Neurocomputing 466 (2021) 58–68

Contents lists available at ScienceDirect

Neurocomputing

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

Attention-based sequence to sequence model for machine remaining useful life prediction

Mohamed Ragab [a,b], Zhenghua Chen [b,*], Min Wu [b], Chee-Keong Kwoh [a], Ruqiang Yan [c], Xiaoli Li [b]

[a] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[b] Machine Intellection Department, Institute for Infocomm Research, A*STAR, Singapore
[c] School of Mechanical Engineering, Xi'an Jiaotong University, China

ARTICLE INFO

Article history:
Received 11 April 2021
Revised 24 July 2021
Accepted 15 September 2021
Available online 20 September 2021
Communicated by Zidong Wang

Keywords:
Remaining useful life
Sequence to sequence with auxiliary task
Attention mechanism

ABSTRACT

Accurate estimation of remaining useful life (RUL) of industrial equipment can enable advanced maintenance schedules, increase equipment availability and reduce operational costs. However, existing deep learning methods for RUL prediction are not completely successful due to the following two reasons. First, relying on a single objective function to estimate the RUL will limit the learned representations and thus affect the prediction accuracy. Second, while longer sequences are more informative for modelling the sensor dynamics of equipment, existing methods are less effective to deal with very long sequences, as they mainly focus on the latest information. To address these two problems, we develop a novel attention-based sequence to sequence with auxiliary task (ATS2S) model. In particular, our model jointly optimizes both *reconstruction loss* to empower our model with predictive capabilities (by predicting next input sequence given current input sequence) and *RUL prediction loss* to minimize the difference between the predicted RUL and actual RUL. Furthermore, to better handle longer sequences, we employ the attention mechanism to focus on all the important input information during the training process. Finally, we propose a new *dual-latent feature representation* to integrate the encoder features and decoder hidden states, to capture rich semantic information in data. We conduct extensive experiments on four real datasets to evaluate the efficacy of the proposed method. Experimental results show that our proposed method can achieve superior performance over 13 state-of-the-art methods consistently.

1. Introduction

Prognostic and Health Management (PHM) is receiving much attention in many industrial applications, as it can potentially reduce equipment downtime and increase system reliability. Typically, PHM systems are leveraged to monitor the condition of mechanical or electrical equipment based on their environmental information and domain knowledge. One key task in PHM is the reliable prediction of *remaining useful life* (RUL) of an equipment. RUL is defined as time interval between the current state and the end-of-life state. With accurate RUL estimation, industries can have predictive maintenance planning and thus prevent catastrophic failures or faults from happening [1]. Yet, with the sophisticated machine design and the dynamic surrounding environment, precise estimation of RUL can be of great challenge. Various approaches have been proposed to estimate the RUL of machines. These approaches can be classified into three major categories, namely, model-based approaches, data-driven approaches, and hybrid approaches. Specifically, model-based approaches require strong theoretical understanding to model the behaviour of equipment and its detailed degradation process [2]. As equipment complexity continues to evolve, it becomes extremely challenging to apply model-based methods in real applications [3]. With increasing data availability in smart manufacturing, data-driven approaches have emerged more promisingly for predicting the RUL of equipment. These methods aim to explore the underlying relationship between the sensor readings and degradation trend, such as hidden Markov model, artificial neural network [4], extreme learning machines [5], and support vector machines [6]. However these approaches require manual feature engineering to extract the corresponding degradation pattern, which can be very laborious task. Hybrid approaches aim to improve the physical model via leveraging the data availability to better detect the deterioration trend [7]. They also suffer from the difficulty of

* Corresponding author.
E-mail addresses: mohamedr002@e.ntu.edu.sg (M. Ragab), chen0832@e.ntu.edu.sg (Z. Chen), wumin@i2r.a-star.edu.sg (M. Wu), asckkwoh@ntu.edu.sg (C.-K. Kwoh), yanruqiang@xjtu.edu.cn (R. Yan), xlli@i2r.a-star.edu.sg (X. Li).

https://doi.org/10.1016/j.neucom.2021.09.022
0925-2312/

building accurate physical models and effectively combining both techniques.

In recent years, with the surge of computational power and the data volume, deep learning with its hierarchical multi-layer representative power can automatically extract silent features without handcrafted feature engineering. As a result, research paradigm of RUL prediction is shifting from conventional machine learning to deep learning based architectures. Various deep learning methods, including convolutional neural networks (CNN) and recurrent neural networks (RNN), have been developed for RUL prediction. In particular, CNN based methods aim to use 1-dimensional convolutional kernel to extract the sequential information from time series data [8,9]. However, CNN-based approaches still have limited capability for RUL prediction, as they are not able to capture long-range sequential dependencies in sensory data.

RNN based approaches were developed to capture the temporal dependency among time series data [10]. However, conventional RNN architectures still suffer from vanishing gradient problem with longer time dependencies. To tackle this issue, the long short-term memory (LSTM), a gated RNN with both long and short memories, was developed to address the vanishing gradient problem and achieved the state-of-the-art performance for RUL prediction [11–15]. Yet, LSTM based methods tend to lose relevant and important historical information when dealing with very long sequences [16], as they only focus on latest sequence information when mapping the whole input sequence into fixed-length vector representation. In addition, all the aforementioned methods only used a single objective, i.e., minimizing the mean square error (MSE) between the predicted and true values for the model training. We argue that using a single objective can limit the generalization performance of the model on unseen test data [17,18].

To address the above two problems, we propose a dual-objective sequence to sequence approach named ATS2S for accurate RUL prediction. First, to address the shortage of LSTM with long sequences, we propose an attention based decoding and focus on the important parts of the input sequence (instead of the latest information in LSTM) that can maximize the decoding performance without losing relevant information. Additionally, we integrate the last hidden state of the decoder with the encoder hidden features as a comprehensive *dual-latent feature representation* for the RUL predictor. Second, inspired by the success of auxiliary tasks in improving the generalization performance [17,18] in computer vision applications, we design a novel auxiliary task to further improve the prediction capability on unseen test data. Particularly, given the current input sequence, we train the model to reconstruct the *future* input sequence in an unsupervised manner. Concurrently, we train the model with a supervised MSE loss between the true RUL labels and the predicted ones.

Overall, our main contributions can be summarized as follows.

- Our model jointly optimizes both *reconstruction loss* of future sequence to empower our model with predictive capabilities (by predicting the next input sequence given current input sequence) and *RUL prediction loss* to minimize the difference between the predicted RUL and actual RUL.
- We design an *attention mechanism* in the encoder-decoder network to handle the long sequences. As such, our model can focus on the most relevant information of the input sequences for RUL prediction.
- We propose a new *dual-latent feature representation* to integrate the encoder features and decoder hidden states, to capture rich semantic information in the data for RUL prediction.
- We conduct extensive experiments on four benchmark datasets to evaluate our proposed approach. The results show that the proposed approach can significantly improve RUL prediction over 13 state-of-the-arts.

2. Related work

Deep learning with the ability of automatic feature extraction has achieved wide success in many fields, including computer vision, natural language processing, and speech recognition [19]. Very recently, various deep learning methods, e.g., CNN and RNN, have also been explored for RUL prediction [20,21]. For instance, Li et al., proposed a CNN with 1-D filters to extract features from input sensor data for RUL prediction and also used window-time approach to prepare data samples for enhanced feature extraction [8]. Yang et al., developed a two-stage approach by using one CNN network to inspect the fault points and another CNN network to estimate the RUL [9]. Zhu et al., proposed a multi-scale CNN to extract features and predict the degradation of bearings [22]. Zhang et al., combined multi-layer perception (MLP) and CNN to extract features from vibration data and predict the health index of machines [23]. As shown in above studies, CNN based methods have achieved good performance for RUL prediction. However, they have limitations when dealing with the sequence data as they ignore the temporal dependency among data points in a given input sequence. Therefore, it is motivated to explicitly handle the temporal dependency of sequence data for RUL prediction.

