axioms

MDPI

Article

Personalized Transfer Learning Framework for Remaining Useful Life Prediction Using Adaptive Deconstruction and Dynamic Weight Informer

Xue Liu [1], Jian Ma [1,2,3] and Dengwei Song [4,*]

[1] School of Reliability and System Engineering, Beihang University, Beijing 100191, China; rseliux@buaa.edu.cn (X.L.); 09977@buaa.edu.cn (J.M.)

[2] Institute of Reliability Engineering, Beihang University, Beijing 100191, China

[3] Science & Technology Laboratory on Reliability & Environmental Engineering, Beihang University, Beijing 100191, China

[4] School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China

* Correspondence: songdengwei@buaa.edu.cn

Abstract: The precise remaining useful life (RUL) prediction of turbofan engines benefits maintenance decisions. The training data quantity and quality are crucial for effective prediction modeling and accuracy improvement. However, the performance degradation process of the same type of turbofan engine usually exhibits different trajectories because of engines' differences in degradation degrees, degradation rates, and initial health states. In addition, the initial part of the trajectory is a stationary health stage, which contains very little information on degradation and is not helpful for modeling. Considering the differential degradation characteristics and the requirement for accurate prediction modeling of the same type of turbofan engines with individual differences, we specifically propose a personalized transfer learning framework for RUL prediction by answering three key questions: when, what, and how to transfer in prediction modeling. The framework tries to maximumly utilize multi-source-domain data (samples of the same type of engines that run to failure) to improve the training data quantity and quality. Firstly, a transfer time identification method based on a dual-baseline performance assessment and the Wasserstein distance is designed to eliminate the worthless part of a trajectory for transfer and prediction modeling. Then, the transferability of each sample in the multi-source domain is measured by an approach, named the time-lag ensemble distance measurement, and then the source domain is ranked and adaptively deconstructed into two parts according to transferability. Ultimately, a new training loss function considering the transferability of the weighted multi-source-domain data and a two-stage transfer learning scheme is introduced into an informer-based RUL prediction model, which has a great advantage for long-time-series prediction. The simulation data of 100 of the same type of turbofan engine with individual differences and five comparison experiments validate the effectiveness and accuracy of the proposed method.

Keywords: turbofan engine; remaining useful life prediction; personalized transfer learning; multi-source-domain adaptive deconstruction

MSC: 68T09

check for updates

Citation: Liu, X.; Ma, J.; Song, D. Personalized Transfer Learning Framework for Remaining Useful Life Prediction Using Adaptive Deconstruction and Dynamic Weight Informer. *Axioms* **2023**, *12*, 963. https://doi.org/10.3390/axioms12100963

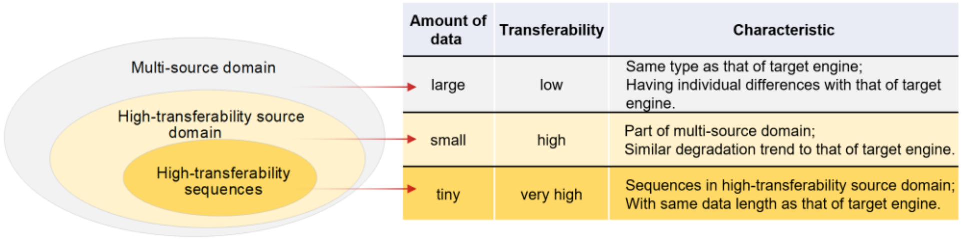
Academic Editors: Nhon Nguyen-Thanh, Lechang Yang, Qingqing Zhai, Rui Peng and Aibo Zhang

Received: 4 September 2023
Revised: 6 October 2023
Accepted: 10 October 2023
Published: 12 October 2023

1. Introduction

As a complex system, an aircraft engine has an exceedingly intricate performance degradation process [1]. The safe and precise maintenance of turbofan engines relies on the accurate prediction of the remaining useful life (RUL) [2], which hinges upon identifying a model capable of discerning the degradation patterns of the target engine. Methods for RUL prediction include physics-based, data-driven, and hybrid approaches [3,4]. However, in most cases, physics-based methods are not applicable because of modeling errors and the inability to comprehensively model. Data-driven methods construct relationships between sensor monitoring data and RUL, which do not require physical-model failure parameters. Data-driven methods have been recently playing an important role in the fault diagnosis and prognosis of industrial systems [5]. The relevant research corroborated the exceptional predictive capabilities of machine learning in forecasting faults within intricate systems [6]. In this context, deep learning methods exhibited significant potential within the realm of RUL prediction [7]. Recurrent Neural Networks, including Long Short-Term Memory (LSTM) and its derivative models, demonstrated exceptional efficacy in the domain of RUL prediction [8,9].

However, one of the difficulties of data-driven methods is that insufficient historical data result in incomplete degradation information. Transfer learning has emerged as a potential solution for addressing the issue of insufficient data in the target domain by leveraging existing knowledge and the data from the source domain. Transfer learning is currently being widely utilized in various fields, such as image and sound classification [10], biomedicine [11], and intelligent diagnosis [12,13]. In addition, some researchers introduced transfer learning to RUL prediction. For instance, it was utilized to predict the RUL of various types of batteries [14–16]. In other studies, data regarding different operating conditions or different individuals were utilized in conjunction with transfer learning to predict the RULs of target bearings [17–25]. Transfer learning was also applied to engine RUL prediction [26–28]. Transfer prediction relies on leveraging degradation information from the source domain to predict the RUL of engines in the target domain. However, variations in individual samples pose challenges to achieving a precise knowledge transfer. The idiosyncratic disparities in engine degradation result in the diverse transferability of data from multi-source-domain engines to the target engine. The data classification is illustrated in Figure 1. Only a part of the multi-source domain has a degradation trend similar to that of the target engine and can be used as a high-transferability source domain. In addition, the transferability of sequences varies, even in the high-transferability source domain. Existing research methods usually transfer the entire source domain as a unified entity. Most researchers focused on how to transfer. Only a few of them focused on what to transfer, and even fewer focused on when to transfer. However, equipment degradation processes are characterized by individual differences. If data are directly transferred without proper selection, the transfer prediction results may be affected. Research gaps exist in three aspects:


Multi-source domain
High-transferability source domain
High-transferability sequences

Amount of data	Transferability	Characteristic
large	low	Same type as that of target engine; Having individual differences with that of target engine.
small	high	Part of multi-source domain; Similar degradation trend to that of target engine.
tiny	very high	Sequences in high-transferability source domain; With same data length as that of target engine.

Figure 1. Relationship among source domain data with different transferability.

(1) Although data in the health stage are not necessary for prediction modeling, the onset times of the engine degradation stage vary widely and can be difficult to identify. However, most existing methods do not consider changes in the engine operating stage and start transfer prediction in the health stage, thus resulting in the insufficient extraction of degradation information.

(2) Engine data exhibit variations in degradation degrees, degradation rates, and initial health states, thereby forming a multi-source domain. However, existing methods either directly transfer all full-life engine data or simply filter source domain data, disregarding the transferability differences and making it challenging to extract sufficient information from the source domain.

(3) Significant differences between target engines lead to the insufficient adaptability of the general transfer prediction model. The transferability of multi-source domains is different among different target engines. However, most methods do not consider the individual differences among the target engines and build general transfer prediction models, resulting in insufficient fitting for the degradation processes of target engines.

Due to individual differences in the health and degeneration stages, the three aforementioned weaknesses in the existing transfer prediction methods can be generalized as three problems: when to transfer, what to transfer, and how to transfer. Therefore, a personalized transfer prediction framework was proposed to solve these problems in engine RUL transfer prediction. A dual-baseline assessment based on the Wasserstein distance (DBA-WD) algorithm was developed to assess engine performance from two terms of engine health and failure, and the degradation rate was calculated based on the performance index to determine when to transfer. To address the issue of what to transfer, we proposed a multi-source-domain deconstruction method based on the time-lag ensemble distance measurement (TL-EDM). This method measures transferability from three perspectives: degradation degree, degradation rate, and initial health degree. By quantifying the measured distance, we can identify high-transferability source domains and assign transferability labels to sequences within low-transferability-density multi-source domains. This approach aims to enhance the efficiency of data utilization and improve overall performance. Regarding the approach of how to transfer, we devised a two-stage transfer prediction scheme to extract and integrate both general and individual degradation information from the engines using source domain data with varying levels of transferability. Within this scheme, we designed a stacked informer model specifically tailored for capturing long-term and slowly changing degradation features in the engines. Building upon this model, we introduced transferability information, as measured by TL-EDM, into the model training process through a dynamic weight mechanism that we developed. The primary contributions of this study are as follows:

(1) We proposed a personalized transfer learning framework for RUL prediction of the same type of turbofan engines with individual differences in performance degradation trajectories caused by degradation degrees, degradation rates, and initial health states. In contrast to most of the common transfer learning methods for prediction, which indiscriminately use the whole multi-source-domain data and the entire trajectories, it was specifically and step-by-step designed by answering three key questions: when, what, and how to transfer in prediction modeling of engines. In this manner, multi-source-domain data were as maximally utilized as possible according to the differential degradation process and their characteristics to improve the training data quantity and quality of the prediction models.

(2) A transfer-time identification method based on dual-baseline performance assessment and the Wasserstein distance was designed to eliminate the worthless part of a trajectory for transfer and prediction modeling. The DBA-WD method combines two representations of engine health and failure based on linear weight fusion, thereby preventing endpoint errors caused by reverse distribution. Utilizing the fused performance index, we identified the transfer timing of engine degradation by calculating its slope.

(3) An adaptive deconstruction method was proposed to solve the issue of transferability differences in the multi-source domain caused by the individual differences among engines. The transferability of each sample in the multi-source domain was measured by the TL-EDM approach, and then the source domain was ranked and adaptively deconstructed into two parts according to transferability. This approach measured transferability by considering three aspects, degradation degrees, degradation rates, and initial health degrees, and was especially suitable for the engine characteristic of slowly evolving degradation processes.

(4) We designed a new training loss function considering the transferability and a two-stage transfer learning scheme and introduced them into the informer-based RUL

prediction model, which has a great advantage in long-time-series prediction. The dynamic training weight loss function can incorporate the transferability information measured by TL-EDM into the prediction process. The two-stage transfer learning scheme facilitates efficient extraction and integration of the general and individual degradation information of engines. We validated the proposed scheme with the C-MAPSS dataset. The results demonstrated that our method outperforms existing deep-learning techniques in RUL prediction.

