Jointly learning of degradation assessment and RUL prediction for aero-engines via dual-task deep LSTM networks

Journal:	*IEEE Transactions on Industrial Informatics*
Manuscript ID	TII-18-2112
Manuscript Type:	Regular Paper
Keywords:	degradation assessment, remaining useful life, prognostics and health management, long short term memory networks, deep learning

Jointly learning of degradation assessment and RUL prediction for aero-engines via dual-task deep LSTM networks

***Abstract*—Health assessment and prognostics are two key tasks within the prognostics and health management (PHM) frame of equipment. However, existing works are performing these two tasks separately and hierarchically. In this paper, we design and establish dual-task deep long short term memory (LSTM) networks for jointly learning of degradation assessment and remaining useful life (RUL) prediction of aero-engines. This enables a more robust and accurate assessment and prediction results making for the increment of operational reliability and safety as well as maintenance cost reduction. Meanwhile, the target label functions that match the network training are constructed in an adaptive way according to the health state of an individual aero-engine. Experiments on the popular C-MAPSS lifetime dataset of aero-engines are employed to verify the accuracy and effectiveness. The performance of our proposed work exhibits superiority over other state-of-the-art approaches and demonstrate its application potential.**

***Index Terms*—degradation assessment, remaining useful life, prognostics and health management, long short term memory networks, deep learning**

I. Introduction

AERO-ENGINES are highly cost complex thermal machines requiring reliable operation under extreme conditions of high temperatures, high pressures, and high rotational speeds and alternating loads. Prognostic and health management (PHM) of aero-engines is an indispensable part of ensuring reliable operation, reducing maintenance cost and avoiding catastrophic accidents once the equipment is put into use [1]. PHM is an integration of multidiscipline and its development is a complex system process. Funded by the US navy and studied by companies like Boeing, Caterpillar and Rockwell Automation from technical standards, OSACBM (open system architecture for condition-based management) [2, 3] is a standard and important reference for building other specific applications, such as, integrated vehicle health management (IVHM) [4], integrated condition assessment system (ICAS) [5]. The OSACBM standard is for data abstraction by a layered approach as seen in Fig. 1(a). Data is acquired from the data generation device at the lowest level and then flows to a higher level to be characterized into states, conditions and then diagnosis. Then it is further projected to prognosis from diagnosis based on a further conditioning, and finally the data is couched in terms of final user activity at the highest level of abstraction. It can be seen that diagnosis and prognostics, usually referring to health state assessment and remaining useful life (RUL) prediction within design parameters, respectively, are the key core parts of OSACBM [6]. In this classical OSACBM architecture, diagnosis and prognostics are carried out in a progressive way. That is, it needs expert experience and artificial design to project diagnosis to prognostics despite the fact that they are rooted in and developed from the same data source. This architecture is designed and established based on the traditional signal processing and machine learning methods such as hidden Markov model (HMM) [7], various neural networks (NN) [8-10], support vector machine (SVM) [11] which are of shallow structures and single objective functions. With the rise and development of deep learning, it provides great possibilities for the improvement of simplicity and versatility of classical PHM architecture. Fig. 1(b) presents an improved architecture based on OSACBM in which health assessment and prognostics intrinsically linked with each other are carried out jointly based on deep learning (DL) [12] and then work corporately for the final activity planning of the highest level data abstraction. DL with deep architectures can learn complex nonlinear hidden information in data which is in accordance with increasingly complex data from equipment, like aero-engines, developing towards higher and higher complexity and working under harsher and harsher conditions. Moreover, multiple purposes can be achieved with an ingenious design of the flexible DL structure.

The DL algorithms developed in recent years can form a more abstract high-level representation of attribute categories or features by combining low-level features and then discover the distributed data feature representation. Deep neural networks (DNNs), including convolutional neural networks (CNNs) [12], deep belief networks (DBNs) [13, 14], auto-encoder [15, 16], recurrent neural networks (RNNs) [12] and sparse deep stacking networks [17] are typical deep learning algorithms, among which are all feedforward neural networks except RNNs. The network topology of a feedforward neural network is directed acyclic structure. That is, connections exist only between adjacent layers and there is no connection among nodes within a same layer. Moreover, there is no propagation from hidden layer at a higher level to that at a lower level in their forward calculation. These typical deep feedforward neural networks all share a common assumption that learning samples are independent stochastic models and are usually applied to tasks like classification [18], target recognition [19]

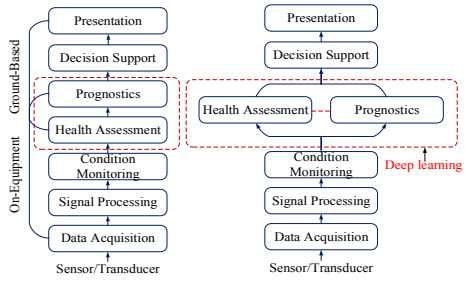

Fig. 1. (a) OSACBM layers [2]; (b) Improved PHM architecture based on OSACBM.

and image processing [20] where outstanding performances have been achieved. However, these feedforward networks feel powerless when they are encountered with sequence inputs for the analysis of overall logic characteristics since history information is not taken into account within their network structure. Meanwhile, equipment health assessment and prognostics are typical sequence problems. Therefore, other solutions are needed to address this sequence issue. Fortunately, RNNs different from those feedforward networks, is tailored for this issue by introducing self-feedback neuros into the network structure which enables it to deal with sequences (in terms of time correlation) with an arbitrary length. Therefore, RNNs is preferred to these prognostics problems.

