Journal of Manufacturing Systems 58 (2021) 109–119

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

Technical Paper

A joint classification-regression method for multi-stage remaining useful life prediction

Ji-Yan Wu [a], Min Wu [a,*], Zhenghua Chen [a], Xiaoli Li [a], Ruqiang Yan [b]

[a] Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore 136682, Singapore
[b] School of Mechanical Engineering, Xi'An Jiaotong University, Xi'An, Shanxi, China

ARTICLE INFO

Keywords:
Prognostic technique
Remaining useful life
Multi-stage
Machine learning

ABSTRACT

Remaining useful life (RUL) prediction plays an important role in increasing the availability and productivity of industrial manufacturing systems. This paper proposes a joint classification-regression scheme for multi-stage RUL prediction. First, the time domain and frequency domain features are extracted from various types of raw sensory data (e.g., acoustic, current, vibration and temperature) to constitute the training data set. Second, the system health stage is classified based on the trained model and real-time sensory data. Third, we perform stage-level RUL prediction with regression algorithm to estimate overall useful life. Distinct from the existing RUL estimation algorithms, the proposed multi-stage remaining useful life (MS-RUL) prediction effectively integrates the machine/deep learning based classification and regression to improve overall estimation accuracy. We conduct the performance evaluation with sensory data from real manufacturing systems. Experimental results demonstrate that the proposed MS-RUL achieves approximately 6.5% accuracy improvements over the state-of-the-art algorithms in the RUL prediction.

1. Introduction

Prognostic and health management schemes are the key technologies in predictive maintenance for various industrial applications [1,2,35,45], e.g., advanced manufacturing, power systems, aircraft engines, vehicles, and heavy industry. In particular, prognostic technologies are commonly adopted to estimate the future status/performance of a subsystem or a component for remaining useful life (RUL) prediction [3–10]. Furthermore, prognostic solutions also evaluate the system/-component degradation based on provided environmental conditions. For instance, accurate prediction of engine failure is critical to make maintenance decisions, so that the maintenance cost can be reduced and the operational activities can be streamlined.

Based on the existing studies [11,12], the current prognostic solutions can be generally divided into the categories of: (1) modeling-based; (2) data-driven; and (3) hybrid methods.

In modeling-based solutions, the prediction models are developed to characterize the degradation level of a component/subsystem. For example, the authors in reference [13] adopt the Paris's fatigue law to model the defect growth rate of a bearing. In literature [14], the crack growth of rotor shaft is analytically modeled based the Forman law in linear elastic fracture mechanics. However, these solutions require special domain knowledge and are not effective in characterizing complex systems.

Data-driven methods typically develop prediction models from sensory data of industrial systems. Conventionally, the statistical and machine learning algorithms are used in these models to capture the system performance. Therefore, these data-driven solutions is able to balance the relationship between applicability, complexity, and precision. The representative data-driven approaches include neural networks [15], Markov models/chains [16–18], autoregressive algorithm [21], stochastic models [22], and Bayesian networks [23].

However, several existing studies already point out the main problems with data-driven prognostic algorithms: (1) sufficient amount of training data is required in most of data-driven methods to learn the degradation model [24]. If the labeled data is absent, the number of healthy states need to be determined manually; (2) a system degradation model is assumed in most of these approaches for the specific component without sufficient evidence/proof; (3) although the methods based on neural network or deep learning entail highly accurate RUL estimations, they do not provide an in-depth understanding of the failure process. On the other side, the stochastic models perform better in exploring failure

* Corresponding author.
E-mail address: wumin@i2r.a-star.edu.sg (M. Wu).

https://doi.org/10.1016/j.jmsy.2020.11.016
Received 1 April 2020; Received in revised form 13 November 2020; Accepted 23 November 2020
Available online 7 December 2020

mechanisms but generally have limited applicability to standard data sets. The hybrid approaches is a combination of modeling-based and data-oriented methods. In general, the hybrid model is developed based on both the specific-domain knowledge of the industrial applications and the parameters trained using real system data.

To remedy the shortcomings of the above prediction methods, this paper proposes a Multi-Stage RUL (MS-RUL) estimation scheme. First, the proposed MS-RUL extracts the time-frequency domain freatures from various sensory data (e.g., acoustic, current, vibration and temperature) to constitute the training data set. Second, the system health stage (e.g., new, medium, heavy) is classified based on the trained model and real-time sensory data. Third, we perform stage-level RUL prediction with machine/deep learning regression algorithm to estimate overall useful life. SVM proves to be effective in RUL estimation [7,12] and this is the main motivation for us to adopt SVM for joint classification-regression. Distinct from the existing RUL estimation algorithms, the proposed solution effectively integrates the machine learning based classification and regression to improve overall estimation accuracy. MS-RUL is an effective and novel integration of the health stage classification and stage-level regression to achieve higher accuracy. The scientific justification of the proposed MS-RUL can be summarized as combining the advantages of machine/deep learning based classification and regression algorithms. First, MS-RUL intelligently divides and diagnoses the machine healthy stages. Second, the special degradation features of different healthy stages are exploited to improve the stage-level prediction accuracy.

The motivation for us combine the two approaches of health stage classification and stage-level regression stems from our practical RUL prediction for tool wear in shaft production line (Section 4.2). In this problem, we observe that the whole life cycle of the tool can be divided into multiple health stages (e.g., new, medium, old) with different sensory data features. This motivates us to design the two-stage approach for RUL prediction and also apply on other representative data-sets (e.g., the C-MAPSS [40]). Experimental results demonstrate that our two-stage combination approach also achieves better performance than other competing algorithms in this turbofan sensory data-set [40].

On the combination of the models for the two stages in MS-RUL, we have evaluated representative algorithms as shown in Tables 3, 5 and 6. These evaluation results demonstrate that the superiority of MS-RUL framework and the effectiveness of the combinations. In different application scenarios, there may be other combinations of approaches/models to achieve high accuracy. Since there is a large number of machine/deep learning models, it is impossible for us to evaluate all the possible combinations. On the selection of the potential algorithms for each stage, we have conducted extensive comparison experiments with representative models (e.g., GB, SVM, LSTM and CNN) in Section 4. The performance is evaluated with commonly-used accuracy metrics, e.g., the RMSE and Score [40]. As the experimental results indicates, the CNN based MS-RUL achieves the highest average accuracy for the C-MAPSS data-set [40]. For other different data-sets, the selection of potential algorithms for each stage can follow the above routine.

The special features and complexity of different sensory data need to be considered for effective combination of the two steps. For instance, the healthy stage classification is ineffective in application scenarios in which there are no distinct pattern differences observed in the raw sensory data of different health stages. To deal with such cases, we need to consider exploring the latent variables (e.g., by extracting frequency or using auto encoder) for accurate RUL prediction. In summary, the combination of different approaches for RUL prediction is dependent on specific conditions and environments.

In particular, the contributions of this research can be summarized as follows.

- We propose a novel RUL prediction scheme that models the multi-stage system/device degradation process with the following modules:
  - –***Classification***: a health stage classification method that maps the continuous sensory data values to discrete machine/component states.
  - –***Regression***: a machine/deep learning based stage-level RUL prediction method to leverage the special features in different stages of useful life.
  - –***Labeling***: an automatic labeling algorithm that leverages membership degree function to process unclassified continuous sensory data.
  - –***Flexibility***: the classification and regression modules can be implemented with different machine/deep learning algorithms (e.g., CNN, LSTM, SVM, etc.) according to the special system/device features to improve prediction accuracy.
- We evaluate the performance with real sensory data from shaft production system and open-source C-MAPSS data for engine degradation. Experimental results demonstrate:
  - –the proposed MS-RUL framework achieves higher prediction accuracy than CNN (Convolutional Neural Network) [1], SVR (Support Vector Regression) [12], LSTM (Long Short-Term Memory), and LR (Linear Regression).
  - –the deep learning algorithms (e.g., LSTM and CNN) achieve higher accuracy shallow learning (e.g., SVM and GB) algorithms in the proposed MS-RUL for joint classification-regression.