2. Methods

2.1. Problem Statement and Proposed Framework

We use $x^k_{i,j} \in R, i = 1, \ldots, N;\ j = 1, \ldots, P$ to denote the j-th parameter value of the k-th engine at the i-th cycle. N is the number of life cycles, which varies among different engines, and P is the number of sensor-monitoring parameters. $X^k \in R^{N\times P}$ denotes all the monitoring values of the k-th source-domain engine. $\mathcal{X} = \left\{X^1, X^2, \ldots, X^k, \ldots\right\}$ denotes the source domain dataset.

Figure 2 illustrates the proposed framework. It is divided into transfer-timing identification, multi-source-domain adaptive deconstruction, and two-stage transfer prediction. This study seeks to answer three primary transfer prediction questions: when, what, and how to transfer. The long-term monitoring data of the target engine are denoted as $X \in R^{(N-RUL)\times P}$. We postulate that in an industrial context, engine failure results in the cessation of operation and is, consequently, regarded as reaching its operational end. N and RUL denote the number of life cycles and remaining cycles of the target engine, respectively. The proposed framework enables the utilization of the source domain, $\mathcal{X}$, to construct a personalized RUL prediction model, M_{target}, for the target engine. And the fitting goal of M_{target} is to predict the most likely RUL given the previous observations.

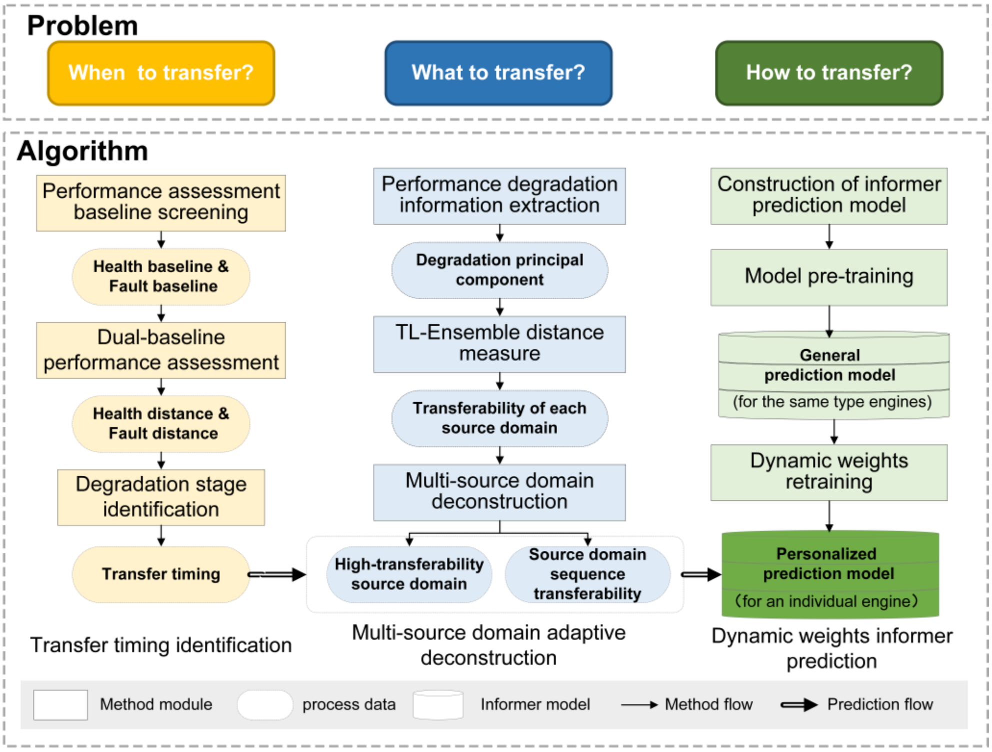

Problem
When to transfer?
What to transfer?
How to transfer?
Algorithm
Performance assessment baseline screening
Health baseline & Fault baseline
Dual-baseline performance assessment
Health distance & Fault distance
Degradation stage identification
Transfer timing
Transfer timing identification
Performance degradation information extraction
Degradation principal component
TL-Ensemble distance measure
Transferability of each source domain
Multi-source domain deconstruction
High-transferability source domain
Source domain sequence transferability
Multi-source domain adaptive deconstruction
Construction of informer prediction model
Model pre-training
General prediction model
(for the same type engines)
Dynamic weights retraining
Personalized prediction model
(for an individual engine)
Dynamic weights informer prediction
Method module
process data
Informer model
Method flow
Prediction flow

Figure 2. Overall process of the proposed framework.

2.2. Transfer-Timing Identification Based on DBA-WD

In practical application, engine performance degradation processes are highly complex, making it challenging to establish a definitive demarcation between the health and degradation stages. Nonetheless, extensively including health-stage monitoring parameters tends to dilute the degraded features within the source domain dataset, $\mathcal{X}$. Therefore, this study presents a method for identifying engine performance degradation stages, thereby facilitating the determination of an optimal transfer timing.

2.2.1. Performance Assessment Baseline Screening

Different monitoring parameters have different responses to engine performance degradation. Some parameters exhibit trends of gradual increase with engine operation, whereas others demonstrate opposite trends. To construct an HI that quantitatively characterizes engine performance, the trends of the monitoring parameters are normalized as follows:

$$x^{k'}{}_{i,j} = \begin{cases} x^{k}{}_{i,j}, \text{ if } x_{1,j} \leq x_{N,j} \\ 1 - x^{k}{}_{i,j}, \text{ if } x_{1,j} > x_{N,j} \end{cases} \tag{1}$$

where $x_{i,j}$ is the value of the j-th monitoring parameter in the i-th cycle after pre-processing, and $x'_{i,j}$ is the corresponding value of trend uniformization.

To analyze health or fault deviation and assess engine performance, the health indices, g_h^k, and fault indices, g_f^k of X^k, are designed as follows:

$$g_h^k = \frac{1}{P}\sum_{j=1}^{P} x^{k}{}_{1,j} \tag{2}$$

$$g_f^k = \frac{1}{P}\sum_{j=1}^{P} x^{k'}{}_{N,j} \tag{3}$$

In the source domain dataset, $\mathcal{X}$, g_h^k and g_f^k are arranged in ascending and descending order, respectively. The top M engines are the health and fault baseline engines, respectively. Their first and last Win_d cycles form the health baseline sets, $G_h \in \mathbb{R}^{win_d \times P \times M}$, and fault baseline sets, $G_f \in \mathbb{R}^{win_d \times P \times M}$.

2.2.2. Dual-Baseline Performance Assessment Based on Wasserstein Distance

Because of engine individual differences, the health and fault degrees at the beginning and the end of life have specific differences. The overlap between the engine parameters and the baselines increases as the engine is closer to the health or fault status. Even a reversed distribution may be observed. To address this problem, the DBA-WD algorithm was developed. The distances between the target engine and baselines are measured based on the WD because of its advantage in distribution measure.

Because of the random fluctuation, the monitoring parameter values in several cycles can better characterize the performance state. Based on the advantages of measuring the distribution distance, the WD algorithm is selected to measure the distance between the target engine and the baseline set. The engine performance can be assessed via measurement of the deviation of the parameter distribution relative to G_h and G_f.

For the target-engine dataset, X, the distance between it and G_h or G_f is calculated as follows:

$$d_{h_j} = \frac{1}{M}\sum_{k=1}^{M} f_{\mathrm{WD}}\left(X_j, g_{h_j}^k\right) \tag{4}$$

$$d_{f_j} = \frac{1}{M}\sum_{k=1}^{M} f_{\mathrm{WD}}\left(X_j, g_{f_j}^k\right) \tag{5}$$

where $X_j \in \mathbb{R}^{N-RUL}$ represents the *j*-th monitoring data of the target engine. d_{h_j} and d_{f_j} denote the health and fault distances, and $f_{WD}(\cdot)$ represents the WD calculation. Further, the arithmetic mean value of the WD is used to represent d_h and d_f.

$$d_h = \frac{1}{P}\sum_{j=1}^{P} d_{h_j} \tag{6}$$

$$d_f = \frac{1}{P}\sum_{j=1}^{P} d_{f_j} \tag{7}$$

where $d_h,\ d_f \in \mathbb{R}^{N-RUL-win_d}$. The linear weight fusion algorithm was designed to convert d_h and d_f into the *HI* of the engine. The fusion equation is expressed as follows:

$$W_h = -\frac{1}{N - RUL - win_d} i + 1 \cdots\cdots (i = 1, 2, \ldots\ldots, N - RUL - win_d) \tag{8}$$

$$W_f = \frac{1}{N - RUL - win_d} i \cdots\cdots (i = 1, 2, \ldots\ldots, N - RUL - win_d) \tag{9}$$

$$HI = \frac{1}{2}\Big[(1 - W_h d_h) + W_f d_f\Big] \tag{10}$$

where W_f and W_h denote the health and fault baseline weights, respectively. And *HI* denotes the health index.

2.2.3. Transfer-Timing Identification of Engine Performance Degradation

During the initial running stages of the engines, the *HI* value exhibits minor fluctuations between adjacent cycles. As the engine runs, degradation gradually accelerates, and the engine performance changes from healthy to degraded. The transfer-timing identification mechanism is described in this section. The *HI* is sliced with a window of width Win_{HI}. Fitting to the *HI* within the window is performed. The fitting equation is expressed as follows:

$$HI' = a_i t + b \tag{11}$$

where HI' is the fitting *HI* of the engine within the window. t represents the cycle number. a_i is the slope in the *i*-th window. b is the intercept term. a_i is defined as a degradation-sensitive feature. When a_i is close to 0, the engine is relatively healthy, whereas when a_i is less than 0, the engine has started to degrade.

When the degradation-sensitive feature is less than the $Thre_a$ set, the target engine is deemed as having degraded. In general, the threshold is slightly less than zero to avoid misjudgments due to small fluctuations in *HI*. When the cycle is characterized by a degradation-sensitive feature below $Thre_a$, the degradation stage starts.

2.3. Multi-Source-Domain Adaptive Deconstruction Based on TL-EDM

The purpose of multi-source-domain adaptive deconstruction is to quantify data transferability and select the source domains and sequences with high transferability. The raw values of monitoring parameters reflect the immediate operating state, while the degradation rate represents the long-term degradation trend of the engine. Therefore, it is challenging for any single measurement method to be able to comprehensively measure the degree of transferability from multiple aspects.

To solve the issue, TL-EDM, a method that takes advantages of both the distribution measure of WD and the angle measure of cosine distance, was designed for transferability measurement. Herein, the entire life cycle of the source domain is traversed through a sliding window time delay. The transferability is measured by calculating the ensemble distances between various initial position sequences in the source domain. In this way, source domains with high transferability are screened from multi-source domains, thus achieving multi-source-domain adaptive deconstruction based on two levels.