Moreover, RNNs has developed into one of the important sub-domain of deep learning with its dynamic characteristics dealing efficiently with tasks involving sequential inputs such as language [21] and text [22]. However, training RNNs is not an easy work as the back propagated gradients either explode or vanish over time. As a result, it is difficult to cope with information of long-term dependencies [12]. Long short-term memory (LSTM) networks with specially designed cell structures can effectively alleviate this problem. Essentially, prognostics is to dig out inherent relations from sequential history data which is well suited for LSTM to process. Therefore, LSTM is the last choice for the problems of health assessment and prognostics of aero-engines presented in this paper. It is also worth mentioning that deep LSTM is of dual deep characteristics in terms of both time and space. Based on the dual deep characteristics, a dual-task deep LSTM network is designed for a simultaneous degradation stage (DS) assessment and RUL prediction of aero-engines in this paper.

The key contributions of this study can be summarized as follows.

1) The proposed framework is a dual-task LSTM network to parallelize tasks of RUL prediction and DS assessment of aero-engines. The architecture is shown in Fig. 3. This parallelization is achieved due to the versatility, deep feature mining ability and time relevance of deep LSTM network. It is executed by optimizing a joint objective function of regression task for RUL prediction and classification task for DS assessment. This parallelization may provide an alternative for the change of the classical layered PHM frame.

2) A new solution is proposed to dynamically construct RUL target label function and DS labels according to the health state of an individual aero-engine which enhances the adaptability of the proposed method. Moreover, the WFRF function adopted in the solution enables to distinguish the initial state of aero-engines from the subsequent deteriorated states.

3) The performance of different LSTM cell structures has been compared and it is found out that the intuition of peephole connection can be captured in this sequence prediction problem.

4) The accuracy, effectiveness and competiveness of the proposed framework have been verified on the dataset from C-MAPSS. Since the result of DS assessment is more stable and that of RUL prediction is more specific, their combination can be more robust and accurate in later guiding decision making.

The remainder of this paper is organized as follows. Section II introduces LSTM theory and the established dual-task deep LSTM network. The adopted data set description and target label construction for network training are presented in Section III. Section IV presents the experimental application. Finally, conclusion is summarized in Section V.

II. Proposed Dual-task Deep LSTM Network

As presented above, diagnosis and prognostics are implemented in a progressive way under the current PHM frame. Seldom RUL prediction work has been accomplished in parallel with the quantitative degradation assessment work. Though certain research has been conducted on the challenging problem of RUL prediction based on DL [23-25], it has been focusing on the direct analysis of monitoring data. To further develop the DL potential in the field of PHM, this paper proposes to implement diagnosis and prognostics in a jointly learning way. To this end, some preliminary study on the paralleled quantitative degradation assessment and RUL prediction of aero-engines is accomplished in this paper. Following benefits can be derived from this paralleled procedure. Firstly, it obviously reduces the time difference between diagnosis and prognostics which can be quite critical at some certain crucial moments. Secondly, it reduces complex feature design and needs less human experience. Lastly, a joint representation of assessment and prediction can achieve a mutual complementation and reinforce the higher level of decision making. Since assessment is usually of high robustness and prediction is of high specificity.

A deep dual-output LSTM network is introduced in this section for the benefits and purposes presented above. In the following, the concept of LSTM is introduced in Section A. Section B presents the configuration of the constructed LSTM network. Optimization algorithm is stated in Section C.

A. Variants of LSTM Network

The basic idea of RNNs is to build connections between units from a directed cycle. RNNs can be seen as very deep feedforward networks once they are unfolded with time where all the layers share the same weights as seen in [12]. The transition function H in RNN is the key to memorize arbitrary-length sequences of input patterns as given in (1):

$$\mathbf{h}_t = \mathrm{H}(\mathbf{x}_t, \mathbf{h}_{t-1}) \quad (1)$$

where H takes in the current time input vector $\mathbf{x}_t$ and the hidden output vector $\mathbf{h}_{t-1}$ which is an internal state of previous inputs' memory to update the current hidden output $\mathbf{h}_t$. Different transition function formulations derive different RNN

models. The basic LSTM and its variants peephole and gated recurrent units (GRUs) are adopted in this paper.

Basic LSTM [26] is developed from traditional RNN by adding gates functions including input gate, forget gate, output gate and one or more memory cells into the specially designed cell structure. A memory cell acting as a short term memory to store dependencies across time-steps controls different gates to alleviate the issue of not capturing long-term dependencies caused by vanishing gradients during back-propagation in RNN.

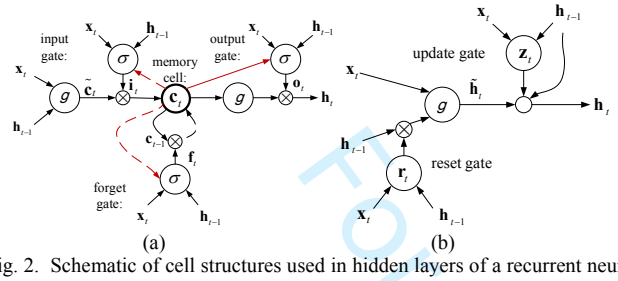

Fig. 2. Schematic of cell structures used in hidden layers of a recurrent neural network: (a) basic LSTM cell (* dashed lines indicate time-lag); (b) GRU cell.