RNN based methods have been shown to be effective in modeling dynamic systems and learning temporal dependency in data. In particular, Long Short-Term Memory (LSTM) is a special type of RNN that can model the dynamics of sequences by introducing the memory cells [24]. It has become increasingly popular for RUL prediction. For example, Zheng et al., have used two layers LSTM network to predict the RUL of turbofan engines [11]. Huang et al., employed a stacked-bidirectional LSTM with auxiliary inputs to model sensor data under multiple operating conditions [12]. For instance, Miao et al., designed a deep LSTM framework to jointly perform degradation assessment and RUL prediction [14]. Chen et al., fused the learned features of the LSTM network with the handcrafted features to boost the RUL prediction performance [13]. Yet, LSTM based approaches tend to only focus on latest information of the sequence and may lose important information at the very beginning of the sequence [16].

More relevant approaches to our work are the encoder-decoder based methods such as LSTM-ED [25] and BiLSTM-ED [26], which leveraged health index estimation to predict the RUL. Our proposed ATS2S is different from them in the following aspects. First, ATS2S is an end-to-end framework, while their methods extract features and predict RUL separately. Second, we propose a novel auxiliary task of reconstructing the future sequence in an unsupervised manner to improve the generalization power of our model to the unseen test data. Last, ATS2S implements an attention mechanism and leverages the *dual-latent feature representation* for RUL prediction, while their methods still use the encoder's last hidden state as features for health index prediction and RUL estimation.

3. Methodology

In this section, we will introduce our proposed attention-based sequence to sequence with auxiliary task (ATS2S) model for RUL prediction.

3.1. Overview of ATS2S

The proposed ATS2S is composed of three main components, namely, encoder, decoder, and RUL predictor, as shown in Fig. 1. Firstly, the encoder maps the whole input sequence into a sequence of hidden states. Unlike conventional encoder-decoder models that compress all the input information into the single fixed-length vector (i.e., encoder's last hidden state), we design

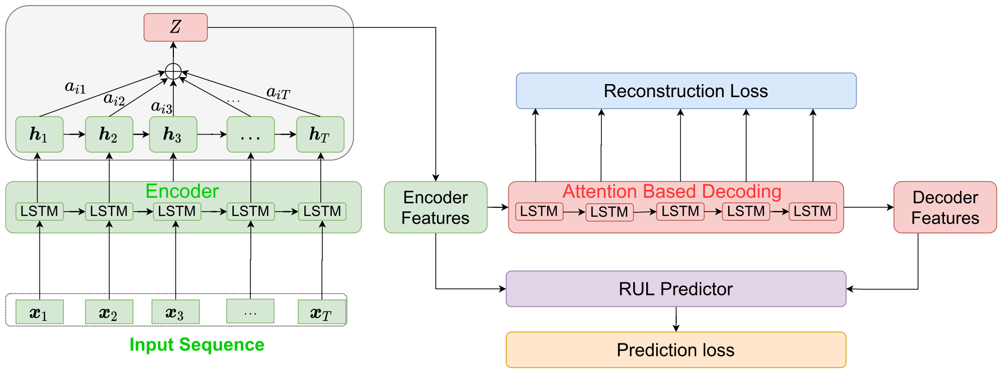


Fig. 1. Attention-based sequence to sequence model for RUL prediction.

an *attention layer* to select the hidden states that are relevant and important for the decoding. Then, we pass the weighted sum of the encoder hidden states (i.e., attention outputs) as *encoder features* to decoder. The decoder is then trained to forecast the *next* input sequence given the current input sequence, in order to give our model more *predictive power*. Finally, the RUL prediction network (a fully connected neural network) takes *dual-latent feature representation* to integrate both the encoder and decoder hidden states/features for RUL prediction. The predictor maps from the feature dimension space to a single value, i.e., predicted RUL.

Note that our ATS2S method jointly optimizes the RUL prediction loss, which is the difference between the predicted RUL label and ground-truth label, as well as the reconstruction loss, which is the difference between predicted and actual sequence. Next, we will introduce each of the three components of ATS2S in details.

3.2. LSTM based encoder

In order to model the input dynamics of sensor signals, we employ the LSTM model as our backbone architecture in the sequence to sequence model. Given an input sample $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) \in \mathbb{R}^{n \times T}, \mathbf{x}_t \in \mathbb{R}^n$ is n-dimensional input vector at each time step t $(1 \leqslant t \leqslant T)$ from n sensors. At each time step t, LSTM takes the input vector $\mathbf{x}_t$ and *previous* hidden state $\mathbf{h}_{t-1}$ to produce *current* hidden state $\mathbf{h}_t$, current long term memory cell $\mathbf{c}_t$ and output $\mathbf{o}_t$. The following equations demonstrate the detailed process in the LSTM cell.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i), \tag{1}$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_t - 1 + \mathbf{b}_f), \tag{2}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_t - 1 + \mathbf{b}_o), \tag{3}$$
$$\mathbf{g}_t = tanh(\mathbf{W}_g\mathbf{x}_t + \mathbf{U}_g\mathbf{h}_t - 1 + \mathbf{b}_g), \tag{4}$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \tag{5}$$
$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t), \tag{6}$$

where σ is nonlinear *sigmoid* function, $\odot$ is an element-wise multiplication operator, $\mathbf{W}_* \in \mathbb{R}^{n \times p}$ (i.e., $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o$ and $\mathbf{W}_g$) are the model parameters that map from input dimension n to hidden dimension $p, \mathbf{U}_* \in \mathbb{R}^{p \times p}$ map from the previous hidden dimension to the current hidden dimension, and $\mathbf{b}_* \in \mathbb{R}^p$ are bias vectors. It worth noting that the parameters are shared across all the time steps. The Encoder model f_{enc} takes the input sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ and produces a sequence of hidden states $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T)$ and a sequence of cell states $(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_T)$ in Eq. (7).

$$[(\mathbf{h}_1, \ldots, \mathbf{h}_T), (\mathbf{c}_1, \ldots, \mathbf{c}_T)] = f_{enc}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T; \theta_{enc}), \tag{7}$$

where $\theta_{enc} = [\mathbf{W}_{enc}, \mathbf{U}_{enc}, \mathbf{b}_{enc}]$ are the parameters of the Encoder model.

3.3. Decoding

The main idea of attention is inspired by human visual systems where human can focus on the relevant part of a scene and ignore irrelevant parts. Similarly, we design an attention mechanism in our sequence to sequence model for the whole sequence of hidden states. In particular, we focus on *all the important* hidden states of the encoder for decoding, while standard sequence to sequence model relies solely on the *last* hidden state and thus loses valuable information.