2.3.1. Transferability Measurement Using Time-Lag Ensemble Distance

In this paper, we adopt principal component analysis (PCA), a statistical method for dimensionality reduction, to reduce error accumulation and extract salient degradation information. Through orthogonal transformation, the original variables that may be correlated are transformed into mutually independent variables. The transformed variables are called principal components. The PCA model identifies the correlation between the monitoring parameters and output independent principal components. The most independent component in the degradation stage is identified as the engine degradation principal component (DPC).

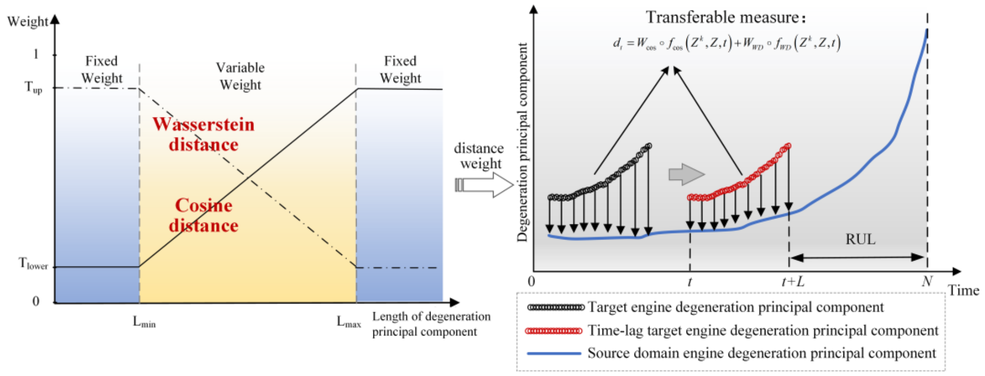
The number of cycles in the degradation stage can significantly vary between target engines. For engines with shorter degradation stages, the monitoring data are more affected by errors and fluctuations, and the degradation rate is less able to characterize degradation. Therefore, we should focus more on the DPC data values. Conversely, for longer degradation stages, the rate value can better represent the engine degradation process than the original data value. WD and cosine distance are selected to represent two transferability evaluation indices, i.e., degradation degree and rate, respectively. These two distance weights are determined by the target DPC length.

To avoid imbalances in the transferable evaluation index caused by excessive weights, upper and lower thresholds are set for the weights of each distance, as shown in Figure 3a. The weights of the two distances are calculated as follows, when the DPC length of the engine is L:

$$\begin{cases} W_{cos} = Ther_{\ d-low} & \text{if } L < L_{min} \\ W_{cos} = \frac{Ther_{d-up} - Ther_{\ d-low}}{L_{max} - L_{min}}(L - L_{min}) + Ther_{\ d-low} & \text{if } L_{min} \le L \le L_{max} \\ W_{cos} = Ther_{d-up} & \text{if } L_{max} < L \end{cases} \tag{12}$$

$$\begin{cases} W_{WD} = Ther_{d-up} & \text{if } L < L_{min} \\ W_{WD} = -\frac{Ther_{d-up} - Ther_{d-low}}{L_{max} - L_{min}}(L - L_{min}) + Ther_{d-up} & \text{if } L_{min} \le L \le L_{max} \\ W_{WD} = Ther_{d-low} & \text{if } L_{max} < L \end{cases} \tag{13}$$

where $Ther_{d\text{-}up}$ and $Ther_{d\text{-}low}$ represent the upper and lower limits of the distance weights, respectively, when $Ther_{d\text{-}up} + Ther_{d\text{-}low} = 1$. L_{max} and L_{min} are the maximum and minimum values of the weight range. The weights of W_{cos} and W_{WD} can be adaptively determined for different target engines. A weighted ensemble distance is proposed to measure the transferability of the degradation trajectory.


Weight
1
T_up
T_lower
0
Fixed Weight
Variable Weight
Fixed Weight
Wasserstein distance
Cosine distance
L_min
L_max
Length of degeneration principal component
distance weight
Transferable measure：
$d_t = W_{cos} \circ f_{cos}(Z^k, Z, t) + W_{WD} \circ f_{WD}(Z^k, Z, t)$
Degeneration principal component
RUL
0
t
t+L
N
Time
Target engine degeneration principal component
Time-lag target engine degeneration principal component
Source domain engine degeneration principal component

(a) Ensemble distance measure (b) Time-lag distance measure

Figure 3. Transferability measurement based on TL-EDM and high-transferability source domain selection.

Because of individual differences, the initial fault degrees among target engines vary. To accurately measure source domain transferability, a time-lag process was added to ensemble distance measurement. Figure 3b displays the application of a time-lag slice on the complete degradation trajectory of the source domain, with the target-engine DPC length L as the window length.

The blue line in Figure 3b is the source domain DPC. The black line is the target-engine DPC. The red line is the time-lag DPC. When the starting point of the sliding window is t, the ending point is $t + L$. Moreover, N denotes the number of life cycles of the source-domain engine. The source-domain DPC is denoted by Z^k, and the target DPC is denoted by Z. The transferable distance is calculated as follows:

$$d_t = W_{\cos} \circ f_{\cos}\left(Z^k, Z, t\right) + W_{WD} \circ f_{WD}\left(Z^k, Z, t\right) \tag{14}$$

where W_{cos} and W_{WD} are the weights of the cosine distance and WD, respectively. $f_{cos}(\cdot)$ and $f_{WD}(\cdot)$ represent calculations for the cosine distance and WD distance, respectively. d_t is the transferable distance. $\circ$ represents Hadamard product. The transferable distance set is denoted as $D = \{d_1, d_2, \ldots d_t \ldots d_{N-L}\}$.

2.3.2. Multi-Source-Domain Adaptive Deconstruction with Transferability

Based on the transferable distance of source domains, we deconstructed the multi-source domain to select high-transferability source domains and label their high-transferability sequences. The high-transferability source domains are pre-selected based on transferability, after which a secondary selection is performed based on RUL labels. The inverse of the transferable distance characterizes the transferability of sequences in the source domain, and its maximum value is defined as the transferability.

$$\gamma_t{}^k = \frac{1}{d_t{}^k} \tag{15}$$

$$\gamma^k = \max(\gamma_t{}^k) = \frac{1}{\min\left(D^k\right)} = \frac{1}{\min\left(d_1{}^k, d_2{}^k, \ldots, d_t{}^k, \ldots, d_{N-L}{}^k\right)} \tag{16}$$

where D^k is the transferable distance set of the k-th source-domain engine. γ^k represents transferability. In the deconstruction mechanism:

(1) Pre-selection based on transferability

In pre-selection, the source-domain engines are arranged in descending order of transferability, γ_k. The top Num_s engines are selected as the pre-selection set.

(2) Secondary selection based on RUL labels

The pre-selected source domains are similar to the target engine. However, the running time of the target engine is relatively short, and the source domains have the potential to be accidentally similar. A secondary selection is, therefore, proposed to exclude the source domains without similar RUL label. Based on the pre-selected source domain, RUL^{*k} is calculated corresponding to t^k:

$$t^k = \text{argmin}_t(D^k) = \text{argmin}_t(d_1^k, d_2^k, \ldots, d_t^k, \ldots, d_{N-L}^k) \tag{17}$$

$$RUL^{*k} = N^k - (t^k + L) \tag{18}$$

where t^k represents the starting time of the highest-transferability sequence in k-th source-domain engine. N^k represents the number of life cycles of the k-th source-domain engine. RUL^{*k} represents the RUL label of the k-th source-domain engine.

Generally, similar sequences should have a uniform RUL label distribution. The mean value (μ) and variance (σ) of the RUL^* label are computed for the pre-selected source domain. The secondary selected range of RUL^* is $[\mu - r{\cdot}\sigma, \mu + r{\cdot}\sigma]$, where r is the ratio coefficient. The engine source domains with RUL^* labels not in the selection range are eliminated. And the remaining source domains are identified as high-transferability source domains.

2.4. Personalized Transfer Prediction Based on Dynamic-Weight Informer Model

2.4.1. Two-Stage Transfer Learning Prediction Scheme

As shown in Figure 1, data transferability can widely vary within a multi-source domain. Similarly, there are also differences between target engines. A general transfer prediction model typically has insufficient adaptability. Moreover, in existing methods, the transfer models are trained using equal training sample weights. These methods neglect differences between target engines, leading to a lack of individuality in the prediction model and limiting its accuracy.

To address this problem, a personalized transfer learning scheme with a two-stage transfer is proposed. The scheme flow and dynamic-weight informer model structure are shown in Figure 4. Through pre-training prediction models using data in the multi-source domain, general degradation information can be mined, which can reduce the data required for the predictive training process. Then, by using the high-transferability source domain, individual degradation information similar to that of the target engine can be mined.

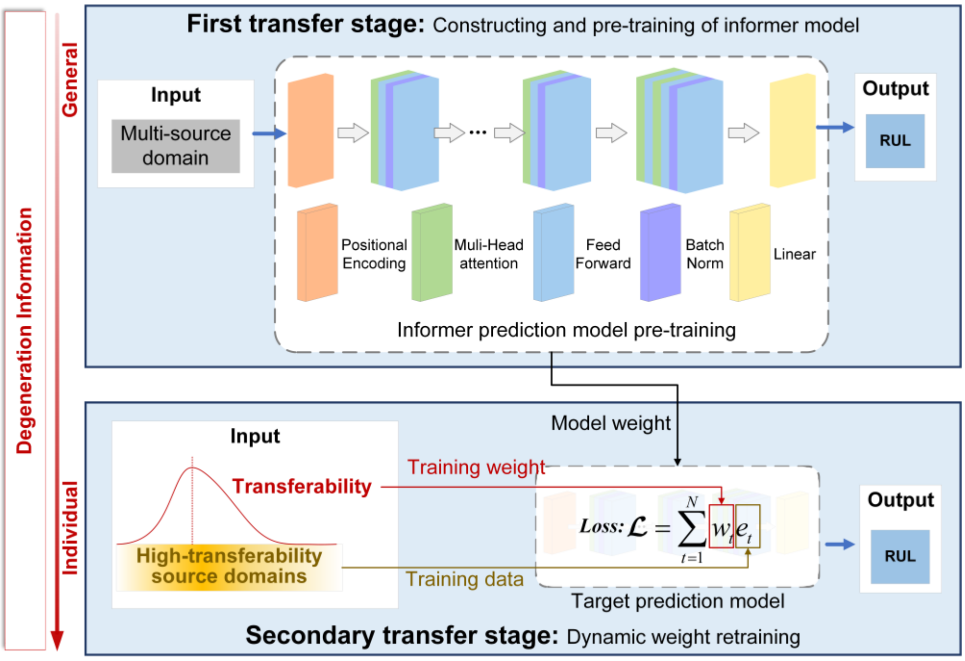

Degeneration Information
General
Individual
First transfer stage: Constructing and pre-training of informer model
Input
Multi-source domain
Output
RUL
Positional Encoding
Muli-Head attention
Feed Forward
Batch Norm
Linear
Informer prediction model pre-training
Model weight
Input
Transferability
High-transferability source domains
Training weight
Training data
Loss: $\mathcal{L} = \sum_{t=1}^{N} w_t e_t$
Target prediction model
Output
RUL
Secondary transfer stage: Dynamic weight retraining

Figure 4. RUL transfer prediction based on dynamic-weight informer model.

