A Deep Branched Network for Failure Mode Diagnostics and Remaining Useful Life Prediction

Zhen Li, Member, IEEE, Yongxiang Li, Member, IEEE, Xiaowei Yue, Member, IEEE, Enrico Zio, Senior Member, IEEE and Jianguo Wu, Member, IEEE

Abstract—In complex systems, the operating units often suffer from multiple failure modes, and each failure mode results in distinct degradation path and service life. Thus, it is critical to perform the failure mode diagnostics and predict the remaining useful life (RUL) accordingly in modern industrial systems. However, most of the existing approaches consider the prognostic problem under a single failure mode, or treat the failure mode classification and RUL prediction as two independent tasks, despite the fact that they are closely related and should be synergistically performed to enhance the generalization performance. Motivated by these issues, we propose a deep branched network (DBNet) for failure mode classification and RUL prediction. In this approach, the two tasks are jointly learned in a sequential manner, wherein the feature extraction layers are shared by both tasks, while the neural network branches into individualized subnetworks for RUL prediction of each mode based on the output of the diagnostic subnetwork. Different from the traditional multi-task learning-based methods, where the diagnostics and RUL prediction are performed in parallel, the proposed DBNet innovatively couples these two tasks sequentially to boost the prognostic accuracy. The effectiveness of the proposed method is thoroughly demonstrated and evaluated on an aircraft gas turbine engine with multiple failure modes.

Index terms—Deep branched network, multi-task learning, remaining useful life prediction, failure mode diagnostics, deep learning.

I. INTRODUCTION

Nowadays, with the rapid development of sensor and information technologies, multiple sensors are commonly embedded in complex systems to form a sensor network for prognostics and health management (PHM) [1], [2]. Various data fusion and feature extraction techniques are developed to handle the high-dimensional sensor data. However, most of the existing approaches only consider one failure mode, or ignore the differences between multiple potential failure modes [3]. In fact, the practical systems like rotating machinery [4], tire plants [5] and communication systems [6] are susceptible to multiple intrinsic failure modes. For different failure modes, the degradation processes may be significantly different, which makes the condition monitoring and prognostics more challenging. Figure 1 gives an example of the degradation profiles for two units undergoing different failure modes, from the C-MAPSS dataset [7]. Clearly, different failure modes have distinct influences on the degradation trajectories. Using only a single unified prognostic model to predict the RUL may result in either low generalization performance across different failure modes, or high model complexity for approximating piecewise functions. Therefore, failure mode diagnostics is an essential first step for achieving accurate and robust prognostics.

To deal with the RUL prediction problem in the case of multiple failure modes, two-step approaches are typically applied. In these approaches, the sensory data are classified into groups for each mode based on a diagnostic model, and then several corresponding prognostic models are developed independently with respect to each mode data. Yan et al. [8] proposed a spectral correlation density combination method for fault type and scale diagnostics, and applied a semi-supervised co-training based approach for RUL prediction. Chehade et al. [9] first estimated the failure mode based on the calculated posterior probability, and then built a Bayesian degradation model and determined a failure threshold in RUL prediction for each mode. All of these methods handle failure mode classification and RUL prediction separately. Actually, these two tasks are highly correlated, and joint learning can leverage each other’s strengths to improve generalization performance via shared feature representation, especially when there are...
limited data.

This paper proposes a deep branched network (DBNet) for failure mode diagnostics and RUL prediction. The major novelties and contributions of the approach can be summarized as follows:

1) Failure mode classification and RUL prediction are jointly learned in the DBNet in a sequential manner, wherein the feature extraction layers are shared by both tasks, whereas the neural network branches into individualized subnetworks for RUL prediction of each mode based on the output of the diagnostic subnetwork.

2) Different from the traditional multi-task learning (MTL) methods, where two or more tasks are learned in parallel through a combined loss function, the proposed joint learning method innovatively couples the classification result with the RUL prediction, thus boosting the prognostic accuracy.

3) In DBNet, individualized models for RUL prediction with respect to each mode are developed; this enables the knowledge transfer among RUL prediction task under different modes, thus realizing implicit data augmentation.

The rest of this paper is organized as follows. Section II provides a literature review on the neural-network-based methods, as well as the traditional multi-task learning approaches for diagnostics and prognostics. Section III presents a general mathematical formulation of the problem, and then describes the detailed methodology of the deep branched network for diagnostics and prognostics. In Section IV, the proposed approach is applied to a high-fidelity aircraft gas turbine engine simulation dataset with multiple failure modes. The effectiveness of the proposed approach is evaluated and compared with several state-of-the-art methods. Section V concludes the paper and outlines future directions.

