

A Risk-Averse Remaining Useful Life Estimation for Predictive Maintenance

Chuang Chen, Ningyun Lu, *Member, IEEE*, Bin Jiang, *Fellow, IEEE*, and Cunsong Wang, *Student Member, IEEE*

Abstract—Remaining useful life (RUL) prediction is an advanced technique for system maintenance scheduling. Most of existing RUL prediction methods are only interested in the precision of RUL estimation; the adverse impact of over-estimated RUL on maintenance scheduling is not of concern. In this work, an RUL estimation method with risk-averse adaptation is developed which can reduce the over-estimation rate while maintaining a reasonable under-estimation level. The proposed method includes a module of degradation feature selection to obtain crucial features which reflect system degradation trends. Then, the latent structure between the degradation features and the RUL labels is modeled by a support vector regression (SVR) model and a long short-term memory (LSTM) network, respectively. To enhance the prediction robustness and increase its marginal utility, the SVR model and the LSTM model are integrated to generate a hybrid model via three connection parameters. By designing a cost function with penalty mechanism, the three parameters are determined using a modified grey wolf optimization algorithm. In addition, a cost metric is proposed to measure the benefit of such a risk-averse predictive maintenance method. Verification is done using an aero-engine data set from NASA. The results show the feasibility and effectiveness of the proposed RUL estimation method and the predictive maintenance strategy.

Index Terms—Long short-term memory (LSTM) network, predictive maintenance, remaining useful life (RUL) estimation, risk-averse adaptation, support vector regression (SVR).

I. INTRODUCTION

IMPROVING operation quality and reducing operation costs are two major concerns in many enterprises [1]–[3]. Unexpected downtimes often mean unbearably high costs. Orienting at Industry 4.0 or made in China 2025, maintenance technology is more attracted to predictability [4]. Based on online condition monitoring information, predictive maintenance (PdM) can output some indicators to show the levels of system health states or the estimation of their remaining useful life (RUL) [5]. These indicators can assist maintenance decision-making, such as to obtain the optimal maintenance time and the ordering time of spare parts. There

Manuscript received April 7, 2020; revised May 20, 2020; accepted June 27, 2020. This work was support by Natural Science Foundation of China (61873122). Recommended by Associate Editor Zhen Song. (*Corresponding author: Ningyun Lu.*)

Citation: C. Chen, N. Y. Lu, B. Jiang, and C. S. Wang, “A risk-averse remaining useful life estimation for predictive maintenance,” *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 2, pp. 412–422, Feb. 2021.

The authors are with College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China (e-mail: chenchuang@nuaa.edu.cn; luningyun@nuaa.edu.cn; binjiang@nuaa.edu.cn; wangcunsong@nuaa.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2021.1003835

is an increasingly common view that PdM outperforms traditional maintenance strategies such as time-based maintenance and corrective maintenance. It has attracted considerable attentions in recent years [6]–[10].

RUL estimation is the most important step in a PdM strategy. With the estimated RUL, one can schedule maintenance activities confidently for a degrading system [11]–[13]. There are many studies attempting to improve the precision of RUL estimation. However, few of them considered it from a maintenance perspective. Maintenance scheduling is the ultimate goal of RUL prognostics. In general, an under-estimated RUL is better than an over-estimated one under the same or close prediction error, because with over-estimated RUL, the risk of unexpected shutdowns will surge and may sometimes cause disastrous consequences. Therefore, it is necessary and significant to properly correct the misleading RUL estimation and then reduce possible wrong decisions of maintenance scheduling.

To do so, a risk-averse RUL estimation method is proposed for predictive maintenance. Firstly, features that can reflect system degradation trend are selected. Then, the latent structure between degradation features and the RUL labels is modeled by a support vector regression (SVR) algorithm and a long short-term memory (LSTM) network separately, since SVR and LSTM are currently two well recognized prognostics tools. To enhance the robustness of prognostics and increase their marginal utility, the SVR model and the LSTM model are combined via three connection parameters to generate a new hybrid model. The risk-averse adaptation is realized by optimizing the three connection parameters to minimize a cost function online. From the perspective of maintenance scheduling, a cost metric is also developed for measuring the benefit provided by risk-averse prognostics.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents a risk-averse RUL estimation method for predictive maintenance. The major steps include selection of degradation features, prognostic model training, risk-averse adaptation, and online prognostics. Verification is conducted using the NASA data set of aero-engines in Section IV. Section V draws the conclusions and discusses the possible future work.

II. RELATED WORK

This section presents the state-of-art of RUL prognostics in recent years. From the perspective of prognostic procedures, these existing RUL prognostics studies can be roughly divided into three branches: degradation state based, regression based, and pattern matching based prognostic methods [14].

For degradation state based methods, RUL is usually estimated in two steps: estimating system health state and then inferring the RUL using a failure threshold. For instance, a fuzzy c-means algorithm was employed to partition the run-to-failure condition monitoring data, and the final RUL was estimated by a multivariate health estimation model and a multivariate degradation prediction model [15]. In [16], the run-to-failure data were divided into four states using several belief functions, and RUL was estimated as the transition duration from the degraded state to the failure state. In [17], RUL was estimated by projection with the combination of degradation modeling and a particle filter algorithm, where the parameters of probability distribution at the last step of online updating were regarded as the final RUL distribution.

The regression based methods are devoted to predicting the evolution behavior of a degradation indicator so that the estimation of RUL can be obtained when the degradation indicator reaches an end-of-life (EOL) criterion. In this branch, support vector regression (SVR) is the mainstream technology. For example, in [18], a particle swarm optimization algorithm was employed to optimize the configuration parameters of SVR; the estimated RUL was obtained by the SVR based one-step time series prediction [19]. In [20], a particle filter and an SVR were combined to estimate the capacity degradation of a lithium-ion battery, and the RUL was defined by the probability distribution parameters.

The above two branches of prognostic methods are strongly dependent on the setting of such failure threshold, which is a challenging issue in practice. The pattern matching based method raises attentions alternatively. They aim to identify the right trend along with its EOL value from the sample datasets. In [21], the degradation was represented as one-dimensional health indicator trajectory using linear regression, and the RUL of the current sample was obtained by comprehensive analysis of similar trajectories. Similar ideas can be found in [22]–[24]. However, the pattern matching method relies on the quality of the sample's datasets. In the absence of matched

patterns, the estimated result is quite wayward.

Recently, two new RUL prognostics methods, which require no estimation of health states or setting of failure threshold, have been published [14], [25]. They attempt to build the direct mapping relationship between degradation features and RUL using data-driven techniques. In [14], SVR was used to build such a direct relationship. In [25], long short-term memory (LSTM) networks were employed. The two models have shown their advantages in higher prognostics precision.

Generally speaking, for a single prediction model, prediction precision shows a trend of diminishing marginal utility [26]. One possible solution is to develop a hybrid model which can maintain the positive parts of each modeling method and prevent the diminishing marginal utility problem of a simple prediction model. For this purpose, a hybrid model that combines the SVR and LSTM models is considered in this paper. Further, on the basis of [14] and [25], the risk-averse adaptation is investigated to contribute to predictive maintenance, which is the main contribution of this paper.

III. METHODOLOGY

A. Key Idea

The scheme of the proposed risk-averse RUL estimation based predictive maintenance strategy is shown in Fig. 1. It contains two phases: offline training and online application.

