



# Multitask learning for health condition identification and remaining useful life prediction: deep convolutional neural network approach

Tae San Kim<sup>1</sup> · So Young Sohn<sup>1</sup>

Received: 21 October 2019 / Accepted: 12 July 2020 / Published online: 24 July 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Predicting remaining useful life (RUL) is crucial for system maintenance. Condition monitoring makes not only degradation data available for RUL estimation but also categorized health status data for health state identification. However, RUL prediction has been treated as an independent process in most cases even though potential relevance exists with health status detection process. In this paper, we propose a convolution neural network based multi-task learning method to reflect the relatedness of RUL estimation with health status detection process. The proposed method applied to the C-MAPSS dataset for aero-engine unit prognostics supported superior performances to existing baseline models.

**Keywords** Prognostics and health management · Remaining useful life · Multi-task learning · Convolution neural network

## Introduction

Engineering maintenance and prognostics are crucial in most industries. Traditionally, breakdown corrective maintenance and scheduled preventive maintenance have been key strategies in maintenance processes (Azadeh et al. 2015). However, as physical systems have become more complex, existing methods no longer meet the industrial requirements for efficiency and reliability (Stringer et al. 2012). Thus, intelligent prognostic and health management (PHM) technologies have been developed to efficiently perform such diagnoses and management by monitoring sensor data (Vogl et al. 2019). The goal of intelligent PHM is to reduce maintenance costs and improve reliability by monitoring the system state (Xia et al. 2018).

With sufficient condition monitoring (CM) data, the PHM process can be utilized to estimate the remaining useful life (RUL) (Babu et al. 2016), and with monitored degradation data, the future performance of the system can be predicted in terms of the RUL. Efficient and accurate prognostic technologies help to make maintenance decisions in advance to avoid failures; thus, RUL estimation based on given

historical sensor measurements has become one of the most important PHM activities.

The sensor measurements can also be used to other PHM processes such as identifying health conditions, anomaly detection or system diagnosis (Lei et al. 2008; Niu and Yang 2010). The integrated PHM system has been introduced to accomplish various PHM processes based on collected sensor data (Khan and Yairi 2018). However, in most cases, each task has been regarded as an independent task. Individual model is constructed for each task, thus the system complexity has become a potential risk to simultaneously consider other PHM process with RUL prediction (Khan and Yairi 2018). Moreover, there is a potential relation between other PHM processes and RUL prediction (Zaidan et al. 2015). Therefore, performing RUL predictions simultaneously with other process can improve the performance of the PHM system while reducing its complexity.

In this paper, we suggest a data-driven approach to simultaneously accomplish health condition identification and RUL estimation of a complex system. The health condition identification is to measure the current system conditions. If the current state can be specified, operable and committable time could be measured, and it can be helpful to evaluate the remaining time of a system from the current state (Patton et al. 2013). Thus, a multi-task learning (MTL) framework is suggested to infer the RUL of a current system, reflecting its correspondence to present health conditions.

✉ So Young Sohn  
sohns@yonsei.ac.kr

<sup>1</sup> Department of Industrial Engineering, Yonsei University,  
134 Shinchon-dong, Seoul 120-749, Republic of Korea

MTL has been used to improve performance by utilizing the domain-specific information involved in the related tasks (Baxter 1997). The simultaneous learning of multiple related tasks has been shown, both theoretically and empirically, to improve performance as compared to the independent learning for each task (Girshick 2015). Traditional MTL based on a neural network structure is composed of two partial networks: a shared network and a task-specific network. A shared network uses the same hidden neurons over different tasks, while task-specific networks are made independent to separate the information of individual tasks. On connecting these networks, shared representation is developed concurrently across tasks. Thus, MTL can reduce the risk of overfitting and improve the single task performance of the original task.

In this study, one shared network and two task-specific networks are organized: the former to capture interdependencies between RUL estimation and health condition identification, and the latter to perform each task. By capturing the interdependencies of related integrated health management processes, the proposed MTL framework is expected to improve prognostic accuracy. In the proposed MTL model, convolution neural network (CNN) layers are employed as the basic structure of shared layers to extract global features from complex signals. Raw sensor data is input directly to the CNN-based MTL structure, and a multi-variate 1-D filter is utilized for feature extraction, reflecting the time varying relationship amongst multi-sensor data. The proposed framework was applied to the C-MAPSS dataset, which has a RUL label, and the system's degradation information per cycle was recorded (Saxena et al. 2008). Our MTL framework is expected to contribute to managing and maintaining system conditions by improving RUL prediction performance.

The composition of this paper is as follows. Second section reviews previous studies on health management. Third and fourth sections explain the data and methodology used in this study, and fifth shows the results of empirical analysis. Finally, sixth section presents the conclusions as well as the contributions of the study.

## Literature review

### Data-driven approaches for the estimation of remaining useful life

In recent years, data-driven approaches have been used to discover the relationship between monitored condition data and RUL (Lee et al. 2014). One of the advantages of data-driven approaches to the RUL estimation in CBM is that extensive prior knowledge of the physical system is not required. Moreover, they can reflect intrinsic correlations and causalities of sensor measurements, thus proving

their good prediction performance. Various approaches using data-driven algorithms aim to utilize these benefits. Riad et al. (2010) modeled a multi-layer perceptron (MLP) approach to estimate the RUL from an aircraft turbofan engine, demonstrating that the prediction performance of MLP is superior to that of a simple linear regression model. In another approach, Tian (2012) proposed an artificial neural network (ANN) based method to estimate the RUL of a physical instrument using monitored data.

Recently, deep neural network-based techniques, also known as deep learning approaches, have been applied to RUL estimation, as they are known for their ability to handle complex and high-dimensional systems (LeCun et al. 2015). Malhi et al. (2011) suggested an RNN-based approach for long-term prognostics of RUL. This model captured the long-term dependency of sequential vibration signals and could thus measure the RUL of rolling bearing products from real-time signals. To improve the drawbacks of traditional RNN such as gradient vanishing or exploding, Yuan et al. (2016) proposed a long-short term memory (LSTM) based framework, showing that the performance of LSTM was superior to RNN in predicting the RUL of an aero-engine. Several other studies have suggested RNN and LSTM based approaches for RUL estimation because the recurrent network structure is more effective than time-series data (Malhotra et al. 2016; Lim et al. 2016; Gugulothu et al. 2017; Guo et al. 2017; Yoon et al. 2017; Wu et al. 2018).