II. LITERATURE REVIEW

A. Neural-Network-based Approaches for Diagnosis and RUL Prediction

Given the readily available multi-sensory data, the data-level fusion approaches have gradually become a major focus in fault diagnostics and RUL prediction [10], [11]. Due to the capability of approximating any functions, the neural-network-based algorithms for diagnostics and RUL prediction are extensively studied in recent years. For cases involving multiple failure modes, several automated fault diagnosis schemes using online neural network learning have been designed. Karpenko et al. [12] developed a multilayer neural network to diagnose actuator faults of interest. Huang and Tan [13] employed a radial basis function (RBF) neural network to capture the non-linear characteristics of the fault function, and designed an additional neural network to identify the failure mode. For RUL prediction, Riad et al. [14] used a classical multi-layer perceptron (MLP) to provide an accurate estimation, demonstrating the superiority of the MLP method over a simple linear regression model. To further improve the prediction performance, many researchers applied artificial neural networks (ANNs) to predict the RUL [15]. For example, Gao et al. [16] proposed a joint prognostic model, where ANN is applied to model the non-linear relationship between the multiple sensory signals and RUL. Li et al. [17] presented a shape-constrained neural network to construct a composite health index, which facilitates the following-up RUL prediction.

Recently, deep neural-network-based techniques have become more and more popular, due to their flexibility in handling high-dimensional data and extracting the temporal features [18]. Yang et al. [19] utilized a deep convolutional neural network (CNN) to extract features for RUL prediction. Since the recurrent neural network (RNN) can effectively deal with sequential data, Malhi et al. [20] proposed an RNN-based approach for capturing the long-term dependency of sequential vibration signals for RUL prediction of rolling bearings. However, because the RNN is fundamentally an extremely deep feedforward network with the same weights across all layers, it’s difficult to keep the long-term information memorized. Therefore, to avoid gradient vanishing or exploding problems that can occur while training the traditional RNN, Yuan et al. [21] proposed a long-short term memory (LSTM) based framework for RUL prediction, and showed that the performance of LSTM is superior to RNN for temporal data fusion. Besides, the graph convolutional network techniques are also powerful for high-dimensional feature representation by considering the topological relationship between the graph data [22], [23]. Li et al. [24] proposed a multi-receptive field graph convolutional network for effective intelligent fault diagnosis. Chen et al. [25] presented a hybrid GCN method by considering the prior knowledge to further improve the fault diagnosis performance. Li et al. [26] constructed a network for multiple sensors and proposed a hierarchical attention graph convolutional network for modeling the sensor network. However, the aforementioned deep learning neural network approaches were developed for only a single task, i.e., either failure mode classification or RUL prediction.

B. Multi-task Learning Approaches

In the existing literature, most methods treat diagnostics and prognostics as two separate tasks, which may reduce the generalization performance and require sufficient data for each task. Recently, several multi-task learning approaches have been proposed to improve generalization performance, with a small number of data [27]. It is an approach that learns the related tasks simultaneously, which can improve the generalization ability by utilizing the mutual information contained in related tasks as an inductive bias. Figure 2 shows a typical framework for dual-task learning, where the shallow layers are shared by both tasks, and then branched out to individual ones. Under this architecture, different tasks are integrated into one framework through joint optimization, achieving an inductive transfer by leveraging the affinities between tasks.

Based on the above MTL framework, Miao et al. [28] designed a dual-task learning approach for degradation stage classification and RUL prediction under the consideration that the RUL is closely related to the degradation stage. However, in this work, the degradation stage classification regards only
the detection of whether a fault occurs or not, therefore is not applicable to fault diagnostics with multiple failure modes. Wang et al. [29] utilized the MTL technique to classify the failure modes and predict the RUL simultaneously. In their method, logistic regression and linear regression are respectively applied for failure classification and RUL prediction, which may not be effective in extracting temporal features and capturing the highly non-linear relationship. Recently, Liu et al. [30] constructed a dual-task learning neural network for fault diagnostics and RUL prediction with high-dimensional time series data. This method can infer whether the failure is resulted from the inner or outer of the bearing race while also performing the RUL prediction. Although these dual-task learning approaches can efficiently utilize the shared feature representation for both tasks to improve the generalization performance, the effectiveness of these MTL-based works is still limited by the parallel structure. In these methods, the diagnostic results are not integrated into the RUL prediction to boost the prognostic accuracy. Instead, a single unified prognostic model is established for RUL prediction for different failure modes. It behaves like using a general model to approximate a piecewise function, with each piece representing a prognostic function for the corresponding failure mode. Consequently, it poses a significant burden on the model complexity to achieve desirable prognostic accuracy.

III. DEEP BRANCHED NETWORK FOR JOINT LEARNING

Targeted at the aforementioned research gap, this paper proposes a deep branched neural network (DBNet) to extract features through deep learning as well as jointly learn the diagnostic and prognostic tasks. In the following subsections, we first present the structural configuration of DBNet and the basic concept of LSTM for the sake of completeness, and then state the objective function and optimization method in detail.