The remainder of this paper includes the following parts. Section 2 briefly reviews the related work close to this paper. In Section 3, we present the proposed algorithms of the MS-RUL in detail. The performance evaluation is provided in Section 4.

2. Related work

The related work close to this paper generally include RUL estimation algorithms and fault detection methods in industrial manufacturing systems/applications. In Table 1, we summarize and compare the representative studies on RUL prediction. The following subsections will discuss these works in detail.

2.1. Data-driven RUL prediction schemes

In reference [1], the authors adopt deep CNN based on regression approach for RUL prediction. The convolutional and pooling filters in [1] are applied along the temporal dimension over the multi-channel sensor data. However, the CNN model [1] is ineffective in capturing the sudden/fast degradation trends caused by faults/anomalies. In reference [12], the relationship between health indicator and sensory data values are directly modeled using SVR. Furthermore, an offline wrapper variable is used for selection before model training. This SVR-based prediction method has the limitations of using the same weight vector for different healthy stages of RUL. Orchard et al. [25] develop a RUL prediction algorithm that merges mutual information obtained from the multiple features and appropriately correlates with the machinery failure. Then, the authors estimate the RUL using a particle filtering-based algorithm. However, reference [25] assumes the same degradation trend for the whole life-cycle. In reference [28], a joint fuzzy mean clustering and neural network method is adopted to classify the sensory data into dynamic health status. And the RUL is calculated based on the time stamp of the failure status. This data-driven prognostic technique [28] is heavily dependent on the accurate failure threshold value and large amount of training data. Chen et al. [44] propose an intelligent and end-to-end health indicator construction approach which integrates the advantages of RNN and CNN. In [46], an attention-model based LSTM is proposed to improve the accuracy of machine RUL prediction.

Table 1
Comparison of the existing works on RUL prediction.

Ref.	Type	Algorithm	Contributions/Features	Remarks
[1]	Data-Driven	CNN	Convolution-pooling applied along with temporal dimensions	Ineffective for sudden degradation
[12]	Data-Driven	SVR	Health indicator to sensory data mapping, offline wrapper for training	Same weight vector for whole life cycle
[28]	Data-Driven	Clustering & NN	Dynamic health status	Dependency on failure threshold
[2]	Model-Driven	Stochastic model & Kalman filter	Stochastic model for degradation process	Limited variability sources
[26]	Model-Driven	Particle filtering (PF)	Combining particle filter & degradation model	Gaussian distribution assumption
[27]	Model-Driven	Belief function	Sensory data classification into four health stages	Fixed health stage number
[37]	Hybrid	SV-based	End of Life estimation	Based on critical features
[39]	Hybrid	SVR-PF	Probability distribution based	Assumption in distribution
MS-RUL	Hybrid	Classification & regression	Automatic stage classification stage-level prediction	Capture various degradation trends

2.2. Model-driven RUL prediction schemes

Lei [2] et al. propose a machine RUL prediction method includes: (1) a stochastic model considering the multiple variability sources of machine degradation process; (2) a Kalman particle filtering algorithm to estimate the system states and predict RUL. The authors in reference [26] combine an unscented particle filter algorithm and degradation modeling to predict the RUL. The probability distributions of the last step for online update are considered as the RUL value distribution. However, it is widely acknowledged that the particle filter algorithm generally assumes a Gaussian distribution of the measured data. The data in practical scenarios may not follow the Gaussian distribution. The belief functions are adopted by Ramasso et al. [27] to classify the original sensory data into four healthy states. Then, the RUL is estimated as the duration from degradation to failure state. The fixed healthy state number in [27] usually cannot capture the degradation stages in practical industrial manufacturing systems/applications. In [19], the anomaly monitoring (raw sensor data change) is used in both RUL estimator training and deployment processes. Wang et al. [20] present tool RUL prediction approach based on Bayesian network and PF to reliably quantify the degradation trend. In [30], the tool wear state is predicted by recursively updating a physics-based tool wear rate model with online measurement, following a Bayesian inference scheme.

In instance-based learning research, the system degradation is expressed as health indicator obtained with the fuzzy model [29], unsupervised kernel regression modeling [31], linear regression [32]. Or using lower and upper envelopes of a trajectory to compute polygons [33]. Then, the RUL is calculated as a weighted moving average of all the useful life values from similar cases. However, the health indicator used for different healthy stages often cannot capture the fast/sudden degradations.

2.3. Hybrid RUL prediction

There are also some researches using pattern matching for prognostic. These models first estimate the variation of a failure signal and the RUL is then estimated as the duration for the signal to approach the end of life (EOL). These approaches generally require adjusting failure threshold values. The Support Vector Regression [34] or Support Vector Machines [36] are adopted to capture degradation models. The minimal value obtained with the three mathematical models in reference [36] is used to estimate the RUL. In [37], the RUL estimation is made only for the state near the EOL and is defined as a percentage of EOL values. The particle swarm optimization is adopted by Qin et al. [34] to obtain SVR configuration parameters. Loutas et al. [38] use Wiener entropy to monitor conditions and locate critical system faults. Based on the establishment of a critical operational threshold, the RUL is then predicted. Dong et al. [39] combine SVR-PF model to estimate the component degradation time. Then, the RUL is estimated by updating the probability distribution when the time stamp changes.

In summary, a failure threshold is generally required in the existing degradation and regression based prognostic methods. However, a static threshold is difficult to estimate and such static parameters are not adaptive to environmental changes. On the other side, the pattern matching methods normally do not require setting threshold values. The RUL estimation is the process of tracing the correct trend following its EOL value. Nonetheless, this approach cannot derive right solutions in the case that matched patterns are absent. This paper leverages the advantages of machine learning based classification and regression. And the special features of different stages are leveraged to improve the prediction accuracy. The idea of integrating the classification and regression schemes is motivated by the fact/observation that the machine/component RUL lifetime can be divided into multiple stages. To leverage this multi-stage degradation feature, we take efforts on developing the joint classification-regression framework including the feature selection, health stage classification, and stage-level RUL prediction.

3. MS-RUL algorithms

3.1. Overview of system framework

The system framework of the proposed scheme is shown in Fig. 1. We consider the problem of system RUL prediction with real-time heterogeneous sensory data (e.g., acoustic, current, vibration, temperature and other sensory values).

First, the time and frequency domains of features are extracted from four types of sensory data (i.e., acoustic, current, vibration and temperature) to constitute the eigenvectors of support vector machine (SVM). In the data pre-processing phase, the time-frequency feature analysis is used for characterizing and converting data rows with variable statistics in time. This step is a refinement and summarization of the Fourier transform for the case of time-varying signal frequency. In practical industrial applications, many sensory data (e.g., acoustic, vibration, current values and temperature) have variable frequency features and time-frequency analysis has wide range of application scenarios.

3.2. Feature engineering

Before the model training phase, it is important and necessary to differentiate the most important information from less important ones. Including the irrelevant/unnecessary features may degrade the training accuracy and delay performance of machine learning algorithms. Consequently, the selection of critical information is of vital significance for developing a model with high accuracy. Thus, the variable determination is a critical step on the improvement of SVR models.