Because recurrent networks have complex computational burdens, CNNs, which are designed to extract features through weight sharing filters, were also utilized for RUL prediction. Babu et al. (2016) designed a two-dimensional (2D) CNN model to consider multivariate relationships of the sensor data; sequential monitoring data was used as the raw input data, and temporal patterns were also captured through the convolution layer. Li et al. (2018b) also designed a similar CNN approach, but by using a univariate 1-D CNN filter to reflect the sequential patterns of each sensor; the independent temporal features were extracted for each sensor and then concatenated in the output layer to estimate the RUL. However, none of these approaches utilized MTL.

### Multi-task learning

MTL is an approach which simultaneously handles multiple tasks (Caruana 1997). One of the purposes of MTL is to improve single task generalization by reflecting the domain-specific information in the related tasks, which are also called auxiliary tasks (Ando and Zhang 2005). An additional aim is to combine the common knowledge of all tasks and simultaneously enhance their performance (Pan and Yang 2010). To achieve MTL objectives, it is necessary to learn related tasks; multiple tasks can be related in various ways, such as if the functions in each task work similarly in some

manner or share a common aim in corresponding domains (Argyriou et al. 2007).

Implicit data augmentation, attention focusing, eavesdropping, and representation bias are some of the properties that allow MTL to perform better than independent learning (Caruana 1997; Ruder 2017; Zhang and Yang 2017). Essentially, deep learning utilizes a large network with numerous parameters, which requires a large amount of training data. One of the drawbacks of deep learning is that it is rendered ineffective by insufficient data samples (Marcus 2018). However, MTL utilizes the datasets from all tasks; it effectively increases the training sample size and extracts a more general representation from multiple tasks with different noise patterns (Ruder 2017). Additionally, it mainly updates model parameters from focused features that are commonly important to all the tasks, thus providing additional information on the relevance of features to other tasks. MTL also allows for most features to be effectively trained by different tasks (Caruana 1997); even if one task can only train certain features, the other features can be learned through other tasks. Finally, the model is trained using general representations from all related tasks, so a trained model can accept new related tasks in the same domain (Ruder 2017). This helps to improve the model's adaptability to new tasks.

## Problem settings

### C-MAPSS data

The commercial modular aero-propulsion system simulation (C-MAPSS) dataset is derived from a turbofan engine simulation program developed by NASA (Saxena et al. 2008). The simulation program monitored the degradation of multiple aero-engines and collected multi-variate time-series sensor measurements from 21 sensors. The initial wear state and manufacturing variation of engine units are unknown but assumed to be healthy. The C-MAPSS dataset consists of four sub-datasets depending on whether the experimental setting has single or different operational conditions for each cycle and what kind of faults (i.e. FAN degradation or HPC degradation) are considered in the engine. Thus, each sub-dataset has run-to-failure sensor records collected under different operating conditions and fault modes, respectively.

Each sub-dataset includes a training set and a test set. In the training set, system degradation is observed as time progresses, and the last recorded data of each engine unit is considered as the fault declaration and the termination of the experiment. Multiple sensor measurements of each cycle are used as inputs for training samples, and all of them are used to predict the RUL. Records in the test set are pruned to stop before failure, and the aim of the C-MAPSS dataset is to predict the true RUL of the last record in the test set.

This implies that only one sensor measurement of the last record per engine unit is considered as the testing sample, and the actual RUL value of it is used for verification. The C-MAPSS dataset information is shown in Table 1.

### Data preprocessing

Although multi-variate temporal data is included in the C-MAPSS dataset, some sensors have constant values which are not useful for the RUL estimation. Constant sensor values cannot be used significantly to predict the target RUL value. Therefore, these sensor values are eliminated, and only 14 out of 21 sensors are considered for the raw input data (sensor 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20 and 21).

For each of the sub-datasets in C-MAPSS, experimental scenarios can differ according to the operating conditions. FD001 and FD003 have single operating conditions, while FD002 and FD004 have six different operating conditions. Because of the conditional relationship between operating conditions and the monitored values, the actual meaning of sensor values can differ according to the operating conditions (Rabiei and Modarres 2013). Thus, condition-wise normalization is performed on the sensor data of FD002 and FD004 to adjust the scale of the sensor values in compliance with the operating conditions. The normalization has an observable effect on the sensor values in the RUL estimation. The normalization equation is as follows:

$$Norm(x^{(o,d)}) = \frac{x^{(o,d)} - \mu^{(o,d)}}{\sigma^{(o,d)}} \quad (1)$$

where  $x^{(o,d)}$  denotes  $d$ -th sensor values of the  $o$ -th operating condition, and  $\mu^{(o,d)}$  and  $\sigma^{(o,d)}$  represent the mean and standard deviations of the sensor values for each operating condition. On the other hand, FD001 and FD003 have only one operating condition; thus, their normalization is performed by each sensor.

The normalized sensor values are used to estimate both the actual RUL value and the present health condition. Because the asset has a very healthy initial state, its RUL does not decrease at an early stage. Therefore, to obtain the RUL value in this study, a healthy state is maintained during

**Table 1** Information of C-MAPSS dataset

Description	FD001	FD002	FD003	FD004
Operating conditions	1	6	1	6
Fault modes	1	1	2	2
# of engine units for training	100	260	100	249
# of engine units for testing	100	259	100	248
# of training samples	20,631	53,759	24,720	61,249
# of testing samples	100	260	100	248

the early stages by using a piece-wise linear degradation model that limits the maximum RUL value (Heimes 2008). Because the degradation is identified after the initial healthy period, this approach prevents overestimation of the RUL (Babu et al. 2016). We set the maximum RUL value to 125 as provided in Malhotra et al. (2016) and Li et al. (2018b). Meanwhile, there is no information related to the present health condition in the C-MAPPS dataset. Thus, three health condition labels are assumed: the first 30% of the samples are healthy, the last 30% are unhealthy and the remaining 40% are deteriorating (Yuan et al. 2016). In this manner, our dataset not only holds the true RUL value, but also the present health condition in every cycle. All functions in the suggested MTL are simultaneously learned from both targets.