The cell structure of basic LSTM is illustrated in Fig. 2 (a). It's worth noting that, the lines in red in Fig. 2 (a) are peephole connections with whom the cell structure derived into peephole cell [27, 28]. Peephole connections project from an internal state in the memory cell to the controlling gate units breaking the rule that only the final value of the memory cell is visible to other units when it is without peephole connections. The transition functions in hidden units of basic LSTM are given step-by-step in the following (2)-(7):

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{V}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \quad (2)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{V}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{V}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \quad (4)$$

$$\tilde{\mathbf{c}}_t = g(\mathbf{W}_c \mathbf{x}_t + \mathbf{V}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \quad (5)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \quad (6)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot g(\mathbf{c}_t) \quad (7)$$

where $\mathbf{x}_t$ is the input vector at time t, $\mathbf{W}_* \in \mathbb{R}^{N\times M}$, $\mathbf{b} \in \mathbb{R}^N$, $\mathbf{V}_* \in \mathbb{R}^{N\times N}$ are shared model parameters at all time steps with M and N denoting the dimension of input vectors and hidden states, respectively. σ and g are pointwise nonlinear activation functions. The logistic sigmoid ($\sigma(x) = 1/(1+e^{-x})$) and hyperbolic tangent ($\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$) are usually adopted as gate activation function and block input output activation function for σ and g, respectively. Operator $\odot$ denotes the element-wise product. Except adding cell state to each gate in transition functions of peephole cell, the rest is the same as those in basic LSTM. Therefore, no more details of peephole cell transition functions given here.

A gated recurrent unit (GRU) (see Fig. 2 (b)) is a simplified version of LSTM architecture introduced in 2014 [29]. It is different from basic LSTM in the following aspects: 1) GRUs containing two gates $\mathbf{r}_t$ and $\mathbf{z}_t$ only and possessing no internal memory $\mathbf{c}_{t-1}$, 2) a second non-linearity g not being applied as that in Fig. 2 (a). The transition functions in hidden units of GRU are given in (8)-(11) with $\mathbf{x}_t$, $\mathbf{W}_* \in \mathbb{R}^{N\times M}$, $\mathbf{V}_* \in \mathbb{R}^{N\times N}$, $\mathbf{b} \in \mathbb{R}^N$, σ and g same as those in basic LSTM functions.

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{V}_r \mathbf{h}_{t-1} + \mathbf{b}_r) \quad (8)$$

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{V}_z \mathbf{h}_{t-1} + \mathbf{b}_z) \quad (9)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{r}_t \odot \mathbf{V}\mathbf{h}_{t-1}) \quad (10)$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \quad (11)$$

Among three LSTM cell structures, only the cell with peephole connections allows its gates to observe the cell states directly. Since this structure is good at timing [28], the peephole structure is adopted in this paper for RUL prediction and DS assessment.

B. *Dual-task deep LSTM network*

Based on the flexibility and versatility of the DL structure, a joint classification and regression is proposed under the framework of multitask RNN in this section. The architecture of a new established end-to-end multitask RNN is presented in Fig. 3. Three main components are included under this framework: a deep LSTM network composed of two-stacked LSTM layers for a deep feature extraction, a classification subnetwork for DS task and a regression network for RUL prediction task. And the peephole cell is taken to deal with this time series problem in this work since its direct inspection of current internal states favors the prediction work.

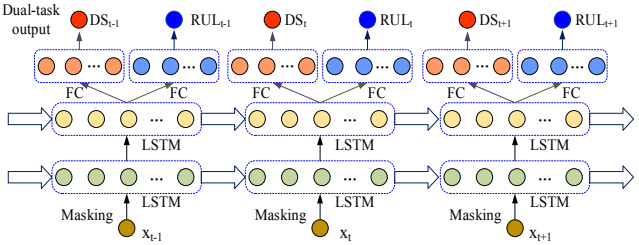

Fig. 3. Dual-task deep LSTM neural networks architecture for DS assessment and RUL prediction.

The following presents a more detailed description of the dual-task deep LSTM networks. First of all, the dynamic RNN cell is adopted to deal with variable length sequences since different individual equipment has different life span. Then, the normalized sensor signals pass through a masking layer to make all sequence samples the same length once they enter into the network. Next, they flow into a two-stacked LSTM layers network (circles in olive stand for the 1st LSTM layer and circles in yellow the 2nd) to learn sequence relations among time series vectors. This two-stacked LSTM layers is designed on one hand to improve the structural spatial characteristics to enhance characterization capabilities and on another to extract temporal dynamic characteristics of the frame. And then, signals divert into two branches from the deep LSTM network output to achieve its assigned tasks of classification and regression. A fully connected (FC1) neural network (circles in orange) with hyperbolic tangent function as activation function and with output encoded into one-hot vector to differentiate different DSs represented in red circles is designed for the classification task. As for the RUL prediction task, a multilayer network with three fully connected nonlinear layers (FC2, FC3 and FC4 represented by circles in light blue) is designed to increase the network complexity and with ReLU expressed by $f(x) = \max(0, x)$ as activation function for the last layer to produce specific RUL value represented by circles in red.

C. *Optimization of the architecture*

Since the whole network is to achieve both tasks of DS assessment and RUL prediction, both losses of these two

subnetworks are calculated into the total loss of the whole network to form the learning objective as:

$$L = L_{RUL} + \alpha L_{DS} \tag{12}$$

where L_{RUL} and L_{DS} are the RUL prediction loss and DS assessment loss, respectively, and the DS loss is weighted by an scalar α to achieve a balance between these two losses.

For the classification subnetwork, loss function of the DS assessment task is the cross entropy between the target probabilities $y_{t,k}$ and the outputs $\hat{y}_{t,k}$ as presented in (13):

$$L_{DS} = -\frac{1}{n}\sum_{t=0}^{n-1}\sum_{k=0}^{C} y_{t,k} \ln \hat{y}_{t,k} \tag{13}$$

where the target probabilities $y_{t,k}$, taking values of one or zero, are the supervised information provided to train the classifier, $\hat{y}_{t,k}$ corresponds to the predicted category at time point t and k is the ground truth label, C is the number of all DS classes, n is the life span of a sample aero-engine and the same below.