3.3.1. Calculation of attention weights

For the decoding process, we employ the hidden states from both encoder and decoder to produce the attention weights. Note that each decoding time step will have different attention weights. For the decoding time step i, the attention weights $a_i = [a_{i1}, a_{i2}, \ldots, a_{iT}]$ can be calculated by the attention module $f_{attn}(\cdot)$, which can be expressed as

$$a_{\mathbf{i}} = f_{attn}(\mathbf{s}_{i-1}, \mathbf{H}), \tag{8}$$

where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T] \in \mathbb{R}^{p \times T}$ represents the encoder hidden states, and $s_{i-1} \in \mathbb{R}^p$ is the previous decoder hidden state which is initialized by the last encoder hidden state at the very beginning of the decoding process. Here, p is the dimension of each hidden state and T is the total number of time steps for one sample. Fig. 2 shows the detailed structure of the attention module $f_{attn}(\cdot)$. Specifically, the decoder hidden state s_{i-1} will be concatenated with each encoder hidden state and then passed through a fully connected layer $FC : \mathscr{R}^{2p} \rightarrow \mathscr{R}$. The outputs of the fully connected layer

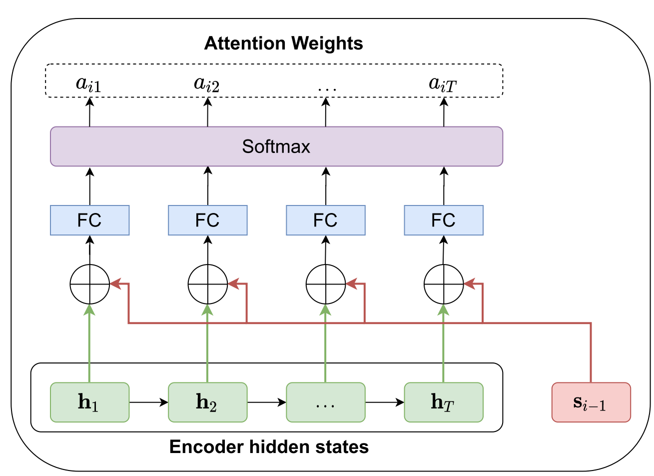


Fig. 2. The detailed structure of the attention module..

will be fed into a softmax layer which produces the final attention weights a_i.

3.3.2. Attention based decoding

For each time step i in the decoding process, we employ the attention weights $\mathbf{a}_i$ and the encoder hidden states $\mathbf{H}$ to calculate the context vector z_i as follows:

$$z_i = \sum_{j=1}^{T} a_{ij} h_j, \tag{9}$$

where $\mathbf{Z} = [z_i, z_{i+1}, \ldots, z_{i+T-1}]$ is a collection of context vectors for all the time steps. For the i-th time step, the context vector z_i will be concatenated with the current input $\hat{y}_i$ which is the prediction of the previous step. Then, the concatenated vector and the previous hidden state s_{i-1} will be passed through the decoder cell, which can be formalized as:

$$s_i = f_{dec}((\hat{y}_i, \mathbf{z}_i, \mathbf{s}_{i-1}); \boldsymbol{\theta}_{dec}). \tag{10}$$

where θ_{dec} represents the parameters of the decoder network. Then, we map from $\mathbf{s}_i$ to the next step of the target $\hat{\mathbf{y}}_{i+1}$ in Eq. (11):

$$\hat{\mathbf{y}}_{i+1} = f_{FC}(\mathbf{s}_i; \boldsymbol{\theta}_{FC}), \tag{11}$$

where f_{FC} is a fully connected layer that maps from the hidden dimension to the output dimension, and θ_{FC} represents the parameters of the fully connected network. It worth noticing that we pass the output of the last time step as the next input. Hence, the decoder is trained to predict the future step given the current input which can be valuable for the RUL prediction.

3.4. RUL predictor

The objective of the RUL predictor is to accurately predict the corresponding RUL value for each input sequence (sensor signals). We first integrate the last hidden state of the decoder with the encoder hidden features, as a comprehensive *dual-latent feature representation*, and then design a function that maps the dual-latent feature representation to a single RUL value. We denote the RUL predictor as $f_{pred} : \mathbb{R}^D \rightarrow \mathbb{R}$ in Eq. (12), where D is the dimension of dual-latent feature representation.

$$\widehat{RUL} = f_{pred}((\mathbf{h}_T, \mathbf{s}_T); \boldsymbol{\theta}_{pred}), \tag{12}$$

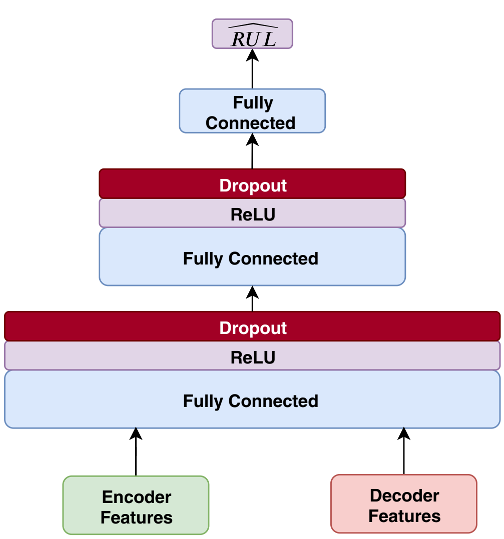


Fig. 3. Architecture of RUL predictor network.

where $\widehat{RUL} \in \mathbb{R}$ is the predicted label, $\mathbf{h}_T$ and $\mathbf{s}_T$ are the features of encoder and decoder respectively. Fig. 3 shows the diagram of the RUL predictor, which is a multi-layer feed-forward network followed by a non-linear activation function (i.e., ReLU).

3.5. Multi-objective optimization

3.5.1. Reconstruction loss

In our ATS2S, we aim to forecast the next input sequence given the current input sequence so that our model has predictive power. Fig. 4 shows the detailed process of the forecasting-based reconstruction loss. Given a predicted sequence by the decoder $\hat{\mathbf{Y}}_i = (\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \cdots, \hat{\mathbf{y}}_T) \in \mathbb{R}^{n \times T}$, and a target sequence $\mathbf{Y}_i = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T) \in \mathbb{R}^{n \times T}$ where $\mathbf{y}_t = \mathbf{x}_{t+1} \in \mathbb{R}^n, T$ is the length of the sequence, and n is the number of sensors. We define the reconstruction loss as the mean square error between the target output and predicted output. Eq. (13) shows the formulation of the reconstruction loss.

$$L_{rec}(\theta) = \frac{1}{N} \sum_{i=1}^{N} ||\hat{\mathbf{Y}}_i - \mathbf{Y}_i||_2^2, \tag{13}$$

where θ is the model parameters, and N is the total number of samples.

3.5.2. RUL prediction loss

The RUL prediction loss is defined as the mean square error between the true RUL label and the predicted RUL label for each input sequence. The RUL loss can be defined as follows:

$$L_{rul}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left(\widehat{RUL}_i - RUL_i \right)^2 \tag{14}$$

where $\widehat{RUL}_i$ is predicted label and RUL_i is the true label.

3.5.3. Joint loss

The proposed model aims to optimize both reconstruction and prediction losses concurrently. We argue that jointly optimizing both losses can not only provide a good and rich latent representation, but also improve the accuracy of RUL prediction. The joint loss can be formulated as follows

$$L(\theta) = \alpha L_{rec}(\theta) + L_{rul}(\theta), \tag{15}$$

where α is a tunable parameter to control the contribution of the reconstruction loss. It can control the contribution from reconstruction loss while maintaining the prediction loss (the major loss for RUL prediction).

4. Experiments and results

We have conducted extensive experiments on benchmark data to evaluate the performance of our proposed model.

4.1. Experimental data

We evaluate our proposed ATS2S method on C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) data [27]. C-MAPSS data describes the degradation process of aircraft engines as shown in Fig. 5. It consists of four benchmark datasets with different number of training/testing engines, operating conditions and fault types. The details about these four datasets are summarized in Table 1.