(1) First transfer stage: constructing and pre-training the informer prediction model

During the initial transfer stage, a substantial amount of sensor data is necessary to extract general degradation information. Deep learning algorithms are widely used in nonlinear feature extraction. Compared to traditional time series prediction methods such as LSTM, the transformer neural network obtains correlation relationships and attention matrices between model inputs and exclusively outputs through the attention mechanism, allowing for more parallelization and resulting in higher processing efficiency and prediction quality [29]. However, high memory usage and time complexity issues prevent the transformer model from being directly applied to RUL prediction [30]. The informer model solves these issues by utilizing the ProbSparse self-attention mechanism to replace the original attention mechanism. To address the computational burden and potential accuracy impact associated with long time series, the informer modified by transformer neural network is ultimately selected as the RUL prediction model for engines.

The informer engine prediction model is constructed utilizing the architecture in Figure 4 and employing a random set of initial weights. The model is then trained on all available multi-source-domain data. The training input consists of sequentially concatenated source-domain DPCs, while the training output corresponds to the RUL value of the DPC sequence. Equal weights are assigned to the training data, enabling the identification of general engine degradation information through the trained model.

(2) Secondary transfer stage: dynamic weight retraining

In the secondary transfer stage, the core principle is to use high-transferability source domains to mine the individual degradation information by retraining. For the target engine, the high-transferability source domains and sequences are determined, and training sample weights are set according to transferability, thus introducing transferability information into the prediction model to mine the individual degradation information.

- Setting the training data weights with transferability

Within the source domain, the DPC sequences corresponding to different initial times t are characterized by distinct transferability. To input transferability time information into the prediction model, the training data weight is designed as follows:

$$w_t = \frac{1}{d_t} \tag{19}$$

where w_t is the training data weight corresponding to the initial time. d_t is the transferable distance.

- Retraining of individual layers of personalized model

The DPCs of the high-transferability source domains are used to retrain the personalized model. The different position sequences of the source domain are assigned corresponding to training data weights during the iterative process of retraining the model. Through retraining, the model weight is determined, and the model training process is completed.

(3) Target-engine RUL prediction using the personalized transfer learning method

With the target-engine DPC as input, the target-engine RUL is predicted using the model trained for the target engine.

2.4.2. Dynamic-Weight Informer Prediction Model

To adapt the transformer model for time series prediction tasks, certain modifications are applied to the traditional model. The model structural diagram is depicted in Figure 5. The model's input embedding layer is removed. Instead, the DPC time series values are utilized as the model input. Moreover, the softmax layer for classification output is excluded. And the mean square error (MSE) function associated with regression is adopted as the loss function. On this basis, the transferable distance is introduced into the prediction model, and the dynamic weight loss function is constructed as

$$\mathcal{L} = \sum_{t=1}^{N} w_t e_t \tag{20}$$

where e_t is the MSE error of the t-th DPC time series. w_t is the corresponding training weight. $\mathcal{L}$ denotes the model training loss. The principal framework of the prediction model is divided into four main parts: position encoding, encoder, decoder, and fully connected layers.

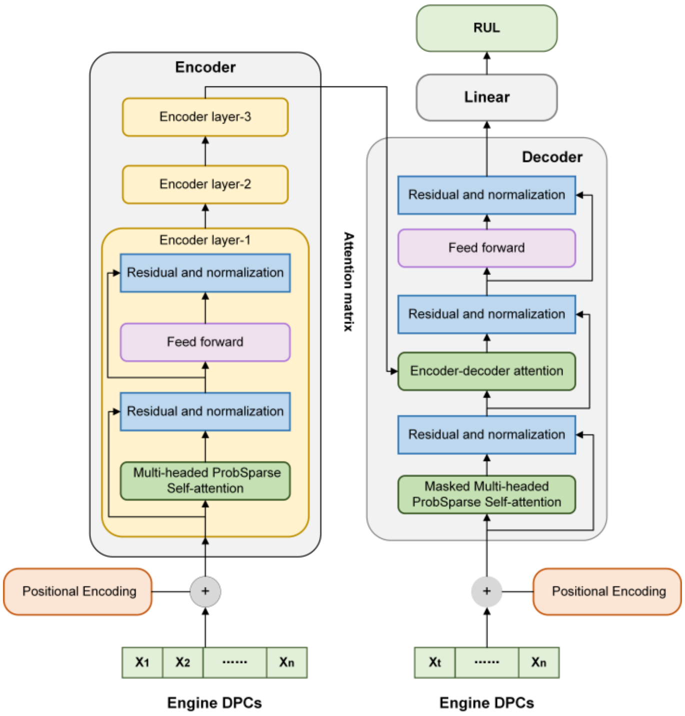

RUL
Linear
Encoder
Encoder layer-3
Encoder layer-2
Encoder layer-1
Residual and normalization
Feed forward
Residual and normalization
Multi-headed ProbSparse Self-attention
Attention matrix
Decoder
Residual and normalization
Feed forward
Residual and normalization
Encoder-decoder attention
Residual and normalization
Masked Multi-headed ProbSparse Self-attention
Positional Encoding
+
X1
X2
......
Xn
Engine DPCs
+
Positional Encoding
Xt
......
Xn
Engine DPCs

Figure 5. Dynamic-weight informer prediction model.

(1) Position encoding: In this study, we choose to utilize absolute positional coding to locate each element in the time series [29]. The inputs are denoted as $X = \{x_1, x_2, \ldots, x_i, \ldots, x_N\}$, where N denotes the length of time series. By position encoding X, we can obtain $PE = \{pe_1, pe_2, \ldots, pe_N\}$. The input of the model is $input = \{x_1 + pe_1, x_2 + pe_2, \ \ldots \ , x_N + pe_N\ \}$.

$$pe_{(i,2j)} = \sin\left(\frac{i}{10,000^{\frac{2j}{P}}}\right) \tag{21}$$

$$pe_{(i,2j+1)} = \cos\left(\frac{i}{10,000^{\frac{2j+1}{P}}}\right) \tag{22}$$

where i denotes the position of X in $input$. j is the position in x_i. P is the dimension of x_i, that is, the number of sensor-monitoring parameters. By applying positional encoding of time series, the transformer can acquire not only the specific data information but also the corresponding position information of the time series.

(2) ProbSparse self-attention mechanism: The attention mechanism is a fundamental component of the transformer model, enabling it to extract essential information from extensive datasets. In time series analysis, the focus is on feature extraction for prediction tasks. For input matrix X, it is transformed into matrix Q (Query), K (Key), and V (Value) by different weight matrices. The ProbSparse self-attention mechanism aims to identify important sparse queries to optimize calculation efficiency. For $Q \in R^{m\times \mathrm{d}}$, $K \in R^{n\times \mathrm{d}}$, and $V \in R^{n\times \mathrm{d}}$, the specific steps are as follows:

- By sampling K, we can obtain k_k. Sampling length is L_k. For each $q_i \in Q$, M is calculated as follows [29]:

$$M(q_i, K) = \max\left(\frac{q_i k_k^T}{\sqrt{d_k}}\right) - \frac{1}{L_k}\Sigma\left(\frac{q_i k_k^T}{\sqrt{d_k}}\right) \tag{23}$$

- Compute the u queries, q, with the greatest M values, and assemble them into a new matrix denoted as $\overline{Q}$.
- Calculate $Attention(Q, K, V) = \text{softmax}\left(\frac{\overline{Q}^T T}{\sqrt{d_k}}\right)V$. Attention extraction identifies feature data with high information quality and strong performance expression capabilities, thus improving the model prediction accuracy.

(3) Residual and normalization

In the informer encoder, a residual and normalization layer are implemented after each module, as shown in Figure 5. Incorporating a residual connection serves to retain the original information and improve the generalization capability of the model. Additionally, the inclusion of layer normalization guarantees a stable data distribution and expedites model convergence.

3. Case Study

3.1. Data Description

In this study, the FD001 dataset provided by NASA for their PHM08 Challenge was used for the framework research and validation [31]. Several researchers proved the authority of the dataset [32–34]. The FD001 dataset includes the following:

(1) Training dataset: full-life monitoring parameter data on degradation processes for 100 engines;
(2) Testing dataset: randomly intercepted data on 100 test engines (target engines), in terms of monitoring parameters and their RULs.

The FD001 dataset includes 21 engine gas-path parameters. As an engine operates, its parameter data gradually change, indirectly reflecting the engine's performance. However, the sensitivity to degradation of each parameter widely varies. Data analyses showed that seven of the parameters (T2, P2, P15, EPR, farB, Nf_dmd, and PCNFR_dmd) do not exhibit change trends. Through the relevant research, eight parameters (T24, T30, Ps30, PHI, P30, T50, BPR, and Nf) with prominent variation trends were selected as the input parameters for engine RUL prediction. The parameter descriptions are shown in Table 1.

Table 1. Engine parameter descriptions.

Symbol	Description
T24	Total temperature at fan inlet (◦R)
T30	Total temperature at LPC outlet (◦R)
Ps30	Static pressure at HPC outlet (psia)
PHI	Ratio of fuel flow to Ps30 (pps/psi)
P30	Total pressure at HPC outlet (psia)
T50	Total temperature at LPT outlet (◦R)
BPR	Bypass ratio
Nf	Physical fan speed (rpm)

3.2. Performance Assessment and Transfer-Timing Identification

We referred to the scale self-optimization smoothing method proposed in [35] and pre-processed the eight monitoring parameters. The degradation trends of the monitoring parameters were normalized, as shown in Figure 6. The degradation directions of the PHI and P30 parameters were reversed to an increasing trend. And the degradation information remained unchanged. Table 2 presents the health and fault baseline engines' selection results.

