Few-shot remaining useful life prognostics through auxiliary training with related data-set

Alaaeddine Chaoub, Alexandre Voisin, Christophe Cerisara, Benoît Iung

Few-shot remaining useful life prognostics through auxiliary training with related data-set

Alaaeddine Chaoub[1,2*], Alexandre Voisin[2,3], Christophe Cerisara[1,3], Benoît Iung[2,3]

[1*]LORIA, Université de lorraine, Nancy, 54000, France.
[2]CRAN, Université de lorraine, Nancy, 54000, France.
[3]CNRS, Nancy, 54000, France.

*Corresponding author(s). E-mail(s): alaaeddine.chaoub@loria.fr;
Contributing authors: alexandre.voisin@univ-lorraine.fr;
christophe.cerisara@loria.fr; benoit.iung@univ-lorraine.fr;

Abstract

Predicting the remaining useful life (RUL) of equipment can help organizations improve efficiency, reduce costs and enhance safety by enabling them to plan maintenance and repairs, and optimize their use of equipment. A major challenge in this field is the development of accurate deep learning models, particularly when data is limited to a few run to failure trajectories. Traditional methods, such as pre-training on a larger data set followed by fine-tuning on the main data, often fall short under these constraints. To address this challenge, we propose an auxiliary training approach that integrates auxiliary objectives from related but distinct data sets. This approach enriches the learning process, utilizing knowledge from a broader data range and acting as a regularization mechanism to improve generalization from limited data. The effectiveness of the proposed method is demonstrated by experiments on two well-known public data sets, CMAPSS and N-CMAPSS, across eight distinct settings, and is shown to outperform state-of-the-art approaches such as single-task learning and pre-training followed by fine-tuning.

Keywords: Prognostics and health management; Remaining useful life prediction; Deep learning; Auxiliary Training; Few-shot learning; Transfer learning

1 Introduction

Predictive maintenance is a key aspect of modern manufacturing and service industries, as it allows organizations to proactively identify and address potential issues with equipment or systems before they lead to costly failures. The task of predicting the remaining useful life of equipment or systems, represents one of the fundamental components of predictive maintenance, as it provides anticipation ability and allows organizations to plan for maintenance and make informed decisions about when to replace or upgrade equipment. Hence the need for accurate prognostic models to leverage predictive maintenance effectively.

Deep learning (DL) has emerged as a powerful tool for a variety of tasks in recent years, including prognostics (Zhang et al (2019b)). Deep learning algorithms are particularly well-suited to prognostics tasks because they can automatically learn features from raw data, allowing them to effectively process and analyze large amounts of data. In addition, they can learn non-linear relationships in data which allows them to model complex relationships between different features and can result in more accurate predictions. There exists a large number of works on building DL models for prognostics. Reviews on this subject can be found in the literature, including (Zhang et al (2019c); Wang et al (2020); Ochella et al (2022)). These literature reviews focus mainly on the diverse range of models employed in this field. Among the last works, one can find approaches based on recurrent neural networks (RNNs) and their variants (Zhang et al (2019a); Luo et al (2020); Chaoub et al (2021); Xiang et al (2020)), convolutional neural networks (CNNs) (Yang et al (2019); Palazuelos et al (2020)), transformers and attention-based approaches (Zhao et al (2023); Su et al (2021); Nie et al (2022)), and hybrid approaches combining the strengths of different techniques (Xia et al (2020); Al-Dulaimi et al (2019); Zhao et al (2022)). As also shown in the reviews, these approaches have been applied to predict the remaining useful life (RUL) of various systems including bearings, batteries, and aircraft turbofans, and have demonstrated good performance.

Deep learning algorithms are typically trained using large amounts of labeled data and can achieve impressive results on a wide range of tasks. However, building such models for prognostics can be challenging, particularly when data is limited on a specific use case. Indeed, in many use cases, data for specific equipment or systems may be scarce for several reasons. Industrial equipment are often maintained in a preventive manner which reduces the occurrence of failures (Chaoub et al (2022)). Additionally, even though there may be a large amount of data available from monitoring of the system, most of the data represents normal operation. The degraded and failures states of the industrial system, as they lead to unwanted product are most of the time largely under represented. Also, the labeling of the data, or identifying which data corresponds to failures, is a time-consuming manual process that requires reviewing maintenance reports and expert knowledge. Finally, it is often not possible to obtain "run-to-failure" data from a "real" process, where the system is allowed to run until it fails, because it's costly and time-consuming (Eker et al (2012)).

In the Prognostics and Health Management field, RUL prognostics face the additional challenge of the lack of pre-trained models that can serve as a foundation for new applications. Indeed, in areas like computer vision or natural language processing,

pre-trained models are the starting point of new applications. One reason for such a situation is the lack of large, publicly accessible prognostics data-sets. The available data consists of multiple small data-sets from various types of equipment, with diverse input features and run-to-failure trajectories of varying lengths and are research oriented rather than from the field. This lack of data availability creates a bottleneck for the progress of this domain.

To overcome this challenge of data scarcity, researchers have introduced the few-shot learning paradigm. In this paradigm, the goal is to learn a model that can perform well on a task using only a small amount of data. This is a challenging problem, especially when the task involves predicting the future behavior of a complex system, such as equipment remaining useful-life prediction. In the landscape of current research, a multitude of techniques such as episodic learning and transfer learning have been leveraged to overcome data scarcity. However, these methods often falter under certain conditions, struggling with a set of issues that we detail in section (2). Therefore, there is a clear need for a more versatile and general approach that can work across diverse situations.

We propose an approach based on auxiliary training (AT), AT is a learning paradigm focused on improving the generalization of a single primary task through the use of additional objectives. The role of auxiliary tasks is to assist the primary task, and at the time of testing, only the primary task is considered. we perform AT by adding auxiliary objectives that are based on related data sets, the goal is to allow the model to learn broader patterns from all tasks and apply this enriched knowledge on the primary one.