And for the regression subnetwork, mean square error (MSE) is used as loss functions of the RUL prediction task as presented in (14).

$$L_{RUL} = \frac{1}{n}\sum_{t=0}^{n-1}(l_t - \hat{l}_t)^2 \tag{14}$$

where $\hat{l}_t$ corresponds to the predicted RUL value at time point t and l_t is the ground truth label.

III. Target Label Construction and Data Visualization

Signals from an aero-engine are of multi-dimensional no-label properties in engineering practice. No label signifies that the degenerate patterns and degradation levels of aero-engines with time are unknown. Therefore, labelling is needed for the training of the dual-task deep LSTM network established before. That is, it needs to construct target label functions for DS assessment and RUL prediction respectively before the sensor data is fed into the designed network.

The remainder of this section is organized as follows. Firstly, the data set adopted in this paper is presented in Section A. Then the data preprocessing procedure as shown in Fig.4 for target label construction is detailed in Section B. Lastly, the constructed target label is visualized in Section C.

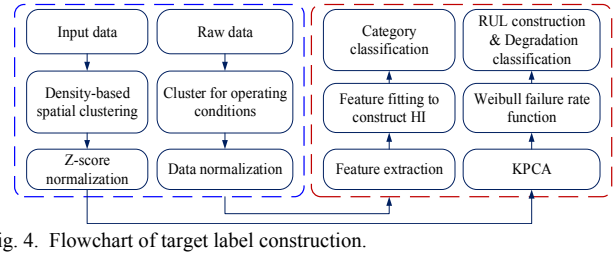

Fig. 4. Flowchart of target label construction.

A. Data overview

The data set adopted in this paper was generated by software simulator commercial modular aero-propulsion system simulation (C-MAPSS) developed by NASA [30]. This data set is comprised of four subsets from different issues as detailed in Table I where each engine trajectory signifies each engine life-cycle. Each engine record data is in the format of $n \times 26$ matrix where n is the run time of a certain engine trajectory represented by cycles and 26 including one column of engine ID number, one column of operational cycle number, three columns of operating settings which can affect the engine performance dramatically and 21 other columns of sensors signals [30]. Each engine is of a certain level of initial wear which is considered to be normal instead of faults to represent individual difference. The fault initiates at some unknown time point in the running process.

TABLE I
C-MPASS DATA SET

Data Set	FD001	FD002	FD003	FD004
Train trajectories	100	260	100	249
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2
Maximum life span(cycles)	362	378	525	543
Minimum life span(cycles)	128	128	145	128

B. Target label construction

1) Operating conditions clustering and data normalization

As listed in Table I, subsets FD002 and FD004 are within discrete multiple operating conditions. Though it is already informed that there are six categories of operating conditions in those two subsets, this work is still dedicated to studying a more general approach. Therefore, the density-based spatial clustering of applications with noise (DBSCAN) [31] method is adopted to cluster data with the same operating conditions into one cluster. DBSCAN is chosen because it enjoys two major advantages over other traditional clustering methods. That is, it is applicable to clustering with unknown number of categories and adaptable to data sets with an arbitrary shape.

The z-score normalization is adopted in this paper. Moreover, considering the multiple operating conditions within subsets FD002 and FD004, a unified z-score normalization is needed for all the sub-data sets as given in (15) [25].

$$x_i' = \frac{x_{i,c} - \mu_{i,c}}{\sigma_{i,c}} \tag{15}$$

where c represents operating conditions, i stands for the i-th sensor, $\mu_{i,c}$ and $\sigma_{i,c}$ are the mean value and corresponding standard deviation of the i-th sensor under operating condition c, respectively. x_i' is the final normalized sensor data.

To have a direct and vivid understanding of the data from C-MAPSS set, more data details are displayed before the final results presented. The sub dataset FD001 is taken to illustrate the effect of data preprocessing. Though there are 21 sensor signals, there remains 14 signals for each engine sample after normalization since there appears not a number for sensor signals keeping constant which are excluded from entering the network. Since more comprehensive degradation information can be retained through a principal component (PC) extraction for a health indicator (HI) construction for the further analysis, all these signals instead of some certain selected sensor signals are adopted in the following.

2) KPCA for deterioration feature extraction

To extract a comprehensive degeneration characteristic from the multi no-label data is a key fundamental for the performance degradation assessment and RUL prediction. Therefore, to adapt the characteristics of degeneration data and to guarantee the nonlinearity of feature extraction, the unsupervised feature extraction method named kernel principal component analysis

(KPCA) [32] is adopted as feature extraction method. The top eigenvectors, that is, the principal components (PCs) represent comprehensive degradation characteristics of an equipment. To facilitate the comparison of different principal components, to reveal the essence of deterioration degree and to construct a unified HI for all the engines, all the components are in normalization.

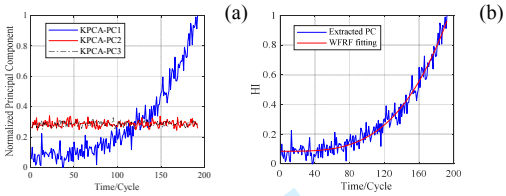

Fig. 5. (a)First three PCs from deterioration feature extraction based on KPCA; (b) Fitted extracted 1st PC as HI

The sensor data of an arbitrary engine from subset FD001 is adopted to display the extracted first three PCs information based on KPCA as shown in Fig. 5 (a). It can be seen that, except for the first PC, other PCs do not exhibit a degeneration trend in the whole lifetime of all the aero-engines in the dataset. Therefore, the first eigenvector is taken as extracted feature for further analysis in this paper. In this way, a comprehensive degeneration characteristic is extracted from these multi no-label signals.