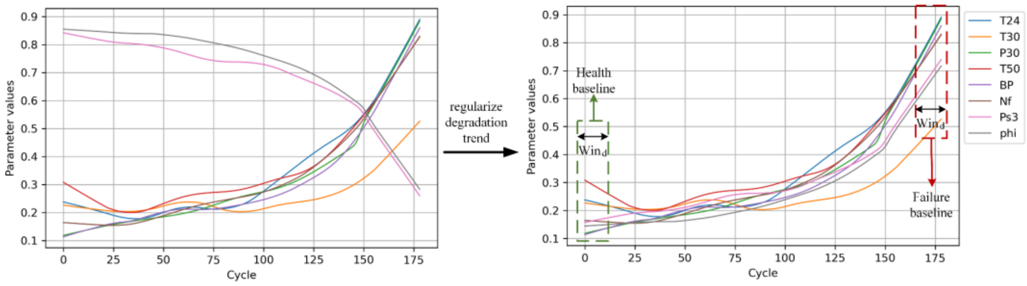


Figure 6. Parameter trend uniformization and dual-baseline construction.

Table 2. Health and fault baseline engines.

	Engine Numbers
Health baseline engines	77#, 82#, 94#, 14#, 8#, 1#, 46#, 60#, 27#, 81#
Fault baseline engines	55#, 61#, 21#, 83#, 7#, 39#, 90#, 72#, 65#, 15#

The performance assessment result is shown in Figure 7. An inverse distribution led to certain measurement errors at the beginning and end of the curves for the health and fault distances. The weights of the initial value for the health WD and the end value for the fault WD were reduced using the linear fusion algorithm. The results indicated that the endpoint measure bias problem caused by inverse distribution is well-solved.

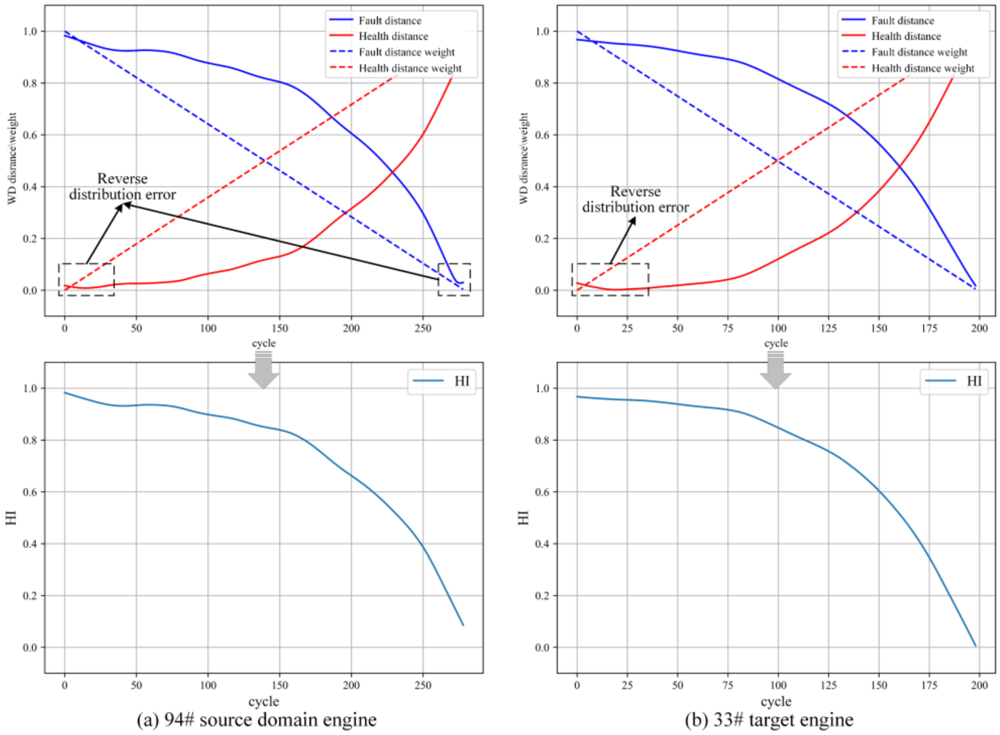


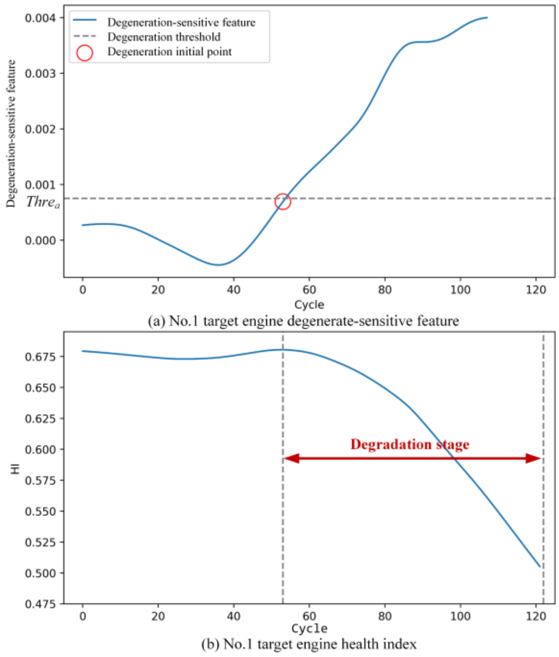
Figure 7. Engine WD measurement results and linear weights.

The results show that HI degradation was slow at the beginning of the engine operation and that the rate of HI degradation gradually increased as the cycle number increased. The time at which the target engine began to significantly degenerate can be determined by identifying the inflection point of the HI. Table 3 lists the parameter settings for engine-health-state assessment and transfer-timing identification.

Table 3. Parameters for engine-health-state assessment and transfer-timing identification.

Parameter Name	Win_d	Win_{HI}	$Thre_a$
Parameter values	5	10	0.00075

The linear fitting slope of the HI in the window with Win_{HI} as its width is a degradation-sensitive feature. Here, $Thre_a$ was used as the discriminant condition to identify when the degradation-sensitive feature exceeded the threshold, which is the transfer timing. Numerous previously published works defined the last 130 cycles of engine operation as the degradation stage, while categorizing the preceding cycles as the healthy stage [36]. In our study, we conducted a statistical analysis on the healthy stage data within the FD001 training dataset, revealing that the average standard deviation of the degradation-sensitive feature during this phase is 0.0007624. Consequently, we set $Thre_a$ to 0.00075, which is close to the value. Using the No. 3 target engine as an example, the results of the degradation stage and transfer-timing identification are shown in Figure 8.



(a) No.1 target engine degenerate-sensitive feature

(b) No.1 target engine health index

Figure 8. Transfer-timing identification for No. 3 target engine.

The results demonstrated that with increased engine operating time, the engine performance deteriorates, which is reflected in a more rapid decline of the HI values. The degradation-sensitive feature increased and exceeded the degradation threshold, $Thre_a$, indicating that the engine entered the degeneration stage. The health statement of the target engine exhibited a slight and stable fluctuation during the early stage of degradation, with the corresponding degradation-sensitive feature fluctuating around 0 within a range of 0.0005–0.001. Therefore, $Thre_a$ was set to 0.00075.

3.3. Transferability Measurement and Multi-Source-Domain Adaptive Deconstruction

Figure 9 shows the PCA results of the source- and target-domain engine. The source-domain engines vary from each other in terms of entire-life length, degradation rate, degradation characteristics, and initial health degree. In addition to these differences, the running-time lengths of the target engines also greatly vary. Therefore, the transferable distances of the source domains and sequences for different target engines greatly vary. Table 4 shows the parameters for transferability measurement and high-transferability source domain selection. Figure 10 shows the transferable distances of the No. 1 source-domain engine for the No. 3 target engine.

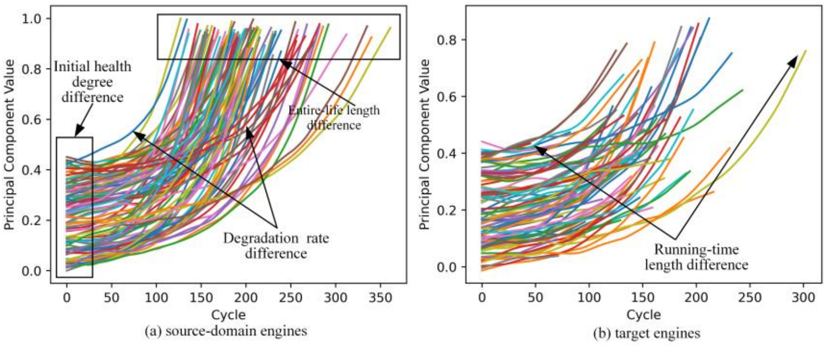


Figure 9. PCA results.

Table 4. Parameters for multi-source-domain adaptive deconstruction.

Parameter Name	$Ther_{d\text{-}up}$	$Ther_{d\text{-}low}$	S	r
Parameter values	1/3	2/3	10	1.5

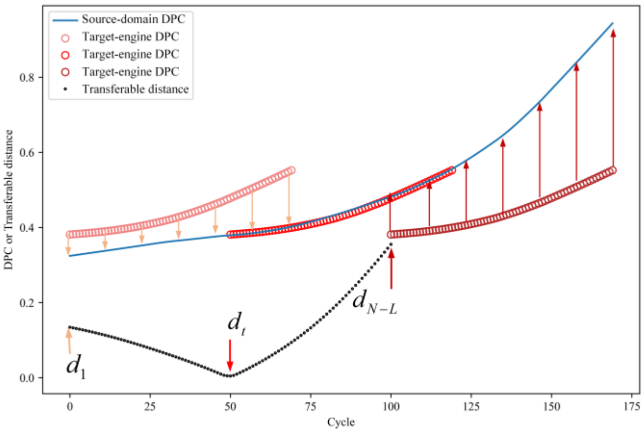
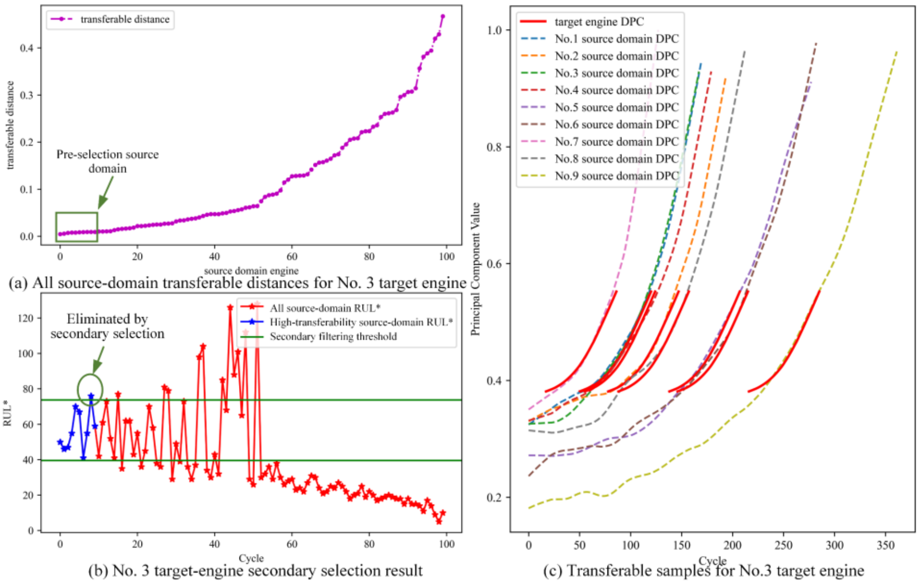


Figure 10. Transferable distances of source-domain sequences for No. 3 target engine.