The main contributions of our work are enumerated as follows:

- **Introduction of AT in Few-shot Prognostic:** We propose the use of auxiliary training with other related data sets to improve learning and performance when dealing with a limited data set.
- **Task-specific projection layers:** Our methodology introduces task-specific projection layers designed to handle the complexities of diverse industrial data-sets.
- **Loss Weighting to Prevent Over-fitting:** We use a strategy for weighting the losses of main and auxiliary tasks, effectively mitigating the risk of over-fitting.
- **Empirical Evaluation:** Our methodology is rigorously evaluated on two distinguished benchmarks for RUL prediction tasks: CMAPSS (Saxena et al (2008)) and N-CMAPSS (Arias Chao et al (2021)). Notably, we assess its robustness under various few-shot scenarios and by varying the selected few samples. Our approach consistently showcases superior predictive performance when compared with existing baseline methods.

This article is structured as follows: section 2 provides a summary of related work, Section 3 introduces the proposed method of utilizing an auxiliary data-set for learning, Section 4 showcases the experimental setup and performance evaluation results for the proposed approach, and finally, Section 6 concludes the article.

2 Related work

Our approach of Auxiliary Training with related datasets goal is to address data scarcity in Remaining Useful Life (RUL) prediction. Initially, we refer to the principles of Few-Shot Learning (FSL), a technique for learning from few examples. We then examine the concepts underpinning Multi-Task Learning (MTL), a strategy bearing similarities and differences with our approach. Following that, we look into auxiliary training, our core method. Finally, we cite works concerning pre-training and fine-tuning, techniques utilized for comparative purposes in this paper.

Few-shot learning (FSL) aims to build a model using only a few labeled examples and has gained significant attention in recent years due to its potential to facilitate the adoption of deep learning in various domains. In this paper, we aim to utilize the FSL paradigm to address the problem of predicting remaining useful life (RUL) in prognostics. A comprehensive survey of these methods can be found in (Wang and Yao (2019)), which categorizes them into three categories: data augmentation approaches, model-based approaches, and algorithm-based approaches. Our works belong to this last category. Some research has also focused on using algorithm-based approaches, such as meta-learning techniques, for prognostics and diagnostics (Zhang et al (2020b); Hu et al (2022)). These techniques, inspired by (Finn et al (2017)), which introduces a model-agnostic meta-learning system. Such a system trains machine learning models across numerous tasks, thus equipping the model to learn new tasks more efficiently. This is often achieved through a process known as "episodic learning". While approaches like this have provided significant advancements in the domain of meta-learning, they come with their own set of challenges. Central to these is the underlying assumption that the data-sets across different tasks share the same input and output structure, given that a shared model architecture is used throughout the optimization process (Finn et al (2017)). This implies a prerequisite of uniformity in the structure and format of data across all tasks used in episodic training. However, this requirement may not be practical or achievable in certain industrial settings. In real-world scenarios, the configuration of sensors can exhibit a high degree of diversity between multiple equipment. Consequently, the data collected from these can vary dramatically in terms of the number of features and trajectory lengths. For instance, consider two public data-sets, C-MAPSS and N-CMAPSS (Saxena et al (2008); Arias Chao et al (2021)) used in this paper, both collected from similar types of equipment (Turbofan engines) but with notably different sensor configurations. Thus, the homogeneity of data structure and format is frequently absent in these situations, posing substantial challenges to their effective implementation.

Multi-task learning (MTL) is a technique that aims to extract shared feature representations or modules for related tasks. Unlike learning separate networks for each task independently, MTL allows for the extraction of correlated information from multiple tasks, leading to substantial enhancement in network performance for each individual task. One such approach is the multi-task deep neural network proposed by (Liu et al (2019b)), which combines MTL and language model pre-training to achieve state-of-the-art (SOTA) results in various natural language understanding tasks compared to the original single-task deep neural network setting. In applications for RUL prediction, both Yan et al (2023) and Wang et al (2022) advocate for MTL.

While one focuses on learning RUL prediction in conjunction with health state (HS) estimation, the latter highlights its simultaneous prediction with fault detection. In the context of few-shot learning, (Weller et al (2022)) demonstrated that MTL can outperform intermediate fine-tuning in natural language processing tasks when the target task is smaller than the supporting task. Furthermore, (Wang et al (2021)) provided theoretical and empirical evidence to support the claim that, under certain conditions, MTL can compete with SOTA gradient-based meta-learning algorithms in few-shot image classification benchmarks.

In contrast to MTL, **Auxiliary training** is focused on improving the generalization of a single primary task by utilizing additional tasks. The role of the auxiliary tasks is to assist the primary task, and at test time, only the primary task is considered. Auxiliary training methods can use simple auxiliary tasks that are based on the primary task data, or entirely different data-sets. For example, (Xu et al (2021)) trained a primary task of semantic segmentation alongside two auxiliary tasks, multi-label image classification, and saliency detection, using only the main task image-level ground-truth labels. In a similar use of primary task data, (Zhang et al (2020a)) employed augmented data as auxiliary tasks to enhance the accuracy and robustness of image classifiers, (Lin et al (2021)) proposed a fault classification-assisted RUL prediction network based on multi-task learning and auxiliary training, (Liu et al (2019a)) proposed a self-supervised approach based on meta-learning and auxiliary training to enhance the performances on multiple image classification benchmarks. Their approach involves training a multi-task network that performs the primary task and the auxiliary task in parallel, along with a label generation network that generates labels for the auxiliary task to improve primary task performance. In another approach to auxiliary training, (Watanabe et al (2022)) proposed using multiple data-sets as auxiliary training tasks for named entity recognition.