## Methodology

### Architecture

In this study, we propose a deep architecture to effectively predict the RUL by simultaneously learning two tasks: RUL estimation and health condition prediction. To simultaneously estimate the RUL value and present health condition, an MTL model is constructed. The proposed model captures the interdependencies of both tasks offering supportive knowledge to the RUL estimation task to obtain improved prognostic performance.

Because sequential sensor measurements have temporal patterns, they can exhibit powerful performance in deep architecture-based MTL on recurrent networks. However, recurrent network-based approaches require relatively high computational time because, unlike CNN, the recurrent structure does not allow for increased parallelization (Vaswani et al. 2017). While MTL has the drawback of high computational complexity because it requires huge networks to reflect the combined information from multiple tasks, CNN effectively reduces computational time by utilizing parallelization. Therefore, CNN architecture was applied as the basic deep-learning structure of our proposed MTL.

Instead of using a recurrent architecture, a multi-variate 1D filter was applied to the CNN architecture to extract a better feature representation in terms of sequential multi-sensor measurements. The 1D filter can capture the underlying relationship between multiple sensors, and the temporal patterns of those sensors can be reflected by sliding the filter. The last feature map extracted by this filter was utilized as the input vector of separate networks for RUL estimation and health condition identification.

During the training process, the true RUL value and health condition were set as the learning objectives of our model, and all functions were updated by both complementary targets. In the testing process, the prediction values of both tasks

were acquired, and the performance of our multi-task model was evaluated by comparing the RUL performance with other baselines. In the following sections, each process is described in detail.

### Proposed multi-task network structure

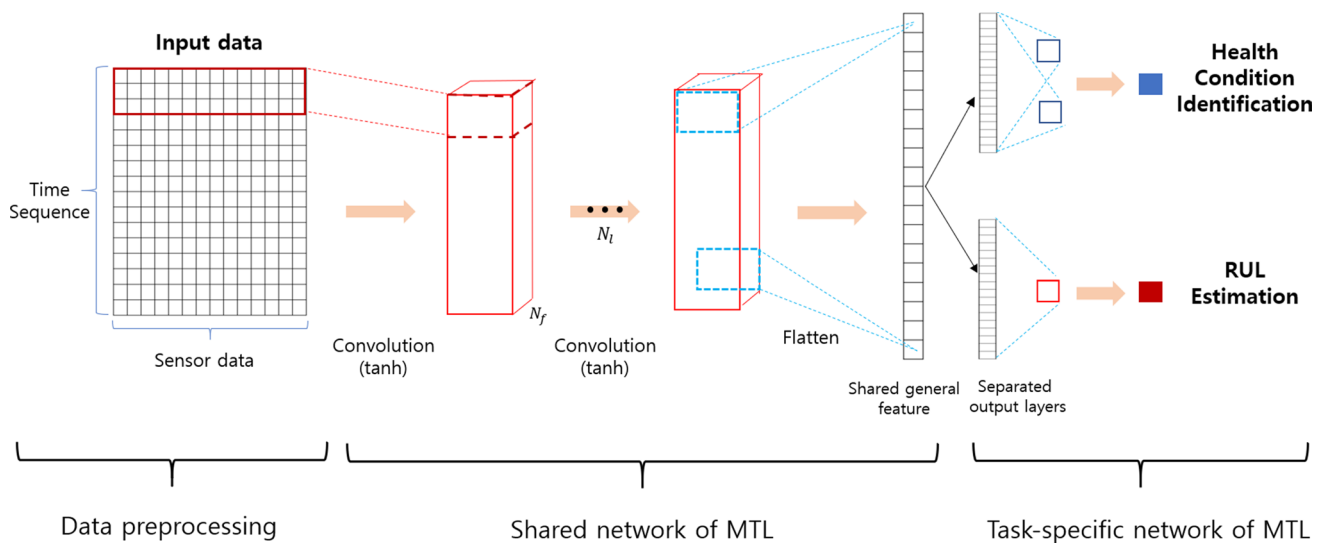
MTL has two layers: shared layers and task-specific output layers. Usually, the bottom layers of the network are for sharing domain-specific information, whereas the top layers are for separating task-specific information. Shared layers share parameters of hidden layers, while task-specific output layers retain their own parameters. The former is to extract the general features of corresponding domains, while the latter is to focus on performing individual tasks. In this study, the shared layers are composed of a fully convolutional CNN layer and two separated fully connected layers (FCLs) of CNN and are regarded as the task-specific output layers, as shown in Fig. 1.

All parameters in the multi-tasking network were jointly optimized to simultaneously improve task generalization and individual model performance. The convolution parameters, which are shared network parameters, were updated to reflect the common knowledge of CBM. Meanwhile, the FCL parameters, which are task-specific network parameters, learned independent information corresponding to the RUL estimation and health condition identification, respectively. The loss function of each task affects both the individual task and the shared parameters.