3) WFRF fitting for HI *construction*

Though the feature extraction is carried out and possesses a global trend, it has a serious local oscillation which is not an ideal output for deep LSTM network training. Therefore, it needs a further fitting procedure for the final HI construction. Weibull distribution (WD) [33] is widely used as a failure model in reliability and life data analysis due to its relative simplicity and versatility. Considering the research focus of this paper is on the deterioration assessment and RUL prediction, the Weibull failure rate function (WFRF) [34] is adopted for the extracted feature smoothing in time domain. Meanwhile, to adapt the WFRF to different levels of initial state, a constant value is added to the classical WFRF as presented in the following form:

$$\lambda(t,\beta,\eta,c,K) = c + K\frac{\beta}{\eta^{\beta}}t^{\beta-1} \tag{16}$$

where c is the constant value to denote the initial failure rate, β is the shape parameter which establishes the curve shape, η is the scale parameter, the introduced parameter K is to scale the fitted values to any range and these four parameters are evaluated from fitting the extracted PC of each aero-engine. The ingenuity of this formula lies in its clear description of the initial static state with c and the follow-up dynamic degradation by $K(\beta/\eta^{\beta})t^{\beta-1}$. Fig. 5 (b) shows the fitting effect of the 1st PC from Fig. 5 (a). The fitted WFRF parameters are within a 95% confidence interval and the estimated values of c, K, β and η for the HI of this engine are 0.0849, 0.0128, 4.3970 and 31.1168, respectively.

4) Target label construction for DS division and RUL prediction

Once the HI is established from the fitting process, the engine system performance degeneration in the whole lifetime is divided according to the fitted results. As different individual has different initial value at its initial state, the system initial value should be set to zero for the following classification work as to unify the classification for all the individual system. That is, only the dynamic degradation part represented by $K\frac{\beta}{\eta^{\beta}}t^{\beta-1}$ is adopted for a unified degeneration classification. Fig. 6 (a) presents the DS division results based on HI where Stage-0 represents the initial healthy state depicted by the green line. From Stage-1 to Stage-3 the system deteriorates at a moderate speed. From Stage-6 to Stage-7, the system performance aggravates at a high rate and degrades to a rather severe state. That is, the system is about to its failure time once the system is upon to or into Stage-7.

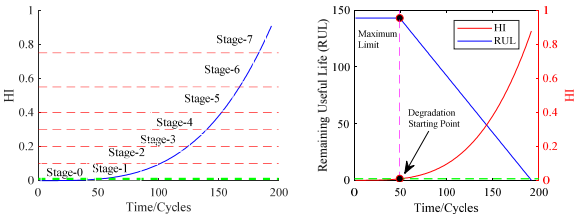

Fig. 6. (a) DS division based on HI; (b) RUL target construction from degradation starting point.

The ideal way to assign values to RUL target is to make RUL decrease linearly with time. It is implicit that the system health state deteriorates linearly with time. However, the system deterioration is usually too slight to be noticeable during its initial using stage and aggravates with time especially when it is upon the end of life. Therefore, a piece-wise linear RUL target label function by limiting the maximum RUL to a constant value was adopted in [23, 25, 35] to better model the RUL change over time. However, they set the maximum limit of all the data samples at a constant value of 130 time/cycles by trial and error. Though this may lead to a better prediction results, a better prediction value is not all the prediction work about. The ultimate aim of the prediction work is to carry out timely maintenance work to prevent further losses. Besides, different system equipment individual has different degradation paths, it is more reasonable to set the maximum limit of time cycles according to the system health state instead of a same certain value which can be too high or too low for different individuals. In this paper, the maximum limit of the time cycles is set at the end cycle of the initial healthy state which is also the degradation starting point as presented in Fig. 6 (b). The right axis in Fig. 6 (b) stands for HI presented by the red line and the left axis is the RUL target piece-wise function presented by the blue line. The cross point between the green line and the red line marks where the degradation starts. In this way, Fig. 6 (b) depicts specifically the proposed method to determine the target RUL function in this paper.

C. *Target label visualization*

Fig. 7 (a) displays the HIs for all the engine samples in FD001 constructed from the normalized extracted 1st PC by KPCA where a radial basis function with a parameter of 18 is adopted. Their fitting ones by WFRF are shown in Fig. 7 (b). Finally, the initial individual health state difference is removed from each engine sample for the target label function construction of DS division and RUL prediction as shown in Fig. 7 (c). From Fig. 7 (d), it can be seen that different engine individual has quite different degradation starting point which

can be much greater or smaller than the setting constant 130 used in [23, 25, 35] which enables an appropriate capture of engine degradation information later on.

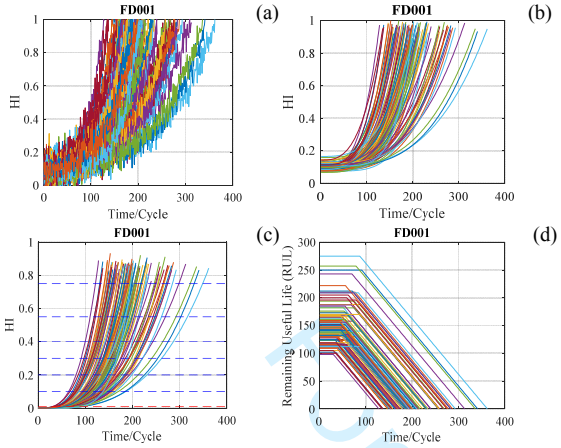

Fig. 7. Procedure of target label function construction: (a) extracted 1st PC by KPCA as HI; (b) HI fitting by WFRF; (c) initial health state removal for DS target label construction; (d) RUL target label construction.