4.1.1. Sensor data selection

Twenty-one sensors are deployed in different locations of the engine to measure temperature, pressure and speed. To select relevant sensors for RUL prediction, we visualize the signals from all

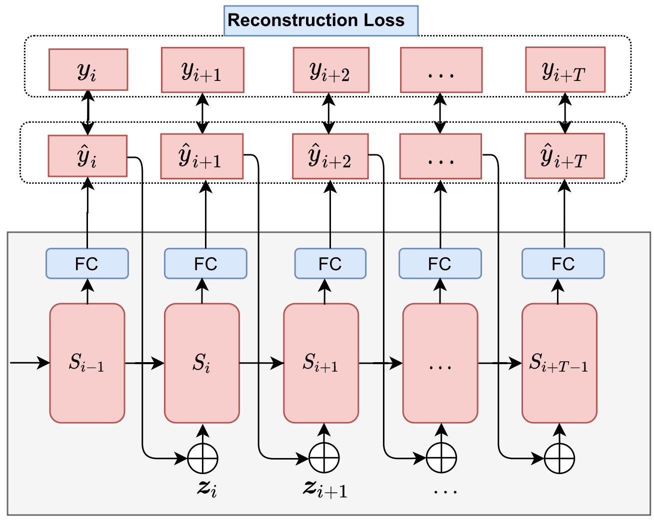


Fig. 4. Details of forecasting based reconstruction task.

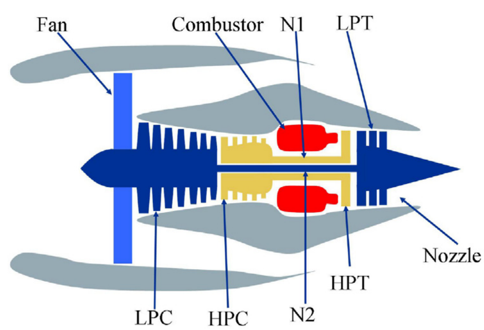


Fig. 5. Diagram of the engines in C-MAPSS data [27].

the 21 sensors for FD001. Fig. 6 shows the sensor readings for a randomly selected engine. While most of sensors have a clear degradation trend, other sensors remain constant in the run-to-fail experiments (i.e., sensors 1, 5, 6, 10, 16, 18 and 19). Therefore, 14 sensors, namely sensors 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20 and 21, are used for RUL prediction. FD003 follows the same degradation patterns as FD001 and thus we use the same subset of sensors for FD001 and FD003. Similar procedure has been done for FD002 and FD004. Eventually we adopt 9 sensors [12], namely sensors 3, 4, 9, 11, 14, 15, 17, 20 and 21, for RUL prediction on FD002 and FD004.

4.1.2. Data segmentation and processing

We follow the sliding window method [28,29] for data segmentation. Fig. 7 shows the process of data segmentation with sliding window, where W is the window size, n is the number of sensors and s is the shifting size. Given that the total number of cycles is T, the RULs for the first and second windows/samples are thus $T-W$ and $T-W-s$, respectively. In our experiments, W and s are set to be 30 and 1, respectively.

Table 1
Properties of C-MAPSS dataset.

Dataset	FD001	FD002	FD003	FD004
# Training engines	100	260	100	249
# Testing engines	100	259	100	248
# Operating conditions	1	6	1	6
# Fault types	1	1	2	2

Moreover, we adopt the piece-wise linear degradation model [12,30] for the RUL labels. In case a sample has RUL value greater than a pre-defined threshold, we re-set the RUL value as the threshold for this sample. In particular, we follow the previous studies [12,30] and set the threshold as 125 for FD001/FD003 and 130 for FD002/FD004.

4.1.3. Data normalization

The prognostic problem of real systems involves different types of sensors and different operating conditions. Directly feeding the raw sensor readings with high variance to the machine learning models may hinder the learning process and affect the model performance. To remedy this issue, we use Min–Max normalization for each sensor restrict the values within $[0,1]$. For datasets with multiple working conditions, we normalize the sensor readings with respect to their corresponding working condition. In particular, we first group the sensors by their corresponding working conditions, then we apply normalization on each cluster independently. To formulate the scaling function, let a vector $\mathbf{Q}_{rm}$ contains all the data points of the r-th sensor under m-th working condition. The normalized vector $\hat{\mathbf{Q}}_{rm}$ is calculated as follows:

$$\hat{\mathbf{Q}}_{rm} = \frac{\mathbf{Q}_{rm} - min(\mathbf{Q}_{rm})}{max(\mathbf{Q}_{rm}) - min(\mathbf{Q}_{rm})}. \quad (16)$$

4.2. Experimental settings and evaluation metrics

4.2.1. Experimental settings

Our architecture is composed of three main parts, namely, encoder network, decoder network, and RUL predictor network. Both encoder and decoder networks rely on LSTM model. To reconstruct the next input sample, the decoder network is followed by a single layer fully connected (FC) network to map from the hidden dimension to the output dimension. The attention mechanism is implemented by two FC networks, i.e., one network computes the attention weights with dimension of $n \times 30$, while the other network generates a weighted sum of the encoder hidden states using attention weights. Finally, the RUL predictor network consists of three FC layers, and each layer is followed by rectified linear unit (ReLU) to increase complexity. Adam optimizer is used to optimize the overall model with learning rate of $3e-4$. Moreover, dropout regularization algorithm is employed to relieve the over-fitting

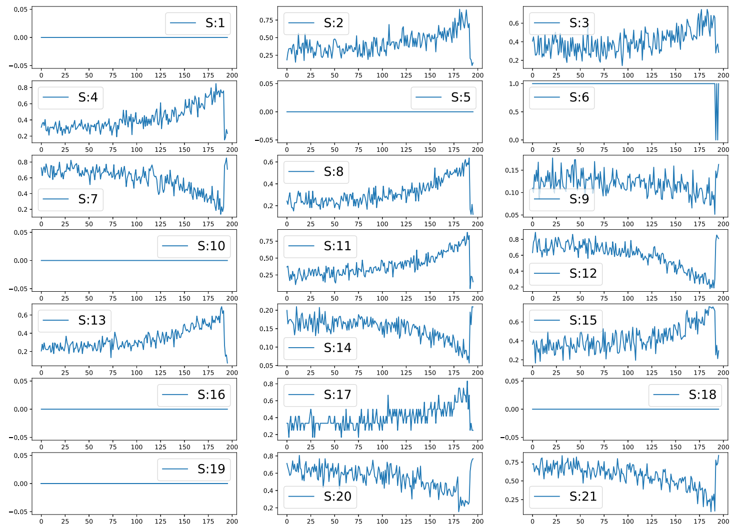


Fig. 6. Degradation trend of one engine across 21 sensors on FD001.

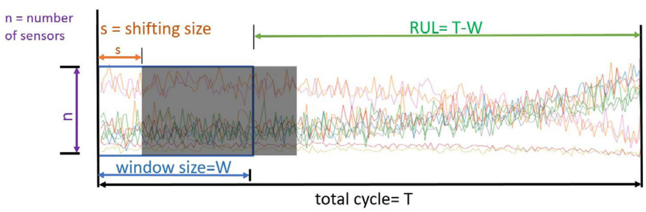


Fig. 7. Data segmentation using sliding window for RUL prediction.

problem. Table 2 summarizes all the hyper-parameters in our ATS2S model.