Based on the dependence on decision algorithms, the feature selection methods can be generally classified into three main groups: wrappers, filters, and embedded methods. Specifically, the filters run independently of decision algorithms. The wrapper methods take into

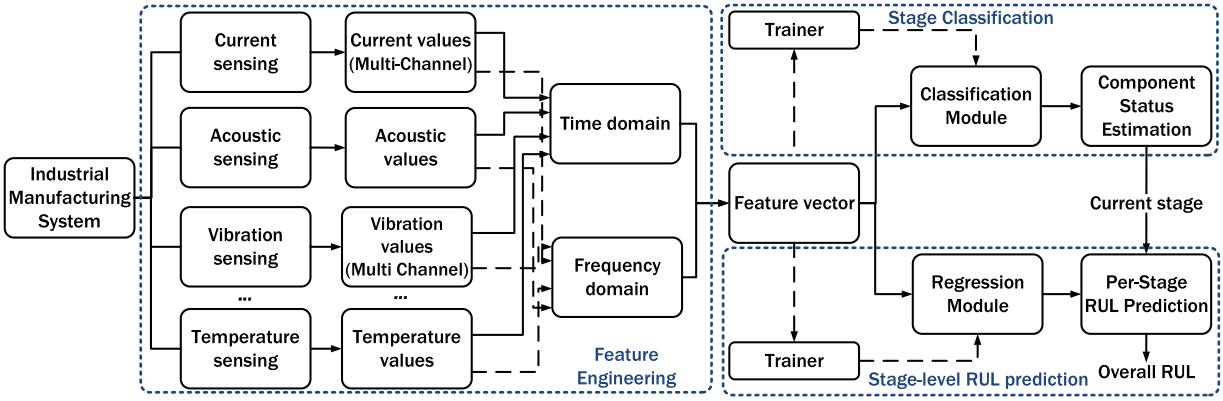

Fig. 1. System design of the proposed multi-stage RUL prediction scheme.

account the performance of the decision algorithm during the criteria selection. Embedded methods are developed based on a decision mining algorithm for the feature filtering. In this system framework, a wrapper sensor selection scheme is adopted as the offline phase of the proposed solution. The performance and score of each type of sensory values, as well as the system-level values are tested first in order to analyze the impact on the trained model. Then, at each duration, a new sub-set of sensory data values are selected to achieve the best performance and this is achieve using the leave-one-out cross validation.

In this work, we consider the features that characterize both the values and the trend of the time-series data in a sliding window. The time series is first decomposed into non-overlapping windows of a fixed size. For each window, we extract the following parameters per dimension: the mean value of the sliding window and the trend parameter on the sliding window. Consequently, the feature vector elements of size 2 times the time-series dimension is obtained for each sliding window. The remaining useful life is then associated with the feature vector elements.

As shown in Table 2, the time-domain features include the mean value, maximum value, minimum value, variance, root mean square, first quartile, and mean absolute value. Specifically, the equations for some of these time domain and frequency domain features are listed below:

$$\text{Mean } (\mu) = \frac{\sum_{i=1}^{n} x_i}{n},$$

$$\text{Variance } (\sigma^2) = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n},$$

$$\text{Root mean square} = \sqrt{\frac{\sum_{i=1}^{n} (x_i)^2}{n}},$$

$$\text{Skewness} = \frac{\sum_{i=1}^{n} (x_i - \mu)^3}{n \times \sigma^3},$$

$$\text{Kurtosis} = \frac{\sum_{i=1}^{n} (x_i - \mu)^4}{n \times \sigma^4} - 3.$$

Table 2
Time and frequency domain feature list.

Time-domain features	Frequence-domain features
Mean	Correlation
Max	Spectral roll-off
Min	Spectral centroid
Variance	Spectral flux
Root mean square	Kurtosis
First quartile	Skewness
Mean absolute value	Amplitude

The fast Fourier transform can be applied to capture the frequency of different signal values. Note that his method also requires the descriptions of the signal features over the whole lifetime. In merging the information across the time domain, it is natural to consider also leveraging the points in the frequency/spectral domain. Table 2 also presents the frequency-domain features including correlation, spectral roll-off, spectral centroid, spectral flux, Kurtosis and Skewness.

We use the data buffer to deal with the non-stationary time-series data in the RUL prediction process. The buffer size can be dynamically adjusted according to the specific requirements (e.g., the data sampling frequency and delay requirement). In our proposed MS-RUL, the buffer size is 30 seconds because of the accuracy results. As shown in Table 8 of Section 4.4, the buffer size of 30 s is able to achieve the highest average accuracy for different data-sets. Please note the RUL prediction problems considered in this research are normally delay-tolerant (i.e., there is no strict deadline). A shorter buffer size can be adopted if the RUL prediction is imposed with tighter delay constraint.

3.3. Health stage classification

The health stage classification step is used to determine the current system/machine status, such that the RUL of the current and remaining stages can be estimated. Basically, this step can be performed using machine/deep learning based classification algorithm. In this section, we describe the health stage classification using the SVM algorithm. The other machine/deep learning algorithms (e.g., CNN, LSTM, GB, etc.) can also be adopted in the MS-RUL for this stage classification.

We consider the problem in which three groups of data (e.g., new, medium and heavy states in this studied problem) have to be classified. Basically, these cases are considered independently, i.e., the multiple classes can be linearly separated. Assume that a training data-set includes sensory data values is available and the status values $\{-1, 0, 1\}$ are allocated to the three classes in which the original sensory data can be classified, e.g.,

$$y_i = \begin{cases} -1, & \text{if } i\text{-th sample} \in -1, \\ 0, & \text{if } i\text{-th sample} \in 0, \\ 1, & \text{if } i\text{-th sample} \in 1. \end{cases} \quad (1)$$

The three classes are assumed to be separable. Then, it is feasible to obtain a function $f(x)$ subject to the condition of

$$y_i \cdot f(x) > 0, \quad \forall i.$$

The largest margin Π within the hyperplane is selected among the three classes: this can also be regarded as the constrained maximization problem

$$\underset{\beta, \beta_0, \|\beta\|}{argmax} \{\Pi\}$$
$$\text{subject to: } y_i \cdot f(x) \geq \Pi, \quad i = 1, \ldots, n.$$

In practical classification issues, the multiple classes may be non-separable, i.e., the three categories interlace. In these cases, SVMs can still be used and some samples may be on the incorrect side of the separation line. For the separable cases, the optimization is updated by revising the condition as

$$y{\cdot}(\beta_0 + x_i{\cdot}\beta) \geq \Pi - \xi = \Pi{\cdot}(1 - \xi_i) \quad \forall i, \tag{2}$$

in which the slack variables ξ_i satisfy the following conditions: $\xi_i \geq 0$ (the points on the incorrect side of the margin area are labeled with $\xi_i^* > 0$, while the points on the correct side are marked as $\xi_i = 0$) and $\sum_{i=1}^{n} \xi_i \leq \mathbb{R}$. Then, the constrained optimization problem is converted to

$$\underset{\beta,\beta_0}{argmin}\left\{\frac{1}{2} \parallel \beta \parallel + \gamma \sum_{i=1}^{n} \xi_i\right\}$$

$$\text{subject to: } y{\cdot}(x_i{\cdot}\beta + \beta_0) \geq 1 - \xi_i, \quad i = 1, \ldots, n.$$

In the above problem, the γ (i.e., regularization parameter) determines the relationship between the sum of values of the slack variables and the margin.

3.4. Labeling: value-to-status mapping

Algorithm 1. Continuous value to discrete status

In the case that the labels of machine/component status is unavailable, the value-to-status mapping algorithm in this subsection is adopted to achieve the target. We consider the membership function set (represented as $\mathbb{MF}$) and the discrete value set (denoted with $\mathbb{DS}$) as correlated arrays whose paired elements including the same index should be treated as an integral unit. A continuous sensory data value can be mapped to multiple discrete values in case that the probability density of the fitted Gaussian distribution includes several terms. This is because an identical numerical characteristic is reflected in each term, i.e., the values indicated by a term are closer in semantic than those represented with other terms.

The proposed Algorithm 1 (Continuous Value to Discrete Status Mapping) is adopted for this sensory data to discrete status mapping process. Specifically, the membership degree of sensory data will be calculated when the raw data is obtained. And the status values $\{-1, 0, 1\}$ closest to the membership degree is selected as the current discrete state. In particular, the hyperplane F_0 in the space $\mathbb{R}^p$ is defined as

$$F_0 = \{x | f(x) = x\beta + \beta_0 = 0\}, \tag{3}$$

where the parameter $\beta \in \mathbb{R}^p$ is of normalization $\beta = 1$. Then, the status classification is according to the selection of $f(x)$ (and as a result of F_0): for a new data sample $x^i \in \mathcal{S}$, we classify

$$y_i = \begin{cases} -1, & \text{if } f(x^{\text{new}}) \in -1, \\ 0, & \text{if } f(x^{\text{medium}}) \in 0, \\ 1, & \text{if } f(x^{\text{heavy}}) \in 1. \end{cases} \tag{4}$$

In the data training process, a sample set of continuous sensory values is collected and the value-to-status mapping algorithm (see Algorithm 1) is invoked to calculate the discrete values (if the status labels are unavailable). In particular, (1) a continuous sensory value is mapped into a discrete one from step 2 to step 8; (2) an basic discrete state is converted into the expected format from step 11 to step 14.