IV. Experiments

A. Evaluation metrics

To illustrate and assess the performance of our proposed approach, the metrics detailed in the following are adopted.

1) RMSE

Root Mean Square Error (RMSE) of estimated RUL is employed as a performance measure for a single engine as given in (17). It defines the RMS of prognostic errors for an engine from life starting point to the end of life, where n is the total number of prognostic points in the whole life cycle of an engine and $h(i)$ is the prognostic error at each life cycle point as given in (18). Moreover, RUL_l presented in (19) denotes $RMSE$ value of the last l cycles within the life span. A series of values as 120, 80, 50, 20, 10, 5 and 1 are chosen for l to demonstrate the prediction performance over time.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} h(i)^2} \tag{17}$$

$$h(i) = RUL_{estimated}(i) - RUL_{actual}(i) \tag{18}$$

$$RUL\_l = \sqrt{\frac{1}{l}\sum_{i=n-l+1}^{n} h(i)^2} \tag{19}$$

2) Classification accuracy

$Accu$ is calculated as the proportion of the number of correctly evaluated degradation levels at each time point to the whole life span for a single engine as given in (20) with n same as that in (17).

$$Accu = \frac{n_{correctly\_evaluated}}{n} \tag{20}$$

What's more, the mean value m_* (21) with * referring to $RMSE$ and $Accu$ on the whole testing samples from each C-MAPSS dataset are also calculated to illustrate the overall performance of the proposed approach, where N is the number of testing engine samples.

$$m\_* = \frac{1}{N}\sum_{j=1}^{N} *_j \tag{21}$$

B. Experimental Settings

Only the training set in the whole C-MAPSS dataset depicted above is used for training and validation of the proposed approach in this paper. A 5-fold cross-validation procedure based on the training set only is adopted to determine the hyper parameters of the established dual-task deep LSTM networks architecture. The determined network structure is L(32, 32)N_R(16, 8, 1)N_D(8) where L(32, 32) stands for LSTM with two layers and each layer with 32 nodes and N_R(16, 8, 1) denotes the regression sub-network with three fully connected (FC) layers and each layer with 16, 8 and 1 node, respectively, and N_D(8) denotes the classification sub-network with 8 classes to classify. The Adam optimizer [36] is chosen to train models with an initial learning rate of 0.001. One dropout regularization (the dropout rate is selected as 0.9 in this paper) is utilized through the whole network to prevent over-fitting. The value of scalar α presented in (12) is chosen as 200. Moreover, 20% engine samples from each sub dataset are randomly picked out as testing data. That is, 20, 52, 20 and 49 engine samples for sub dataset FD001, FD002, FD003 and FD004, respectively.

C. Results presentation and discussion

The statistical mean RMSE values within a series of RULs of different cell structures and sub datasets are reported in Table II.

1) Different LSTM cell structures performance comparison

A comparison has been drawn on the performance of different LSTM cell structures. Here the most commonly used three LSTM cell structures, the basic LSTM, GRU and peephole cell, as mentioned before are tested on sub dataset FD001 from C-MAPSS. It can be seen from the first 3 columns of Table II that, all of these three LSTM cells can achieve a relatively good results in RUL prediction in general. However, the peephole cell outperforms other two cells in terms of both RUL prediction and degradation assessment. The good performance of peephole cell benefited from its own sequence-memory-enhanced structure is beneficial for the RUL prediction and DS assessment problem presented in this paper. Therefore, the peephole cell is adopted in the following.

2) Comprehensive performance validation

To demonstrate the comprehensive performance of the proposed framework, the tested results of the whole dataset of C-MAPSS are listed in Table II in bold. It can be seen that the RMSE of RUL prediction decreases dramatically with the decrease of l. That is, the later the running time is, the more accurate the RUL prediction result is. Meanwhile, the overall degradation assessment accuracy is above 80% as listed in the last row of Table II in bold. It signifies that the proposed approach can not only accurately predict RUL value but also correctly assess the DS at the same time. Moreover, the comparison results of the proposed framework with other popular approaches are listed in Table III where the IMP in the last row signifies the improvement of our proposed framework over the state of the art that proposed in [23] (calculated by *IMP* = 1- *Dual-task LSTM/Deep LSTM*). It can be seen that, the deep learning methods outperforms traditional shallow structure machine learning methods and LSTM with its self-feedback structure and sequence memory characteristic outperforms the

feedforward structure CNN and our proposed frame work outperforms the deep LSTM method presented in [23]. This improvement can be attributed to the following two reasons. One is that the peephole cell structure is adopted instead of the traditional basic one enabling the memory enhancement of time sequence series and another is that the target RUL prediction function is constructed adaptively from an individual aero-engine state instead of being assigned to a same certain value to all the aero-engines. It's worth mentioning that, the accompanied DS assessment output by our proposed frame work can be considered as a plus and a supplement of the RUL prediction results since the DS results are more stable which allows the final results more robust and reliable. More details will be illustrated in the following individual analysis.