In the offline phase, the raw run-to-failure data (with known EOLs) are first transformed into a set of features using the degradation feature selection module. The features should reflect the degradation trend of the system. Next, the SVR and LSTM models are separately trained for learning the relationship between the features and the RULs. Also, the two models are fused by several connection parameters, and the parameters are determined using an evolutionary algorithm.

In the online phase, the online condition monitoring data (with unknown EOL) are collected and appended to get a section of trajectories, and then processed in the same manner as that in the offline phase to extract degradation features.

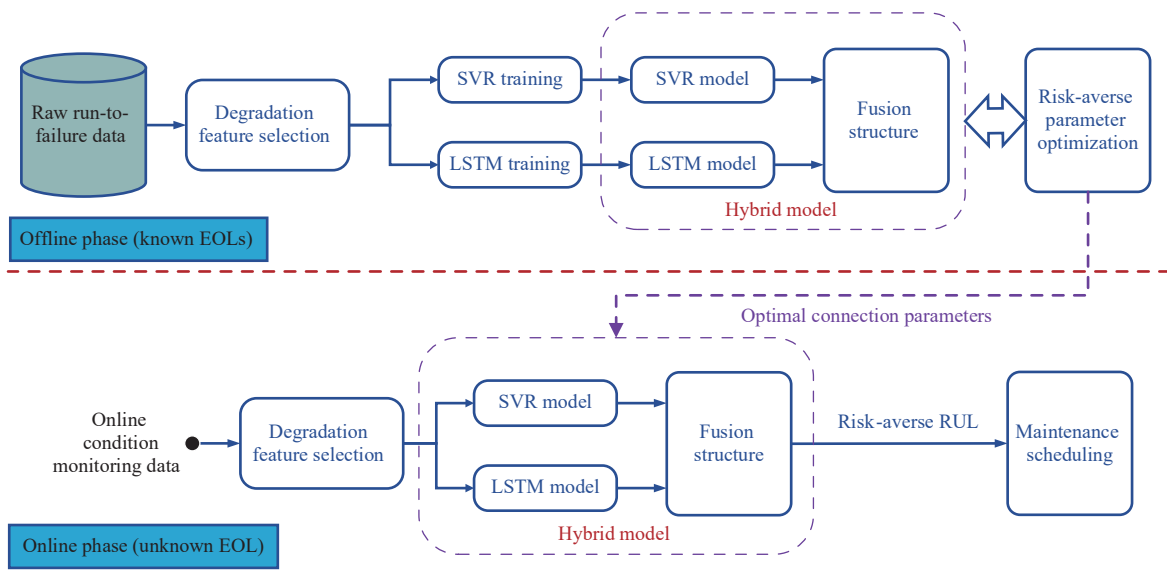


Fig. 1. Framework of the proposed PdM strategy.

Right after, the features are fed into the well-trained SVR and LSTM models, and the two models output the estimated RULs independently. The ultimate RUL estimation is obtained by the risk-averse hybrid model for maintenance decision-making.

B. Extracting Degradation Features

The process of degradation feature selection can be illustrated in Fig. 2. Usually, system degradation is a slow and monotonic process, which means that the degradation features should exhibit certain tendencies [27]. Moving average is a simple but effective skill for tendency extraction. Given a time series $\{s_1, s_2, \dots, s_l\}$, using moving average, the degradation trend (namely the degradation features discussed below) can be obtained, i.e.,

$$\gamma_i = \frac{s_i + s_{i-1} + \dots + s_{i-n+1}}{n}, \quad i = n, n+1, \dots, l \quad (1)$$

where s_i denotes the monitored indicator of system at moment i , and n is the moving window size.

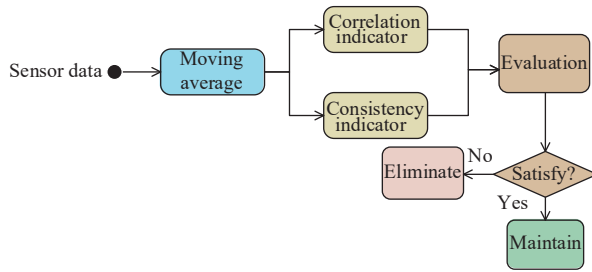


Fig. 2. Process of degradation feature selection.

Irrelevant or redundant features affect the performance of learning algorithms. Therefore, it is necessary to eliminate such features and only maintain contributory ones. Correlation and consistency indicators [15] can be employed for such purposes.

The correlation indicator reflects the linear correlation between a feature and the length of an observed sequence, which is formulated as

$$Corr = 1 - \frac{6 \sum_{i=1}^l d_i^2(\gamma_i)}{l(l^2 - 1)} \quad (2)$$

where $d_i(\gamma_i)$ denotes the difference between ranks for each γ_i and i . More precisely, the data sets $\{\gamma_i\}_{i=1}^l$ and $\{i\}_{i=1}^l$ can be rearranged in numerical order from large to small, and the difference between ranks for each γ_i and i is the position difference of γ_i and i in the sorted data sets. The correlation indicator ranges from -1 to 1 , and a feature with greater absolute value of $Corr$ implies stronger linear correlations.

The consistency indicator reflects the consistent tendency of a feature in all samples, which can be calculated by

$$Con = \frac{1}{N} \sum_{j=1}^N \rho(Corr_j) \quad (3)$$

where N is the number of samples and $\rho(x) = \begin{cases} 1 & x > 0 \\ 0.5 & x = 0 \\ 0 & x < 0. \end{cases}$

The consistency indicator ranges from 0 to 1, and a feature with a consistency indicator of 0 or 1 indicates that the degradation trend is completely identical for all samples.

Thus, the rule for extracting the degradation features is to choose the features that satisfy the conditions, $|Corr| \geq \vartheta$ and $Con = 0$ or 1 . Here, ϑ is the correlation threshold and is generally set between 0.5 and 1.

C. Modeling Features-RUL Relationship

This step is to develop a model that can accurately describe the relationship between the degradation features and the corresponding RUL, i.e.,

$$RUL = F(\text{features}). \quad (4)$$

To the best of our knowledge, SVR and LSTM models are currently two well recognized prognostics models. SVR model was developed from support vector machine (SVM). With an error range, the predicted value falling within the error range is regarded as the correct prediction [28]. SVR is effective in establishing nonlinear mapping relationships because its learning algorithm comes from a convex optimization problem. However, it cannot handle dependencies amongst multiple time series, so it lacks the overall perception ability. LSTM model is a special type of neural network which can handle dependency between successive steps. The mapping relationship is realized by the cooperation of three gates (input gate, forget gate and output gate) [29], [30]. Nevertheless, the LSTM model structure is difficult to optimize, and as a result, its performance is difficult to improve.

As stated before, a hybrid model can maintain the positive aspects of each modeling method and prevent the diminishing marginal utility problem of simple prediction models. Thus, a hybrid model that combines the SVR and LSTM models is developed via three connection parameters (w_1 , w_2 , b), as illustrated in Fig. 3. It is noted that to be exact, the ϵ -SVR and vanilla LSTM models are chosen according to [14] and [25]. The fusion structure enables information exchange between the two models. Through such information exchange, a better RUL estimation can be achieved.

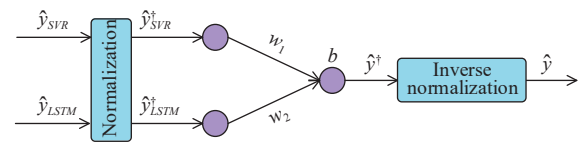


Fig. 3. Fusion structure of SVR and LSTM models.