Pre-training and fine-tuning have been shown to be effective in a variety of application. It involves copying the weights from a pre-trained network and tuning all/-part of them on a downstream task. In the Prognostics and health management field, several studies (Deng et al (2021); Zhang et al (2018a); Yao et al (2023)) have demonstrated the usefulness of this approach, using it to improve the performance of a DL model on the prognosis and diagnosis of multiple industrial equipments. For instance, Couture and Lin (2022) employed a pre-trained Convolutional Neural Network (CNN) for feature extraction, opting to retrain just the final layer to align with their target outputs. In a similar vein, Behera and Misra (2023) utilized three different pre-trained CNNs in a multi-modal fashion to enhance the performance of RUL prediction. This is related to our work in terms of information transfer between support (auxiliary) data-set and the target (primary) one. However, the features learned during pre-training are not always tailored to the target task. Furthermore, tasks with insufficient training data often encounter rapid over-fitting during the fine-tuning process.

The characteristics of industrial data-sets and use cases, as well as the lack of a large pre-trained model for this kind of problems, motivated us to propose a solution under this paradigm. Our approach addresses these challenges and allows us to effectively use related data-sets to improve the performance of prognostics models.

3 Proposed approach

This paper presents a method for few-shot prognostics, which leverages auxiliary training to augment model performance, i.e. the use of related data as auxiliary task, alongside the limited number of samples from the main task. The aim is to achieve improved outcomes by leveraging the insights and patterns captured from the auxiliary data.

For simplicity and also to compare the approach with others, we consider that there is one single related auxiliary data-set. Let $D_s{=}(x^i_{aux}, y^i_{aux})_{i=1}^{I}$ and $D_{main}{=}(x^i_{main}, y^i_{main})_{i=1}^{I'}$ be the auxiliary and main sets, containing I and I' data samples respectively, where $I' < 20$. The choice of having less than 20 samples in the target set, aligns with the common practice in Few-Shot Learning. Furthermore, this choice is designed to simulate the conditions often encountered in many real-world scenarios for RUL prediction. Let f be a neural network model with parameters θ that takes as input a data sample x and outputs a predicted value y. We define the loss function $\mathcal{L}$ for auxiliary training as:

$$\mathcal{L} = \ell(y_{aux}, f(x_{aux})) + \ell(y_{main}, f(x_{main})) \tag{1}$$

where ℓ is the mean squared error (MSE). The first term in the loss function corresponds to the auxiliary task, and the second term corresponds to the main task. By jointly optimizing this loss function on both tasks, we aim to find the optimal model parameters θ^* for the target task.

When developing a model for auxiliary learning with distinct data sets, several problems can arise. One is that the auxiliary task may have different input characteristics to the main task, making it difficult to use the same model/parameters for both tasks. In addition, the operating conditions of different manufacturing processes or machines may change, leading to variations in data set distributions. These variations can hamper the development of a model capable of learning efficiently from both data sets, particularly if the differences are large. In addition, inconsistency in the length of run to failure trajectories (RTF) or outputs between data sets adds complexity to the creation of a model capable of exploiting the available knowledge.

One way to exploit them is through task-specific parts of the model that act as adapters to project the features or representations of different data-sets into the needed space. These modules can be used for the input, projecting different features from different tasks into the same space, thus solving the problem of varying input sizes and/or operating conditions between data-sets. In addition, they can also be used as task-specific prediction heads to address the issue of varying lengths of trajectories i.e., useful lifetimes of equipment.

Mathematically, we can represent the model with multiple branches as follows (Eq. 2):

$$\hat{y}_k = f_k^{out}(h^{sh}(f_k^{in}(x_k))) \tag{2}$$

where $k \in \{s, t\}$ represents the task index, For each task k: $(x_k, \hat{y}_k)$ stand for the input and output values, f_k^{in} and f_k^{out} represent the input and output adapters parameterized by ϕ and ψ, respectively. Furthermore, h^{sh} denotes the shared backbone