The target RUL variable is an interval type, but the health condition is an ordinal type target with three levels, represented by 1 for healthy, 2 for deteriorating, and 3 for unhealthy ( $k = 1, 2, 3$ ). To directly reflect this ordinal target into neural networks, we utilized 2 binary classifiers (Niu et al. 2016) with two dummy variables that represent the target vectors of healthy, deteriorating and unhealthy with  $\{0,0\}$ ,  $\{1,0\}$  and  $\{1,1\}$ , respectively. The neural networks were trained to predict each binary target (Cheng et al. 2008). In the testing process, unseen samples could be ranked by scanning from the first to the second classifier. Each sample had two probabilities; if they were higher than a certain threshold, they were set to 1, or else to 0. The threshold value was set to 0.5. The sample ranks were judged based on the number of 1s in all classifiers. Estimating RUL and classifying health condition respectively yield regression loss and ordinal regression loss, as follows,

$$L_{cls}(\hat{H}_i^r, H_i^r) = \frac{1}{N} \sum_{i=1}^N \sum_{r=1}^2 \{H_i^r \log \hat{H}_i^r + (1 - H_i^r) \log(1 - \hat{H}_i^r)\}$$

$$L_{reg}(\hat{R}_i, R_i) = \frac{1}{N} \sqrt{\sum_{i=1}^N (\hat{R}_i - R_i)^2}$$
(2)



**Fig. 1** Proposed multi-task network structure, where  $N_t$  is the number of convolution layers and  $N_f$  is the number of kernel filter for each convolution layer

where  $N$  represents the number of data samples, and  $L_{reg}$  is the root mean squared error (RMSE) loss between target RUL  $R$  and predicted RUL  $\hat{R}$ , while  $L_{cls}$  is the combined sigmoid loss information of two binary classifiers where  $H^r$  is the true health condition and  $\hat{H}^r$  is the predicted health condition for each sample. For multi-tasking learning, the loss function of each task affected both corresponding task layer and shared layer. Thus, parameters of task-specific layer were updated to improve performance of each task, while parameters of shared layer were changed to extract good general feature of CBM. Our proposed entire model is learned by the weighted sum of these two losses. Total loss is equally affected by both losses at a ratio of 1:1.

**Multi-variate one-dimensional filter**

Although the CNN was originally designed for image processing, it has become one of the most popular and successful deep learning methodologies in many research fields. In RUL estimation, CNN has been used in various studies and shown excellent prediction results (Babu et al. 2016; Li et al. 2018b). The CNN identifies the intrinsic patterns of input data through localized filters and spatial pooling (Krizhevsky et al. 2012). The convolution layer extracts abstract spatial features of the signals, while the pooling layer obtains significant values in local features.

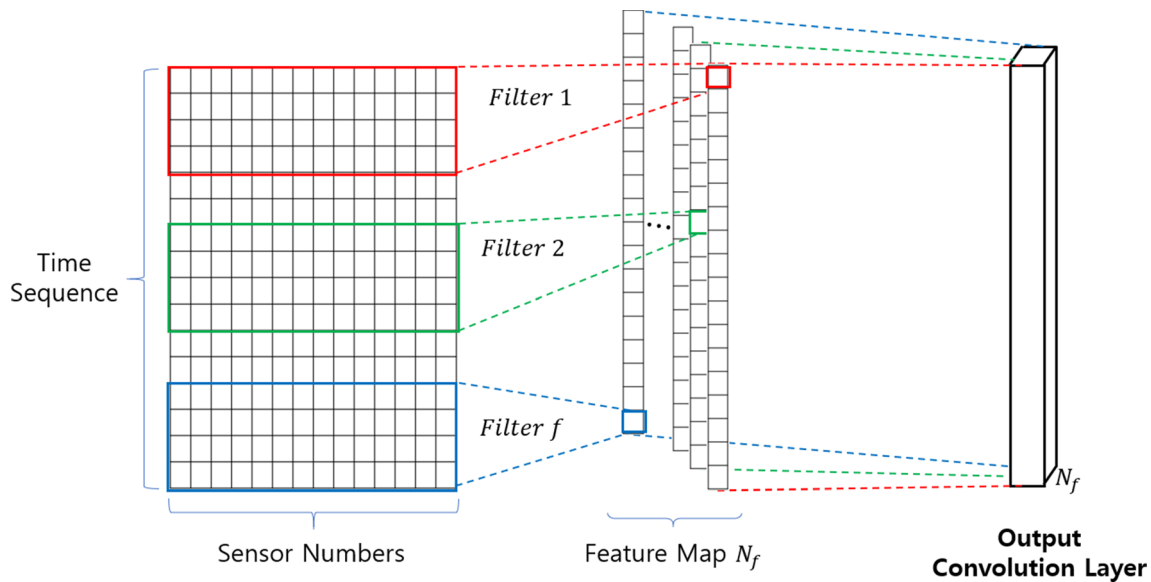
As shown in Fig. 2, our raw data were pre-processed to a 2D matrix containing  $D$  raw sensor signals with an  $N_w$  time sequence.  $D$  represents 21 sensor numbers, and  $N_w$  indicates the time window size for the number of sensor measurement cycles to consider. Thus, the input data has two intrinsic properties: the potential relationship amongst multiple

sensors and that in the time-series. Generally, CNNs extract the feature representation by sliding horizontally and vertically across the input map with convolution filter, essentially reflecting local correlations among pixels, assuming they are related to their neighbors. However, because the physical position of the sensors was unknown, there was no information about sensor correlations in our dataset. Thus, the size of one filter dimension was fixed to 21 sensor numbers. This captured the correlation between multiple sensors at once, leaving only the filter length  $N_t$ , that is, the number of cycles to be considered by the filter, to be decided. By vertically sliding the filter, the temporal properties of the incorporated multi-sensor values could be captured simultaneously.

Because this multi-variate one-dimensional filter can simultaneously reflect all the sensor values, the output value represents the combined values of multiple sensors. The convolution operation based on this filter is as follows:

$$Z^l = [X^{l-1} * k_1^l + b_1^l, \dots, X^{l-1} * k_{N_f}^l + b_{N_f}^l], \quad X^l = ELU(Z^l) \tag{3}$$

where  $*$  represents the convolution operator,  $l$  denotes the  $l$ -th convolution layer,  $X^{l-1}$  and  $Z^l$  represent the input and output of  $l$ th convolution layer.  $ELU$  is the activation function for non-linear transformation between layers,  $k^l$  and  $b^l$  represent the filter kernel and bias, respectively, in the  $l$ -th convolution layer. The concatenation of all the output feature vectors was conducted for the next convolution layer; at the end of the convolution layer, the separated FCL were employed for each task. However, both FCLs had the same input feature vector from a shared network. The feature maps from the filter kernels of the last convolution layer



**Fig. 2** Structure of CNN layer with one-dimensional filter

were flattened and concatenated as an input feature vector of each FCL. Following this, the dropout technique was utilized to address the overfitting problem, and the output neurons for ordinal classification and regression, respectively, were attached at the end of both FCLs for RUL estimation and health condition identification.