4.2.2. Performance metrics

We employ two standard metrics, namely root mean square error (RMSE) and the Score, to evaluate the performance of various methods for RUL prediction. RMSE is defined following Equation:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\widehat{RUL}_i - RUL_i\right)^2} \tag{17}$$

where $\widehat{RUL}_i$ and RUL_i are the predicted RUL and true RUL respectively, and N is the total number of samples. For machine prognosis and RUL prediction, late prediction of RUL (e.g., the predicted RUL is longer than the actual RUL) can lead to catastrophic consequences compared to early prediction. However, RMSE is not able to distinguish between early and late predictions. Hence, it requires an asymmetric evaluation function to give larger penalty for overestimation. To address this issue, a score metric has been developed, which was firstly proposed by the PHM community during the 2008 PHM data challenge competition [27]. Recently, many related research has adopted the score metric to evaluate the performance of a model on the RUL prediction task [12,29]. The score metric can be formalized as follows:

$$Score = \begin{cases} \sum_{i=1}^{N}\left(e^{-\frac{error_i}{13}} - 1\right), & \text{if } (error_i \le 0) \\ \sum_{i=1}^{N}\left(e^{\frac{error_i}{10}} - 1\right), & \text{if } (error_i > 0) \end{cases} \tag{18}$$

where $error_i = \left(\widehat{RUL}_i - RUL_i\right)$ is the difference between the predicted RUL $\widehat{RUL}_i$ and the true RUL_i.

Table 2
Hyper-parameters of proposed approach.

Hyper-parameters	Range
Batch size	{10}
Learning rate	{0.0003}
Training epochs	{10, 20}
Dropout rate	{0.2, 0.5}
Sequence length	{30}
α	{1}
Number of layers (Encoder and Decoder)	{1}
Number of hidden units (Encoder and Decoder)	{18, 32}
Number of layers (Attention Model)	{2}
Number of hidden units (Attention Model)	L1{30}, L2{9, 14}
Number of layers (RUL predictor)	{2, 3}
Number of hidden units (RUL predictor)	L1{18,32}, L2{18,1}, L3{1}

4.3. Comparison against state-of-the-arts

In this section, to comprehensively evaluate our proposed ATS2S method, we compare against 13 state-of-the-art methods, which can be classified into 6 categories as follows.

- Traditional machine learning (ML) methods. Three shallow models are employed in the comparison, including support vector machine (SVM) [29], random forest (RF) [29], and gradient boosting (GB) [29].
- CNN based methods. A 2D CNN network was used in [28] to predict the RUL for turbofan engines, while Li et al., used 1D CNN with multiple channels for RUL prediction [8].
- LSTM based methods. A standard LSTM network [11] and a bi-directional LSTM [12] were developed for RUL prediction. In [31], LSTM is augmented with a bootstrap algorithm to predict the RUL values.
- Ensemble methods. A deep belief network (DBN) is used together with ensemble techniques for the RUL prediction task [29].
- Hybrid CNN-LSTM based methods. Combination of CNN and LSTM models has been used for RUL prediction. CNN and LSTM can be cascaded in a sequential manner, e.g., CNN-LSTM [32]

Table 3
Comparison among various methods in terms of RMSE and Score.

Category	Method	RMSE				Score			
		FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
Traditional ML	SVM [28,29]	40.72	52.99	46.32	59.96	7703	316483	22542	141122
	RF [28,29]	17.91	29.59	20.27	31.12	480	70457	711	46568
	GB [28,29]	15.67	29.09	16.84	29.01	474	87280	577	17818
CNN methods	2D CNN [28]	18.45	30.29	19.82	29.16	1287	13570	1596	7886
	1D CNN [8]	**12.61**	22.36	12.64	23.31	274	10412	284	12466
LSTM methods	D-LSTM [11]	16.14	24.49	16.81	28.17	338	4450	852	5550
	LSTMBS [31]	14.89	26.86	15.11	27.11	481	7982	493	5200
	BLSTM [12]	N/A	25.11	N/A	26.61	N/A	4793	N/A	4971
Ensemble methods	MODBNE [29]	15.04	25.05	12.51	28.66	334	5585	422	6558
Encoder-decoder methods	BiLSTM-ED [26]	14.74	22.07	17.48	23.49	273	3099	574	3202
Hybrid CNN-LSTM methods	CNN-LSTM [32]	14.4	27.23	14.32	26.69	290	9869	316	6594
	BLCNN [33]	13.18	19.09	16.76	20.97	302	1558	381	3859
	HDNN [30]	13.02	15.24	12.22	18.16	245	1282	288	1527
Proposed	**ATS2S**	12.63	**14.65**	**11.44**	**16.66**	**243**	**876**	**263**	**1074**
IMP			**3.87%**	**6.4%**	**8.3%**	**0.82%**	**31.6%**	**8.7%**	**29.7%**

put CNN in the first stage, while BLCNN [33] reversed the order. In addition, HDNN [30] combined both the features from CNN and LSTM to generate the final predictions.

- Encoder-decoder based methods. BiLSTM-ED [26] first extracts health index and then estimates the health index curves using linear regression model. Finally it uses curve-similarity matching to estimate the RUL.

Table 3 shows the comparison among the above methods for RUL prediction. Note that the highest score in each column is in **bold**, while the second best score is underlined. We have used the same datasets and experimental settings of the compared approaches to ensure fair comparison. Hence, in Table 3, we have directly reported their published results.

Table 4
Comparison of the number of parameters between the proposed method and some state-of-the-arts.

Model	ATS2S	BLSTM	HDNN	BLCNN	BiLSTM-ED	D-LSTM
Number of model Parameters	**13628**	29053	54766	16196	345600	14865

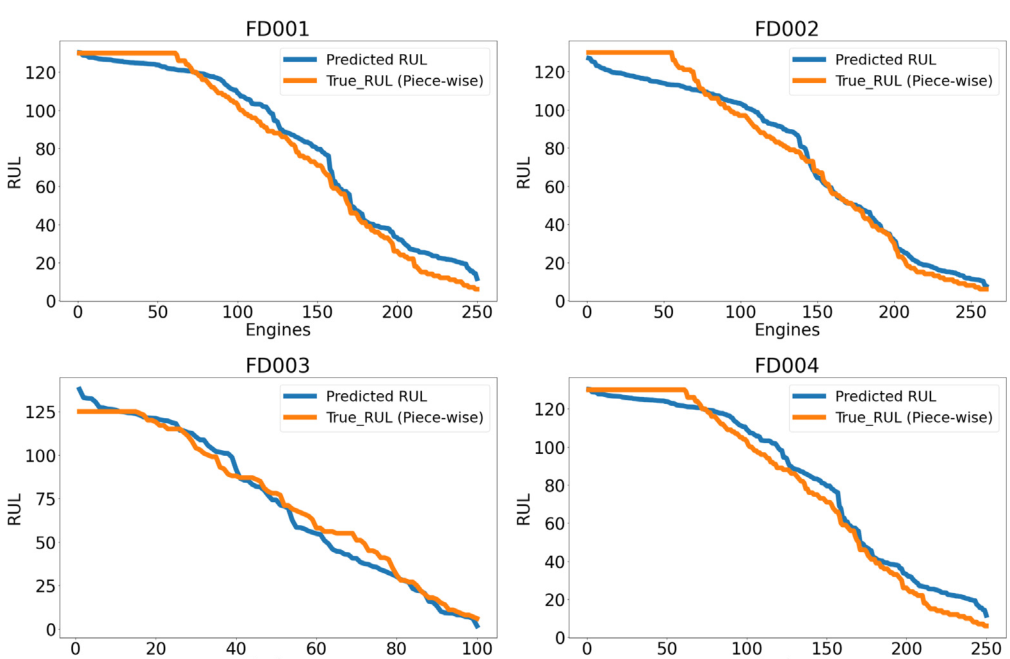


Fig. 8. Comparison between predicted RULs of the proposed model and the actual RULs. Each point represents a test engine and its corresponding RUL. The test engines are sorted in descending order based on their RUL values.