In Fig. 3, \hat{y}_{SVR} and \hat{y}_{LSTM} denote the estimated RULs of SVR and LSTM models respectively; while \hat{y}_{SVR}^\dagger and \hat{y}_{LSTM}^\dagger denote the normalized values calculated by

$$\hat{y}_{SVR}^\dagger = \frac{\hat{y}_{SVR} - \min(\hat{y}_{SVR})}{\max(\hat{y}_{SVR}) - \min(\hat{y}_{SVR})} \quad (5)$$

$$\hat{y}_{LSTM}^\dagger = \frac{\hat{y}_{LSTM} - \min(\hat{y}_{LSTM})}{\max(\hat{y}_{LSTM}) - \min(\hat{y}_{LSTM})} \quad (6)$$

where $\min(\cdot)$ and $\max(\cdot)$ denote the minimum and maximum

operations, respectively. Besides, the parameters w_1 and w_2 denote the weights of SVR and LSTM models, respectively, while the parameter b denotes the bias of the hybrid model. Based on a hyperbolic tangent activation function, the estimated RUL, denoted by \hat{y}^\dagger , is gained as

$$\hat{y}^\dagger = \tanh\left([w_1, w_2] \begin{bmatrix} \hat{y}_{SVR}^\dagger \\ \hat{y}_{LSTM}^\dagger \end{bmatrix} + b\right) \\ = \frac{e^{w_1\hat{y}_{SVR}^\dagger + w_2\hat{y}_{LSTM}^\dagger + b} - e^{-(w_1\hat{y}_{SVR}^\dagger + w_2\hat{y}_{LSTM}^\dagger + b)}}{e^{w_1\hat{y}_{SVR}^\dagger + w_2\hat{y}_{LSTM}^\dagger + b} + e^{-(w_1\hat{y}_{SVR}^\dagger + w_2\hat{y}_{LSTM}^\dagger + b)}}. \quad (7)$$

It should be noted that, the activation function of (7) is adopted to handle the nonlinearity of the fusion structure. The fusion structure can be a simple linear expression. However, linear expressions often do not perform well compared with the above nonlinear form. Observed from (7), the range of hyperbolic tangent activation function is between -1 and 1 , and this also explains the purpose of normalization using (5) and (6).

Lastly, the RUL estimation is obtained after an inverse normalization, i.e.,

$$\hat{y} = \hat{y}^\dagger(U - L) + L \quad (8)$$

where U and L denote the upper and lower bounds associated with \hat{y}_{SVR} and \hat{y}_{LSTM} , and it is obvious that U is between $\max(\hat{y}_{SVR})$ and $\max(\hat{y}_{LSTM})$, while L is between $\min(\hat{y}_{SVR})$ and $\min(\hat{y}_{LSTM})$. For simplification, one can take their midpoints, i.e.,

$$U = \frac{\max(\hat{y}_{SVR}) + \max(\hat{y}_{LSTM})}{2} \quad (9)$$

$$L = \frac{\min(\hat{y}_{SVR}) + \min(\hat{y}_{LSTM})}{2}. \quad (10)$$

Now, the remaining problem is to determine the three parameters (w_1 , w_2 and b) to meet the design requirement of risk-averse adaptation.

D. Risk-Averse Adaptation

It is a common view that underestimated RUL is relatively better than the overestimated one under the same or close prediction error. Accordingly, the hybrid model should consider the objective of reducing the overestimation rate while maintaining a reasonable underestimation level. In this context, the error function below is given to incline more to the underestimated RUL, i.e.,

$$e = \begin{cases} \hat{y} - y, & \text{if } \hat{y} - y \leq 0 \\ (C_c/C_p)(\hat{y} - y), & \text{if } \hat{y} - y > 0 \end{cases} \quad (11)$$

where y is an actual RUL, \hat{y} is the predicted one, C_c is the corrective maintenance cost, and C_p is the preventive maintenance cost. Here, C_p refers to all costs associated with the preventive maintenance actions, such as replacing the worn parts with new ones, system cleaning and adjustment, and the inventory cost of spare parts. C_c refers to the total cost due to sudden failure, including human damage cost, equipment damage cost, and out-of-stock cost in the case of unavailable spare parts. C_c/C_p is the penalty multiplier for the overestimation error. For example, one has to pay ¥1000 for corrective maintenance when the system suddenly fails,

whereas preventive maintenance only costs ¥100 if well-timed preventive maintenance activities are taken. To be reasonable, the extra cost should be borne by the overestimated RUL, and the penalty multiple for overestimation error should be 10.

Thereupon, the minimization error function (cost function) is defined as

$$\min z = \frac{\sum_{j=1}^N \sum_{i, \hat{y}_i^j - y_i^j \leq 0} |\hat{y}_i^j - y_i^j|}{\sum_{j=1}^N l_j} \\ + \frac{\sum_{j=1}^N \sum_{i, \hat{y}_i^j - y_i^j > 0} (C_c/C_p)(\hat{y}_i^j - y_i^j)}{\sum_{j=1}^N l_j} \quad (12)$$

where l_j is the RUL sequence length of the j -th sample. With this penalty mechanism, the cost function forces the prognostics to shift from overestimation to underestimation. The values of w_1 , w_2 , and b are selected by minimizing the function (12).

Considering the complexity of (12), it is difficult to get a closed-form solution. From a usability standpoint, one can obtain a group of feasible sub-optimal solutions rather than a closed-form optimal solution by using evolution algorithms [31]. In this paper, the grey wolf optimization (GWO) algorithm is selected to estimate the values of w_1 , w_2 , and b .

GWO algorithm is an emerging technique in the field of evolutionary algorithms. It is inspired by the intelligent activities of grey wolf population hunting and has been proved to be superior to traditional algorithms, such as genetic algorithm (GA), particle swarm optimization (PSO) algorithm, and differential evolution (DE) algorithm [32]. Unfortunately, the GWO algorithm generates an initial population in a random manner. Although the initial population has a certain population diversity, the population level may not be good, affecting the convergence speed and precision of the algorithm. After randomly generating the initial population, embedding a selection operator will help improve the optimization performance of the GWO algorithm [33]. In addition, the GWO algorithm must balance the global searching and the local searching. Without an effective balance mechanism, the algorithm may fall into local optimum. To overcome this problem, one can modify the convergence factor. More specifically, the linear convergence factor is replaced by a nonlinear convergence factor. The nonlinear convergence factor will drive the algorithm to search globally in the early stage, and turn to local searching later. This improved algorithm is named as GWO-II algorithm in this paper.

Fig. 4 shows the implementation process of the GWO-II algorithm. Based on the GWO-II algorithm, the three parameters (w_1 , w_2 , b) can be solved by the following steps.

Step 1: Initialize the parameters, including the population size S , the maximum number of iterations T , and each individual position $Q_k = (w_1, w_2, b)$ with $k = 1, 2, \dots, S$.

Step 2: Embed a selection operator. The fitness values of all individuals in the population are first calculated according to (12). Subsequently, these fitness values are arranged in

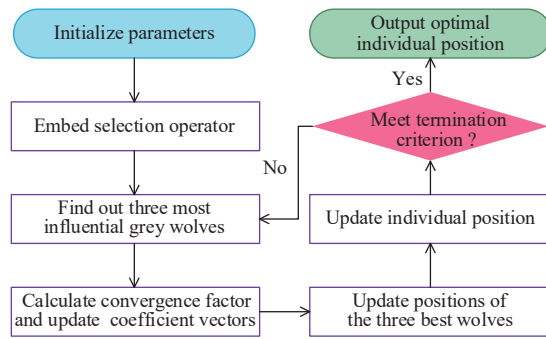


Fig. 4. Implementation process of GWO-II algorithm.

ascending order. All individuals are divided into the front, middle, or back segment. Individuals in the front are better solutions. Finally, each segment is randomly selected according to the proportion of 1.0, 0.8, and 0.6. For the individuals who lost 20% in the middle segment and 40% in the back segment, the individuals in the front segment are supplemented to form a new population with the same population size.