## Evaluation metrics

In this study, a multitasking model for RUL prediction and health condition identification is proposed. The performance of the RUL prediction is basically measured by RMSE metric. In addition, a scoring function was also used as another evaluation metric for RUL estimation. The following scoring function introduced in the PHM 2008 challenge has been used in many studies (Babu et al. 2016; Zhang et al. 2017; Li et al. 2018b):

$$L_{score} = \begin{cases} \sum_{i=1}^N \left( e^{-\frac{(\hat{R}_i - R_i)}{13}} - 1 \right) & \text{for } (\hat{R}_i - R_i) < 0 \\ \sum_{i=1}^N \left( e^{\frac{(\hat{R}_i - R_i)}{10}} - 1 \right) & \text{for } (\hat{R}_i - R_i) \geq 0 \end{cases} \quad (4)$$

Meanwhile, we evaluate the health condition identification using zero–one error. Zero–one error is the evaluation metric measuring the percentage of wrong assignments of ordinal categories (Cheng et al. 2008). However, zero–one error does not take into account prediction performance of each class. Thus, the precision, recall and f1 value are additionally considered for evaluating performance of

health condition identification. The measures are defined as follows:

$$\begin{aligned} Precision_i &= \frac{True\ positive_i}{True\ positive_i + False\ positive_i} \\ Recall_i &= \frac{True\ positive_i}{True\ positive_i + False\ neagtive_i} \\ F1\ score_i &= \frac{2 * Precision_i * Recall_i}{Precision_i + Recall_i} \end{aligned} \quad (5)$$

where  $i$  is one of the health condition classes.

## Experimental results

### Experiment setting

In this study, the proposed multi-task framework utilizes the CNN architecture. A pre-processed matrix format of sensor measurements was used as the input data. The matrix was composed of sequential 21 sensor signals with  $N_w$  time window size; at each time step, the matrix was scanned by multi-variate one-dimensional filters of length  $N_f$ , and the filters extracted high-level feature vectors capturing the temporal patterns of multi-signals.

Our CNN architecture consisted of four convolution layers and two FCLs. The number of kernel filters  $N_f$  was different for each convolution layer, while the two FCLs that serve as output layers for RUL estimation and healthy state identification had the same number of neurons  $N_o$ . The dropout technique was imposed on each output layer to reduce the overfitting problem.

The activation function “exponential linear unit” (*ELU*) was used to perform non-linear transformation on the high-level latent vector. The results of the transformation were used as the inputs for the next layer. The *ELU* function is family of rectified linear units (*ReLU*) which speeds up learning and obtains a high performance result. *ReLU* has a linear shape in the positive arguments, alleviating the gradient vanishing problem (Nair and Hinton 2010). However, in the negative arguments, the gradient of the *ReLU* function is zero. The weights cannot be updated once they have negative values. Meanwhile, the *ELU* function decreases the lower-bound threshold to  $-1$  (non-zero) and approximates to an exponential function (Clevert et al. 2015). This allows the *ELU* function to have negative values, which drives sped-up learning.

To find the optimal solution, gradient-descent and back-propagation were adopted in our framework to update all the parameters in the network. For stochastic learning, the Adam optimizer and mini-batch technique were used. The mini-batch size  $N_b$  was determined to be 256, and an optimization technique based on the loss function was performed on each batch sample. The total number of training epochs  $N_e$  was 500, and early stopping was implemented for efficient learning. The learning rate was 0.001 for the first 250 epochs, after which it was 0.0001. All the hyperparameter values, set through trial and error, are presented in Table 2.

### Baselines with hyper-parameter setting

The proposed multi-tasking CNN model (MT-CNN) is able to provide an accurate RUL estimation. To demonstrate the effectiveness of simultaneous learning, we first compared our MT-CNN to single-task CNN (ST-CNN), which is applied only for RUL estimation. Then, four regression-based architectures were used as baselines: deep neural network (DNN), support vector regression (SVR), extreme gradient boosting algorithm (XGBoost), recurrent neural network (RNN) and LSTM. Additionally, eight different methods in the prognostic domain were compared with our model, such as the RNN-based approach (Malhotra et al. 2016; Lim et al. 2016; Yoon et al. 2017), CNN-based approach (Babu et al. 2016; Li

et al. 2018a, b), and ensemble-based approach (Zhang et al. 2017).

#### DNN

The ANN is a basic neural network technique that reflects the nonlinear relationship between the variables (Rosenblatt 1958). These complex nonlinear relationships can be considered by adding deeper layers. Five hidden layers were constructed in this study; 16, 32, 64, 128, and 256 are defined as the number of neurons in each hidden layer, respectively. To avoid the overfitting problem, a dropout technique was employed, with a dropout rate of 0.5.

#### XGBoost

The XGBoost is an ensemble tree learning algorithm with a parallel decision tree structure with different weights (Chen and Guestrin 2016). A classification and regression tree (CART) is used as a basic tree model in XGBoost, and the tree weights are updated by a gradient boosting algorithm. For the tree structure in this study, a maximum of 1000 leaves and a maximum individual tree depth 100 were used. A learning rate of 0.06 and an iteration count of 300 were also applied to update the tree weights.

#### RNN

The RNN has a feedback connection from input to output and can thus handle sequential information (Williams and Zipser 1989). Because the sensor value of C-MAPSS dataset has the sequence, RNN can have superiority to perform time-series RUL prediction task. In this study, an RNN model with 5 recurrent layers was organized to have a structural depth that was identical to our proposed multi-task model. An Adam optimizer, a learning rate of 0.01 and 500 epochs were adopted to obtain optimal parameter values.