A. DBNet Architecture

Multiple sensors installed on a system collect the condition monitoring data from various physical aspects. Let $x_{n,t,s}^k$ denote the sensor data for unit $n$, sensor $s$ at observation epoch $t$, resulting from failure mode $k$, then we have the vector of sensor measurements $x_{n,t}^k = (x_{n,t,1}^k, \ldots, x_{n,t,S}^k)'$ for $S$ sensors. Suppose the total number of historical units is $N$, and let $T = (T_1, \ldots, T_N)$ represent the overall lifetime cycle durations of these $N$ units. Then, the multi-sensor data $X_n^k$ for unit $n$ under failure mode $k$ can be expressed as

$$
X_n^k = (x_{n,1}^k, x_{n,2}^k, \ldots, x_{n,T_n}^k)',
$$

where

$$
[ x_{n,1}^k \ x_{n,2}^k \ \cdots \ \ x_{n,T_n}^k ]
= [ x_{n,1}^k \ x_{n,2}^k \ \cdots \ x_{n,2}^k ]
= [ x_{n,1}^k \ x_{n,2}^k \ \cdots \ x_{n,S}^k ].
$$

Given the multiple sensors data, $D = \{ X_n^k, n = 1, \ldots, N, k = 1, \ldots, K \}$, the key issue is to identify the potential failure mode based on the condition monitoring signals and further predict the RUL according to the failure mode estimation.

In order to make full use of the mutual information across different tasks, a joint learning approach for classification and regression is proposed. Unlike the traditional multi-task learning, in which two or more tasks are simply combined and optimized in parallel order, the proposed method further incorporates the classification result into the construction of the regression model to improve the prediction accuracy. The overall architecture of the sequential branched network model is presented in Figure 3.

This framework mainly includes four parts, i.e., masking/sliding, shared representation layers, classification layers, and regression layers. Specifically, the first part is to perform necessary data preprocessing on the raw multi-channel monitoring data. In general, the established model should be able to make classifications and predictions for units with varying data durations. In this work, to describe the sequential degradation process and capture the long-term dependency of sensor signals, we adopt the LSTM mechanism in the shared representation layers to deal with the temporal data. However, the input sequence length of the LSTM model is always fixed for the convenience of model training. In the early stages, the target unit may not have sufficient observations, or the unit fails rapidly, with very limited data before failure. Thus, except for the basic preprocessing operation, such as normalization, a masking layer is also required to deal with the sequence batches whose durations are less than the predetermined sequence length. In this layer, the sequence batch is padded to the fixed length with a mask value, which is pre-set to be very different from the values of the features. As a result, the padded batch sequence can be directly discriminated by the learned network model through the masking technique, which informs the sequence-processing layers that certain timesteps in input are missing, and thus should be skipped during processing. Therefore, the masking operation in DBNet is to make the LSTM only act on the sequence of its actual
length, without processing the useless padding part. After this operation, the multi-sensory time series data is processed into $\sum_{n=1}^{N} Q_n$ time windows of a fixed length $L$ with $S$ dimensions, where $Q_n = \max(T_n - L + 1, 1)$.

The second part is the shared representation layers, which are composed of stacked LSTM layers and fully connected (FC) layers with ReLU ($f(x) = \max(0, x)$) as the activation function. As the different temporal patterns captured by LSTM are crucial for task learning, it is well-suited for the classification and prediction tasks on long time series data. The detailed mechanism of LSTM can be depicted in Figure 4.

To be specific, a common LSTM unit is composed of a cell state $c_t$, a forget gate $Γ_f$, an update gate $Γ_u$, a candidate state $\tilde{c}_t$, an output gate $Γ_o$ and finally the output value $h_t$. The cell remembers values over arbitrary time intervals, whereas the gate selects the valuable information to transfer to the LSTM. Specifically, the step-by-step information transition in LSTM can be illustrated as follows. Let $W$ and $b$ respectively denote the weights and biases in the neural network, while the footnote $f, u, c,$ and $o$ represent forget gate, update gate, and candidate cell, and output. The first step for LSTM is to determine which information in the cell state should be abandoned through the forget gate. The gate consists of a Sigmoid neuron layer and a point-wise multiplication operation. The Sigmoid function works on the output of the $f$th layer and the current input $x_t$ to generate a vector. Since the output range for the Sigmoid function (denoted as $σ$) is between 0 and 1, the forget gate regulates the flow of information that should be out of the cell according to the following operation,

$$Γ_f = σ(W_f(h_{t-1}, x_t) + b_f).$$

The next step is to decide what information should be stored in the cell state, which includes the following two parts. First, a Sigmoid layer decides the values to update, and then the tanh layer creates a candidate vector $\tilde{c}_t$, which will be added to the state. The above operations lead to the following formulations:

$$Γ_u = σ(W_u(h_{t-1}, x_t) + b_u),$$

$$\tilde{c}_t = \tanh(W_c(h_{t-1}, x_t) + b_c).$$

Then the new cell state $c_t$ can be updated according to the control gates as follows

$$c_t = Γ_u \odot \tilde{c}_t + Γ_f \odot c_{t-1}. \quad (4)$$

The final output $h_t$ is based on the cell status, but will be filtered by the Sigmoid function to determine which part of the unit state to output in the following Eq. 5 and 6,

$$Γ_o = σ(W_o(h_{t-1}, x_t) + b_o), \quad (5)$$

$$h_t = Γ_o \odot c_t. \quad (6)$$

Through the aforementioned mechanism of cell memory function, the LSTM cell vector has the ability to artificially forget its previously stored memory, as well as to add new information during the information transmission process. In this mechanism, the output of LSTM cell at the previous time point $t - 1$ is combined with the observation at the current time point $x_t$ and passed into the next cell. These sequential cells form the LSTM network to process the temporal data for multiple sensors. Therefore, the LSTM layers are better at extracting temporal features and characterizing the degradation pattern from time series data, whereas the FC layers can effectively compile all the neurons and learn a potentially non-linear function between the input and output features of the FC layers.