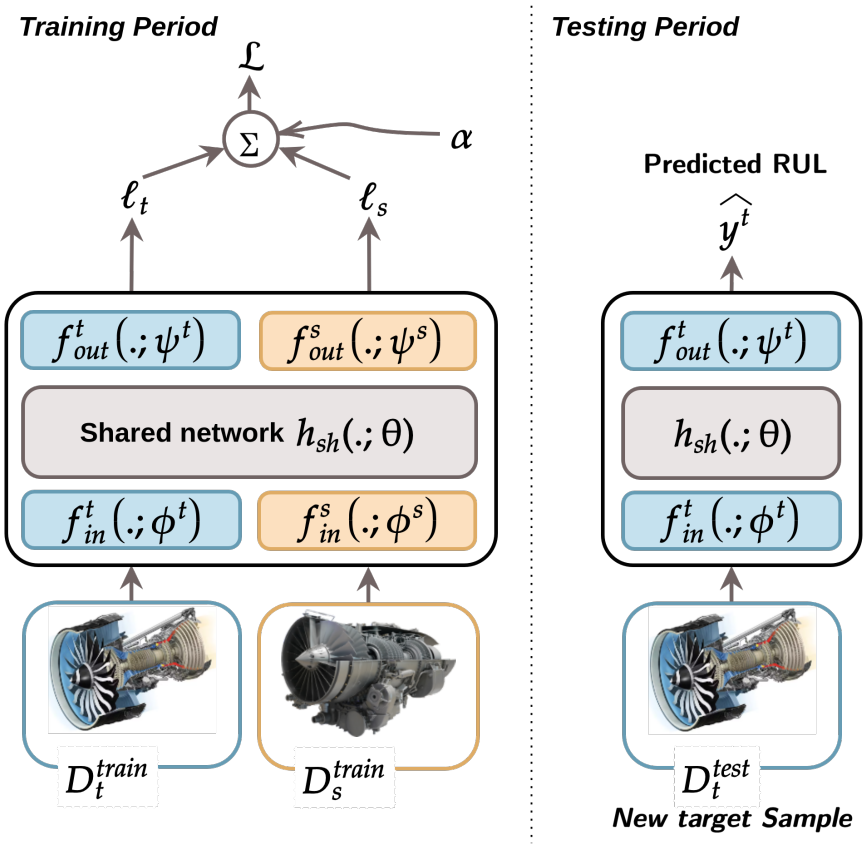

Fig. 1 Illustration of the proposed auxiliary training approach. (a) Training period: The data involved in the training include run to failure trajectories from the main set D^{train}_{main} and the trajectories from the auxiliary data-set D^{train}_{aux}. (i) the samples from each task are fed to their respective input adapter f^{in} to obtain similar dimensions and related features that can be used by the shared layers after. (ii) The features of both tasks are fed into the same layers h^{sh} where knowledge from both data-sets is learned. (iii) The output of the shared network for each task are fed into their main output adapter f^{out}. (vi) losses from both tasks are combined using a weighting parameter, α, to get a total loss $\mathcal{L}$ which is used for back-propagation. (b) Testing period: Prediction of the RUL of a new sample from main task is done by the main adapters, and the auxiliary adapters can be dropped to reduce model parameters.

parameterized by θ used for both tasks. A schematic illustration of this structure is depicted in Figure (1).

To solve the few shot learning problem, our approach consists of weighting the two losses of auxiliary and main tasks during training in a way that aims to prevent over-fitting on the few samples from main task. To do so, we propose a linear combination of the two losses through significantly reducing the weight of the main one, we add a

target loss weight parameter $\alpha << 1$ to the joint loss $\mathcal{L}$ as shown in Eq. 3 :

$$\mathcal{L} = \ell(y_{aux}, \hat{y}_{aux}) + \alpha * \ell(y_{main}, \hat{y}_{main}) \tag{3}$$

This approach might initially seem counter intuitive to our primary objective; nevertheless, empirical evidence from our experiments substantiates its validity. Specifically, by empirically demonstrating the advantages through our results, we show that reducing the weight of the main loss, denoted by a smaller alpha, the model becomes less prone to over-fitting to the samples of the main task. This allows the auxiliary task to serve effectively as a regularization mechanism during training. Our empirical findings contrast this method with the conventional two-step approach of pre-training and fine-tuning, which tends to make the model more susceptible to over-fitting on the main task samples, leading to potential loss of auxiliary information. Furthermore, our joint learning approach allows the shared network to learn and generalize mainly from the auxiliary task while the representations learned are biased/tailored towards the main task without over-fitting. Consequently, the main task adapters benefit by utilizing this shared knowledge, which guides the optimization process and improves overall performance.

Algorithm 1 Algorithm of the proposed auxiliary training approach

Input: main training data-set D_{main}, auxiliary training data-set D_{aux}
$D'_{\text{main}} \leftarrow$ duplicate D_{main} to match the number of samples in D_{aux}
$Model \leftarrow$ initialize model
for $i = 1$ to $EPOCH$ **do**
  for $j = 1$ to $ITERATION$ **do**
    $Batch_{\text{main}} \leftarrow \text{sample}(D'_{\text{main}}, BatchSize)$
    $Batch_{\text{aux}} \leftarrow \text{sample}(D_{\text{aux}}, BatchSize)$
    $Loss_{main}, Loss_{aux} \leftarrow \text{Compute}_{\text{Loss}}(Model, [Batch_{\text{main}}, Batch_{\text{aux}}])$
    $TotalLoss \leftarrow Loss_{aux} + \alpha \times Loss_{main}$
    $Model \leftarrow$ update model parameters
  end for
end for
$select\_best\_model_{\text{main}}(Model)$
Output: trained $Model$

The proposed auxiliary training Algorithm 1 begins by equalizing the number of samples between the main training data-set and the auxiliary data-set. To achieve this, we duplicate both data and corresponding labels in the target data-set, essentially creating replicated pairs of (sample, label), until its size matches that of the auxiliary data-set, given the auxiliary data-set is larger. It's important to note that the extent of this duplication is dependent on the specifics of the task at hand, as the number may vary. Then, the model is initialized, and the main and auxiliary data-sets are iterated over for a specified number of epochs. Within each epoch, the data is partitioned into batches and fed into the model. Next, the algorithm computes the main loss and

auxiliary loss for each batch. The two losses are then combined into a total loss using the hyper-parameter α, and the model's parameters are updated via backpropagation.

Once all the epochs have been completed, the $select\_best\_model_t$ function selects the model with the lowest error on the target task validation set. To accomplish this, the function evaluates the model's performance on the validation set after each epoch and saves the model's parameters if its performance is better than the previous best model.

4 Experimental setup

4.1 Model architecture

In the present study, we utilize a modified version of the MLP-LSTM-MLP architecture (Chaoub et al (2021)) as the foundation of our approach. The modification involves the addition of supplementary branches into the base architecture (Figure (1)) as discussed in section 3. This latter has a number of advantageous properties that make it well-suited to our needs. One key benefit is that it can be easily trained in an end-to-end manner, eliminating the need for feature engineering or selection. This architecture is composed of a first multi-layer perceptron (MLP) stage, which has the possibility to handle the raw inputs and learn good representations for each time frame, while the LSTM shall capture the dependencies through time, then a final regression head, composed of another MLP to predict the RUL from these temporally smoothed representations. This can simplify the training process, as analyzing both data-sets can be time-consuming. Additionally, the single-layer MLP adapters are relatively easy to adjust in terms of hyper-parameters compared to other types of layers, making this model easily adaptable to multiple data-sets.