#### LSTM

Because RNN has a vanishing or increasing gradient tendency during back propagation, it has difficulty of capturing long-term dependencies (Mikolov et al. 2010). In LSTM, a gate structure is employed, which helps to weaken gradient vanishing or increasing tendencies (Hochreiter and

**Table 2** Description of hyperparameter values

Hyperparameter	Value	Hyperparameter	Value
Number of kernel filter $N_f$	32/64/128/256	Filter length $N_l$	3
Time window size $N_w$	30	Number of neurons $N_o$	256
Activation function	<i>ELU</i>	Dropout rate	0.5
Mini-batch size $N_b$	256	Learning rate	0.001/0.0001

Schmidhuber 1997). Thus, LSTM is regarded as an alternative to an RNN. Here, an LSTM model was constructed with the same structure as an RNN to provide a fair comparison.

### Comparison with baselines

The comparison results with other baselines are shown in Table 3. Our proposed method MT-CNN and ST-CNN demonstrated promising performance in both RMSE and scoring evaluations, and MT-CNN showed the best performance on all the subsets. This result indicates that simultaneous learning of related tasks is a more effective method than individual learning. In particular, our model is more effective in handling complex problems (i.e., FD002 and FD004) than simple ones (i.e., FD001 and FD003). Because the multi-tasking architecture requires sufficient network capabilities to reflect all the information from various tasks, our model achieved significant improvement in a more elaborate experimental environment.

The RNN-based models show the next best performance in capturing temporal patterns in sequential sensor values. LSTM produces a better result than RNN because of its modified internal structure. However, we used the basic structure of the recurrent layer, and variations of them have been further optimized in several studies (Malhotra et al. 2016; Lim et al. 2016; Yoon et al. 2017). The XGBoost is not designed to consider temporal patterns, and it thus demonstrates the least efficient performance despite its reputation of powerful performance.

For the health condition identification, the zero–one error and precision, recall and f1 score are applied to measure the performance of another task. In this study, we have three categories for health condition. If we select one of them randomly, the performance of random selection might be 0.33. As shown in Table 4, our proposed model shows twice as much zero–one error performance as random selection. From the zero–one error, it is confirmed that our proposed model has the reasonable performance of auxiliary task.

Additionally, in the terms of the precision and recall results, our proposed model has the low precision and high recall, high precision and low recall, and high precision and high recall at the healthy, deteriorating and unhealthy state,

**Table 4** Performance of health condition identification

Metric	FD001	FD002	FD003	FD004
Zero–one error	0.71	0.72	0.73	0.69
Healthy state				
Precision	0.16	0.3	0.13	0.25
Recall	0.8	0.84	0.75	0.7
F1 score	0.26	0.44	0.23	0.37
Deteriorating state				
Precision	0.9	0.94	0.96	0.87
Recall	0.5	0.42	0.49	0.45
F1 score	0.64	0.58	0.65	0.6
Unhealthy state				
Precision	0.71	0.92	0.87	0.84
Recall	1	0.99	1	0.88
F1 score	0.83	0.96	0.93	0.86

respectively. The results of healthy and deteriorating state indicate that our model predict more healthy state and less deteriorating state than actual number of them. Since the sensor measurements of the end periods have a characteristic pattern rather than start periods, there are only few characteristic sensor patterns that distinguish between both states. This is because there are only few characteristic sensor patterns that distinguish between both states. However, from the results of high precision and recall of unhealthy state, our proposed model specializes in capturing characteristic patterns of the unhealthy state. Diagnosing near-failure condition is also helpful in estimating RUL which predicts when failures occur. Thus, our health condition identification task will eventually help to improve the performance of RUL prediction.

### Comparison with related works

The comparison results with previous studies that evaluated their framework on the C-MAPSS dataset are presented. Table 5 summaries the experiment result of latest research on the four sub-datasets of C-MAPSS (Malhotra et al. 2016; Lim et al. 2016; Yoon et al. 2017; Babu et al. 2016; Li et al. 2018a, b; Zhang et al. 2017). They applied various neural

**Table 3** Comparison performance with other baselines

Model	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
ANN	19.62	640	23.55	3461	19.66	699	24.35	4134
XGBoost	30.78	12,085	35.39	62,241	28.63	9899	32.00	40,965
RNN	13.36	412	23.82	3663	14.11	440	24.06	3971
LSTM	13.12	331	23.53	3417	13.68	411	24.05	3951
ST-CNN	12.72	279	21.24	2247	12.51	386	21.86	2388
MT-CNN	12.48	224	19.77	2023	12.11	334	19.98	2097



**Table 5** Compared result with related works

Related work	RMSE			
	FD001	FD002	FD003	FD004
CNN with full-dimensional filter	18.45	30.29	19.82	29.16
Random Forest	17.91	29.59	20.27	31.12
Time window based NN	15.16	N/A	N/A	N/A
Multi-objective DBN ensemble	15.04	25.05	12.51	28.66
LSTM with semi-supervised learning	13.96	N/A	N/A	N/A
Light GBM	13.45	N/A	N/A	N/A
LSTM encoder–decoder	12.81	N/A	N/A	N/A
CNN with univariate 1-d filter	12.61	22.36	12.64	23.31
Proposed ST-CNN	12.72	21.24	12.51	21.86
Proposed MT-CNN	12.58	<b>19.77</b>	12.11	<b>19.98</b>