Step 3: Find out the three most influential grey wolves in the new population. They are noted as α , β , and δ in turn. The three wolves will lead the population to surround, hunt and attack the prey (target solution).

Step 4: Calculate the convergence factor and update the coefficient vectors. The proposed convergence factor ϖ is reduced from 2 to 0 in a nonlinear way of (13). After that, the coefficient vectors A and C can be updated by (14) and (15).

$$\varpi = 2 - 2 \left(\frac{e^{t/T} - 1}{e - 1} \right)^2 \quad (13)$$

$$A = 2\varpi \cdot r_1 - \varpi \quad (14)$$

$$C = 2 \cdot r_2 \quad (15)$$

where t is the current number of iterations, and r_1 and r_2 are the random vectors between $[0,1]$.

Step 5: Update the positions of the three best wolves by

$$Q_\alpha(t+1) = Q_\alpha(t) - A_1 |C_1 Q_\alpha(t) - Q(t)| \quad (16)$$

$$Q_\beta(t+1) = Q_\beta(t) - A_2 |C_2 Q_\beta(t) - Q(t)| \quad (17)$$

$$Q_\delta(t+1) = Q_\delta(t) - A_3 |C_3 Q_\delta(t) - Q(t)|. \quad (18)$$

Step 6: Update the individual position by

$$Q(t+1) = \frac{Q_\alpha(t+1) + Q_\beta(t+1) + Q_\delta(t+1)}{3}. \quad (19)$$

Step 7: Terminate the algorithm if the termination criterion is reached and output the optimal individual position. Otherwise, go back to Step 3 and continue the algorithm.

E. Online RUL Estimation

The well-trained SVR and LSTM models together with the obtained optimal connection parameters (w_1 , w_2 , and b) can result in a high-performance hybrid model, which will be used to predict the RUL online. With the online monitoring data of an in-service system, we seek to predict how long the system can be operated safely, that is, to predict the time span

between the current moment and the future true system failure moment.

Fig. 5 illustrates the RUL estimation for an in-service system. Suppose that there are currently available monitoring data from time 1 to K and three feature sequences (denoted by x_1 , x_2 , and x_3) are extracted from the available monitoring data. The SVR and LSTM models constitute the hybrid model, whereas the two base models have different prediction strategies. As for the LSTM model, it needs to handle the dependency between successive steps, but the SVR model does not. Therefore, the two models must be treated differently when predicting the RUL. In fact, the SVR model directly receives the feature vector (x_1^K, x_2^K, x_3^K) of the K -th moment and produces the estimated RUL, \hat{y}_K^{SVR} . Regarding the LSTM model, it feeds on the three feature sequences and outputs a RUL sequence with K predictions, $\hat{y}_1^{LSTM}, \dots, \hat{y}_k^{LSTM}, \dots, \hat{y}_K^{LSTM}$. Among them, the \hat{y}_K^{LSTM} represents the estimated RUL of the LSTM model. When the \hat{y}_K^{SVR} and \hat{y}_K^{LSTM} are given to the fusion structure with the obtained optimal connection parameters (w_1 , w_2 , and b), the final estimated RUL \hat{y}_K is obtained. Hence, a failure may occur at time $K + \hat{y}_K$.

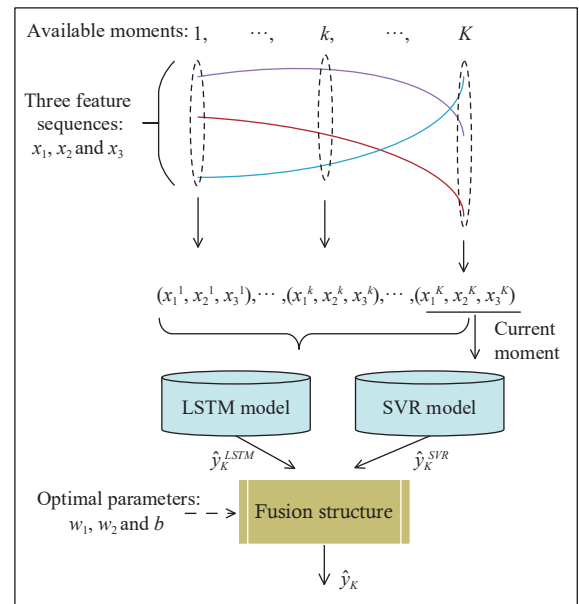


Fig. 5. Illustration of RUL estimation for an in-service system.

With the estimated RUL of the in-service system, one can schedule the maintenance activity, as shown in Fig. 6. Clearly, preventive maintenance activity should be implemented before the estimated failure time. Assuming that the time margin of the reliability is η , the maintenance activities will be taken at time $K + \hat{y}_K - \eta$.

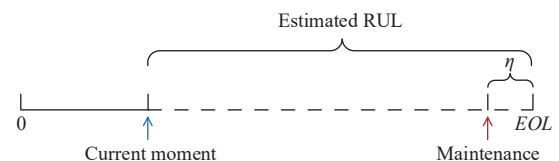


Fig. 6. Illustration of maintenance scheduling.

There are two possible scenarios in real-world maintenance activities. If the scheduled preventive maintenance moment is conducted earlier than the actual failure moment of the in-service system, the preventive maintenance is effective. On the other hand, if the in-service system fails earlier than the scheduled preventive maintenance moment, the preventive maintenance is void and corrective maintenance must be taken.

The maintenance cost rate (MCR), defined as the ratio between the total maintenance cost and total life cycle duration, can be used to estimate the performance of the RUL estimation. The estimated RUL with lower MCR is considered to have better performance. It intends to measure the benefit provided by risk-averse prognostics. When the in-service system still works but the preventive maintenance is performed, the MCR is given by

$$MCR_p = \frac{C_p}{K + \hat{y}_K - \eta}. \quad (20)$$

On the other hand, if the system fails, corrective maintenance must be carried out. In this event, the total maintenance cost is C_c and the total life cycle duration is $K + y$. Thus, according to the definition of MCR, the MCR of the corrective maintenance is given by

$$MCR_c = \frac{C_c}{K + y}. \quad (21)$$

In general, the prediction accuracy of a model is between 50% and 100%. The 100% prediction accuracy means perfect prognostics, whereas the 50% prediction accuracy implies unsatisfactory prediction results. In this sense, the estimated failure time in (20) is supposed to satisfy $0.5(K + y) \leq K + \hat{y}_K \leq K + y$. Thus, for the ideal PdM case with perfect prognostics, the MCR becomes

$$MCR_{ideal} = \frac{C_p}{K + y}. \quad (22)$$

It is worth noting that the ideal PdM case is only an ideal hypothesis which cannot be achieved in practice. However, the MCR_{ideal} can help understand the gap between reality and ideal. In addition, this inequality, $0.5(K + y) \leq K + \hat{y}_K - \eta$, can also be satisfied considering that the time margin η is usually small. As mentioned before, the corrective maintenance is more expensive than the preventive maintenance and this means that $C_c > 2C_p$ is reasonable. Through $0.5(K + y) \leq K + \hat{y}_K - \eta$ and $C_c > 2C_p$, one can get $(C_c)/(K + y) > (C_p)/(K + \hat{y}_K - \eta)$. Therefore, the proposed cost metric tends to favor lead prediction. In other words, the overestimated RUL will be more costly than the underestimated RUL.