4.2 Experimental data-sets

4.2.1 The C-MAPSS data-set

The C-MAPSS data-set (Saxena et al (2008)) is a widely used benchmark in the literature for evaluating approaches for remaining useful life (RUL) prediction of turbofan engines, i.e., the remaining time before a failure appears. It is divided into four sub-data-sets (FD001, FD002, FD003, FD004), with a varying number of operating conditions and fault modes (see Table (4.2.1)). Each sub-data-set is further divided into development and test subsets. The development set is composed of input time series which are assumed to go on until failure. In the test set, time series are truncated arbitrarily and the objective is to estimate the number of remaining operational cycles before the system failure occurs.

In each subset, engine number, operational cycle number, three operational settings, and 21 sensor measurements reflect turbofan engine degradation. The detailed description of sensor data can be found in Saxena et al (2008).

Table 1 Data-sets

Dataset		FD001	FD002	FD003	FD004	N-CMAPSS
Nb Train trajectories		100	260	100	249	66
Nb Test trajectories		100	259	100	249	43
Nb Operating conditions		1	6	1	6	-
Nb Fault modes		1	1	2	2	7
Trajectories length distribution	Max	360	378	525	543	100
	Mean	206	206	247	245	75
	Min	128	128	145	128	48

4.2.2 The N-CMAPSS data-set

The N-CMAPSS data-set (Arias Chao et al (2021)) represents further improvements and developments of the original CMAPSS data-set. This data was also synthetically generated using the CMAPSS engine simulator by using more models that simulate other important factors such as the atmosphere and the power management system. In addition, actual flight conditions recorded on board a commercial aircraft were used as input to the simulation model, providing a more realistic data-set with greater fidelity of the degradation and operating conditions.

The N-CMAPSS data-set comprises multivariate time series representing multiple sensor measurements across entire flights until engine failure (sensor description and analysis can be found in Arias Chao et al (2021) and in Chatterjee and Keprate (2021)). To facilitate analysis of flight data, we summarize each flight's data by calculating its mean, standard deviation, minimum, and maximum values. This uniform approach is applied across all methods using the same base architecture for fair comparison. Also, this data is divided into two subsets by default, namely development data and test data.

4.3 Baselines

We consider the following baselines to assess the proposed approach (See Figure (2)).

- **Single task training.** (**Single**) simply train the network on the **few samples** from the main data-set independently (Figure 2 (a)).
- **Pre-training + Fine-tuning.** (**PT-FT**) The second baseline is derived from the popular pre-training fine-tuning paradigm. It involves pre-training the model on the auxiliary data-set, and then copying all the model parameters to the fine-tuning stage on the main task, except the first MLP layers (adapter input) which will be initialized from scratch. This is done because the input features between the auxiliary and main data-sets could be different. During fine-tuning, the pre-trained model is re-trained on the small number of samples from the target set (Figure 2 (b)). This is one of the standard approaches in the literature used for transfer learning for RUL prediction Zhang et al (2018b); Yao et al (2023).

Table 2 Comparison of approaches in terms of their use of information from primary and auxiliary data sources. The quantity of primary data samples is variable and depends on the specific experiment conducted.

Approach	Number of primary samples	Number of auxiliary samples
Single Task learning	$\{3, 5, 10, 20\}$	0
PT-FT	$\{3, 5, 10, 20\}$	100
PT-R-in_out	$\{3, 5, 10, 20\}$	100
Our approach (AT)	$\{3, 5, 10, 20\}$	100

- **Pre-training + retraining input and output layers.** (**PT-R-in-out**) It involves pre-training the model on the auxiliary data-set, freezing the hidden layers parameters, then re-initializing the parameters of the first and last layers (the adapters), and finally training them on the main set (Figure 2 (c)).

All approaches except single task learning use the same quantity of samples (see Table (2)), we study how different ways of leveraging this knowledge affects the performances on the main task.

4.4 Settings

To verify the performance of the proposed approach, we conduct a series of experiments on the data-sets presented in Table (3). The auxiliary data-set used is a CMAPSS data-set (FD001) and the main task data can be either sub-sampling of FD004 or N-CMAPSS. To be in a FSL configuration, we randomly select a limited number (3,5,10,20) of Run-to-Failure trajectories from the selected main data-set. The goal is to obtain the best predictive performance over the main test set. We chose this configuration for the auxiliary and main sets because FD001 data-set is the simplest, thus evaluating the approach on the most challenging configurations possible using these data-sets (Table (3)).

Table 3 experimental configurations

Configuration	**Support set**	**Target set**	**Operating conditions**	**Fault modes**
C_{FD004}	FD001	FD004	1→6	1→2
$C_{N-CMAPSS}$		N-CMAPSS	1→ -	1→7

Due to the random selection of Run-to-Failure trajectories from the target set and also due to random initialization, the performance may vary across different training runs and sub-sampled D_{main}'s. In order to demonstrate the variability in our