We can observe that our proposed ATS2S outperforms all the other methods consistently, except that it achieves a comparable RMSE with 1D CNN [8] on FD001 dataset. In particular, our ATS2S achieves significant improvement over the state-of-the-arts on FD002 and FD004, which are two complex datasets with multiple working conditions and thus indicate more practical scenarios. For example, ATS2S is able to achieve improvements over the second best performer on FD004 by 8.3% and 29.7% in terms of RMSE and Score, respectively. Such improvements on FD002 and FD004 demonstrate that ATS2S has clear advantages over the competing methods to handle the complex datasets. In addition, compared with the RMSE metric, our ATS2S achieves even better improvements in terms of the Score metric, indicating that we can better address the issue of late predictions.

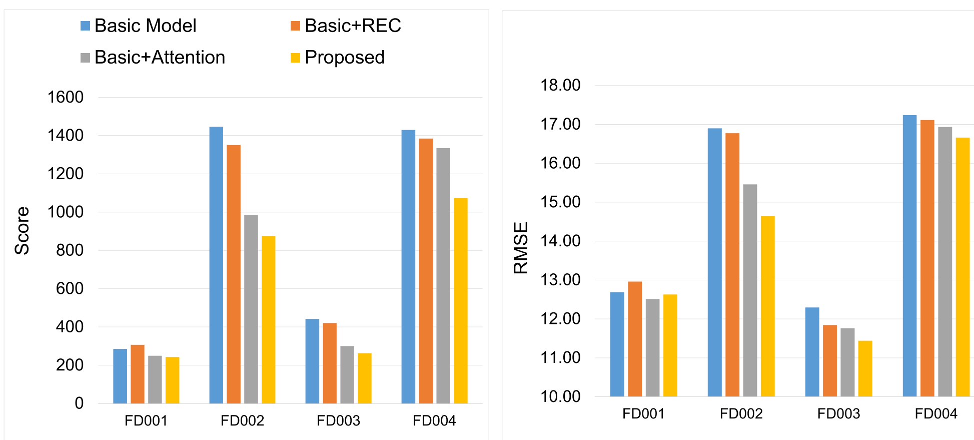
To further evaluate the complexity of our proposed model, we have compared it with some state-of-the-art methods for RUL prediction, i.e., BLSTM, HDNN, BLCNN, BiLSTM-ED, and D-LSTM, in terms of the number of model parameters. The results are shown in Table 4. It can be clearly seen that our model requires less number of parameters, which indicates its efficiency. In a nutshell, our ATS2S outperforms existing state-of-the-arts for RUL prediction in terms of RMSE and Score without requiring additional computational burden.

To further show the efficacy of our proposed approach, we have visualized the predicted RUL against the true RUL for test engines among four different datasets, as shown in Fig. .8. It worth noting that we have sorted the RUL values of test engines in descending order for clearer visualization. It can be clearly seen that our predicted RUL values are well aligned with the true RUL values for all the four datasets.

4.4. Model analysis

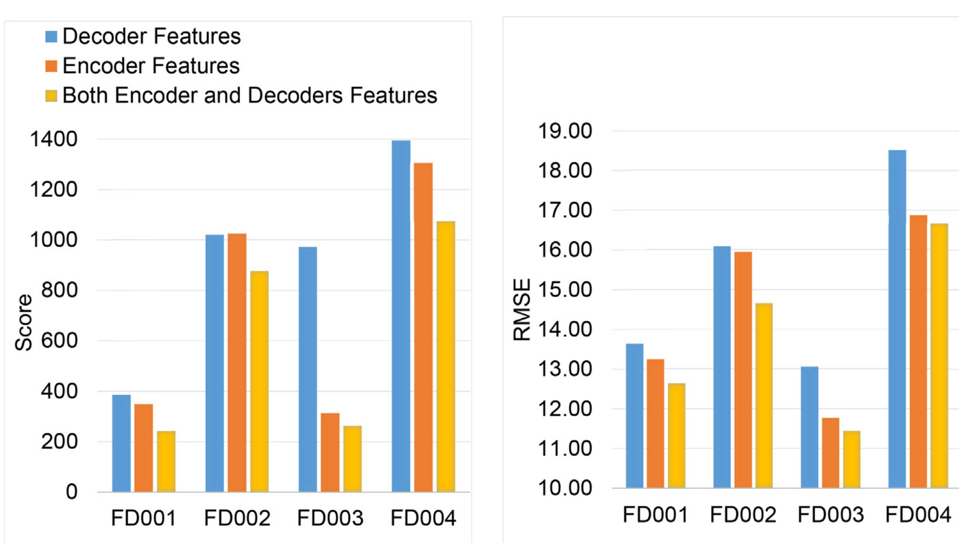
4.4.1. Ablation study

In this section, we disentangle the contribution of each part of the ATS2S model. In addition to the ATS2S model, we further derive three variants, namely (1) Basic sequence to sequence model without reconstruction or attention, (2) Basic model with reconstruction, (3) Basic model with attention. Fig. 9 shows the comparison between these 3 variants and the proposed ATS2S model. Based



(a) (b)

Fig. 9. Ablation study for the proposed ATS2S method.



(a) (b)

Fig. 10. Study of feature importance of the proposed method.

on the comparison shown in Fig. 9, we can further draw two conclusions.

Firstly, our proposed ATS2S model with both attention mechanism and reconstruction architecture achieves the best performance over 4 datasets in terms of both metrics, showing that it is indeed more effective for RUL prediction than basic sequence to sequence model. This demonstrates that learning from most relevant information from long sensor signals by attention mechanism (not just focusing on the latest information), as well as enabling predictive power and capturing temporal dependencies by reconstruction architecture, are critical for improving RUL prediction.

Secondly, the model with attention mechanism outperforms the model with reconstruction architecture, indicating that attention mechanism has larger impact than reconstruction task in our ATS2S model. Without the attention mechanism, we squash the whole input sequence into a single hidden vector (i.e., the *last* hidden state of the encoder). Instead, attention mechanism can consider all the hidden states with different weights and help to learn better comprehensive *dual-latent feature representation* from both encoder and decoder for RUL prediction.

4.4.2. Feature importance analysis

As shown in Fig. 3, we use the *dual-latent feature representation* to integrate features from both encoder and decoder for RUL prediction. To study the importance of the features used in our ATS2S, we conduct experiments using three different feature sets, namely, encoder features (i.e., encoder hidden states), decoder features and integrated features, i.e., encoder-decoder features (dual-latent feature representation). Fig. 10 shows the detailed model