Then, unlike the traditional neural networks, which are usually trained to handle different tasks isolatedly, the established DBNet framework starts with the aforementioned shared layers to extract features, after which different tasks branch the number of layers into their own sequence. Specifically, the failure mode diagnostics step is set to be the third part in DBNet, where a classification subnetwork after the shared layers is designed to be a fully connected neural network with ReLU as the activation function for the middle layers, and the Softmax operation for the output of the last FC layer. The Softmax activation function for $h_{n,t}^k$ is given as follows,

$$p_{n,t}^k = \frac{e^{h_{n,t}^k}}{\sum_{j=1}^{K} e^{h_{n,t}^j}}. \quad (7)$$

This equation yields the probability $p_{n,t}^k$ that the instance $n$ at time $t$ belongs to failure mode $k$. For the RUL prediction task,
since different failure modes may lead to distinct degradation patterns, the prerequisite of precise RUL prediction is to group the data into corresponding categories, and then model the degradation processes respectively. The shared representation layers before the classification layers are then branched into multiple regression subnetworks based on different failure modes. Each subnetwork is composed of multiple fully connected non-linear layers to increase the network representation capability. The ReLU is employed as the activation function for middle layers, while between the last hidden layer and the output layer, the identity activation function is used to produce a specific RUL estimation \( \hat{q}_{k, n,t} \) under mode \( k \). Therefore, the established DBNet model constructed \( K \) regression models in total to achieve more precise RUL prediction results for each mode.

Finally, the RUL estimation generated by each regression subnetwork for different failure modes \( q_{k, n,t} \) is multiplied with the probabilistic output of the classification subnetwork \( p_{n,t} \), and combined to boost the prediction accuracy. Similar to the law of total expectation, the final output is computed as

\[
\hat{\tau}_{n,t} = \sum_{k=1}^{K} p_{n,t} q_{k, n,t}.
\] (8)

There are four advantages by using the above-mentioned architecture in the proposed DBNet: first, the branched subnetworks for RUL prediction become specialized for each failure mode, which is expected to provide a more accurate prediction with simple subnetworks; second, it can effectively achieve implicit data augmentation and knowledge transfer not only between diagnostics and RUL prediction, but also between RUL prediction tasks under different failure modes; third, due to the layer sharing of related tasks, the resulting memory cost is typically much lower [31], [32]; finally, since the features in the shared layers do not need to be calculated repeatedly for the different tasks (classification and RUL prediction for each mode), it can save computing power and improve the training efficiency [31].

B. Joint Optimization of the DBNet

With the DBNet developed for failure mode diagnostics and RUL prediction, the key task remaining is to train the model through joint learning or optimization. Let \( \theta = [\theta_c, \theta_r] \) denote the model parameters, where \( \theta_c \) represent all the parameters for the shared representation layers and classification layers, and \( \theta_r \) represent all the parameters for regression layers. As for the RUL prediction task, the root mean square error (RMSE) between the ground truth of RUL \( \tau_{n,t} \) and estimation \( \hat{\tau}_{n,t} \) is employed to define the loss of the prediction task, which is given by

\[
L_R(X_{n,t}^k, \theta) = \sqrt{\frac{1}{T_n} \sum_{t=1}^{T_n} (\hat{\tau}_{n,t} - \tau_{n,t})^2}.
\] (9)

Since the failure mode diagnostics task is fundamentally a multi-class classification problem, the objective of this task is set to minimize the cross-entropy between the ground truth class labels \( y_{n,t}^k \) and the established failure mode distribution \( p_{n,t} \)

\[
L_C(X_{n,t}^k, \theta) = -\frac{1}{T_n} \sum_{t=1}^{T_n} \sum_{k=1}^{K} y_{n,t}^k \ln p_{n,t}^k.
\] (10)

Then, the proposed DBNet is then trained by maximizing a combined loss function of these two interrelated tasks. The overall objective is to minimize

\[
L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \left[ L_R(X_{n,t}^k, \theta) + \lambda L_C(X_{n,t}^k, \theta) \right]
\] (11)

where \( \lambda \) is the tuning parameter to balance the two losses. This objective function \( L(\theta) \) can be optimized through the gradient descent (GD) with back-propagation algorithms. Considering the computational efficiency, stochastic gradient descent (SGD) is applied to minimize the loss function in this work. During each iteration, the SGD algorithm only extracts a random subset of training samples to compute the gradient, which can considerably reduce computational consumption. The adaptive moment estimation algorithm (Adam) [33] is an SGD variant that has lately gained popularity in deep learning applications for model training. It is an algorithm for first-order gradient-based optimization of objective functions, based on the adaptive estimates of lower-order moments. The customized Adam algorithm is applied to minimize the loss function in our work, since it has many advantages over the

![Diagram of an LSTM sequence.](image)
typical SGD approaches, (for example, easy to implement, high computational efficiency, and less memory requirements). The key step in Adam is to compute the following partial derivatives of Eq. 11 at each iteration:

$$
\left[ \frac{\partial L(X^k_n, \theta)}{\partial \theta_r}, \frac{\partial L(X^k_n, \theta)}{\partial \theta_c} \right],
$$