IV. VERIFICATION

A. Dataset Description

To verify the performance of the proposed risk-averse RUL estimation method and the PdM strategy, a turbofan engine degradation simulation dataset [34] is considered. The dataset consists of multivariate time series signals generated by C-MAPSS. Fig. 7 shows a sketch of the main components of the turbofan engine, including the fan, low-pressure compressor

(LPC), high-pressure compressor (HPC), high pressure turbine (HPT), and low pressure turbine (LPT). A total of 21 signals that can describe the degradation process were generated, as shown in Table I. The first nine signals were obtained by direct measurement of Sensors 1–9, while the rest were gained by soft measurement of Sensors 10–21 [35].

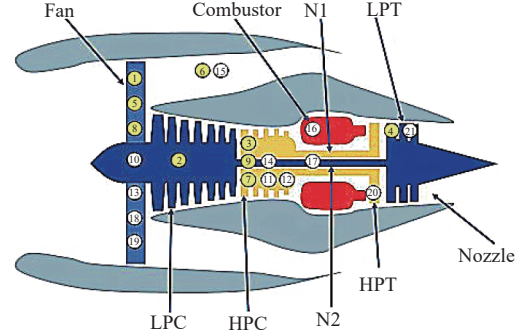


Fig. 7. Sketch of the main components of the engine [35].

TABLE I
DESCRIPTION OF THE SENSOR DATA [35]

ID	Symbol	Description	Units
1	T2	Total temperature at fan inlet	°R
2	T24	Total temperature at LPC outlet	°R
3	T30	Total temperature at HPC outlet	°R
4	T50	Total temperature at LPT outlet	°R
5	P2	Pressure at fan inlet	psia
6	P15	Total pressure in bypass-duct	psia
7	P30	Total pressure at HPC outlet	psia
8	Nf	Physical fan speed	rpm
9	Nc	Physical core speed	rpm
10	epr	Engine pressure ratio (P50/P2)	–
11	Ps30	Static pressure at HPC outlet	psia
12	phi	Ratio of fuel flow to Ps30	pps/psi
13	NRf	Corrected fan speed	rpm
14	NRc	Corrected core speed	rpm
15	BPR	Bypass ratio	–
16	farB	Burner fuel-air ratio	–
17	htBleed	Bleed enthalpy	–
18	Nf_dmd	Demanded fan speed	rpm
19	PCNfR_dmd	Demanded corrected fan speed	rpm
20	W31	HPT coolant bleed	lbm/s
21	W32	LPT coolant bleed	lbm/s

Dataset “FD001” is used in this paper. It describes the gradual degradation process of HPC under one failure mode and one operating condition. The dataset contains three.txt files: “train_FD001.txt”, “test_FD001.txt” and “RUL_FD001.txt”. “train_FD001.txt” records the complete life cycles of 100 engines from the beginning to failure, “test_FD001.txt” is composed of 100 incomplete time series ending up prior to failure, and “RUL_FD001.txt” provides the actual RUL values.

B. RUL Estimation Results

Based on the provided dataset, the first step is to extract the degradation features of the engines. According to Section III-B, the moving average technique is utilized for tendency extraction, and the moving window size n is set to 20. Next, the correlation and consistency indicators of each sensor are calculated using (2) and (3), as shown in Table II. In this paper, the sensor with absolute value of correlation indicator greater than 0.7 is regarded as a stronger linear correlation. Thus, the sensors that satisfy $|Corr| \geq 0.7$ and $Con = 0$ or 1 will be selected.

TABLE II
CORRELATION AND CONSISTENCY INDICATORS FOR 21 SIGNALS

Sensor ID	<i>Corr</i>	<i>Con</i>	Action
1	[0, 0]	0.5	
2	[0.76, 0.99]	1	√
3	[0.69, 0.98]	1	
4	[0.84, 0.99]	1	√
5	[0, 0]	0.5	
6	[-0.11, 0.90]	0.8	
7	[-0.99, -0.71]	0	√
8	[0.12, 1]	1	
9	[-0.95, 1]	0.71	
10	[0, 0]	0.5	
11	[0.83, 1]	1	√
12	[-0.99, -0.85]	0	√
13	[-0.13, 0.99]	0.99	
14	[-0.99, 1.00]	0.6	
15	[0.79, 0.99]	1	√
16	[0, 0]	0.5	
17	[0.73, 0.99]	1	√
18	[0, 0]	0.5	
19	[0, 0]	0.5	
20	[-0.99, -0.82]	0	√
21	[-0.99, -0.73]	0	√

Considering that sensors 1, 3, 5, 6, 8, 9, 10, 13, 14, 16, 18, and 19 do not meet the limitation of the correlation indicator, they will be eliminated. Subsequently, the remaining sensors will be tested by the consistency indicator. These sensors that have passed the test will be maintained. Thus, the remaining nine sensors are selected, i.e., the sensors 2, 4, 7, 11, 12, 15, 17, 20, and 21.

With the obtained degradation features, the SVR and LSTM models are separately trained to establish the nonlinear mapping relationship between the features and the RUL. In this experiment, the number of hidden units, learning rate, and dropout probability of the LSTM model are specified as 200, 0.01, and 0.5, respectively. The SVR hyper-parameters, such as box constraint, Gaussian kernel scale parameter and ϵ , are set to 1, 1, and 0.1, respectively.

Considering that the degradation of the engine is usually not noticeable before running for a while, this means it may not be

reasonable to estimate the RUL at this stage [36]. Thus, this paper clips the responses at the RUL threshold 150, which is determined via several trials with best performance on the test dataset. That is to say, the models will treat the instances with higher RULs as equal, which enables the models to learn more essential mapping relationship when the engines are close to failing. Fig. 8 shows the parts of prediction results on the test set.

For the test engines ID 31 and ID 38, the predicted values of the LSTM model are closer to the actual values than those of the SVR model. However, for test engine ID 78, the SVR model outperforms the LSTM model. Besides, it is observed from the test engine ID 91 that the prediction curve of the LSTM model is approaching the actual curve prior to Cycle 155. After that, the SVR model has the upper hand. These results indicate that each of the studied SVR and LSTM models has its own benefits and drawbacks, and there is no clear-cut winning method. Nevertheless, it must be pointed out that late prediction is intolerable. For example, there are obvious late predictions for the test engine ID 78, and the consequences of the late predictions may be catastrophic. Therefore, the risk-averse adaptation is necessary, and the next section will present the verification results for the risk-averse RUL estimation.

C. Risk-Averse RUL Estimation Results on Test Data

The GWO-II algorithm is used to determine the values of w_1 , w_2 , and b . To enable the GWO-II algorithm to reach a convergence state, the population size is set to 20 and the maximum number of iterations is set to 50. The search spaces of the three parameters are set between -10 and 10 . Also, the corrective maintenance cost and preventive maintenance cost are assumed to be 500 and 100, respectively. Generally speaking, it is difficult to describe a four-dimensional space, such as w_1 , w_2 , b , and z . For visualization, the three-dimensional view of w_1 , w_2 , and z is depicted in Fig. 9. This figure shows the distribution of the grey wolf population during the optimization process.