In the multi-source-domain adaptive deconstruction process, the pre-selection source-domain quantity, Num_S, was set to 10. The secondary selection ratio, r, was then set to 1.5. Thus, the source domains with RUL^* in the interval $[\mu - 1.5\sigma, \mu + 1.5\sigma]$ were identified as the high-transferability source domains. With respect to $Ther_{d\text{-}up}$ and $Ther_{d\text{-}low}$, as shown in Equations (13) and (14), the distance weight was partitioned into three equal segments. Accordingly, $Ther_{d\text{-}up}$ and $Ther_{d\text{-}low}$ were configured as 1/3 and 2/3, respectively. The multi-source-domain adaptive deconstruction results for the No. 3 target engine are shown in Figure 11.

The deconstruction pre-selection results for the No. 3 target engine are shown in Figure 11a. Figure 11b shows the secondary selection results based on the RUL^* label. The results show that, as the transferability of the source-domain sample decreases, the distribution of RUL^* gradually becomes unstable. Through secondary selection, source-domain samples without stable RUL^* distributions are removed. Finally, the high-transferability source domains and sequences for the target engine are determined. As shown in Figure 11c, the proposed multi-source-domain adaptive deconstruction algorithm can select source domains having degradation trajectories similar to that of the target engine.



(a) All source-domain transferable distances for No. 3 target engine

(b) No. 3 target-engine secondary selection result

(c) Transferable samples for No.3 target engine

Figure 11. Multi-source-domain adaptive deconstruction for No. 3 target engine.

3.4. RUL Prediction Using the Personalized Transfer Learning

3.4.1. Construction of the Informer-Based RUL Prediction Model

A precise model relies on reasonable parameter settings for accurate predictions. Table 5 shows the parameter settings of the RUL prediction model using informer. Table 6 presents the training parameters. To reduce overfitting, we set the L2 regular coefficient as 0.05 and set the dropout as 0.08.

Table 5. Structural parameters of the prediction model.

Parameter Name	Input Layer Neuron	Encoder-1 Neuron	Encoder-2 Neuron	Encoder-3 Neuron	Encoder-4 Neuron	Decoder-1 Neuron
Parameter values	14	14	10	8	6	4
Parameter name	Decoder-2 neuron	Output layer neuron	Encoder activation function	Decoder activation function	Loss function	Optimizer
Parameter values	2	1	LeakyRelu	tanh	MSE	Adam

Table 6. Training parameters of the prediction model.

Parameter Name	L2 Regularization Coefficient	Epoch	Batch Size	Dropout
Parameter values	0.05	1000	200	0.08

3.4.2. Personalized Transfer Learning for RUL Prediction Model Using DPCs

In the first transfer stage, the DPCs of all multi-source-domain data were used as training data. The input was the normalized DPC, and the training weights were equal. The output was the normalized RUL. During the secondary transfer stage, the input data for retraining the model were selected from high-transferability source domains. The weights for the training data were determined by transferability based on the multi-source-domain adaptive deconstruction. Figure 12 shows the training data and weights for the No. 3 target engine.

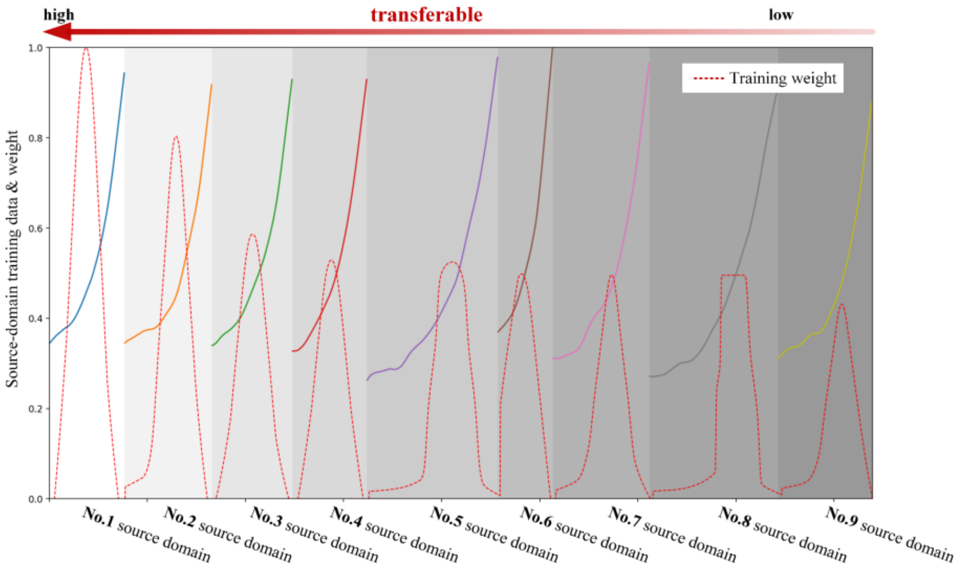


Figure 12. Retraining data and weights for No. 3 target engine.

Here, the training weights of the source domain gradually decreased as the transferability decreased. Furthermore, for a source domain, the weights of different position sequences exhibited a Gaussian-like distribution, which signified that a sequence with higher transferability has a higher weight and that the weights of other position sequences gradually decrease.

3.5. Comparative Analysis of the Prediction Results

3.5.1. Evaluation Index

This paper comprehensively evaluated the capability of the proposed transfer prediction framework using the following five indices.

- Prediction error

The prediction error is the difference between the predicted RUL value, $\hat{r}_t$, and the real RUL value, r_t.

$$error_t = r_t - \hat{r}_t \tag{24}$$

- Score

The score is an index proposed by the dataset provider [31], which is calculated based on $error_t$. The equation for score S is as follows:

$$s_n = \begin{cases} e^{-error/10} - 1 & \text{if } error_t \leq 0 \\ e^{error/13} - 1 & \text{if } error_t > 0 \end{cases} \tag{25}$$

$$S = \sum_{n=1}^{N} s_n \tag{26}$$

This index evaluates the capability of the method in terms of the percentage of correct predictions. When $error_t$ is in $[-10, 13]$, the prediction is considered to be correct. Otherwise, the prediction result is considered to be excessively early or excessively late.

- Relative accuracy

The relative accuracy is a general evaluation index for RUL prediction. The relative accuracy is defined as follows:

$$\text{Re}Acc = 1 - \frac{|L_t - \hat{L}_t|}{L_t} \tag{27}$$

where L_t is the real life, and $\hat{L}_t$ is the predicted life.

- RMSE

The RMSE is also a general evaluation index. For multiple prediction results, the RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=n}^{N} error_n{}^2} \tag{28}$$

3.5.2. Comparison of Source-Domain Numbers

In practice, determining the number of transfer source domains requires professional experience. However, the quantity and quality of the model training data are both affected by the number of transfer source domains. To compare the impact of the number of transfer source domains on the prediction results, a total of 40 RUL prediction experiments were conducted. The results are presented in Table 7 and Figure 13. When there are few transfer source domains, e.g., 1–5, the model is not sufficiently trained because of insufficient data, resulting in low prediction accuracy. As the number of transfer source domains increases, the accuracy increases to a peak and then gradually decreases because of the impact of data quality.

Table 7. Prediction results comparison.

Method/Index	Total Score	Acceptable Rate	Mean Relative Accuracy	RMSE
Proposed framework	**278.15**	**70%**	**95.24**	**12.34**
Without setting transfer timing	1567.77	44%	91.18	21.95
Without the general model transfer	312.05	55%	94.11	13.92
Without setting training data weights	326.64	69%	94.40	15.00
Without transfer scheme	1682.28	40%	90.96	22.32
DLSTM [36]	655	-	-	18.33
BHSLSTM [37]	376.64	63%	-	-
C-LSTM [38]	303	-	84.66%	16.127

The excessive selection of transfer source domains does not have a significantly negative impact on prediction accuracy. This is because the proposed transfer prediction framework limits the influence of low-transferability source domains on the prediction model via training weights. As shown in Figure 14, as the transferability of the source domain decreases, the weights of the training data significantly decrease. The proposed personalized transfer learning scheme has strong stability and parameter adaptability, making it suitable for practical industrial applications.

3.5.3. Methods Comparison

The RULs of 100 target engines in the C-MAPSS dataset were predicted using the framework, and the results are shown in Figure 15. The real life cycles of the target engines were distributed within the range of 141–341. Most prediction results were significantly close to these values. However, for some target engines, the life cycles were relatively small, thus increasing the prediction difficulty and causing some deviations in the results.