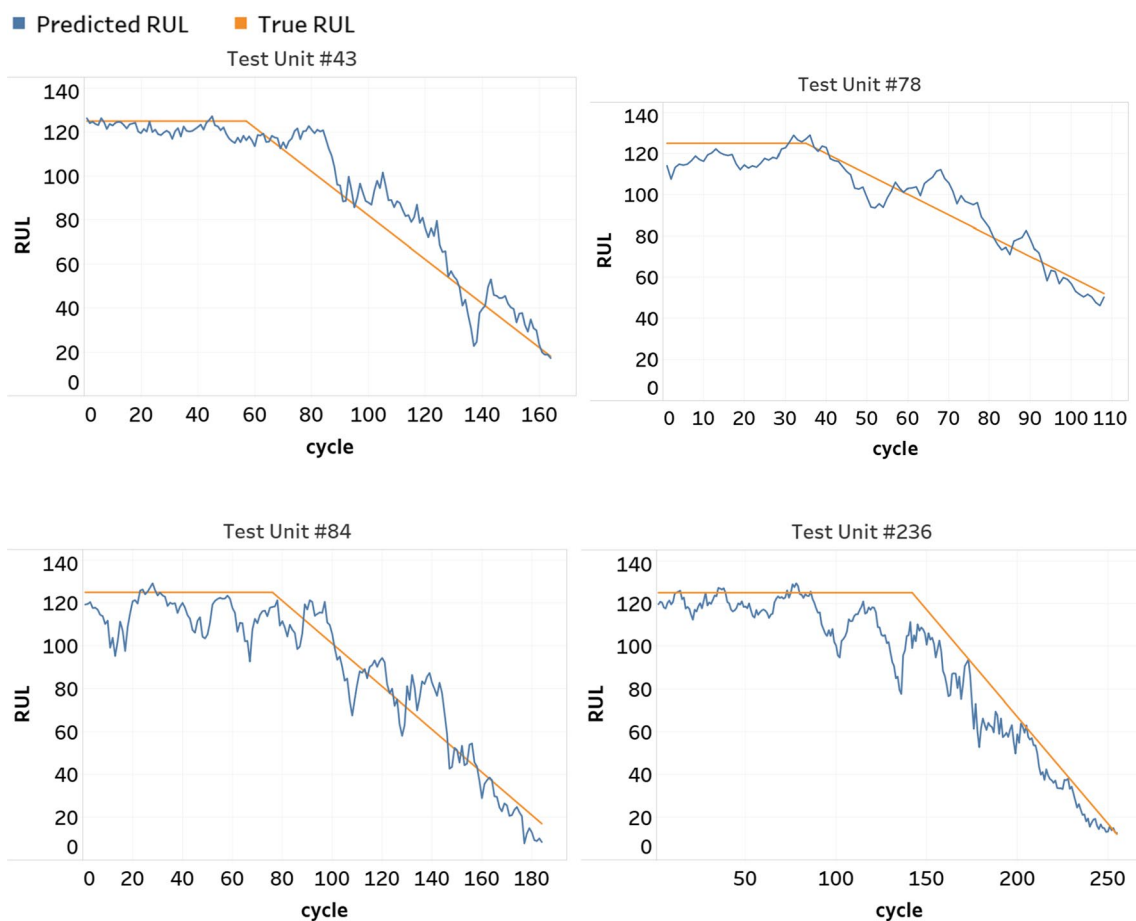
network architectures including recurrent, convolution and ensemble networks.

Our proposed multitasking model demonstrates outstanding performance compared to other related works. The CNN-based approach with univariate one-dimensional

filter produces very promising results, but unlike our model, it only considers RUL information in the learning process. With the exception of this model, most time-dependent models have better results than others because of their advantage of capturing the temporal pattern of sequential sensor measurements. However, recurrent network-based approaches require relatively high computational time to be trained (Vaswani et al. 2017). Since RNN is needed to be processed sequentially, parallel processing of GPU cannot be adopted to the computation (Bradbury et al. 2016). Hence, in integrated health management systems, our proposed multitasking architecture has great potential in terms of both performance and efficiency in making RUL estimations.

**Prediction power of unit-level**

The purpose of our study is to increase prognostic performance for each engine unit. Even if the RMSE is very small for the total RUL, the difference between the predicted and actual RUL values could be large for some units. To demonstrate the forecasting power for the engine unit level, the prediction results of testing engine units in the FD002 subset



**Fig. 3** Unit-level results of RUL predictions for testing units

are presented in Fig. 3. Among the units, four testing units with the longest cycles are selected.

It can be observed that the predicted values in every cycle are close to actual RUL values. Specifically, the difference between actual and predicted values tends to be small when the cycles are close to the last record. This shows that the sensor measurements of the end periods have characteristic pattern. In particular, as proven in performance of health condition identification, our model has good ability to differentiate the last period from normal ones. A good evaluation in the last period can effectively save maintenance costs and enhance system reliability. This is the reason for the superior performance of our MTL framework as compared to independent learning, which distinguishes unique properties in healthy and unhealthy conditions. Our MTL framework can effectively reflect the characteristics of unhealthy conditions through healthy condition identification tasks.

## Conclusion

In most industries, achieving good prognostic performance has great benefits in making maintenance decisions. In this paper, a CNN-based MTL framework was proposed to accurately estimate RUL by simultaneous learning during health state identification. The health state labeling process was carried out before our MTL process. Health information was combined to establish an MTL model for RUL estimation, where interdependencies of both tasks were considered via a network for RUL estimation by using general feature extracted in the shared network.

The proposed method was evaluated with the C-MAPSS dataset, and the prediction results on each sub-dataset was obtained and compared with those from methods in related literatures. Because the health state identification network could capture distinct characters of sensor measurements very close to health and failure, our model showed outstanding performance among various single task models. Our CNN architecture-based model achieved good prediction accuracy, and it has the additional advantage of improved computational complexity as compared to the recurrent neural architecture. Thus, the proposed framework shows promise for effective maintenance decision-making.

Specifically, various supportive techniques were employed in each part of our framework for effective and accurate prediction. Sensor selection and condition-wise normalization were applied to refine data. Multi-tasking learning architecture was exploited to capture salient patterns of both tasks from a complex system. Here, a multivariate one-dimensional filter was established to consider temporal properties among various sensor measurements. A dropout technique and *ELU* function were used to avoid

the overfitting problem and promote effective learning, respectively.

In future work, the proposed method will be extended to solve multi-objective problems in integrated health management systems. In addition, a state-of-the-art shallow network can be employed to reduce the number of parameters and accelerate computational speed. We anticipate further studies addressing such limitations in the future.