3.5. Stage-level RUL prediction

The objective of this module is to estimate the variable value y (i.e., the RUL value of the current stage) using prediction variables x resulted from the regression calculations. Similar to the health stage classification step, this process can also be implemented with different machine/deep learning algorithms according to special features/requirements.

```
   Input: a set of sample data with continuous values ;
   Output: a discrete status value;
1  DS ⇐ S;
2  if Is_Cotinuous(D) then
3      < MF, DC >⇐ MF − DV(S)  ▷ calculate the discrete data sets and determine
       membership degrees;
4      for each sample s ∈ S do do
5          f_DC(s) ← max f_DC(s)|DC ∈ DC, f_DC ∈ MF   ▷ Find the membership
           function with largest value;
6          s ↔ DC                  ▷ s is changed into discrete value;
7          s ← DC          ▷ The sample is replaced with discrete value set;
8      end
9  end
10 else
11     for sample s ∈ S do do
12         DC ← REFINE(S)          ▷ Convert into expected form DC;
13         s ↔ DC;
14     end
15 end
16 return DS;
```

Therefore, the main idea is to obtain a function that models the tradeoff between the target variables and the predictor: $y \simeq f(x)$.

This section provides an example of adopting $\in$-SVR for stage-level RUL prediction. In $\in$-SVR, the symbol $\in$ determines the number of errors permitted in the algorithm. The objective of $\in$-SVR selection is to obtain an affine function $f(x)$ satisfied to: for all the training samples, there are maximum $\in$ deviations of $f(x)$ from the targets y and this function should also be possibly flat. We introduce the slack variables of ζ_i and ζ_i^* into the optimization problem and it can be converted into

$$\left\{ \parallel \omega \parallel^2 + C \sum_{i=1}^{N} (\zeta_i + \zeta_i^*) \right\} \tag{5}$$

$$\text{subject to}: \begin{cases} y_i - < \omega, x_i > -b \leq \in + \zeta_i \\ < \omega, x_i > +b - y_i \leq \in + \zeta_i^*, \\ \zeta_i, \zeta_i^* \geq 0. \end{cases} \quad (6)$$

Specifically, the parameter $C > 0$ determines the tradeoff between the errors tolerance (larger than $\in$) and the flatness of target function. The kernel-trick similar to SVM is adopted to analyze the nonlinear relationships between the parameters x and y. Namely, the input data is mapped into a feature-space of higher dimension and a linear $\in$-SVR is used on this space of new high dimension. The result of this SVR calculation is a model to be employed to estimate the RULs in each stage.

The SVR model trained from the health stage classification step is now adopted to estimate the RUL of the current stage of the specific components. The context is stated as follows: the sensory value of the target component is continuously monitored and the objective is to estimate when this component is likely to fail based on the historical data. The monitoring of the target component is modeled using a time series data $U = u_1, \ldots, u_l(U)$. However, the information about the last monitoring event $l(U)$ is provided in the previous subsection (status classification). The per-status RUL estimation process introduced in this subsection aims at estimating the duration between the failure instant and $l(U)$ of the testing component.

Firstly, we split the time series U into sliding windows of size L. For these sliding windows, we should also note the cases of overlapping as follows: By converting the previous time instant, a new window is created. Totally, the time series U has $n_U = l + l(U) - L$ such windows. The trend features are extracted following the steps in the previous subsection to obtain a feature vector for each of these sliding windows. The SVR model predicts the time (from the last event of the sliding window) at which a failure is supposed to occur when the feature vector x_k of the kth window of U is provided. Therefore, the model prediction value y_k for window k indicates the failure time instant of $f_k = L + k - 1 + y_k$. Finally, this will result in n_U predictions: $f_1, \ldots, f_{n_U}$ for the estimated failure occurrence events. We select a mean of these values the final RUL prediction of U.

In prognostic techniques, it is also fairly important to add a confidence interval measurement to the RUL prediction. Specifically, the error covariance matrix is used to calculate the confidence interval of the RUL prediction in the estimation phase. Given the uncertainty of the estimated results between the actual measurements and predicted state, the covariance matrix of estimation error is updated after each phase. This can be either the variance or entropy. $P(1,1)$ includes the uncertainty of the conjectured state. Provided this uncertainty, a confidence interval of 95% can be introduced for this estimation. The lower and upper bounds of $x_{\text{lbd}} = x(l) - 2.567 \cdot P(1,1)$ and $x_{\text{ubd}} = x(l) + 2.567 \times P(1,1)$ can be calculated. Besides, these values can also be predicted as the failure threshold to determine the lower and upper confidence intervals.

4. Performance evaluation

This section presents the performance evaluation of the MS-RUL framework using both open-source and real production-system data.

4.1. Experimental setting

To validate the efficacy of the proposed solution, we implement MS-RUL using SVM, GB (Gradient Boost), LSTM and CNN for performance comparison. The implemented CNN architecture is based on our previous work [1] on RUL prediction. The results of the following reference algorithms are also included:

- LR (Linear Regression): linear regression is the basic linear approach to characterize the relationship between a explanatory variables (or independent variables) and scalar response (or dependent variable).
- CNN (Convolutional Neural Network): In reference [1], the CNN is firstly adopted for the RUL prediction. This network architecture is composed of multi-variate time-series input, 2 convolutional filtering layers, 2 polling filtering layers and one fully connected layer.
- LSTM (Long Short Term Memory) [7]: In the LSTM algorithm, we set the sliding window size as 50. All the simulations for this algorithm are conducted using original data.
- SVR (Support Vector Regression) [17]: The SVR uses the same RBF kernel as the SVM, with the objective to find the objective RUL as a continuous value (instead of a discrete value as the status classification). Note that this SVR algorithm also represents the solution without using multiple stages.

For the CNN-based MS-RUL, the network structure is adjusted for the multi-variate time-series data. Specifically, we use two pairs of convolutional layers, two pairs of pooling layers and one fully-connected multi-layer perceptron. All end layer feature maps are concatenated into a vector as the MLP input for RUL estimation. Training stage involves the CNN parameters estimation by standard back propagation algorithm using stochastic gradient descent method to optimize objective function, which is cumulative square error of the CNN model.

Theoretically, the optimal SVM parameters are obtained by adopting grid search technique (cross validation) over the training data. However, the grid search algorithm incurs polynomial complexity of $M \times N$ times, in which M denotes the optional amount of C and N represents the optional amount of γ. To reduce the time complexity, a two-phase grid search is conducted to guarantee calculation efficiency and optimize the parameter selection process:

- In the first stage, we select a larger step size and testing interval. For instance, the step size is configured as 4 and the testing interval is $\{2^{-15}, 2^{15}\}$. Within a large range, this step is critical to generally locate the optimal values $(C_{\text{opt}}, \gamma_{\text{opt}})$ of the kernel.
- In the second stage, we adopt a testing range covering $(C_{\text{opt}}, \gamma_{\text{opt}})$ and smaller step size. Experimental results demonstrate that the optimal values $(C_{\text{opt}}, \gamma_{\text{opt}})$ of the kernel is $(2^0, 2^1)$. Prior studies suggest that the optimal value of C is often located in the interval larger than 2^0. And the optimal value of γ is more likely to appear in interval smaller than 2^0. The search intervals of γ and C are $\gamma = \{2^{-9}, 2^{-8}, \ldots, 2^2\}$ and $C = \{2^{-2}, 2^{-1}, \ldots, 2^9\}$, respectively. Then, the optimal parameter setting can be obtained by comparing and selecting the combination with the highest classification accuracy, e.g., $\text{accuracy} = 0.932$ for $C = 2^0, \gamma = 2^{-1}$, $\text{accuracy} = 0.926$ for $C = 2^1, \gamma = 2^{-2}$, and $\text{accuracy} = 0.936$ for $C = 2^0, \gamma = 2^1$.