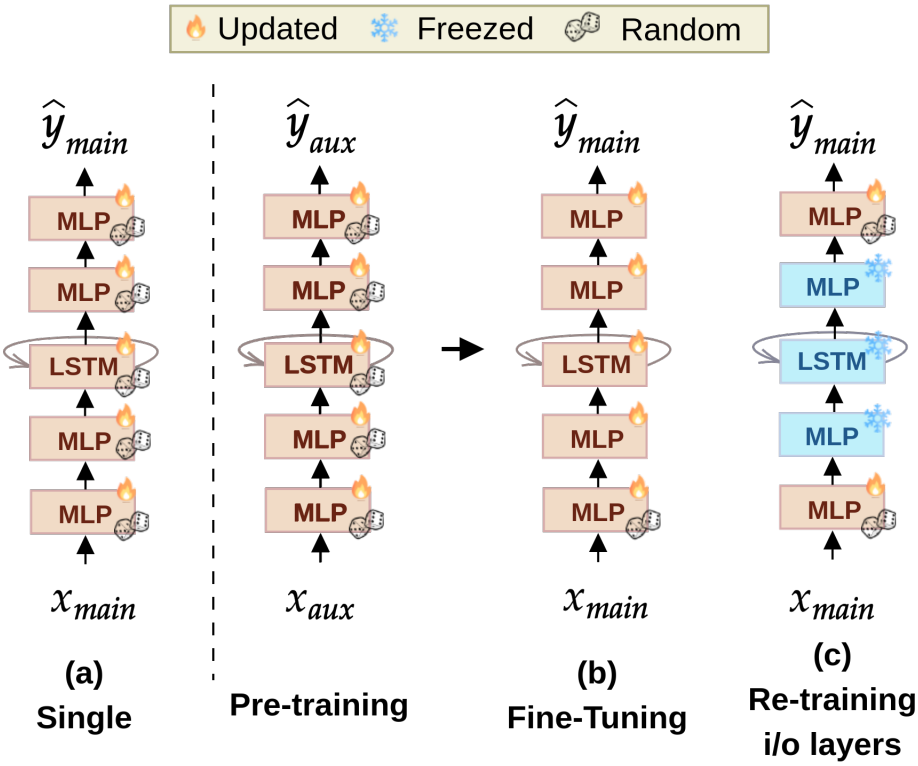

Fig. 2 baseline approaches used for comparison. Single task training (a), Pre-training followed by Fine-tuning (b), and Pre-training followed by Retraining Input and Output Layers (c). Random icon indicate layers initialized from scratch. In (b) and (c), input adapters are also initialized from scratch to accommodate different input structure.

results, we first conduct random sub-sampling five separate times. Each instance of sub-sampling is treated as a unique experiment. Within each experiment, we execute the process of training for each of our four distinct approaches (AT, Single, PT-FT and PT-R-in-out). These processes are run five times independently within each experiment. This procedure provides us with an array of results that reflects the range of potential outcomes from both the auxiliary training and baseline approaches. To quantify our findings, we report two key statistical measures: the average (mean) and the standard deviation. The average gives us a central tendency for each approach, while the standard deviation informs us about the dispersion or variability in our results.

4.5 Training details

To optimize the performance of the model, we conducted several experiments to adjust various hyper-parameters. The selection process was guided by the lowest root mean squared error (RMSE) obtained from multiple runs on the validation set of the main

Table 4 Hyper-parameters used in grid search for each approach

Hyper-parameter	**Approaches**	
	Single/PT-FT/PT-R-in-out	**AT**
learning rate	$\{ 5 \times 10^{-4}, 1 \times 10^{-4}, 1 \times 10^{-5} \}$	$\{ 5 \times 10^{-4}, 1 \times 10^{-4}, 1 \times 10^{-5} \}$
number of epochs	{ 30, 100, 300 }	{ 100, 300, 500 }
dropout	{ 0.0, 0.4, 0.6 }	{ 0.0 }
α	-	$\{ 1 \times 10^{-3}, 1 \times 10^{-5}, 1 \times 10^{-7} \}$

task, which represented 20% of the development subset while 80% were used for training. It's important to note that the train-validation split remained consistent across all runs, ensuring that our results were not influenced by changes in the data split.

We methodically fine-tuned a series of hyper-parameters using a grid search approach, which essentially performs a comprehensive brute force testing of different combinations. These hyper-parameters include the learning rates, dropouts, epochs, and target loss weight, and evaluated their impact on the model's performance. Refer to Table (4) for specifics on each approach.

We maintained a consistent model architecture throughout our testing, consisting of input adapters projecting features to a 10-dimensional space, a 3-layer MLP with 50 neurons per layer, a single-layer LSTM with 50 cells, two additional MLP layers with 50 and 10 neurons, and output adapters consisting of a one-layer MLP with one neuron. By keeping the same model architecture while testing multiple hyper-parameter values, we were able to single out the impact of each hyper-parameter on the model's performance. This methodology allowed us to accurately evaluate and identify the optimal combination of hyper-parameters to achieve the best results.

Given the nature of our study, the specific optimal hyper-parameters varied across the different sub-sampled target data. Consequently, a comparative study of hyper-parameters may not be meaningful, as they are contextually dependent on the specific instance of the training run and data subset. It is crucial to highlight that the selection of hyper-parameters is intrinsically tied to the unique conditions of each experiment, making a generalized comparison less informative than might be traditionally expected.

5 Results and discussion

This section presents the results of a comparative study of four different approaches for few-shot learning tasks: Single task training (Single), Pre-training + Fine-tuning (PT-FT), Pre-training + retraining input and output layers (PT-R-in-out), and Auxiliary Training (AT). The performance evaluation was carried out as follows: we selected various amounts of samples, ranging from 3 to 20, from the development sets of the target data. These selected samples were used to train the models using the approaches discussed. After the training phase, the approaches were then tested using the test data from the corresponding target data-set, and the results were collected.

Table (5) (top) and figure (3) show the results on the FD004 test set, demonstrating that the Auxiliary Training approach consistently outperforms the other methods

Table 5 Few-Shot Remaining Useful Life (RUL) Prediction on FD004 (top) and on N-CMAPSS (bottom): RMSE on Test Data shows that Auxiliary Training (AT) predominantly outperforms Single task training (Single), Pre-Training + Fine-Tuning (PT-FT), and Pre-Training + Retraining Input/Output Layers (PT-R-in-out). The standard deviation across multiple selections and runs is represented by ±.