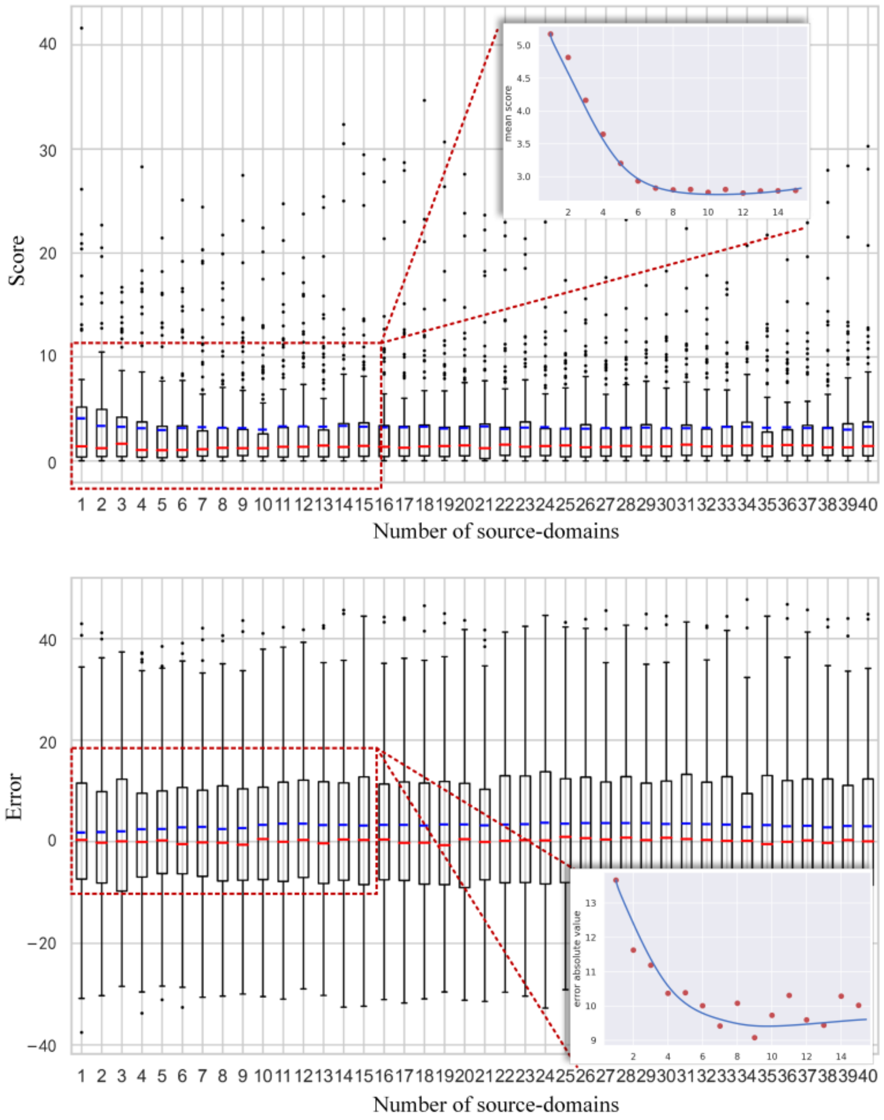


Figure 13. RUL prediction results for different transfer source-domain numbers.

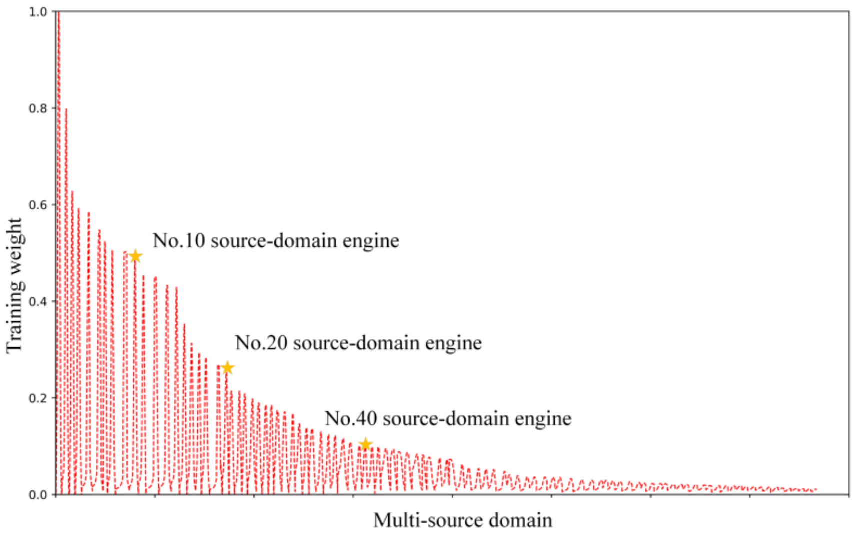


Figure 14. Training data weights for multi-source domain.

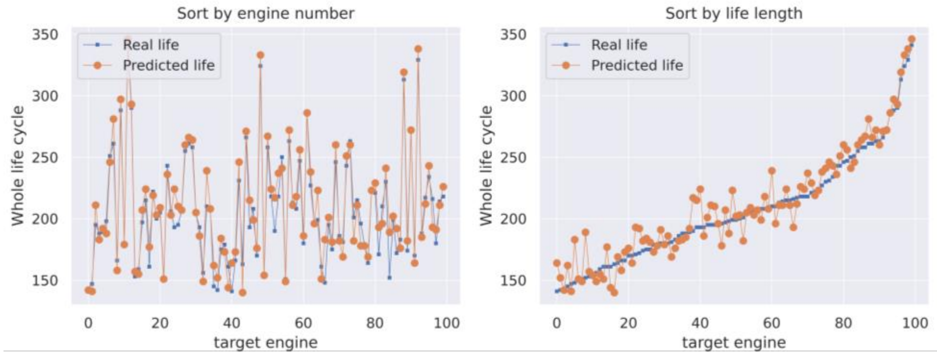


Figure 15. Target-engine RUL prediction results.

The errors and scores of the results are shown in Figure 16. The errors for 67% of the target engines were less than 10, and the errors for most of the engines were within the acceptable range [−10, 13]. The acceptable number was 70. And the scores for 88% of the engines were less than 5. Therefore, the proposed transfer prediction framework yielded accurate prediction results for the 100 target engines in the C-MAPSS dataset.

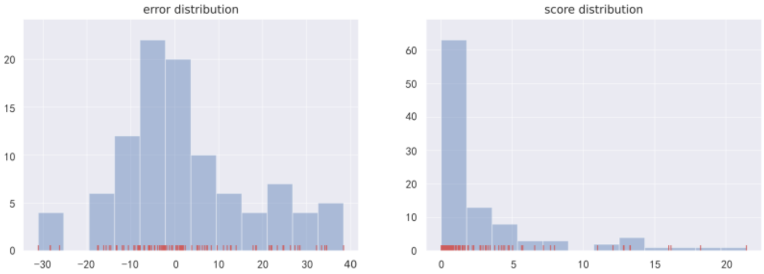


Figure 16. Target-engine RUL prediction error and score distributions.

Five comparative experiments were designed to verify the effects of the transfer time, sample, and scheme on the prediction results. In addition, we compared the results to those of other RUL prediction methods. The results are presented in Table 7.

- Comparative experiment without identifying transfer timing

The comparative experiment results are outlined in Table 7. All monitoring data of the target engine were used to select high-transferability source domains, thereby influencing the effectiveness of the multi-source-domain adaptive deconstruction. The scores and RMSEs indicated that the prediction accuracy is significantly affected for some target engines, which verified the importance of transfer-timing identification.

- Comparative experiment without using general model transfer

The experiment was conducted by excluding the pre-training process of the prediction model while keeping the data conditions, network structure, and model parameters consistent. The results demonstrated that the pre-training process contributes to the preservation of a greater amount of degradation information in the model. Furthermore, it was observed that the overall degradation information from the source domain has a positive impact on the RUL prediction.

- Comparative experiment without setting training data weight as transferability

During the same transfer process, equal training data weights were employed, resulting in decreased accuracy. These findings demonstrated that training weights play a pivotal role in incorporating transferable information into the prediction model, thereby improving its individualization and prediction accuracy.

- Comparative experiment without using transfer learning scheme

An experiment without a transfer learning scheme was conducted to verify the performance of the personalized transfer framework. The informer prediction model was trained by all the multi-source-domain data. And retraining with high-transferability source domains was canceled. Even though the model has powerful feature extraction and life prediction abilities, the results still showed significant deterioration. The experiment indicated that the transfer framework can significantly enhance the performance of the prediction model.

- Comparative experiment with commonly used deep-learning prediction methods

As shown in Table 7, relative to some commonly used prediction methods, such as DLSTM [36], bidirectional handshaking LSTM (BHSLSTM) [37], and convolution and LSTM (C-LSTM) [38] hybrid deep neural networks, the proposed personalized transfer prediction framework produced further breakthroughs. The results indicated that the proposed prediction framework in this paper achieves higher accuracy.

Through a series of comparative analyses, the effectiveness of the personalized transfer learning framework was verified for mining performance degradation information and engine RUL prediction. In addition, each part of the transfer learning framework—the transfer-timing identification based on DBA-WD, the multi-source-domain adaptive deconstruction based on TL-EDM, and the transfer prediction scheme based on the dynamic-weight informer model—contributed to improvements in the prediction accuracy.

4. Discussion

We predicted the RULs of 100 target engines using the proposed personalized transfer learning framework and obtained accurate results. There are three main reasons for these improvements: (1) the dual-baseline assessment accurately identifies the transfer timing, which is the key premise of transfer prediction; (2) the multi-source-domain adaptive deconstruction based on TL-EDM effectively screens out the high-transferability source domains; (3) the personalized transfer prediction scheme enhances individualization and ensures the accuracy of the prediction model. The comparison results indicated that all components of the framework contribute to the improvement of prediction accuracy.

(1) When to transfer: dual-baseline performance assessment can accurately identify the performance degradation stage of the target engine, which is the key premise for excluding unnecessary health-stage data.

The quantitative description of the engine performance state is key to determining when the decline begins and to identifying the transfer timing, which affects the selection effectiveness of the high-transferability source domains. However, the description of the performance state by a single metric can cause a reverse distribution error. The results indicated that the dual-baseline assessment can resolve this. In addition, the comparative experiment results showed that the transfer timing significantly affects RUL prediction accuracy.

(2) What to transfer: the multi-source-domain adaptive deconstruction based on TL-EDM can effectively mine high-transferability source domains, thus balancing transferable data quantity and information quality.

The retraining data screening results verified the accuracy of the transferability measurement. The results indicated that the RUL labels of sequences with high transferability were centrally distributed, proving that the screened sequences have similar degradation features. These observations can be attributed to three main reasons: (1) the transferable distances are measured by ensemble distance from the two perspectives of the raw values and degradation rate to guarantee the comprehensiveness of the multi-source-domain adaptive deconstruction; (2) the distance-measure weights are adjusted according to the DPC length to reduce measurement errors caused by fluctuation; (3) the time-delay measure deconstructs the multi-source domain into independent sequences, avoiding the effects of initial differences on the transferability measure. The prediction results showed that screening the high-transferability source domains and sequences from the multi-source-domain adaptive deconstruction is an essential basis for RUL transfer prediction.

(3) How to transfer: the personalized transfer learning framework effectively utilizes the general and individual information from multi-source domains of the same type engines, thereby improving the individualization and accuracy of the transfer prediction model for each target engine.