4.2. Evaluation with real sensory data

In this subsection, we present and discuss the evaluation results measured from a real shaft production system. Specifically, this production system includes the acoustic, current, and vibration sensors to provide real-time feedback information. The data profile is presented in Fig. 2. The original sensory data from different channels need to be merged as the input for model training. In this production system, tool wear (caused by surface cutting operations) is one of main factors that impact on the shaft quality (e.g., surface roughness, roundness, and diameter). Therefore, we use the evaluation algorithms to predict the tool RUL and the tool conditions are classified into three states (new, medium, heavy).

For this evaluation data set from real shaft production system, the labels to classify the machine states are available and these states represent the tool conditions.

Table 3 presents the average prediction accuracy of the different evaluated algorithms. As show in the table, the proposed MS-RUL is able to achieve approximately 90% accuracy. While the reference schemes

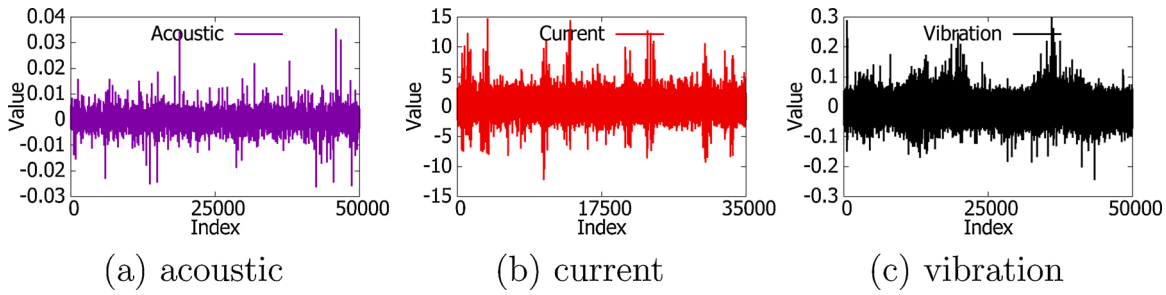

Fig. 2. Profile of the real sensory data from production system.

Table 3
Average prediction accuracy.

Algorithm	Avg. prediction accuracy (%)
LR	71.5 ± 3.2
SVR	76.5 ± 3.5
LSTM	81.3 ± 2.9
CNN	83.5 ± 2.7
MS-RUL	**Avg. prediction accuracy (%)**
MS-RUL (GB)	88.2 ± 2.1
MS-RUL (SVM)	90.3 ± 1.8
MS-RUL (LSTM)	90.5 ± 1.5
MS-RUL (CNN)	91.2 ± 1.1

are around 71–83 percent. This is because the per-state RUL estimation can overcome the uncertainties in different stages, e.g., caused by cutting system dynamics, tool mis-alignment, and other faulty conditions.

Fig. 3 presents the accuracy comparison in terms of RUL versus the number of cycles. This accuracy is expressed as 1 − MAPE (Mean absolute percentage error), in which $\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{Actual}-\text{Predicted}}{\text{Actual}}\right|$. Specifically, each cycle is approximately 6.13 s according to the average RUL of tool data. As shown in the results: (1) LR models the RUL as linear-type variation trend and the result is normally too large/small than the actual value; (2) SVR uses a single weight vector to determine the whole RUL and thus is often unable to capture the special tool conditions in different stages. For our proposed MS-RUL solution, we present two typical results of MS-RUL (SVM) and MS-RUL (CNN) in Fig. 3. The microscopic results obtained from MS-RUL (GB) and MS-RUL (LSTM) are very close to the above two, respectively. As it can be observed, CNN performs slightly better than SVM in the estimation accuracy.

An important module in our proposed MS-RUL is the value-to-status mapping algorithm (Algorithm 1 in Section 3.4). This module is used to classify the data into different states if the labels are unavailable. Therefore, the accuracy of the Algorithm 1 is critical to the performance of MS-RUL. In our experiment, we evaluate the classification accuracy of Algorithm 1 for different types of real sensory data by comparing with the ground truth of the labels provided. As indicated in the results, the overall state classification accuracy is 89.3% by integrating different types of data.

Table 4 presents the accuracy of the linear, RBF (Radial Basis Function) and polynomial kernels. As the results indicate, the classification accuracy of the RBF kernel is approximately 2% higher than the polynomial and linear. Therefore, the RBF kernel is selected as the SVM

Table 4
Classification accuracy of different SVM kernels.

Kernel	Avg. classification accuracy (%)
Linear	90.3 ± 0.35
RBF	93.7 ± 0.21
Polynomial	91.3 ± 0.32

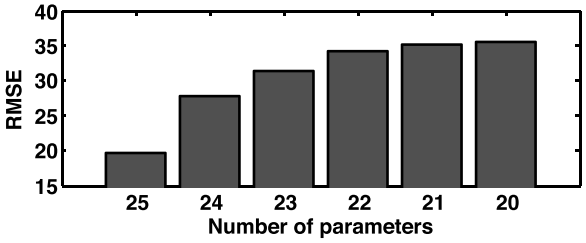

Fig. 4. RMSE for different numbers of input parameters.

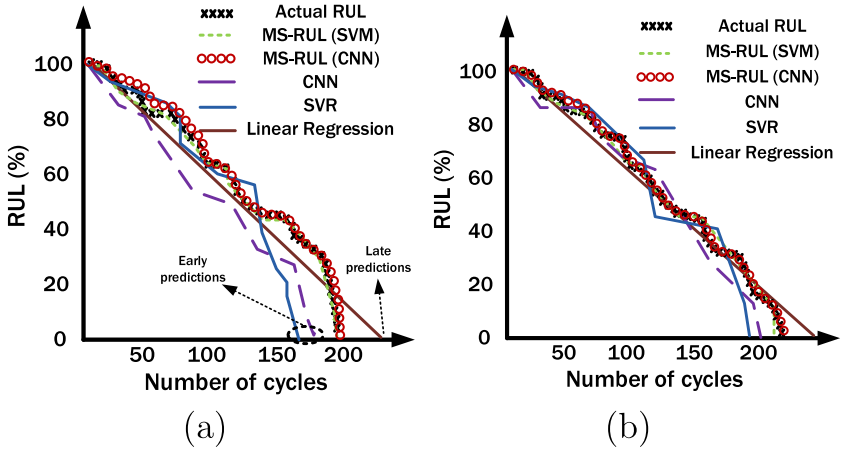

Fig. 3. RUL degradation trends predicted by the evaluated algorithms.

kernel in our proposed MS-RUL.

SVM performance is known to get affected by sensational variations or peaks in time series data [43]. As far as we can see, such influence can be mitigated by involving more useful sensory data in the RUL prediction. This result in Fig. 4 is evaluated with the FD003 dataset of C-MAPSS. As it can be observed, a higher accuracy can be generally achieved if more types of sensory data values are involved.

4.3. Evaluation with C-MAPSS data [40]

The C-MAPSS data set is commonly used in existing studies [2,40] for performance comparison. This data set includes multiple multivariate time-series signals generated by a simulation model developed based on the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). Totally, 26 signals/inputs (e.g., temperature, pressure, rotation speed, and other sensory values) are generated for the open-loop system operations. Specifically, three of these inputs represent the configuration of the system environments, twenty one are the sensory data records and the remaining ones denote the number of cycles and engine IDs.

Each data set stands for a different engine from the same operation system. Each engine includes different system elements, e.g., high/low pressure compressors, turbines, fan and other components. The flowchart that describes how various subroutines are assembled and main elements of the engine model in the simulation are presented in Fig. 1 of reference [40]. Interested readers can refer to [40] for more information. The engine operations are normal at the start and a fault is introduced at a time before the system failure.

The dataset is featured by single failure mode and operating condition. The text files for this data set are listed as follows:

- "TRAIN FD001-4": include 100, 260, 100, 248 training units. The measurement data starts at a degradation level with similar patterns that is considered healthy and stop when failure is reached.
- "TEST FD001-4": include 100, 259, 100, 249 testing units. The time series data is incomplete and finishes before engine failure. The target is to predict the remaining lifetime.
- "RUL FD001-4": specify the actual RUL values which are used as the baseline (standard values) for comparison. Fig. 5 presents the actual values used as the ground truth for comparison.