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT LSTM CELL STRUCTURES ON C-MAPSS DATA IN TERMS OF RMSE AND ACCURACY

Data Set / Cell	FD001 basic LSTM	FD001 GRU	FD001 **peep-hole**	FD002 **peep-hole**	FD003 **peep-hole**	FD004 **peep-hole**
Rul_all	12.41	12.35	**12.29**	**17.87**	**14.34**	**21.81**
Rul_120	9.51	10.34	**7.62**	**10.37**	**8.99**	**9.66**
Rul_80	7.95	7.66	**4.87**	**7.09**	**6.70**	**6.77**
Rul_50	7.03	6.37	**3.58**	**5.56**	**5.23**	**5.43**
Rul_20	5.01	6.09	**3.16**	**4.36**	**5.87**	**4.60**
Rul_10	4.17	4.86	**3.37**	**3.58**	**5.02**	**3.47**
Rul_5	2.32	2.45	**2.38**	**2.38**	**2.45**	**2.17**
Rul_1	0	0	**0**	**0.29**	**0**	**0.082**
Accu	0.61	0.81	**0.85**	**0.831**	**0.813**	**0.85**

TABLE III
RMSE COMPARISON ON C-MAPSS DATA WITH OTHER BASELINES

Data Set	FD001	FD002	FD003	FD004
MLP[25]	37.56	80.03	37.39	77.37
SVR[25]	20.96	42.00	21.05	45.35
RVR[25]	23.80	31.30	22.37	34.34
CNN[25]	18.45	30.29	19.82	29.16
Deep LSTM[23]	16.14	24.49	16.18	28.17
Dual-task LSTM	12.29	17.87	14.34	21.81
IMP	**23.85%**	**27.03%**	**11.43%**	**22.58%**

3) *Individual performance analysis*

The detailed individual prediction result is presented in the following. One testing example of whole life span per sub dataset from C-MAPSS is taken as example to demonstrate the RUL prediction and DS results shown in Fig. 8. Fig. 8 shows the estimated RUL and DS at any time/cycle according to time sequences inputted into the LSTM network through the whole lifetime. It can be seen that, there is a relatively large prediction error at the initial healthy stage of an engine in terms of RUL prediction. However, the assessed degradation level of this period can serve as a supplement due to its stability and accuracy during the most time of this period.

The predicted RUL comes closer and closer to the piece-wise RUL as the engine health state deteriorates. This phenomenon is important and beneficial to the equipment health monitoring since an accurate prediction in the later period is more critical to monitoring decisions. This benefits from the LSTM network. Assuming the life span of an engine is T, the LSTM network constructed in this paper is to use a time sequence $[x_1, x_2, x_3, \cdots x_t]$ (x_i as the preprocessed sensor vector at time point i) to predict the RUL value as $T-t$ and its corresponding DS at time t. The longer the sequence is, that is, the larger t is, more accurate prediction results are, since the LSTM network can learn more information from a longer sequence effectively. On the other hand, this result also confirms the effectiveness of the proposed approach in this paper. Moreover, degradation classification confusion matrixes corresponding to these four testing examples are presented in Fig. 8. It can be seen clearly the correct classification rate at each DS during the whole lifetime for an individual tested engine. In general, it reflects the overall deterioration trend of an engine. The DS assessment is roughly correct at most of the time and misclassification usually occurs at two adjacent stages (Fig. 8 as reference). On one hand, as the natural degradation of the engine state is a continuous process, it is insignificant to carry out an exact separation at the junction of two stages. On the other hand, the pursuit of degradation assessment is more of reflecting equipment status rather than a high value of accuracy. In this sense, the purpose of degradation assessment has been reached.

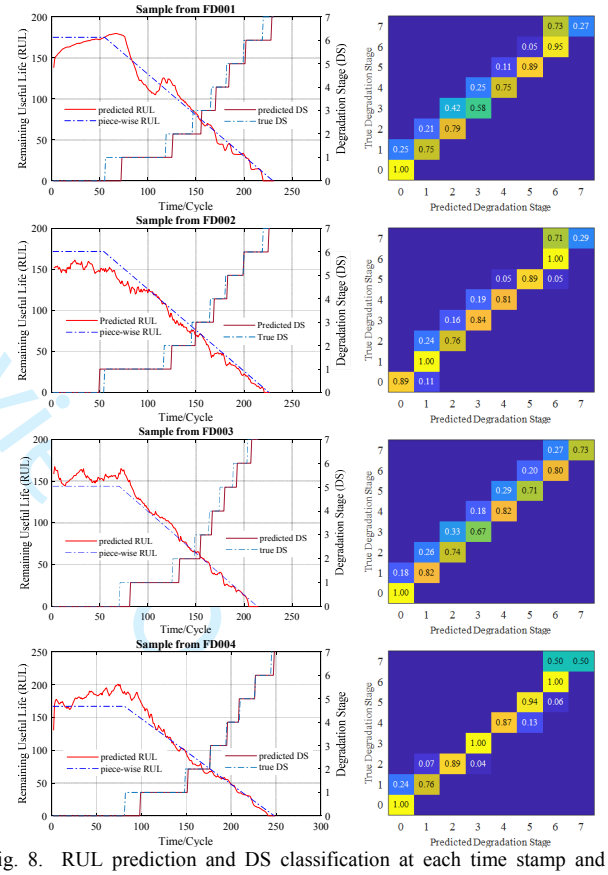

Fig. 8. RUL prediction and DS classification at each time stamp and the corresponding DS confusion matrix for each sample

Overall, the proposed framework can achieve satisfactory performance both in RUL prediction and DS assessment. This jointly learning contributes to a more robust and accurate results since DS is more stable and RUL is specific and becoming more accurate when it comes to the later running period.

V. CONCLUSION

This paper proposes a dual-task deep LSTM network for a jointly learning of RUL prediction and DS assessment. Meanwhile, a general solution is proposed to construct the dual training objective functions based on DBSCAN, KPCA and

WFRF which allows DS division and RUL labels to be assigned adaptively according to the equipment health state. Hyper parameters of the established dual-task deep LSTMs network are determined from a five-fold cross validation. Based on the determined network, performance of different LSTM cell structures, that is, basic LSTM, peephole and GRU, has been compared. The results show that peephole outperforms the other two due to its structure including the original prior state in the calculation of gates thus resulting in the enhancement of sequence memory. Finally, the experimental application on the widely used dataset C-MAPSS achieved a satisfactory result in terms of both DS assessment and RUL prediction. It reflects the accuracy and effectiveness of the proposed approach in this paper.