The prediction results for 100 target engines reveal that (1) even when the multi-source domain has low transferability, it still has the common degradation characteristics of engines. Mining general degradation information can provide better support for prediction models and reduce the data required for model retraining. (2) The DCPs of high-transferability source domains are similar to those of the target engine. By using them to retrain the prediction model, the individual degradation information can be mined. (3) Setting the training data weights can introduce the transferability information of the training data into the training process. The comprehensive utilization of the deconstructed multi-source-domain data can increase individuation and improve accuracy.

5. Conclusions

We proposed a personalized transfer learning framework for predicting turbofan engine RUL, thus improving the prediction accuracy from when, what, and how to transfer. In the meantime, the framework can maximize the utilization of similar degradation information contained in engines of the same type and balance the quantity and quality of transferable information. The prediction results validated that the proposed framework is necessary and applicable for RUL predictions of turbofan engines with individual differences.

The performance of the proposed transfer prediction framework was verified using an international general simulation dataset with a total score of 278.15 and an average prediction accuracy of 95.24%. The comparative results indicated that the joint application of the transfer-timing identification, the multi-source-domain adaptive deconstruction, and the personalized transfer prediction based on the dynamic-weight informer model can exclude unnecessary health-stage data, guarantee transferable data quantity and information quality, improve individuation, and increase prediction accuracy. In this regard, this transfer prediction framework can be extended to other time series predictions.

For future research, our main task will be to address the transfer prediction problem under multiple operating conditions and multiple failure modes. Additionally, we will continue to explore prediction methods for other types of equipment.

Author Contributions: X.L., J.M. and D.S. contributed to the study conception and design. Material preparation, data collection, and analysis were performed by X.L. The first draft of the manuscript was written by X.L. and J.M. All authors contributed to the revision of the paper. All authors commented on previous versions of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Foundation of State Key Laboratory (grant number 6142004200501), a Civil Aircraft Special Research Project (grant number MJ-2018-Y-58), the Fundamental Research Funds for the Central Universities (grant number YWF-22-L-516), and the National Natural Science Foundation of China (grant No. 51575021).

Data Availability Statement: The data used in this article are publicly available on the web.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kayid, M.; Alshagrawi, L.; Shrahili, M. Stochastic Ordering Results on Implied Lifetime Distributions under a Specific Degradation Model. *Axioms* **2023**, *12*, 786. [CrossRef]
2. Xue, B.; Xu, H.; Huang, X.; Zhu, K.; Xu, Z.; Pei, H. Similarity-based prediction method for machinery remaining useful life: A review. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 1501–1531. [CrossRef]
3. Ahmadzadeh, F.; Lundberg, J. Remaining useful life estimation: Review. *Int. J. Syst. Assur. Eng. Manag.* **2014**, *5*, 461–474. [CrossRef]

4. Camci, F.; Chinnam, R.B. Health-state estimation and prognostics in machining processes. *IEEE. Trans. Autom. Sci. Eng.* **2010**, *7*, 581–597. [CrossRef]
5. Askari, B.; Bozza, A.; Cavone, G.; Carli, R.; Dotoli, M. An Adaptive Constrained Clustering Approach for Real-Time Fault Detection of Industrial Systems. *Eur. J. Control* **2023**, 100858. [CrossRef]
6. Atrigna, M.; Buonanno, A.; Carli, R.; Cavone, G.; Scarabaggio, P.; Valenti, M.; Graditi, G.; Dotoli, M. A Machine Learning Approach to Fault Prediction of Power Distribution Grids under Heatwaves. *IEEE Trans. Ind. Appl.* **2023**, *59*, 4835–4845. [CrossRef]
7. Wang, Y.; Zhao, Y.; Addepalli, S. Remaining useful life prediction using deep learning approaches: A review. *Procedia Manuf.* **2020**, *49*, 81–88. [CrossRef]
8. Wang, Y.; Zhao, Y. Multi-Scale Remaining Useful Life Prediction Using Long Short-Term Memory. *Sustainability* **2022**, *14*, 15667. [CrossRef]
9. Wang, Y.; Zhao, Y.; Addepalli, S. Practical options for adopting recurrent neural network and its variants on remaining useful life prediction. *Chin. J. Mech. Eng.* **2021**, *34*, 69. [CrossRef]
10. Mou, Q.; Wei, L.; Wang, C.; Luo, D.; He, S.; Zhang, J.; Xu, H.; Luo, C.; Gao, C. Unsupervised domain-adaptive scene-specific pedestrian detection for static video surveillance. *Pattern Recognit.* **2021**, *118*, 108038. [CrossRef]
11. Alhudhaif, A.; Polat, K.; Karaman, O. Determination of COVID-19 pneumonia based on generalized convolutional neural network model from chest X-ray images. *Expert. Syst. Appl.* **2021**, *180*, 115141. [CrossRef]
12. Deng, Z.; Wang, Z.; Tang, Z.; Huang, K.; Zhu, H. A deep transfer learning method based on stacked autoencoder for cross-domain fault diagnosis. *Appl. Math. Comput.* **2021**, *408*, 126318. [CrossRef]
13. Yang, B.; Xu, S.; Lei, Y.; Leu, C.G.; Stewart, E.; Roberts, C. Multi-source transfer learning network to complement knowledge for intelligent diagnosis of machines with unseen faults. *Mech. Syst. Signal Process.* **2022**, *162*, 108095. [CrossRef]
14. Kim, S.; Choi, Y.Y.; Kim, K.J.; Choi, J.L. Forecasting state-of-health of lithium-ion batteries using variational long short-term memory with transfer learning. *J. Energy Storage* **2021**, *41*, 102893. [CrossRef]
15. Pan, D.; Li, H.; Wang, S. Transfer learning-based hybrid remaining useful life prediction for lithium-ion batteries under different stresses. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3501810. [CrossRef]
16. Chen, H.; Zhan, Z.; Jiang, P.; Sun, Y.; Liao, L.; Wan, X.; Du, Q.; Chen, X.; Song, H.; Zhu, R.; et al. Whole life cycle performance degradation test and RUL prediction research of fuel cell MEA. *Appl. Energy* **2022**, *310*, 118556. [CrossRef]
17. Li, J.; Lu, J.; Chen, C. Tool wear state prediction based on feature-based transfer learning. *Int. J. Adv. Manuf. Technol.* **2021**, *113*, 3283–3301. [CrossRef]
18. Ding, Y.; Jia, M.; Miao, Q.; Huang, P. Remaining useful life estimation using deep metric transfer learning for kernel regression. *Reliab. Eng. Syst. Saf.* **2021**, *212*, 107583. [CrossRef]
19. Ding, Y.; Ding, P.; Jia, M. A novel remaining useful life prediction method of rolling bearings based on deep transfer auto-encoder. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3509812. [CrossRef]
20. Shen, F.; Yan, R. A new intermediate domain SVM-based transfer model for rolling bearing RUL prediction. *IEEE ASME Trans. Mechatron.* **2021**, *27*, 1357–1369. [CrossRef]
21. Mao, W.; Liu, J.; Chen, J.; Liang, X. An interpretable deep transfer learning-based remaining useful life prediction approach for bearings with selective degradation knowledge fusion. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3508616. [CrossRef]
22. Xia, P.; Huang, Y.; Li, P.; Liu, C.; Shi, L. Fault knowledge transfer assisted ensemble method for remaining useful life prediction. *IEEE Trans. Ind. Inform.* **2021**, *18*, 1758–1769. [CrossRef]
23. Cheng, H.; Kong, X.; Wang, Q.; Ma, H.; Yang, S. The two-stage RUL prediction across operation conditions using deep transfer learning and insufficient degradation data. *Reliab. Eng. Syst. Saf.* **2022**, *225*, 108581. [CrossRef]
24. Zhuang, J.; Jia, M.; Ding, Y.; Ding, P. Temporal convolution-based transferable cross-domain adaptation approach for remaining useful life estimation under variable failure behaviors. *Reliab. Eng. Syst. Saf.* **2021**, *216*, 107946. [CrossRef]
25. Miao, M.; Yu, J.; Zhao, Z. A sparse domain adaption network for remaining useful life prediction of rolling bearings under different working conditions. *Reliab. Eng. Syst. Saf.* **2022**, *219*, 108259. [CrossRef]
26. Miao, M.; Yu, J. A deep domain adaptative network for remaining useful life prediction of machines under different working conditions and fault modes. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3518214. [CrossRef]
27. Li, X.; Li, J.; Zuo, L.; Zhu, L.; Shen, H.T. Domain adaptive remaining useful life prediction with transformer. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3521213. [CrossRef]
28. Fan, Y.; Nowaczyk, S.; Rögnvaldsson, T. Transfer learning for remaining useful life prediction based on consensus self-organizing models. *Reliab. Eng. Syst. Saf.* **2020**, *203*, 107098. [CrossRef]
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural. Inf. Process. Syst.* **2017**, *30*, 1–11.
30. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021.
31. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 6 October 2008.
32. Hu, C.; Youn, B.D.; Wang, P.; Yoon, J.T. Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliab. Eng. Syst. Saf.* **2012**, *103*, 120–135. [CrossRef]

33. Ramasso, E.; Saxena, A. Performance benchmarking and analysis of prognostic methods for CMAPSS datasets. *Int. J. Progn. Health. Manag.* **2014**, *5*, 1–15. [CrossRef]
34. Ma, J.; Su, H.; Zhao, W.-L.; Liu, B. Predicting the remaining useful life of an aircraft engine using a stacked sparse autoencoder with multilayer self-learning. *Complexity* **2018**, *2018*, 3813029. [CrossRef]
35. Ma, J.; Liu, X.; Zou, X.; Yue, M.; Shang, P.; Kang, L.; Jemei, S.; Lu, C.; Ding, Y.; Zerhouni, N.; et al. Degradation prognosis for proton exchange membrane fuel cell based on hybrid transfer learning and intercell differences. *ISA Trans.* **2021**, *113*, 149–165. [CrossRef] [PubMed]
36. Wu, J.; Hu, K.; Cheng, Y.; Zhu, H.; Shao, X.; Wang, Y. Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network. *ISA Trans.* **2020**, *97*, 241–250. [CrossRef] [PubMed]
37. Elsheikh, A.; Yacout, S.; Ouali, M.-S. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing* **2019**, *323*, 148–156. [CrossRef]
38. Kong, Z.; Cui, Y.; Xia, Z.; Lv, H. Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. *Appl. Sci.* **2019**, *9*, 4156. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.