**Funding** This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2020R1A2C2005026).

## References

- Ando, R. K., & Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov), 1817–1853.
- Argyriou, A., Evgeniou, T., & Pontil, M. (2007). Multi-task feature learning. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems* (Vol. 19, pp. 41–48). Cambridge: MIT Press.
- Azadeh, A., Asadzadeh, S. M., Salehi, N., & Firoozi, M. (2015). Condition-based maintenance effectiveness for series-parallel power generation system—A combined Markovian simulation model. *Reliability Engineering & System Safety*, 142, 357–368.
- Babu, G. S., Zhao, P., & Li, X. L. (2016, April). Deep convolutional neural network based regression approach for estimation of remaining useful life. In *International conference on database systems for advanced applications* (pp. 214–228). Cham: Springer.
- Baxter, J. (1997). A Bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28(1), 7–39.
- Bradbury, J., Merity, S., Xiong, C., & Socher, R. (2016). Quasi-recurrent neural networks. arXiv preprint [arXiv:1611.01576](https://arxiv.org/abs/1611.01576).
- Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1), 41–75.
- Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). ACM.
- Cheng, J., Wang, Z., & Pollastri, G. (2008, June). A neural network approach to ordinal regression. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1279–1284). IEEE.
- Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint [arXiv:1511.07289](https://arxiv.org/abs/1511.07289).
- Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).
- Gugulothu, N., TV, V., Malhotra, P., Vig, L., Agarwal, P., & Shroff, G. (2017). Predicting remaining useful life using time series embeddings based on recurrent neural networks. arXiv preprint [arXiv:1709.01073](https://arxiv.org/abs/1709.01073).
- Guo, L., Li, N., Jia, F., Lei, Y., & Lin, J. (2017). A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, 240, 98–109.
- Heimes, F. O. (2008, October). Recurrent neural networks for remaining useful life estimation. In *International conference on prognostics and health management, 2008. PHM 2008* (pp. 1–6). IEEE.

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107, 241–265.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems, NIPS* (pp. 1097–1105). <https://doi.org/10.1145/3065386>.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436.
- Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 42(1–2), 314–334.
- Lei, Y., He, Z., & Zi, Y. (2008). A new approach to intelligent fault diagnosis of rotating machinery. *Expert Systems with Applications*, 35(4), 1593–1600.
- Li, X., Ding, Q., & Sun, J. Q. (2018a). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172, 1–11.
- Li, F., Zhang, L., Chen, B., Gao, D., Cheng, Y., Zhang, X.,... & Peng, J. (2018a, November). A light gradient boosting machine for remaining useful life estimation of aircraft engines. In *2018 21st international conference on intelligent transportation systems (ITSC)* (pp. 3562–3567). IEEE.
- Lim, P., Goh, C. K., & Tan, K. C. (2016, July). A time window neural network based framework for remaining useful life estimation. In *2016 international joint conference on neural networks (IJCNN)* (pp. 1746–1753). IEEE.
- Malhi, A., Yan, R., & Gao, R. X. (2011). Prognosis of defect propagation based on recurrent neural networks. *IEEE Transactions on Instrumentation and Measurement*, 60(3), 703–711.
- Malhotra, P., TV, V., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder. arXiv preprint [arXiv:1608.06154](https://arxiv.org/abs/1608.06154).
- Marcus, G. (2018). Deep learning: A critical appraisal. arXiv preprint [arXiv:1801.00631](https://arxiv.org/abs/1801.00631).
- Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh annual conference of the International Speech Communication Association*.
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807–814).
- Niu, G., & Yang, B. S. (2010). Intelligent condition monitoring and prognostics system based on data-fusion strategy. *Expert Systems with Applications*, 37(12), 8831–8840.
- Niu, Z., Zhou, M., Wang, L., Gao, X., & Hua, G. (2016). Ordinal regression with multiple output CNN for age estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4920–4928).
- Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.
- Patton, R. J., Frank, P. M., & Clark, R. N. (Eds.). (2013). *Issues of fault diagnosis for dynamic systems*. Berlin: Springer.
- Rabiei, M., & Modarres, M. (2013). A recursive Bayesian framework for structural health management using online monitoring and periodic inspections. *Reliability Engineering & System Safety*, 112, 154–164.
- Riad, A., Elminir, H., & Elattar, H. (2010). Evaluation of neural networks in the subject of prognostics as compared to linear regression model. *International Journal of Engineering & Technology*, 10(6), 52–58.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386.
- Ruder, S. (2017). An overview of multi-task learning in deep neural networks. arXiv preprint [arXiv:1706.05098](https://arxiv.org/abs/1706.05098).
- Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008, October). Damage propagation modeling for aircraft engine run-to-failure simulation. In *International conference on prognostics and health management, 2008. PHM 2008* (pp. 1–9). IEEE.
- Stringer, D. B., Sheth, P. N., & Allaire, P. E. (2012). Physics-based modeling strategies for diagnostic and prognostic application in aerospace systems. *Journal of Intelligent Manufacturing*, 23(2), 155–162.
- Tian, Z. (2012). An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. *Journal of Intelligent Manufacturing*, 23(2), 227–237.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems, NIPS* (pp. 5998–6008).
- Vogl, G. W., Weiss, B. A., & Helu, M. (2019). A review of diagnostic and prognostic capabilities and best practices for manufacturing. *Journal of Intelligent Manufacturing*, 30(1), 79–95.
- Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2), 270–280.
- Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, 275, 167–179.
- Xia, T., Dong, Y., Xiao, L., Du, S., Pan, E., & Xi, L. (2018). Recent advances in prognostics and health management for advanced manufacturing paradigms. *Reliability Engineering & System Safety*, 178, 255–268.
- Yoon, A. S., Lee, T., Lim, Y., Jung, D., Kang, P., Kim, D., et al. (2017). Semi-supervised learning with deep generative models for asset failure prediction. arXiv preprint [arXiv:1709.00845](https://arxiv.org/abs/1709.00845).
- Yuan, M., Wu, Y., & Lin, L. (2016, October). Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. In *IEEE international conference on aircraft utility systems (AUS)* (pp. 135–140). IEEE.
- Zaidan, M. A., Relan, R., Mills, A. R., & Harrison, R. F. (2015). Prognostics of gas turbine engine: An integrated approach. *Expert Systems with Applications*, 42(22), 8472–8483.
- Zhang, C., Lim, P., Qin, A. K., & Tan, K. C. (2017). Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2306–2318.
- Zhang, Y., & Yang, Q. (2017). A survey on multi-task learning. arXiv preprint [arXiv:1707.08114](https://arxiv.org/abs/1707.08114).