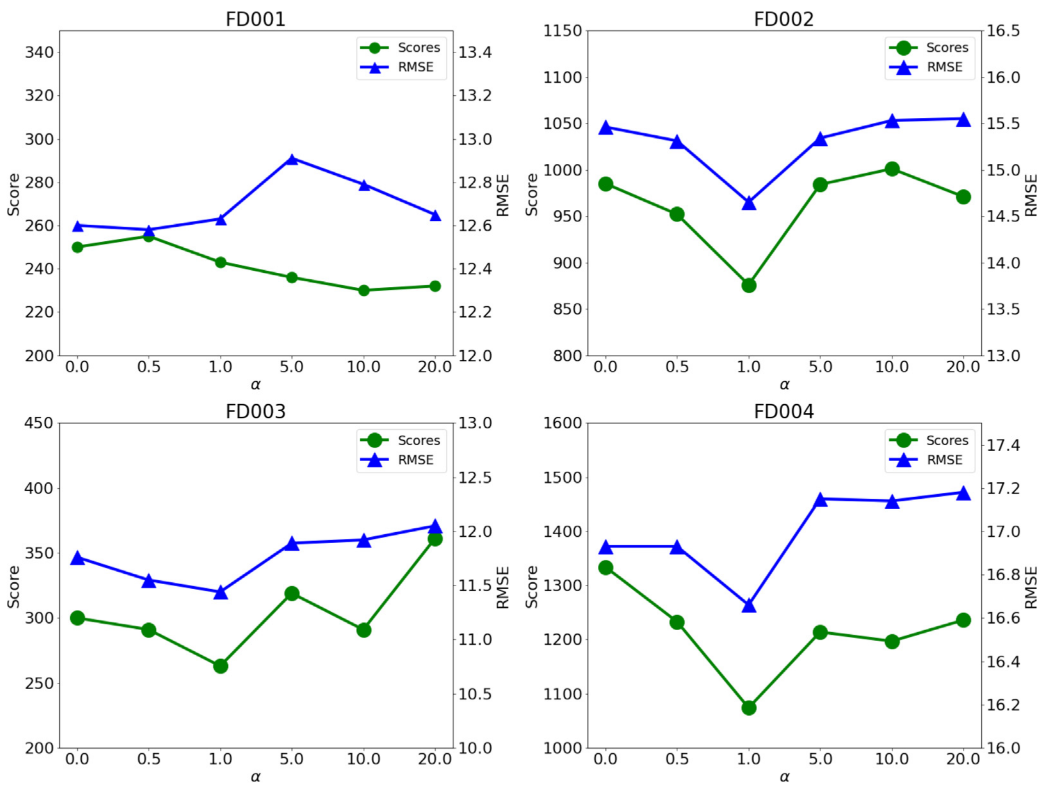


Fig. 11. Sensitivity analysis of reconstruction weight.

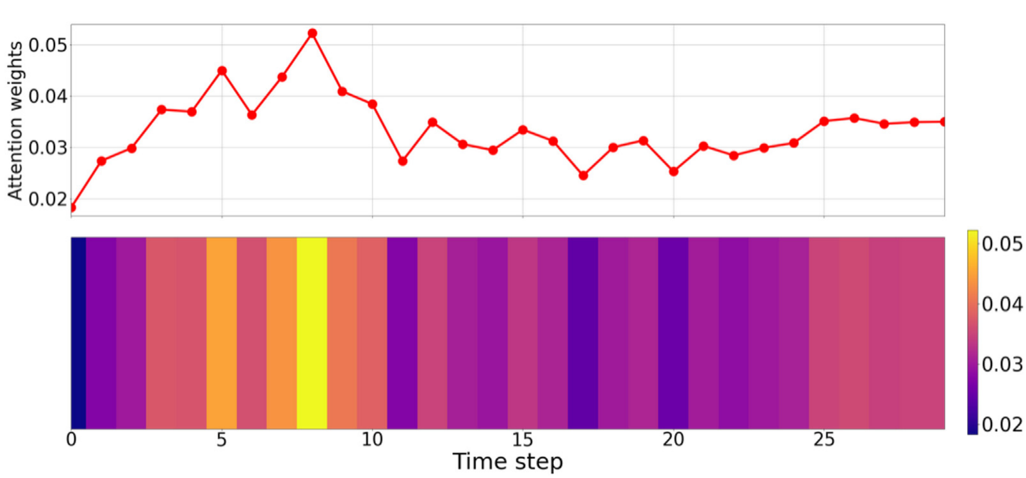


Fig. 12. Illustration of attention weights for a randomly selected sample.

performance with three different feature sets. We can observe that dual-latent feature representation achieves the best performance over all four data subsets consistently, indicating the importance of a comprehensive representation with rich semantics from both encoder and decoder features.

4.4.3. Sensitivity analysis

As shown in Eq. (15), the parameter α controls the contribution of reconstruction loss in the final joint loss. In this section, we perform the sensitivity analysis for this parameter α. Fig. 11 shows the performance of ATS2S model across four datasets with different values for α. Overall, it can be clearly observed that equal contribution from both reconstruction and prediction loss (i.e., $\alpha = 1$) achieves the best performance, demonstrating that both of them are critical for accurate RUL predictions.

4.4.4. Attention weights

To demonstrate our model capability on capturing long-term dependencies, we have visualized the attention weights among different time steps. Fig. 12 shows the attention weights of a randomly selected sample at one decoding time step. It can be found that the model pays more attention to previous time steps. This indicates that the attention mechanism helps the model to capture long-term dependencies of the data.

5. Conclusion

In this work, we presented a novel attention-based sequence to sequence model ATS2S to accurately predict equipment RUL, which has huge impact for many real-world applications. In particular, we designed a novel framework that learns to reconstruct the next sequence and predict the RUL labels concurrently. In addition, we showed our attention mechanism can better capture all the relevant historical information from long sensor sequences than standard LSTM approach which focuses on the latest information only. Finally, our *dual-latent feature representation* which integrate both the encoder and decoder features is very effective for RUL prediction. Our extensive experimental results demonstrate that our proposed ATS2S significantly outperforms 13 state-of-the-arts for RUL prediction across 4 benchmark datasets consistently.

One limitation of our work, as well as existing studies, is the assumption that a large amount of labeled data are always available. This assumption may not be feasible for many real applications. As annotating time series data can be very labor intensive even for experts. Hence, as a future work, we are aiming to rely on the unsupervised input forecasting or another self-supervised learning task to learn feature representation in an unsupervised manner [34]. After that, we can finetune the model with few labeled data to learn the corresponding prognostic task. Reducing the required amount labels can be of great importance towards more practical data-driven RUL prediction.

CRediT authorship contribution statement

Mohamed Ragab: Methodology, Writing - original draft. **Zhenghua Chen:** Supervision. **Min Wu:** Supervision, Writing - review & editing. **Chee-Keong Kwoh:** Supervision. **Ruqiang Yan:** Writing - review & editing. **Xiaoli Li:** 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] C. Liu, L. Zhang, J. Niu, R. Yao, C. Wu, Intelligent prognostics of machining tools based on adaptive variational mode decomposition and deep learning method with attention mechanism, Neurocomputing 417 (2020) 239–254.

[2] J.I. Aizpurua, V.M. Catterson, I.F. Abdulhadi, M.S. Garcia, A model-based hybrid approach for circuit breaker prognostics encompassing dynamic reliability and uncertainty, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (9) (2018) 1637–1648.

[3] A. Elsheikh, S. Yacout, M.-S. Ouali, Bidirectional handshaking lstm for remaining useful life prediction, Neurocomputing 323 (2019) 148–156.

[4] J.B. Ali, B. Chebel-Morello, L. Saidi, S. Malinowski, F. Fnaiech, Accurate bearing remaining useful life prediction based on weibull distribution and artificial neural network, Mechanical Systems and Signal Processing 56 (2015) 150–172.

[5] Z. Liu, Y. Cheng, P. Wang, Y. Yu, Y. Long, A method for remaining useful life prediction of crystal oscillators using the bayesian approach and extreme learning machine under uncertainty, Neurocomputing 305 (2018) 27–38.

[6] Z. Xue, Y. Zhang, C. Cheng, G. Ma, Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression, Neurocomputing 376 (2020) 95–102.

[7] X. Peng, C. Zhang, Y. Yu, Y. Zhou, Battery remaining useful life prediction algorithm based on support vector regression and unscented particle filter, in: 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, 2016, pp. 1–6..

[8] X. Li, Q. Ding, J.-Q. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, Reliability Engineering & System Safety 172 (2018) 1–11.

[9] B. Yang, R. Liu, E. Zio, Remaining useful life prediction based on a double-convolutional neural network architecture, IEEE Transactions on Industrial Electronics 66 (12) (2019) 9521–9530.

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

[11] S. Zheng, K. Ristovski, A. Farahat, C. Gupta, Long short-term memory network for remaining useful life estimation, in: 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), 2017, pp. 88–95.