Approach	Number of samples			
	3	5	10	20
Single	46.11 ± 5.78	38.36 ± 7.37	44.41 ± 5.89	36.01 ± 8.19
PT-FT	49.40 ± 7.39	41.23 ± 7.68	46.48 ± 6.53	39.27 ± 7.49
PT-R-in-out	48.07 ± 7.64	38.37 ± 7.90	44.15 ± 7.25	39.95 ± 8.38
AT	$\mathbf{36.58 \pm 5.93}$	$\mathbf{31.94 \pm 4.93}$	$\mathbf{27.92 \pm 3.73}$	$\mathbf{24.82 \pm 2.93}$

Approach	Number of samples			
	3	5	10	20
Single	$\mathbf{17.74 \pm 2.71}$	17.12 ± 4.99	13.94 ± 2.44	14.20 ± 1.85
PT-FT	19.67 ± 5.42	16.29 ± 4.80	14.72 ± 3.93	15.46 ± 2.29
PT-R-in-out	18.54 ± 6.58	$\mathbf{15.79 \pm 4.80}$	13.85 ± 2.96	14.47 ± 1.75
AT	18.64 ± 5.17	16.14 ± 6.10	$\mathbf{12.82 \pm 1.91}$	$\mathbf{12.80 \pm 1.54}$

across all shot settings. AT achieved the lowest error values, with mean RMSE of 36.58 ± 5.93 for 3-shot, 31.94 ± 4.93 for 5-shot, 27.92 ± 3.73 for 10-shot, and 24.82 ± 2.93 for 20-shot. Conversely, the Single task training approach consistently achieved the highest error values, with mean RMSE of 46.11 ± 5.78 for 3-shot, 38.36 ± 7.37 for 5-shot, 44.41 ± 5.89 for 10-shot, and 36.01 ± 8.19 for 20-shot. The PT-FT and PT-R-in-out approaches showed intermediate results, with mean RMSE values ranging from 39.27 ± 7.49 to 46.48 ± 6.53, but did not achieve significant improvements compared to IT. These findings suggest that fine-tuning and retraining the input/output layers alone may not be enough to enhance the model's performance in few-shot learning scenarios. On the other hand, leveraging auxiliary data during training has a positive impact on the model's ability to generalize to the target task.

To the best of our knowledge, no studies in the existing literature entirely replicate the constraints and settings under which our research is conducted. This circumstance makes direct comparisons somewhat difficult. However, there are works in this field that have utilized the same data-sets and related techniques, and these can serve as points of reference. For instance, Zhang et al (2018a) investigates transfer learning methodologies on the same data (CMAPSS). One relevant result from this study employed the FD002 subset for pre-training, which contains six operational conditions (OC), and select samples from FD004 for fine-tuning. Despite the fact that FD002 has a higher correlation with the target, given the greater number of OCs compared to FD001, the results yield an RMSE of 29.21 and 29.14 respectively when utilizing 10 and 20 samples from the target. These figures, while impressive, still don't surpass the

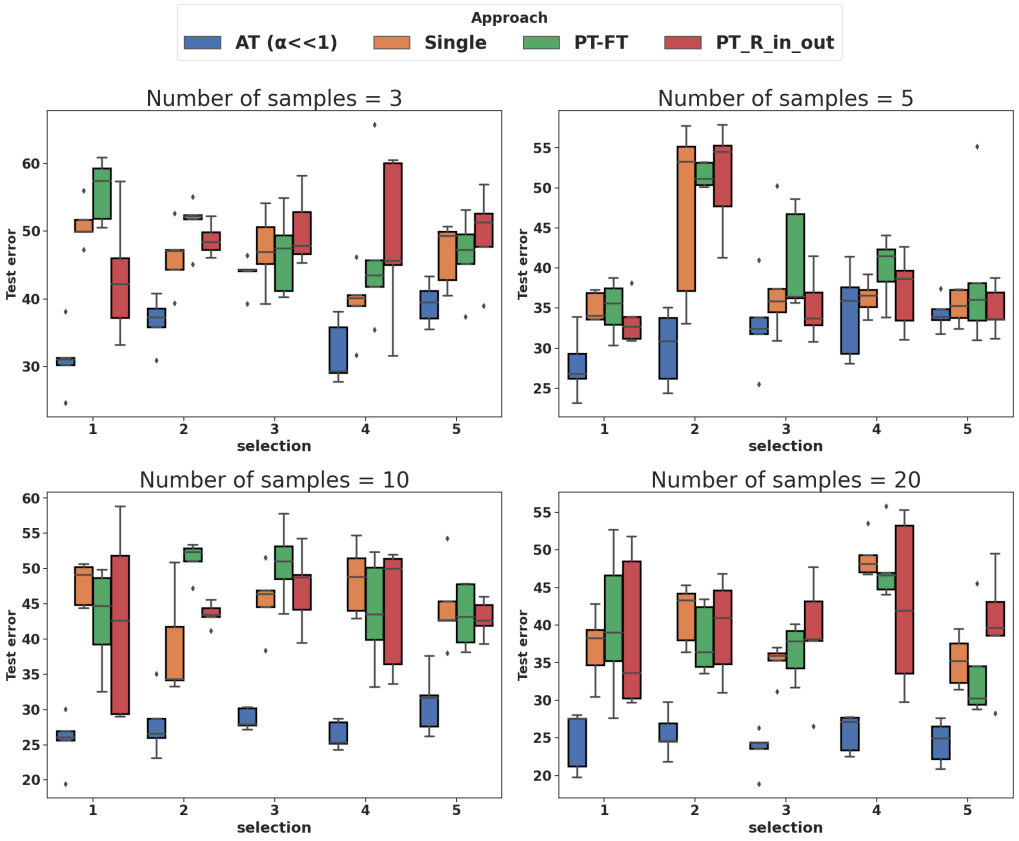

Fig. 3 Boxplots illustrating the RMSE distributions across the various approaches applied to the FD004 data-set. Each plot corresponds to a different subset size—3, 5, 10, and 20 samples—across five selections. The approaches are color-coded allowing for an evaluative assessment of each method's predictive performances.

performance exhibited by our proposed approach, even when the less correlated FD001 subset is used as the support set due to its singular OC and fault model. Moreover, another study worthy of mention is Ragab et al (2020), wherein the authors propose an adversarial transfer learning strategy to grapple with the problem of unlabeled target data. Their work, despite being tangentially related to ours, demonstrated an RMSE of 31.78, achieved by using the entirety of the unlabeled FD004 dataset. This performance, while commendable, also falls short compared to our model's results.