(12)

where

$$
\frac{\partial L(X^k_n, \theta)}{\partial \theta_r} = \sum_{t=1}^{T_n} \sum_{k=1}^{K} \left[ \left( \hat{r}_{n,t} - \tau_{n,t} \right) p_{n,t}^k g_{n,t}^k \left( (1 - p_{n,t}^k) \frac{\partial h_{n,t}^k}{\partial \theta_r} - \sum_{j \neq k} p_{n,t}^j \frac{\partial h_{n,t}^j}{\partial \theta_r} \right) \right] \frac{1}{\sqrt{\frac{1}{T_n} \sum_{t=1}^{T_n} (\hat{r}_{n,t} - \tau_{n,t})^2}} - \lambda g_{n,t}^k \left( \frac{\partial h_{n,t}^k}{\partial \theta_c} - \sum_{j=1}^{K} p_{n,t}^j \frac{\partial h_{n,t}^j}{\partial \theta_c} \right),
$$

(13)

$$
\frac{\partial L(X^k_n, \theta)}{\partial \theta_c} = \sum_{t=1}^{T_n} \sum_{k=1}^{K} \left[ \left( \hat{r}_{n,t} - \tau_{n,t} \right) p_{n,t}^k g_{n,t}^k \left( (1 - p_{n,t}^k) \frac{\partial h_{n,t}^k}{\partial \theta_c} - \sum_{j \neq k} p_{n,t}^j \frac{\partial h_{n,t}^j}{\partial \theta_c} \right) \right] \frac{1}{\sqrt{\frac{1}{T_n} \sum_{t=1}^{T_n} (\hat{r}_{n,t} - \tau_{n,t})^2}} - \lambda g_{n,t}^k \left( \frac{\partial h_{n,t}^k}{\partial \theta_r} - \sum_{j=1}^{K} p_{n,t}^j \frac{\partial h_{n,t}^j}{\partial \theta_r} \right)
$$

(14)

Algorithm 1: Joint Learning of the DBNet Model Using Tailored Adam.

Parameters default settings: $\eta = 0.001$, $\gamma_1 = 0.9$, $\gamma_2 = 0.999$ and $\epsilon = 10^{-8}$.

Input: training samples $D$, default Adam algorithm parameters: step size, $\eta$; time step, $i \leftarrow 0$; exponential decay rates for the two moment estimates, $\gamma_1, \gamma_2$; the 1st and 2nd moment vector, $m_0 \leftarrow 0$, $v_0 \leftarrow 0$; the initial value for $\theta$ to be optimized, $\theta_0$.

Output: optimized model parameters $\theta_i$.

while $\theta_i$ not converged do

    $i \leftarrow i + 1$

    Calculate the gradient or sub-gradient according to Eq. 13 and Eq. 14

    $m_i \leftarrow \gamma_1 \cdot m_{i-1} + (1 - \gamma_1) \cdot g_i$ (update biased 1st moment estimate)

    $v_i \leftarrow \gamma_2 \cdot v_{i-1} + (1 - \gamma_2) \cdot g_i^2$ (update biased 2nd moment estimate)

    $\hat{m}_i \leftarrow \frac{m_i}{1 - \gamma_1^i}$ (compute bias-corrected 1st moment estimate)

    $\hat{v}_i \leftarrow \frac{v_i}{1 - \gamma_2^i}$ (compute bias-corrected 2nd moment estimate)

    $\theta_i \leftarrow \theta_{i-1} - \eta \cdot \left( \frac{\hat{m}_i}{\sqrt{\hat{v}_i} + \epsilon} \right)$ (update the model parameters)

end

Return $\theta_i$.

From Eq. 13 and Eq. 14, we can find that the training of the two tasks is optimized jointly. Unlike the traditional multi-task learning, where the loss function simply adds the losses of each task, DBNet associates the classification output $p_k$ with the RUL prediction loss. As a result, the capability of DBNet is further enhanced through the optimization of the joint loss function. With the established partial derivatives of Eq. 11, the tailored Adam algorithm for the joint model learning is provided in Algorithm 1, in which $g_i^k$ represents the element-wise square.

IV. CASE STUDY

In this section, the proposed DBNet framework is applied to a new turbine engine dataset generated from a high-fidelity simulation software. Extensive experiments are carried out on this dataset to evaluate and compare the DBNet with other state-of-the-art approaches.

A. Turbofan Engine Degradation Dataset with Multiple Failure Modes

The experimental dataset is generated by the TEACHES simulation software, which was developed by the laboratory of thermal turbo-machines [34]. Figure 5 shows a schematic diagram of a commercial aircraft gas turbine engine. In this software, multiple machine failures can be implanted into different components of the engine, and their impact can be studied. The virtual working performance is continuously collected by multiple sensors embedded in different components, which are simulated according to the characteristic maps of the condition and the real-time indicator of health status.

Fig. 5. Simplified engine diagram simulated in the software.