4.3.1. Evaluation metrics

Score: This evaluation metric is defined by the authors of the C-MAPSS data set [40] and is widely used in existing studies [12,28] for performance comparison. And the mathematical expression is as follows.

$$\text{Score} = \begin{cases} \sum_{i=1}^{n} exp\left\{-\frac{d}{a_1}\right\} - 1, & \text{if } d < 0, \\ \sum_{i=1}^{n} exp\left\{-\frac{d}{a_2}\right\} - 1, & \text{if } d \geq 0, \end{cases} \tag{7}$$

where n is the total number of units under test, d = Estimated RUL − True RUL, $a_1 = 10$ and $a_2 = 13$.

Root Mean Square Error (RMSE): RMSE is also a commonly-used metric to compare the accuracy of competing algorithms. In particular, the mathematical expression of RMSE is as follows.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\text{Estimated RUL} - \text{True RUL})^2}{n}}.$$

Table 5
Score [Equation (7)] of the evaluated algorithms.

Baseline algorithms	FD001	FD002	FD003	FD004
LR	2200	19,012	2629	14,173
SVR	2394	18,074	2137	8210
LSTM	983	17,568	1539	9038
CNN	1372	16,048	1546	8843
MS-RUL Performance	FD001	FD002	FD003	FD004
MS-RUL (SVM)	765	14,192	1335	6568
MS-RUL (GB)	772	13,893	1346	6423
MS-RUL (LSTM)	748	13,761	1293	6375
MS-RUL (CNN)	732	13,566	1262	6126

Table 6
RMSE of the evaluated algorithms.

Algorithm	FD001	FD002	FD003	FD004
LR	23.45	33.23	23.36	34.79
SVR	21.74	31.76	22.42	30.87
LSTM	18.98	30.87	20.62	29.83
CNN	19.7	30.39	20.18	29.69
MS-RUL Performance	FD001	FD002	FD003	FD004
MS-RUL (SVM)	17.12	29.79	19.57	28.27
MS-RUL (GB)	17.92	28.63	20.56	28.13
MS-RUL (LSTM)	17.26	28.15	19.13	27.78
MS-RUL (CNN)	16.89	27.52	18.62	27.12

4.3.2. Evaluation results

Table 5 lists the score of the different algorithms for performance comparison. In general, the proposed MS-RUL achieves the lowest score (i.e., the highest accuracy) for all the data sets. For all the algorithms, the score results of FD002 and FD004 are obviously higher than those of FD001 and FD003. This is because of the larger number of engines and more faulty conditions in simulations. Due to the higher complexity, the difficulty in RUL estimation also substantially increases. The superiority of our proposed solution indicate the importance of identifying the health stage in RUL prediction.

Table 6 presents the RMSE of different algorithms for the turbofan datasets. The pattern is similar to that shown in Table 5. Our proposed MS-RUL still achieves the lowest RMSE among all the evaluation methods. In particular, the deep learning algorithms (CNN and LSTM) perform better than shallow learning (GB and SVM) in the MS-RUL framework. Both the RMSE and score values reflect the accuracy improvement of using deep learning algorithms in MS-RUL. This is because both the CNN and LSTM are able to explore more system degradation information than the other shallow learning methods [1].

In order to have a microscopic view of the evaluation results, the comparison of FD001 results are shown in Fig. 6. It can be seen from the

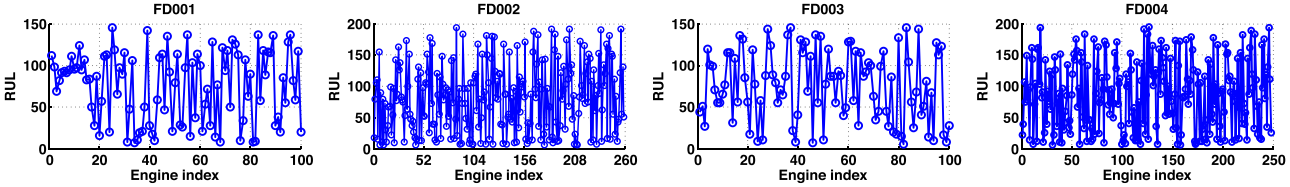

Fig. 5. Actul RUL plots of the C-MAPSS data set.

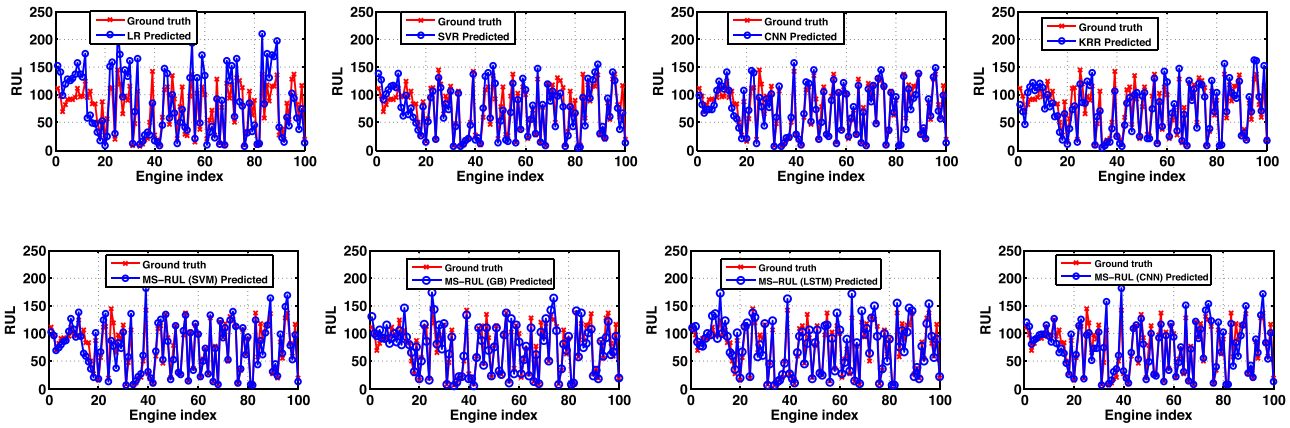

Fig. 6. Ground truth versus predicted values for the FD001 dataset.

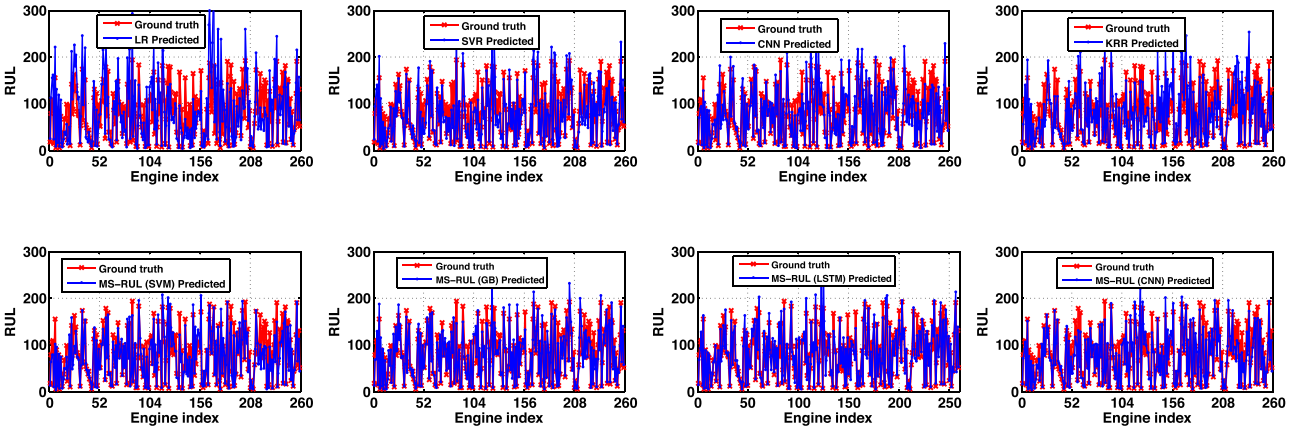

Fig. 7. Actual versus predicted values for the FD002 dataset.