Though some preliminary work has been carried out in this paper, deep learning reveals its powerful application potentials within the PHM architecture frame. Further research can be carried out in the following aspects. Firstly, how to include the operating conditions directly into the network. Secondly, how to construct the target label function directly by designing a more flexible network structure.

REFERENCES

[1] A. J. Volponi, "Gas turbine engine health management: past, present, and future trends," *Journal of Engineering for Gas Turbines and Power,* vol. 136, p. 051201, 2014.

[2] F. M. Discenzo, W. Nickerson, C. E. Mitchell, and K. J. Keller, "Open systems architecture enables health management for next generation system monitoring and maintenance," *Development Program White Paper,* 2001.

[3] K. Swearingen, W. Majkowski, B. Bruggeman, D. Gilbertson, J. Dunsdon, and B. Sykes, "An open system architecture for condition based maintenance overview," in *Aerospace Conference, 2007 IEEE*, 2007, pp. 1-8.

[4] O. Benedettini, T. Baines, H. Lightfoot, and R. Greenough, "State-of-the-art in integrated vehicle health management," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering,* vol. 223, pp. 157-170, 2009.

[5] B. Finley and E. A. Schneider, "ICAS: the center of diagnostics and prognostics for the United States navy," in *Component and Systems Diagnostics, Prognosis, and Health Management*, 2001, pp. 186-194.

[6] J. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mechanical Systems and Signal Processing,* vol. 25, pp. 1803-1836, 2011.

[7] G. Chen, J. Chen, Y. Zi, and H. Miao, "Hyper-parameter optimization based nonlinear multistate deterioration modeling for deterioration level assessment and remaining useful life prognostics," *Reliability Engineering & System Safety,* vol. 167, pp. 517-526, 2017.

[8] Z. Zhao, B. Liang, X. Wang, and W. Lu, "Remaining useful life prediction of aircraft engine based on degradation pattern learning," *Reliability Engineering & System Safety,* vol. 164, pp. 74-83, 2017.

[9] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks," *IEEE Transactions on Industrial Electronics,* vol. 65, pp. 1539-1548, 2018.

[10] M. Ma, C. Sun, and X. Chen, "Discriminative Deep Belief Networks with Ant Colony Optimization for Health Status Assessment of Machine," *IEEE Transactions on Instrumentation and Measurement,* vol. 66, pp. 3115-3125, 2017.

[11] C. Sun, Z. Zhang, X. Luo, T. Guo, J. Qu, and B. Li, "Support vector machine-based Grassmann manifold distance for health monitoring of viscoelastic sandwich structure with material ageing," *Journal of Sound and Vibration,* vol. 368, pp. 249-263, 2016.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature,* vol. 521, p. 436, 2015.

[13] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, pp. 14-22, 2012.

[14] H. Shao, H. Jiang, H. Zhang, and T. Liang, "Electric Locomotive Bearing Fault Diagnosis Using a Novel Convolutional Deep Belief Network," *IEEE Transactions on Industrial Electronics,* vol. 65, pp. 2727-2736, 2018.

[15] M. Ma, C. Sun, and X. Chen, "Deep Coupling Autoencoder for Fault Diagnosis With Multimodal Sensory Data," *IEEE Transactions on Industrial Informatics,* vol. 14, pp. 1137-1145, 2018.

[16] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement,* vol. 89, pp. 171-178, 2016.

[17] C. Sun, M. Ma, Z. Zhao, and X. Chen, "Sparse Deep Stacking Network for Fault Diagnosis of Motor," *IEEE Transactions on Industrial Informatics,* 2018.

[18] J. Pan, Y. Zi, J. Chen, Z. Zhou, and B. Wang, "Liftingnet: a novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification," *IEEE Transactions on Industrial Electronics,* vol. 65, pp. 4973-4982, 2018.

[19] Z. Liu, Z. Wu, T. Li, J. Li, and C. Shen, "GMM and CNN hybrid method for short utterance speaker recognition," *IEEE Transactions on Industrial Informatics,* 2018.

[20] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, pp. 2481-2495, 2017.

[21] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, *et al.*, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 23, pp. 530-539, 2015.

[22] R. Nallapati, B. Xiang, and B. Zhou, "Sequence-to-sequence rnns for text summarization," 2016.

[23] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long Short-Term Memory Network for Remaining Useful Life estimation," in *Prognostics and Health Management (ICPHM), 2017 IEEE International Conference on*, 2017, pp. 88-95.

[24] 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, 2018.

[25] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *International conference on database systems for advanced applications*, 2016, pp. 214-228.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, pp. 1735-1780, 1997.

[27] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of machine learning research,* vol. 3, pp. 115-143, 2002.

[28] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, 2000, pp. 189-194.

[29] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078,* 2014.

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

[31] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*, 1996, pp. 226-231.

[32] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International Conference on Artificial Neural Networks*, 1997, pp. 583-588.

[33] W. Weibull, "A statistical distribution function of wide applicability," *Journal of applied mechanics,* vol. 18, pp. 293-297, 1951.

[34] J. B. Ali, B. Chebel-Morello, L. Saidi, S. Malinowski, and F. Fnaiech, "Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network," *Mechanical Systems and Signal Processing,* vol. 56, pp. 150-172, 2015.

[35] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, 2008, pp. 1-6.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.