This program receives different operating condition parameters as optional inputs, including flight altitude (0-15000), Mach number (0-0.90), and throttle resolver angle (20-100), ambient temperature difference from ISA (-20 to 20), ram pressure recovery (0.99-1), fuel flow (0.55-1.5), low pressure shaft speed (0-4750), high pressure shaft speed (7300-13280), high pressure turbine inlet temperature (900-1400), engine pressure ratio (1.2-1.8), thrust (15000-230000), and accordingly simulates the working performance of the engine.
This enables the users to examine the influence of different operation conditions on engine performance by freely setting the corresponding parameters. Besides, the simulation software also allows users to input indicative parameters of health condition to control the degradation modes, including fan outer flow/efficiency drop, fan inner flow/efficiency drop, HP compressor flow/efficiency drop, HP turbine flow/efficiency drop, LP turbine flow/efficiency drop, nozzle area change. By adjusting these parameters, the users can purposely generate faults and simulate the working state of the engine with specific symptoms. Thus, this software provides the opportunity to examine the algorithm for identifying potential failure modes, that leads to machine deterioration. As shown in Table I, in response to 10 operation conditions and 11 health status indicator parameters in the inputs, there are in total 16 variables in the outputs.

### Table I

**Description of the Outputs from the Simulated Dataset**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>LP shaft speed</td>
<td>rpm</td>
</tr>
<tr>
<td>NH</td>
<td>HP shaft speed</td>
<td>rpm</td>
</tr>
<tr>
<td>P13</td>
<td>Fan outlet pressure</td>
<td>bar</td>
</tr>
<tr>
<td>P26</td>
<td>HP compressor inlet pressure</td>
<td>bar</td>
</tr>
<tr>
<td>T26</td>
<td>HP compressor inlet temperature</td>
<td>C</td>
</tr>
<tr>
<td>P3</td>
<td>HP compressor outlet pressure</td>
<td>bar</td>
</tr>
<tr>
<td>T3</td>
<td>HP compressor outlet temperature</td>
<td>C</td>
</tr>
<tr>
<td>P42</td>
<td>HP turbine outlet pressure</td>
<td>bar</td>
</tr>
<tr>
<td>T42</td>
<td>HP turbine outlet temperature</td>
<td>C</td>
</tr>
<tr>
<td>P5</td>
<td>LP turbine outlet pressure</td>
<td>bar</td>
</tr>
<tr>
<td>T41</td>
<td>HP turbine inlet temperature</td>
<td>C</td>
</tr>
<tr>
<td>Thrust</td>
<td>Thrust</td>
<td>Nt</td>
</tr>
<tr>
<td>Wf</td>
<td>Fuel flow rate</td>
<td>Kg/s</td>
</tr>
</tbody>
</table>

Since the main purpose of this work is a joint optimization for failure mode diagnostics and prognostics tasks, a new dataset involving three randomly selected failure modes (e.g., fan outer flow drop, fan inner efficiency drop, and HP turbine flow drop) under the same operation condition (e.g., Ram Pressure Recovery = 0.99, Fuel Flow = 1.239) is simulated to model the degradation processes. The health index of each degradation mode is described by a linear model with the following polynomial form,

\[
f_{n,t}^k = \sum_{m=0}^{M} \beta_{k,m} t^m, \tag{15}\]

where \(f_{n,t}^k\) represents the health index for unit \(n\) at time \(t\) under the failure mode \(k\), \(M\) is the order of the polynomial, \(\beta_k = (\beta_{k,0}, \beta_{k,1}, \ldots, \beta_{k,M})^T\) is the corresponding regression parameters of dimension \(M + 1\) for mode \(k\). To further demonstrate the effectiveness of transfer learning between different modes, here we intentionally generate an imbalanced dataset, in which the sample size for some failure modes is significantly larger than for others. Table II lists the detailed regression parameters, the number of units \(N_k\) (the training and testing dataset both contain \(N_k\) units), as well as the total number of samples (time points of measurement) for training and testing with respect to the three failure modes. The total number of time points for both training and testing datasets is shown in the last column.

### Table II

**The Parameters of the Three Health Indices and Dataset Sizes.**

<table>
<thead>
<tr>
<th>Mode</th>
<th>(\beta_k)</th>
<th>(N_k)</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0.02)</td>
<td>(0.01)</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>(0.02)</td>
<td>(0.01)</td>
<td>70</td>
</tr>
<tr>
<td>3</td>
<td>(0.01)</td>
<td>(0.003)</td>
<td>50</td>
</tr>
</tbody>
</table>

The health indices corresponding to the three potential failure modes are shown in Figure 6. The units of mode 1 and 2 are set to degrade in a linear manner (\(M = 1\)), while for failure mode 3, the unit degrades quadratically (\(M = 2\)). Based on the input health indices, the software simulates the operating process of each unit accordingly. Figure 7 displays the generated multi-channel measurements for six randomly selected sensors. In this dataset, there are in total 150 units for training and 150 units for testing. In the training dataset, each unit is run until it fails, while in the testing dataset, the unit is terminated at a random time before the real failure time. This dataset is now available at https://github.com/kernellZ/Turbofan-Engine-Degradation-Dataset-with-Multiple-Failure-Modes.git.

