2014. p. 1–15.
[40] Ramasso E, Saxena A. Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset. In: Proc. Annu. Conf. Progn. Heal. Manag; 2014. p. 612–22.
[41] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. In: Int. Conf. Progn. Heal. Manag; 2008. p. 1–9.
[42] Chehade A, Duffie N. Optimal dynamic behavior of adaptive WIP regulation with multiple modes of capacity adjustment. In: Procedia CIRP. 19; 2014. p. 168–73.

Zunya Shi received the B.S. degree in business administration and the M.S. degree in industrial engineering from Xi'an Jiaotong University, Xi'an, China, in 2014 and 2016, respectively. At present, she is currently working toward the Ph.D. degree in the Department of Industrial and Manufacturing Systems Engineering at the University of Michigan–Dearborn. Her research interests focus on prognostics and degradation analysis. She is a member of IISE.

Abdallah Chehade received the B.S. degree in mechanical engineering from the American University of Beirut, Beirut, Lebanon, in 2011 and the M.S. degree in mechanical engineering, the M.S. degree in industrial engineering, and the Ph. D. in industrial engineering from the University of Wisconsin–Madison in 2014, 2014, and 2017, respectively. Currently, he is an assistant professor in the Department of Industrial and Manufacturing Systems Engineering at the University of Michigan– Dearborn. His-research interests are AI-driven solutions for smart and connected systems, data fusion for prognosis and degradation analysis, and safe implementation of AI. Dr. Chehade is a member of INFORMS, IEEE, and IISE.