In the initial stage, the grey wolves are randomly distributed in the space. Through the surrounding, hunting, and attacking operations, the grey wolf population is able to obtain the information about the prey (optimal solution) [37]. Thus, as the number of iterations increases, the grey wolves gradually gather into the optimal solution area. At the 30th iteration, the GWO-II algorithm finds the optimal solution, that is $w_1 = 0.22$, $w_2 = 0.89$, and $b = -0.05$.

With the obtained values of w_1 , w_2 , and b , the hybrid model is used to estimate the RULs on the test set. Fig. 10 shows the estimated RUL results on 100 test samples using the hybrid model. It is observed that the estimated values can not only follow the changes in actual values but also approach the actual ones. This implies that the use of the hybrid model is feasible to predict the RULs.

Fig. 11 shows the distribution of errors for SVR, LSTM and hybrid models. The prognostics on the left side of the red dotted line refers to the lead prediction. It is obvious that the hybrid model has more lead predictions compared with the SVR and LSTM models. Specifically, the numbers of lead

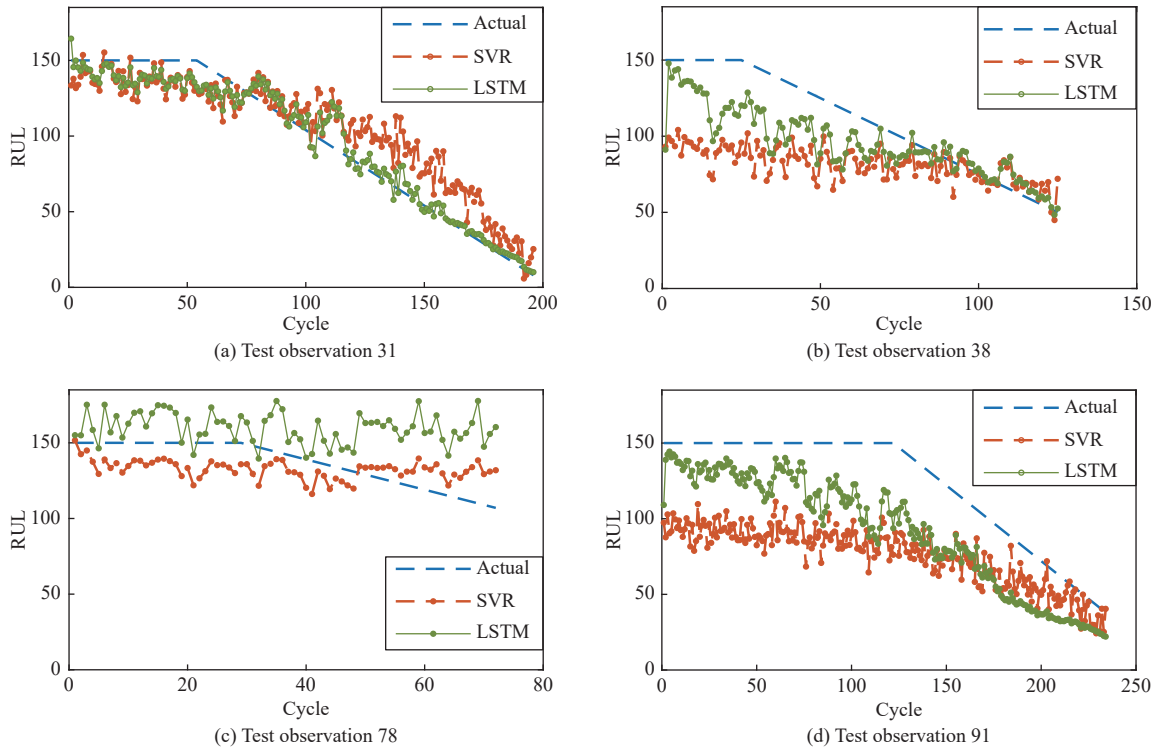


Fig. 8. Prediction results on several test samples.

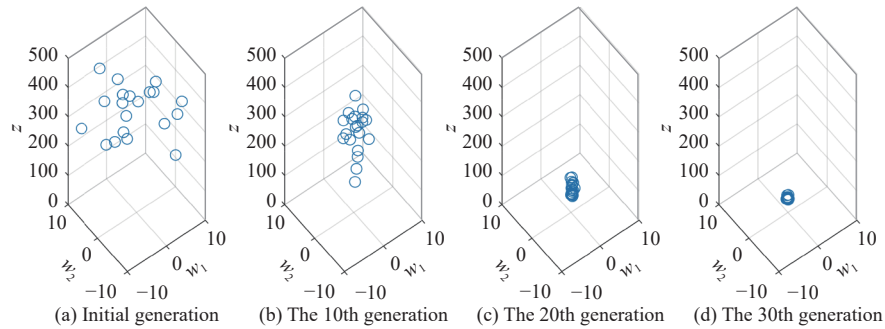


Fig. 9. Search process of GWO-II algorithm for parameters w_1 and w_2 .

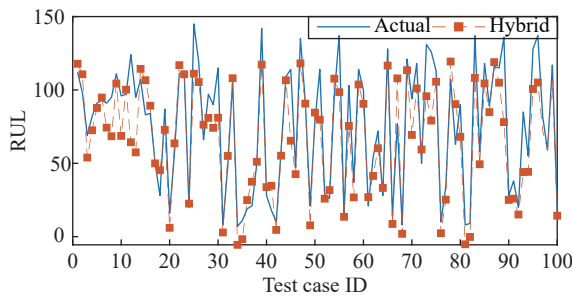


Fig. 10. Estimated RUL results for 100 test samples.

predictions for SVR, LSTM, and hybrid models are 40, 44, and 70, respectively. The results imply that both SVR and LSTM models tend to late predictions. However, as mentioned before, such late predictions are intolerable due to the potential for catastrophic consequences. Unlike the simple SVR or LSTM models, their combination gives good results. This should be attributed to the fusion structure proposed in Section III-C and the penalty mechanism proposed in

Section III-D.

It should be pointed out that the lead prediction without considering the prediction error is meaningless. Thus, the root mean square errors (RMSEs) of the three models are calculated and shown in Fig. 11. The RMSE of the hybrid model is 19.11, which is lower than 24.61 of the SVR and 20.16 of the LSTM, indicating that the hybrid model is superior in prediction precision as well. To sum up, the lead prediction number and the prediction precision of the hybrid model meet the expected goal, which is to reduce the over estimation rate while maintaining a reasonable under estimation level, showing that the proposed risk-averse RUL estimation approach is effective.

D. Performance Evaluation

In this section, the proposed RUL estimation method will be evaluated from the perspective of predictive maintenance. According to the maintenance scheduling in Section III-E, as an illustration, Table III gives the maintenance cost rates of three RUL prognostic strategies for test engines ID 1–10.

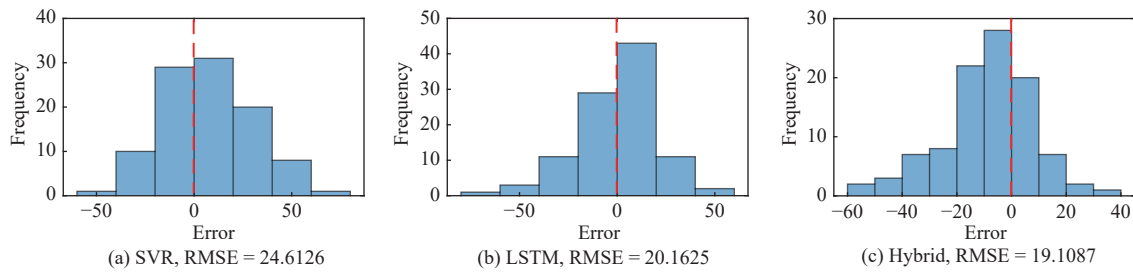


Fig. 11. Distribution of errors for SVR, LSTM and hybrid models.