![Fig. 6. Health indices of the 150 training units under 3 failure modes.](image)

**B. Performance Evaluation and Comparison**

To demonstrate the superiority of the DBNet with sequential multi-task learning, we compare it thoroughly with other representative approaches for failure mode identification and RUL prediction, including support vector regression (SVR) [35], LSTM neural network [36], two-step neural network, and MTL-based neural network. SVR is a regression-oriented variation of the support vector machine technique. The LSTM neural network is used as a baseline to show the performance of the LSTM-based methods for RUL prediction. For the MTL method, an LSTM neural network is delicately designed.
Measurements reduce the classification error. Besides, compared with the increasing the penalty weight of classification could effectively display in Figure 8. From this figure, we can find that λ parameter as the training parameter. The influence of the weighting model parameter with the smallest fitting loss is selected over the randomly generated initial weights and biases, and the performance of each neural model structure is repeated 100 times using L learning rate adopted in this paper is 0.01. The sequence length is set to 50. To reduce the randomness, the optimization is considered, imposing the classification task on the training objective also reduces the prediction errors significantly. This phenomenon indicates that introducing the classification task into the prediction task can improve the generalization capability, thereby decreasing the overfitting risk. Furthermore, accurate classification results also contribute to the construction of specialized regression submodels dealing with mixed data under multiple failure modes. Therefore, integrating the classification task with the prediction task can effectively improve the prediction performance. When λ = 0.12, the developed DBNet achieves the best prediction accuracy on the RUL prediction task. Then the prediction error increases along with the increment of the weighting parameter λ for classification. This phenomenon indicates that when λ further increases to infinity, DBNet will focus more on the classification task instead of the RUL prediction, resulting in an increase in prediction error. To strike a balance between proper classification and prediction performance, we select λ = 0.12 as the weighting parameter based on the analysis of Figure 8. Note that this parameter selection method is similar to [30].

The specific model settings and the corresponding model performances on RUL prediction are presented in Table III. From this table, we can find that the proposed method outperforms all others. For the failure mode classification and RUL prediction tasks, the MLP method has higher accuracy than SVM/SVR methods, demonstrating the advantages of utilizing the neural network approach over the traditional statistical methods dealing with multiple dimensional data. Moreover, when compared to the MLP model, the LSTM approach obtains a 6% accuracy improvement on prediction task, which indicates its excellent performance in sequential data processing. With the adoption of the LSTM neural network structure, the two-step strategy provides more accurate or comparable RUL prediction results compared with the LSTM regression.
model for the units of mode 2 and mode 3. However, the performance of this method for mode 1 is inferior to most methods. The reason behind this phenomenon can be explained from two aspects. Firstly, since the subsequent regression model is chosen incorrectly, a wrong classification result will inevitably lead to a higher prediction error. Secondly, only a few samples are available (30 units vs. 50 and 70 under mode 2 and 3) to train an independent regression model for mode 1, which may result in an overfitting issue. For the MTL method, the classification and prediction accuracy increase markedly, which illustrates that combining two tasks can enhance the generalization ability and consequently improves the model performance. Compared with MTL model, the F1 score of DBNet is improved by 0.06, and the average precision of RUL prediction is improved by almost 4%. This demonstrates that the established DBNet model can effectively promote the classification and prediction performances, which is achieved through the knowledge transferring between two tasks. Besides, based on the accurate failure mode identification results, the specialized regression models can be selected correctly to better characterize the piecewise degradation function, thus leading to a more accurate RUL prediction. In addition, integrating all of the RUL regression submodels in one composite model for joint learning enables the knowledge to be transferred between different modes, thereby realizing implicit data argumentation for the failure mode with insufficient data.

In order to illustrate the prediction performance of the proposed method over time, Figure 9 displays the RUL prediction curve with the $\alpha$-$\lambda$ metric on four randomly selected units (#2, 8, 59, 84). The $\alpha$-$\lambda$ metric has been widely adopted for evaluating the accuracy of RUL predictions performance [37], where $\alpha$ specifies the error bound for RUL prediction, i.e., $[(1-\alpha)\tau_n \leq \hat{\tau}_n \leq (1+\alpha)\tau_n]$, $\lambda$ represents the percentage of predicted time and actual failure time. Here, we set $\alpha = 15\%$. From this figure, we can find that the actual RUL prediction curves are mostly within 15% error and converge as $\lambda$ decreases. Due to the accumulated data, the RUL prediction gets better as $\lambda$ gets smaller [17]. We further show the prediction error at different levels of actual RUL in Figure 10. The bars denote the evaluation metric, which is based on the mean of the absolute percentage error. The $x$-axis represents different levels of RUL. For example, “ALL” means all the time series for 150 testing units, while other numbers represent testing units with

![Fig. 8. The classification and prediction errors of DBNet with different weighting parameter $\lambda$.](image)