Table 7
Efficiency (execution time in units of seconds) of the evaluated algorithms.

Algorithms	FD001	FD002	FD003	FD004
LR	4.3	17.5	4.6	18.8
SVR	15.7	78.6	18.3	80.5
LSTM	35.3	92.2	43.6	97.6
CNN	22.6	85.2	26.3	91.2
MS-RUL (CNN)	29.7	91.8	30.2	97.5

results' trend that the predicted values of our proposed MS-RUL methods (both in SVM and CNN) track closely to the actual RUL values, and achieve the lowest deviations. For other competing solutions, we can observe large differences from the ground truth.

Table 8
Accuracy results for different buffer sizes.

Data-set	Real data from shaft production systems				
Buffer size (s)	10	20	30	40	50
Accuracy (%)	87.6 ± 2.6	89.3 ± 2.2	91.2 ± 1.1	88.7 ± 1.9	87.3 ± 2.5
Data-set	FD002				
Buffer size (s)	10	20	30	40	50
RMSE	13965	13773	13566	13792	13916
SCORE	28.47	28.13	27.52	28.26	28.74

The experimental results for the FD002 dataset is shown in Fig. 7 for comparison. As the number of engines increases, larger variations and differences are reflected in the RUL value comparisons. For our proposed solution, the difference and variation is obviously lower than the reference algorithms. The results reflect our proposed algorithm is able to achieve higher performance superiority in more complex operation conditions.

Based on the above evaluation results using both real sensory and C-MAPSS data, we can have the following conclusions: (1) MS-RUL outperforms the baseline algorithms and the superiority becomes larger for complex data set; (2) the deep learning algorithms (CNN and LSTM) perform better than shallow learning (SVM and GB) in MS-RUL for the joint classification-regression.

4.3.3. Execution time (efficiency)

Table 7 presents the execution time (efficiency) of the evaluated algorithms. As it can be observed from the table that LR algorithm achieves the lowest execution time. This is because the complexity of LR is obviously lower than other algorithms. However, the average prediction accuracy of LR is also the lowest. The deep learning algorithms (e.g., CNN, LSTM, MS-RUL) generally incur higher execution time and achieve higher prediction accuracy.

4.4. Impact of buffer size

This subsection presents the results to validate the impact of buffer data size in MS-RUL. We evaluate the buffer sizes from 10 to 50 s using both data-sets from the shaft production system and C-MAPSS (FD002). As shown in Table 8, the buffer size of 30 s is able to achieve the highest

Table 9
Summary of experimental results.

Data-set	Tool wear in shaft production line				
Algorithm	MS-RUL (CNN)	CNN	LSTM	SVR	LR
Accuracy	91.2 ± 1.1%	83.5 ± 2.7%	81.3 ± 2.9%	76.5 ± 3.5%	71.5 ± 3.2%
Data-set	FD002 in C-MAPSS [36]				
Algorithm	MS-RUL (CNN)	CNN	LSTM	SVR	LR
Score	13,566	16,048	17,568	18,074	19,012
RMSE	27.52	30.39	30.87	31.76	33.23

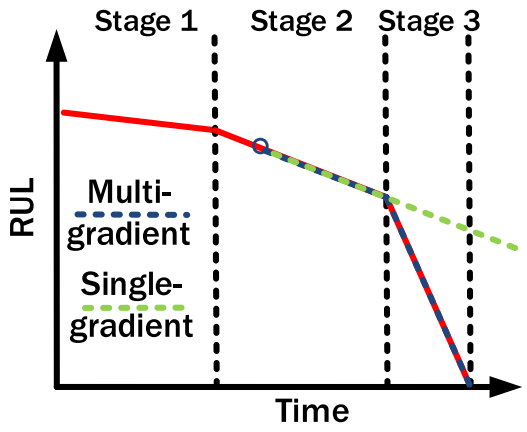

Fig. 8. Multi-gradient versus single-gradient RUL estimation.

accuracy and this buffer size is adopted as the default setting in our experiments.

4.5. Scientific justification

In summary, the scientific justification for our proposed MS-RUL can be summarized as follows:

- **Empirical/Experimental evidence**: The evaluation results with the real sensory data and C-MAPSS data-set [40] prove that MS-RUL is able to achieve higher RUL prediction accuracy than the existing models/algorithms. Those two data-sets of shaft and turbofan represent the typical industrial application scenarios with machinery components. As summarized in Table 9, experimental results demonstrate that MS-RUL achieve more than 6% accuracy improvement than the existing deep learning models (e.g., CNN and LSTM) in the evaluated data-sets. Therefore, it is desirable to adopt our MS-RUL in these real problems to improve the efficiency in predictive maintenance to guarantee system reliability and reduce the cost.
- **Scientific idea**: The scientific idea to develop this multi-stage framework is that RUL variation is a complex process, and multi-gradient based estimation is more accurate than the single-gradient estimation as shown in Fig. 8. A typical example of single-gradient based algorithm is the linear regression (LR). Since the only gradient is used to predict the RUL, the complex RUL variations cannot be captured and early/late predictions often incur high estimation errors.
- **Prior experiences**: The prior experiences from the existing studies (e.g., [12,27,28,37]) demonstrate that RUL variation in many industrial systems/components is a multi-stage process. However, the stage number and degradation trend are significantly different in various application scenarios. Thus, we first develop the feature extraction and value-to-status mapping (Algorithm 1) to automatically estimate the stage number. Then, the stage-level RUL estimation is performed according to classified healthy stages.
- **Plausible mechanisms**: Recent advances in machine/deep learning algorithms provide effective solutions for the classification and regression tasks. The healthy stage classification and stage-level RUL estimation are basically classification and regression problems, respectively. These machine/deep learning algorithms/models are effective in our health stage classification and stage-level RUL estimation problems. Furthermore, the experimental results of MS-RUL (e.g., MS-RUL (CNN) and MS-RUL (SVR)) demonstrate that our proposed framework is able to overcome the limitations of using stand-alone classification/regression models.

5. Conclusion and discussion

RUL prediction is vitally significant to the predictive maintenance in industrial systems. This paper develops an accurate prediction scheme dubbed MS-RUL that effectively integrates the classification-regression algorithms. The objective is to accurately diagnose the faults and reduce the cost in predictive maintenance. This paper proposes a Joint Classification-Regression Method that predicts RUL in multiple states to improve the estimation accuracy. The approach was evaluated using turbofan engines and shaft production system data. Results demonstrate an accuracy improvement around 6.5% over the competing algorithms in RUL prediction. As the future work, we will consider: (1) using different kernels/algorithms for the classification and regression phases to achieve better accuracy; (2) developing new solution to cope with the missing label problem; (3) adopt transfer learning [41,42] algorithms for accurate multi-stage RUL prediction; (4) introducing pre-processing techniques like data cleaning and integration to cope with data corruption/noise problems.

Acknowledgement

This work is supported by the A*STAR Industrial Internet of Things Research Program under the RIE2020 IAF-PP Grant A1788a0023, and partially supported by National Natural Science Foundation of China (No. 51835009). Min Wu is the corresponding author.

Conflict of interest: The authors declare no conflict of interest.

References

[1] Babu GS, Zhao P, Li XL. Deep convolutional neural network based regression approach for estimation of remaining useful life. In: International conference on database systems for advanced applications. Springer, Cham; 2016. p. 214–28.
[2] Lei Y, Li N, Gontarz S, Lin J, Radkowski S, Dybala J. A model-based method for remaining useful life prediction of machinery. IEEE Trans Rel 2016;65(3):1314–26.
[3] Ren L, Sun Y, Cui J, Zhang L. Bearing remaining useful life prediction based on deep autoencoder and deep neural networks. J Manuf Syst 2018;48:71–7.
[4] Ren L, Cui J, Sun Y, Cheng X. Multi-bearing remaining useful life collaborative prediction: a deep learning approach. J Manuf Syst 2017;43:248–56.
[5] Qian Y, Yan R, Hu S. Bearing degradation evaluation using recurrence quantification analysis and Kalman filter. IEEE Trans Instrum Meas 2014;63(11):2599–610.
[6] Lall P, Lowe R, Goebel K. Extended Kalman filter models and resistance spectroscopy for prognostication and health monitoring of leadfree electronics under vibration. IEEE Trans Rel 2012;61(December (4)):858–71.