TABLE III
MAINTENANCE COST RATES FOR TEST ENGINES ID 1–10

Test engine ID	Failure moment	SVR model			LSTM model			Hybrid model		
		T_m	MAT	MCR	T_m	MAT	MCR	T_m	MAT	MCR
1	143	149.09	C	3.50	167.94	C	3.50	138.84	P	0.72
2	147	135.91	P	0.74	176.34	C	3.40	149.65	C	3.40
3	195	184.50	P	0.54	173.46	P	0.58	169.90	P	0.59
4	188	170.99	P	0.58	175.57	P	0.57	168.50	P	0.59
5	189	178.67	P	0.56	186.06	P	0.54	175.73	P	0.57
6	198	198.39	C	2.53	201.94	C	2.53	189.85	P	0.53
7	251	264.87	C	1.99	223.72	P	0.45	224.32	P	0.45
8	261	233.20	P	0.43	229.75	P	0.44	224.44	P	0.45
9	166	146.02	P	0.68	169.00	C	3.01	149.36	P	0.67
10	288	284.26	P	0.35	250.86	P	0.40	250.69	P	0.40

Here, “ T_m ” denotes the planned maintenance moment, “P” indicates that the maintenance action taken (MAT) is preventive maintenance, and “C” indicates that the MAT is corrective maintenance. The time margin of the reliability, η , is considered as 10.

From the given MCRs, it is obvious that the corrective maintenance is more costly than the preventive maintenance. From the maintenance activities taken, the use of hybrid model, instead of the SVR or LSTM model, allows improving the accuracy of maintenance decisions. For example, for the test engines ID 1, 6, and 7, the scheduled maintenance actions using SVR model are unreasonable. The similar situation can be found in the LSTM model, such as the test engines ID 1, 2, 6, and 9. Best of all, the use of hybrid model leads to correct the maintenance decisions for the test engines ID 1, 6, 7, and 9, which is significant for ensuring the operation safety and reliability of the engines. Besides, it is observed that the MCR of the hybrid model is sometimes higher than that of SVR and LSTM models under the preventive maintenance, but it is allowable compared with the corrective maintenance.

The MCRs for the 100 test samples can be visualized in Fig. 12. The test samples are divided into five groups, and each group has 20 engines. The average MCR of each group is calculated based on different prognostic approaches (SVR, LSTM, hybrid models, and ideal PdM case). Evidently, no matter which group, the average MCR of the hybrid model is lower than those of the SVR and LSTM models. Based on this, the average MCRs of the four prognostic approaches for all test samples are calculated. In detail, the average MCR for the SVR model is 1.49, 1.20 for the LSTM model, 0.80 for the

hybrid model and, 0.50 for the ideal PdM case. Compared with the SVR and LSTM models, the average MCR of the hybrid model is reduced by 46.44% and 33.41%, respectively. Also, there is only a MCR gap of 0.3 between the hybrid model and the ideal model. These results show that the benefits the risk-averse RUL estimation brings are considerable, and the proposed approach can significantly reduce the maintenance cost rate and is approaching the ideal one.

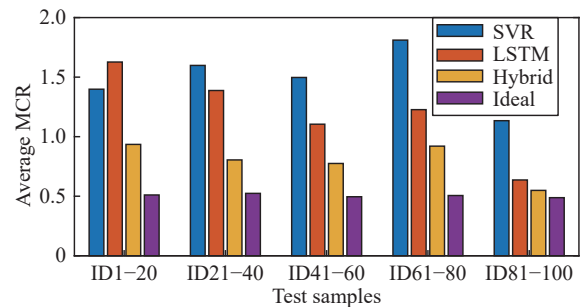


Fig. 12. MCRs for 100 test samples using SVR, LSTM and hybrid models.

V. CONCLUSIONS

In this work, an RUL estimation method with risk-averse adaptation has been proposed. It hybridizes two mainstream models, SVR and LSTM, in the field of RUL prognosis. The proposed estimation method can enhance the robustness of prediction, increase the marginal utility, and reduce the over-estimation rate while maintaining a reasonable under

estimation level. Besides, an embedded degradation feature selection module can obtain crucial features that reflect system degradation trends, and a proposed maintenance cost rate can measure the benefit provided by risk-averse prognostics.

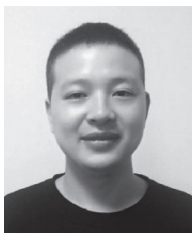
The verification results using NASA data repository reveal the feasibility and effectiveness of the proposed method. According to the number of lead predictions, the prediction errors, the accuracy of maintenance decisions, and the maintenance cost rate, the proposed hybrid model is indeed better than the single models. Above all, the precise and risk-averse RUL estimation is significant for a system to keep safe and stable operation.

The proposed method can also be applied to other engineering systems subjected to single failure modes. Considering that the failure of some engineering systems may be caused by coupling actions of various failure modes, our future work will focus on the investigation of the systems with multiple failure modes.