**TABLE III**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Hyper-parameters</th>
<th>Failure mode classification</th>
<th>RUL prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy Precision Recall F1 score</td>
<td>Mode 1 Mode 2 Mode 3 Average</td>
</tr>
<tr>
<td>SVM/SVR model</td>
<td>-</td>
<td>0.73 0.72 0.76 0.74</td>
<td>20.12 15.23 16.31 16.13</td>
</tr>
<tr>
<td>MLP model</td>
<td>50-20-5-3</td>
<td>0.79 0.82 0.73 0.77</td>
<td>11.80 12.61 20.19 15.20</td>
</tr>
<tr>
<td>LSTM model</td>
<td>50-30-15-3</td>
<td>0.81 0.87 0.74 0.80</td>
<td>9.11 7.89 11.08 9.14</td>
</tr>
<tr>
<td>Two-step model</td>
<td>50-30-15-3</td>
<td>0.81 0.87 0.74 0.80</td>
<td>18.76 7.20 11.90 10.09</td>
</tr>
<tr>
<td>MTL model</td>
<td>30-15-3</td>
<td>0.12 0.92 0.92 0.93</td>
<td>3.71 2.93 4.47 3.56</td>
</tr>
<tr>
<td>DBNet model</td>
<td>30-10-3</td>
<td>0.08 0.85 0.92 0.83 0.87</td>
<td>8.24 7.81 6.59 7.42</td>
</tr>
</tbody>
</table>

![Fig. 9. $\alpha$-$\lambda$ performance metric for four randomly selected units.](image)
V. CONCLUSIONS

This paper proposes a joint learning approach for failure mode diagnostics and RUL prediction. The LSTM network is firstly applied to extract features from the multi-channel degradation data. Then, the failure mode classification and the RUL prediction tasks are jointly learned through a shared representation and a combined loss function, which improves the generalization and individual model performances via transfer learning. Unlike the traditional multi-task learning approaches, which simply combine the two tasks in parallel, the developed DBNet organizes the failure mode diagnostics and RUL prediction tasks sequentially, i.e., the classification result is further used as an input in the RUL prediction model to boost the accuracy of estimation results. Besides, the branched subnetworks predict the degradation process with respect to each failure mode, resulting in more accurate prediction results. This mechanism also enables the knowledge transfer between the RUL prediction tasks across different failure modes, achieving implicit data augmentation for the failure modes with insufficient data. The experimental results indicate that the proposed model not only provides accurate mode classification and RUL prediction results, but also further enhances the ability on dealing with imbalanced data.

There are still several problems worthy of further investigation. First, the evolution of the degradation process may exhibit a multi-phase nature, which is difficult to describe with a single regression model: thus, to further improve the prediction accuracy, the proposed joint learning approach can be extended to include the assessment of the degradation stage. Secondly, during the service of machine in real application, new failure modes may occur and cannot be recognized correctly. The proposed model may be further improved through unsupervised learning approaches. Finally, one general case in practice is that the degradation can result from multiple competing failure modes: this brings difficulties in the failure modes identification and the specialized modeling of the degradation processes regarding the potential modes; therefore, extensions to competing multiple failure modes are also desirable for practical application.

REFERENCES


Zhen Li (S’18) received the Ph.D. degree in data science from the City University of Hong Kong, Hong Kong, in 2019. He is currently an Assistant Professor with the Department of Industrial Engineering and Management and the Chinese Institute for Quality Research, Shanghai Jiao Tong University, Shanghai, China. His research interests comprise both applied and theoretical aspects of data science integrated with domain knowledge, including computer experiment, quality, reliability, statistical learning.

Yongxiang Li received the Ph.D. degree in data science from the Georgia Institute of Technology, Atlanta, USA, in 2016 and 2018. Currently, he is an assistant professor at the Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, USA. His research interests are focused on engineering-driven data analytics for advanced manufacturing. Dr. Yue is a member of IEEE, ASME, INFORMS and IIESE.

Enrico Zio received the Ph.D. degree in nuclear engineering from Politecnico di Milano, Milan, in 1996, and the Ph.D. degree in probabilistic risk assessment from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1998. He is currently a Full Professor with the Centre for Research on Risk and Crises, Ecole de Mines, ParisTech, PSL University, Paris, France, a Full Professor and the President of the Alumni Association, Politecnico di Milano. He has authored or coauthored 7 books and more than 500 papers in international journals. His research interests include modeling of the failure-repair-maintenance behavior of components, and complex systems for the analysis of their reliability, maintainability, prognostics, safety, vulnerability, resilience, and security characteristics, and the development and use of Monte Carlo simulation methods, artificial techniques, and optimization heuristics.

Xiaowei Yue received the Ph.D. degree in Industrial Engineering with minor Machine Learning from the Georgia Institute of Technology, Atlanta, USA, in 2016 and 2018. Currently, he is an assistant professor at the Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, USA. His research interests are focused on engineering-driven data analytics for advanced manufacturing. Dr. Yue is a member of IEEE, ASME, INFORMS and IIESE.

Jianguo Wu (M’18) received the Ph.D. degree in Industrial and Systems Engineering in 2015, from University of Wisconsin-Madison. Currently, he is an Assistant Professor in the Dept. of Industrial Engineering and Management at Peking University, Beijing, China. His research interests are mainly in quality control and reliability engineering of intelligent manufacturing and complex systems through engineering-informed machine learning and advanced data analytics. He is an Associate Editor of the Journal of Intelligent Manufacturing, and a member of IEEE, INFORMS, ISE, and SME.