[7] Zhang J, Wang P, Yan R, Gao RX. Long short-term memory for machine remaining life prediction. J Manuf Syst 2018;48:78–86.
[8] Qian Y, Yan R. Remaining useful life prediction of rolling bearings using an enhanced particle filter. IEEE Trans Instrum Meas 2015;64(10):2696–707.
[9] Soualhi A, Medjaher K, Zerhouni N. Bearing health monitoring based on Hilbert-Huang transform, support vector machine, and regression. IEEE Trans Instrum Meas 2014;64(1):52–62.
[10] Jin X, Sun Y, Que Z, Wang Y, Chow TW. Anomaly detection and fault prognosis for bearings. IEEE Trans Instrum Meas 2016;65(9):2046–54.
[11] Singleton RK, Strangas EG, Aviyente S. Extended Kalman filtering for remaining-useful-life estimation of bearings. IEEE Trans Ind Electron 2015;62(3):1781–90.
[12] Khelif R, Chebel-Morello B, Malinowski S, Laajili E, Fnaiech F, Zerhouni N. Direct remaining useful life estimation based on support vector regression. IEEE Trans Ind Electron 2017;64(3):2276–85.
[13] Li Y, Kurfess T, Liang S. Stochastic prognostics for rolling element bearings. Mech Syst Signal Process 2000;14(September (5)):747–62.
[14] Oppenheimer CH, Loparo KA. Physically based diagnosis and prognosis of cracked rotor shafts. Proc AeroSense 2002;(July):122–32.
[15] Gebraeel N, Lawley M, Liu R, Parmeshwaran V. Residual life predictions from vibration-based degradation signals: a neural network approach. IEEE Trans Ind Electron 2004;51(June (3)):694–700.
[16] Soualhi A, Razik H, Clerc G, Doan DD. Prognosis of bearing failures using hidden Markov models and the adaptive neuro-fuzzy inference system. IEEE Trans Ind Electron 2014;61(6):2864–74.
[17] Kim H-E, Tan ACC, Mathew J, Choi B-K. Bearing fault prognosis based on health state probability estimation. Expert Syst Appl 2012;39(April (5)):5200–13.
[18] Zaidi SSH, Aviyente S, Salman M, Shin KK, Strangas EG. Prognosis of gear failures in DC starter motors using hidden Markov models. IEEE Trans Ind Electron 2011;58(May (5)):1695–706.
[19] Aydemir G, Acar B. Anomaly monitoring improves remaining useful life estimation of industrial machinery. J Manuf Syst 2020;56:463–9.
[20] Wang P, Gao RX. Adaptive resampling-based particle filtering for tool life prediction. J Manuf Syst 2015;37:528–34.
[21] Wu W, Hu J, Zhang J. Prognostics of machine health condition using an improved ARIMA-based prediction method. In: Proc. 2nd IEEE conf. ind. electron. appl.; 2007. p. 1062–7.
[22] Ali JB, Chebel-Morello B, Saidi L, Malinowski S, Fnaiech F. Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network. Mech Syst Signal Process 2015;56(May):150–72.
[23] Mosallam A, Medjaher K, Zerhouni N. Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction. J Intell Manuf 2016;27(October (5)):1037–48.
[24] Gebraeel N, Elwany A, Pan J. Residual life predictions in the absence of prior degradation knowledge. IEEE Trans Rel 2009;58(March (1)):106–17.
[25] Orchard ME, Hevia-Koch P, Zhang B, Tang L. Risk measures for particle-filtering-based state-of-charge prognosis in lithium-ion batteries. IEEE Trans Ind Electron 2013;60(11):5260–9.
[26] Miao Q, Xie L, Cui H, Liang W, Pecht M. Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. Microelectron Rel 2013;53(6):805–10.
[27] Ramasso E, Rombaut M, Zerhouni N. Joint prediction of continuous and discrete states in time-series based on belief functions. IEEE Trans Cybern 2013;43(June (1)):37–50.
[28] Javed K, Gouriveau R, Zerhouni N. A new multivariate approach for prognostics based on extreme learning machine and fuzzy clustering. IEEE Trans Cybern 2015;45(January (12)):2626–39.
[29] Xue F, Bonissone P, Varma A, Yan W, Eklund N, Goebel K. An instance-based method for remaining useful life estimation for aircraft engines. J Fail Anal Prev 2008;8(April (2)):199–206.
[30] Wang J, Wang P, Gao RX. Enhanced particle filter for tool wear prediction. J Manuf Syst 2015;36:35–45.
[31] Khelif R, Malinowski S, Morello BC, Zerhouni N. Unsupervised kernel regression modeling approach for RUL prediction. In: Proc. Eur. conf. prognostics health manage. soc.; 2014. p. 1–7.
[32] Khelif R, Malinowski S, Chebel-Morello B, Zerhouni N. RUL prediction based on a new similarity-instance based approach. In: Proc. IEEE int. symp. ind. electron.; 2014. p. 2463–8.
[33] Emmanuel. Investigating computational geometry for failure prognostics. Int J Progn Health Manag 2014;005(November):1–18.
[34] Qin T, Zeng S, Guo J. Robust prognostics for state of health estimation of lithium-ion batteries based on an improved PSOCSVR model. Microelectron Rel 2015;55(September (9–10)):1280–4.
[35] Wu T, Ma X, Yang L, Zhao Y. Proactive maintenance scheduling in consideration of imperfect repairs and production wait time. J Manuf Syst 2019;53:183–94.
[36] Soualhi A, Medjaher K, Zerhouni N. Bearing health monitoring based on Hilbert-Huang transform, support vector machine, and regression. IEEE Trans Instrum Meas 2015;64(January (1)):52–62.
[37] Patil MA, Tagade P, Hariharan KS, Kolake SM, Song T, Yeo T, Doo S. A novel multistage support vector machine based approach for lithium-ion battery remaining useful life estimation. Appl Energy 2015;159(December):285–97.
[38] Loutas TH, Roulias D, Georgoulas G. Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression. IEEE Trans Rel 2013;62(December (4)):821–32.
[39] Dong H, Jin X, Lou Y, Wang C. Lithium-ion battery state of health monitoring and remaining useful life prediction based on support vector regression-particle filter. Power Sources 2014;271(December):114–23.
[40] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. In: PHM 2008. International conference on prognostics and health management 2008; 2008. p. 1–9.
[41] Sun C, Ma M, Zhao Z, Tian S, Yan R, Chen X. Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. IEEE Trans Ind Inform 2019;15(4):2416–25.
[42] Ragab M, Chen Z, Wu M, Foo CS, Kwoh CK, Yan R, Li X. Contrastive adversarial domain adaptation for machine remaining useful life prediction. IEEE Trans Ind Inform 2020. https://doi.org/10.1109/TII.2020.3032690.
[43] Davo F, Vespucci MT, Gelmini A, Grisi P, Ronzio D. Forecasting Italian electricity market prices using a neural network and a support vector regression. Proc. of AEIT international annual conference 2016:1–6.
[44] Chen L, Xu G, Zhang S, Yan W, Wu Q. Health indicator construction of machinery based on end-to-end trainable convolution recurrent neural networks. J Manuf Syst 2020;54:1–11.
[45] Wang J, Ma Y, Zhang L, Gao RX, Wu D. Deep learning for smart manufacturing: methods and applications. J Manuf Syst 2018;48:144–56.
[46] Chen Z, Wu M, Zhao R, Guretno F, Yan R, Li X. Machine remaining useful life prediction via an attention based deep learning approach. IEEE Trans Ind Electron 2020. https://doi.org/10.1109/TIE.2020.2972443.