[12] C.-G. Huang, H.-Z. Huang, Y.-F. Li, A bidirectional lstm prognostics method under multiple operational conditions, IEEE Transactions on Industrial Electronics 66 (11) (2019) 8792–8802.

[13] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, X. Li, Machine remaining useful life prediction via an attention based deep learning approach, IEEE Transactions on Industrial Electronics (2020).

[14] H. Miao, B. Li, C. Sun, J. Liu, Joint learning of degradation assessment and rul prediction for aeroengines via dual-task deep lstm networks, IEEE Transactions on Industrial Informatics 15 (9) (2019) 5023–5032.

[15] J.-Y. Wu, M. Wu, Z. Chen, X.-L. Li, R. Yan, Degradation-aware remaining useful life prediction with lstm autoencoder, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–10.

[16] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder–decoder approaches, in: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014, pp. 103–111..

[17] T. Trinh, A. Dai, T. Luong, Q. Le, Learning longer-term dependencies in rnns with auxiliary losses, in: ICML 2018: Thirty-fifth International Conference on Machine Learning, 2018, pp. 4965–4974.

[18] L. Le, A. Patterson, M. White, Supervised autoencoders: Improving generalization performance with unsupervised regularizers, Advances in Neural Information Processing Systems 31 (2018) 107–117.

[19] Y. Bengio, I. Goodfellow, A. Courville, Deep learning, Citeseer 1 (2017).

[20] L. Guo, Y. Lei, N. Li, T. Yan, N. Li, Machinery health indicator construction based on convolutional neural networks considering trend burr, Neurocomputing 292 (2018) 142–150.

[21] Y. Wu, M. Yuan, S. Dong, L. Lin, Y. Liu, Remaining useful life estimation of engineered systems using vanilla lstm neural networks, Neurocomputing 275 (2018) 167–179.

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

[23] D. Zhang, E. Stewart, J. Ye, M. Entezami, C. Roberts, Roller bearing degradation assessment based on a deep mlp convolution neural network considering outlier regions, IEEE Transactions on Instrumentation and Measurement (2019), 1–1.

[24] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.

[25] P. Malhotra, V. Tv, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder, arXiv preprint arXiv:1608.06154, 2016..

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

[27] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: 2008 International Conference on Prognostics and Health Management, IEEE, 2008, pp. 1–9..

[28] G.S. Babu, P. Zhao, X.-L. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, in: International Conference on Database Systems for Advanced Applications, 2016, pp. 214–228.
[29] C. Zhang, P. Lim, A.K. Qin, K.C. Tan, Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics, IEEE Transactions on Neural Networks and Learning Systems 28 (10) (2016) 2306–2318.
[30] A. Al-Dulaimi, S. Zabihi, A. Asif, A. Mohammadi, A multimodal and hybrid deep neural network model for remaining useful life estimation, Computers in Industry 108 (2019) 186–196.
[31] Y. Liao, L. Zhang, C. Liu, "Uncertainty prediction of remaining useful life using long short-term memory network based on bootstrap method," in, in: 208 IEEE International Conference on Prognostics and Health Management (ICPHM), 2018, pp. 1–8.
[32] Z. Wu, S. Yu, X. Zhu, Y. Ji, M. Pecht, A weighted deep domain adaptation method for industrial fault prognostics according to prior distribution of complex working conditions, IEEE Access 7 (2019) 139802–139814..
[33] H. Liu, Z. Liu, W. Jia, X. Lin, A novel deep learning-based encoder-decoder model for remaining useful life prediction, in: 2019 International Joint Conference on Neural Networks (IJCNN), 2019.
[34] L. Jing, Y. Tian, Self-supervised visual feature learning with deep neural networks: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).

Mohamed Ragab received the B.Sc. degree (First Class Hons.) and M.Sc. degree from the Department of Electrical Engineering, Aswan University, in 2014 and 2017, respectively. He is currently pursuing his Ph.D. degree from the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. Concurrently, he is with Machine Intellection (MI) department at the Institute of Infocomm Research (I2R), A*STAR. His research interests include deep learning, transfer learning, and intelligent fault diagnosis and prognosis.

Zhenghua Chen received the B.Eng. degree in mechatronics engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2011, and Ph.D. degree in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2017. He has been working at NTU as a research fellow. Currently, he is a scientist at Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. He has won several competitive awards, such as First Place Winner for CVPR 2021 UG2+ Challenge, A*STAR Career Development Award, First Runner-Up Award for Grand Challenge at IEEE VCIP 2020, Finalist Academic Paper Award at IEEE ICPHM 2020, etc. He serves as Associate Editor for Elsevier Neurocomputing and Guest Editor for IEEE Transactions on Emerging Topics in Computational Intelligence. He is currently the Vice Chair of IEEE Sensors Council Singapore Chapter and IEEE Senior Member. His research interests include smart sensing, data analytics, machine learning, transfer learning and related applications.

Min Wu is currently a senior scientist in Data Analytics Department, Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. He received his Ph.D. degree in Computer Science from Nanyang Technological University (NTU), Singapore, in 2011 and B.S. degree in Computer Science from University of Science and Technology of China (USTC) in 2006. He received the best paper awards in InCoB 2016 and DASFAA 2015. He also won the IJCAI competition on repeated buyers prediction in 2015. His current research interests include machine learning, data mining and bioinformatics.

Chee-Keong Kwoh received the bachelor's degree in electrical engineering (first class) and the master's degree in industrial system engineering from the National University of Singapore, Singapore, in 1987 and 1991, respectively. He received the Ph.D. degree from the Imperial College of Science, Technology, and Medicine, University of London, in 1995. He has been with the School of Computer Engineering, Nanyang Technological University (NTU), since 1993. He is the Deputy Executive Director of PaCE at NTU. His research interests include data mining, soft computing and graph-based inference; application areas include bioinformatics and engineering. He has done significant research work in his research areas and has published many quality international conferences and journal papers. He is a member of the Association for Medical and Bio-Informatics, Imperial College Alumni Association of Singapore. He has provided many services to professional bodies in Singapore and was conferred the Public Service Medal by the president of Singapore in 2008.

Ruqiang Yan (M'07-SM'11) received the M.S. degree in precision instrument and machinery from the University of Science and Technology of China, Hefei, China, in 2002, and the Ph.D. degree in mechanical engineering from the University of Massachusetts Amherst, MA, USA, in 2007.
From 2009 to 2018, he was a Professor of the School of Instrument Science and Engineering at the Southeast University, Nanjing, China. He joined the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China, in 2018. His research interests include data analytics, machine learning, and energy-efficient sensing and sensor networks for the condition monitoring and health diagnosis of large-scale, complex, dynamical systems. He holds 28 patents, published two books and over 200 papers in technical journals and conference proceedings.
Dr. Yan is a Fellow of ASME (2019). His honors and awards include IEEE Instrumentation and Measurement Society Technical Award (2019), the New Century Excellent Talents in University Award from the Ministry of Education in China (2009), and multiple Best Paper Awards. He is an Associate Editor-in-Chief for the IEEE Transactions on Instrumentation and Measurement and Associate Editor for the IEEE Systems Journal and IEEE Sensors Journal.

Xiaoli Li is currently a principal scientist at the Institute for Infocomm Research, A*STAR, Singapore. He also holds adjunct professor positions at Nanyang Technological University. His research interests include data mining, machine learning, AI, and bioinformatics. He has been serving as a (senior) PC member/workshop chair/session chair in leading data mining and AI related conferences (including KDD, ICDM, SDM, PKDD/ECML, WWW, IJCAI, AAAI, ACL and CIKM). Xiaoli has published more than 200 high quality papers and won numerous best paper/benchmark competition awards.