REFERENCES

- [1] C. Hu, H. Pei, X. Si, D. Du, Z. Pang, and X. Wang, "A prognostic model based on DBN and diffusion process for degrading bearing," *IEEE Trans. Ind. Electron.*, to be published. DOI: 10.1109/TIE.2019.2947839.
- [2] P. C. Lopes Gerum, A. Altay, and M. Baykal-Gürsoy, "Data-driven predictive maintenance scheduling policies for railways," *Transp. Res. Pt. C-Emerg. Technol.*, vol. 107, pp. 137–154, Oct. 2019.
- [3] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech. Syst. Signal Proc.*, vol. 104, pp. 799–834, May 2018.
- [4] K. T. Nguyen and K. Medjaher, "A new dynamic predictive maintenance framework using deep learning for failure prognostics," *Reliab. Eng. Syst. Saf.*, vol. 188, pp. 251–262, Aug. 2019.
- [5] K. Wang, "Intelligent predictive maintenance (IPdM) system—Industry 4.0 scenario," *WIT Trans. Eng. Sci.*, vol. 113, pp. 259–268, 2016.
- [6] F. Civerchia, S. Bocchino, C. Salvadori, E. Rossi, L. Maggiani, and M. Petracca, "Industrial Internet of Things monitoring solution for advanced predictive maintenance applications," *J. Ind. Inf. Integr.*, vol. 7, pp. 4–12, Sept. 2017.
- [7] J. Wang, L. Zhang, L. Duan, and R. X. Gao, "A new paradigm of cloudbased predictive maintenance for intelligent manufacturing," *J. Intell. Manuf.*, vol. 28, no. 5, pp. 1125–1137, Jun. 2017.
- [8] M. Baptista, S. Sankararaman, I. P. de Medeiros, C. Nascimento Jr, H. Prendinger, and E. M. Henriques, "Forecasting fault events for predictive maintenance using data-driven techniques and ARMA modeling," *Comput. Ind. Eng.*, vol. 115, pp. 41–53, Jan. 2018.
- [9] B. O. Gombé, G. G. Mérou, K. Breschi, H. Guyennet, J. M. Friedt, V. Felea, and K. Medjaher, "A SAW wireless sensor network platform for industrial predictive maintenance," *J. Intell. Manuf.*, vol. 30, no. 4, pp. 1617–1628, Apr. 2019.
- [10] K. A. Nguyen, P. Do, and A. Grall, "Joint predictive maintenance and inventory strategy for multi-component systems using Birnbaum's structural importance," *Reliab. Eng. Syst. Saf.*, vol. 168, pp. 249–261, Dec. 2017.
- [11] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mech. Syst. Signal Proc.*, vol. 107, pp. 241–265, Jul. 2018.
- [12] C. Zhang, C. Wang, N. Lu, and B. Jiang, "An RBMs-BN method to RUL prediction of traction converter of CRH2 trains," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 46–56, Oct. 2019.
- [13] K. T. Huynh, I. T. Castro, A. Barros, and C. Berenguer, "On the use of mean residual life as a condition index for condition-based maintenance decision-making," *IEEE Trans. Syst. Man Cybern. -Syst.*, vol. 44, no. 7, pp. 877–893, Jul. 2014.
- [14] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, "Direct remaining useful life estimation based on support vector regression," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2276–2285, Mar. 2016.
- [15] C. Wang, N. Lu, Y. Cheng, and B. Jiang, "A data-driven aero-engine degradation prognostic strategy," *IEEE T. Cybern.*, to be published. DOI: 10.1109/TCYB.2019.2938244.
- [16] E. Ramasso, M. Rombaut, and N. Zerhouni, "Joint prediction of continuous and discrete states in time-series based on belief functions," *IEEE T. Cybern.*, vol. 43, no. 1, pp. 37–50, Feb. 2013.
- [17] Q. Miao, L. Xie, H. Cui, W. Liang, and M. Pecht, "Remaining useful life prediction of lithium-ion battery with unscented particle filter technique," *Microelectron. Reliab.*, vol. 53, no. 6, pp. 805–810, Jun. 2013.
- [18] T. Qin, S. Zeng, and J. Guo, "Robust prognostics for state of health estimation of lithium-ion batteries based on an improved PSO–SVR model," *Microelectron. Reliab.*, vol. 55, no. 9–10, pp. 1280–1284, Sept. 2015.
- [19] A. Soualhi, K. Medjaher, and N. Zerhouni, "Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 1, pp. 52–62, Jan. 2015.
- [20] H. Dong, X. Jin, Y. Lou, and C. Wang, "Lithium-ion battery state of health monitoring and remaining useful life prediction based on support vector regression-particle filter," *J. Power Sources*, vol. 271, pp. 114–123, Dec. 2014.
- [21] T. Wang, J. Yu, D. Siegel, and J. Lee, "A similarity-based prognostics approach for remaining useful life estimation of engineered systems," in *Proc. Int. Conf. Prognostics Health Manage.*, Denver, CO, USA, 2008, pp. 1–6.
- [22] S. Malinowski, B. Chebel-Morello, and N. Zerhouni, "Remaining useful life estimation based on discriminating shapelet extraction," *Reliab. Eng. Syst. Saf.*, vol. 142, pp. 279–288, Oct. 2015.
- [23] F. Xue, P. Bonissone, A. Varma, W. Yan, N. Eklund, and K. Goebel, "An instance-based method for remaining useful life estimation for aircraft engines," *J. Failure Anal. Prevention*, vol. 8, no. 2, pp. 199–206, Apr. 2008.
- [24] E. Zio and F. D. Maio, "A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system," *Reliab. Eng. Syst. Saf.*, vol. 95, no. 1, pp. 49–57, Jan. 2010.
- [25] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.
- [26] M. Esmaelian, H. Shahmoradi, and M. Vali, "A novel classification method: A hybrid approach based on extension of the UTADIS with polynomial and PSO-GA algorithm," *Appl. Soft. Comput.*, vol. 49, pp. 56–70, Dec. 2016.
- [27] X. Jia, M. Zhao, Y. Di, Q. Yang, and J. Lee, "Assessment of data suitability for machine prognosis using maximum mean discrepancy," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5872–5881, Jul. 2018.
- [28] S. Sharifian, and M. Barati, "An ensemble multiscale wavelet-GARCH hybrid SVR algorithm for mobile cloud computing workload prediction," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 11, pp. 3285–3300, Nov. 2019.
- [29] C. Wang, N. Lu, S. Wang, Y. Cheng, and B. Jiang, "Dynamic long short-term memory neural-network-based indirect remaining-useful-life prognosis for satellite lithium-ion battery," *Appl. Sci.-Basel*, vol. 8, no. 11, Article ID 2078, Nov. 2018. DOI: 10.3390/app8112078.

- [30] M. Martínez-García, Y. Zhang, K. Suzuki, and Y. Zhang, "Measuring system entropy with a deep recurrent neural network model," in *Proc. IEEE 17th Int. Conf. Ind. Informat.*, Helsinki, Finland, 2019, pp. 1253–1256.
- [31] M. Martínez-García, Y. Zhang, and T. Gordon, "Memory pattern identification for feedback tracking control in human-machine systems," *Hum. Factors*, pp. 1–17, Oct. 2019.
- [32] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [33] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, Jul. 2018.
- [34] NASA. "Prognostic Data Repository". [Online]. Available: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>. Accessed on: Sept. 2019.
- [35] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. Int. Conf. Prognostics Health Manage.*, Denver, CO, USA, 2008, pp. 1–9.
- [36] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Proc. Int. Conf. Prognostics Health Manage.*, Denver, CO, USA, 2008, pp. 1–6.
- [37] R. E. Precup, R. C. David, and E. M. Petriu, "Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 527–534, Jan. 2017.



Chuang Chen received the M.S. degree in control engineering from Nanjing Technology University, Nanjing, China in 2018. He is currently a Ph.D. candidate in control theory and control engineering at the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, China. His current research interests include data-driven fault prognosis and health management.



processes.

Ningyun Lu (M'07) received the Ph.D. degree from Northeastern University, Shenyang, China in 2004. From 2002 to 2005, she worked as a research associate and post-doctoral fellow in the Hong Kong University of Science and Technology, China. She is currently a Full Professor in College of Automation Engineering at Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her research interests include data-driven fault prognosis and diagnosis and their applications to various industrial



diagnosis and fault tolerant control and their applications in aircrafts, satellites and high-speed trains. He currently serves as Associate Editor or Editorial Board Member for a number of journals such as the *IEEE Transactions on Control Systems Technology*; *IEEE Transactions on Cybernetics*; *Journal of The Franklin Institute*; *Neurocomputing*; *International Journal of Control, Automation and Systems*; *Acta Automatica Sinica*; *Control and Decision*; *Systems Engineering and Electronics*. He is the Chair of Control Systems Chapter in IEEE Nanjing Section, a member of IFAC Technical Committee on Fault Detection, Supervision, and Safety of Technical Processes.

Bin Jiang (SM'05–F'19) received the Ph.D. degree in automatic control from Northeastern University, Shenyang, China in 1995. He had ever been post-doctoral fellow, research fellow, and a visiting professor in Singapore, France, USA and Canada, respectively. He is currently a Chair Professor of Cheung Kong Scholar Program in Ministry of Education, and the Vice President of Nanjing University of Aeronautics and Astronautics, Nanjing, China. His current research interests include fault



Cunsong Wang received the M.S. degree in automation and control theory and control engineering from Nanchang Hangkong University, Nanchang, China in 2016. He is currently a Ph.D. candidate in control theory and control engineering at the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, China. His current research interests include data-driven fault prognosis and health management.