Table (5) (bottom) and Figure (4) presents the results on the N-CMAPSS test set. The results show that AT stands out as a promising approach, achieving the lowest RMSE values of 12.82 $\pm$ 1.91 and 12.80 $\pm$ 1.54 in the 10-shot and 20-shot settings, respectively. In addition, AT achieves competitive results in the 3-shot and 5-shot scenarios, with RMSE values of 18.64 $\pm$ 5.17 and 16.14 $\pm$ 6.10, respectively. In the

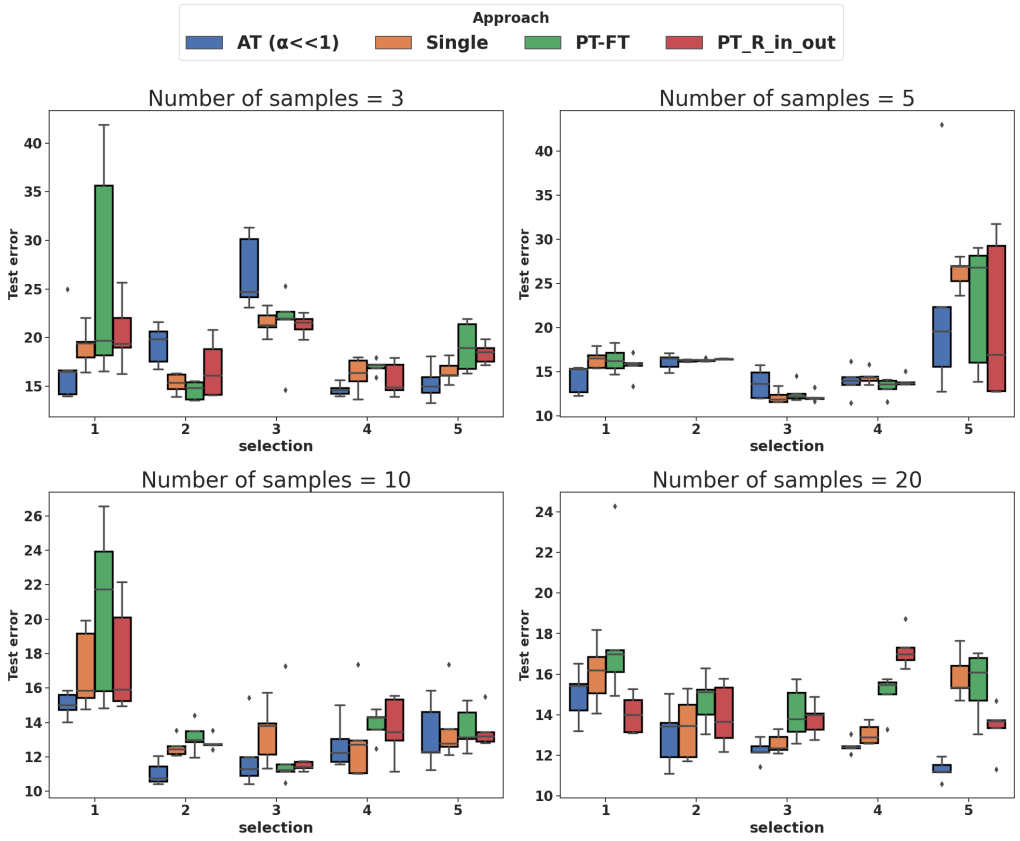

Fig. 4 Boxplots illustrating the RMSE distributions across the various approaches applied to the N-CMAPSS data-set. Each plot corresponds to a different subset size—3, 5, 10, and 20 samples—across five selections. The approaches are color-coded allowing for an evaluative assessment of each method's predictive performances.

5-shot scenario, AT's result is slightly worse than PT-R-in-out's, but still outperforms Single and PT-FT.

To better understand the sub-optimal results observed in the 3 and 5 shot scenarios, we analyzed the results for each selection individually, which are presented in Table 6. The table shows the mean RMSE values over 5 runs for each selection. Our analysis of the 3-shot scenario revealed that the AT approach outperformed the other methods in three of the selections, while PT-FT achieved the lowest mean error in the other two. In contrast, for the 5-shot scenario, AT outperformed the baseline approaches in two selections, while different approaches showed the best performance in the remaining three selections. This performance difference could be attributed to several factors. One possibility is that the N-CMAPSS (target) and FD001 (support) data-sets are less similar, indicating that the method may require more run-to-failure trajectories or relatively similar ones to align the representations between the two tasks. Another

Table 6 3 and 5-shot RUL prediction over multiple selection of the few samples from N-CMAPSS Data-set.

RMSE values					
Selection	1	2	3	4	5
Approach	3-shot				
Single	19.09	15.30	21.55	16.23	16.53
PT-FT	26.39	**14.58**	**21.30**	17.00	19.07
PT-R-in-out	20.47	16.77	21.33	15.70	18.40
AT	**17.2**	19.27	26.70	**14.65**	**15.32**
	5-shot				
Single	16.46	16.28	**12.19**	14.45	26.21
PT-FT	16.38	16.37	12.64	**13.28**	22.81
PT-R-in-out	15.63	16.46	12.20	13.95	**20.72**
AT	**14.23**	**16.16**	13.68	13.93	22.68

possibility is that the adapters used for the auxiliary task are too simple (one layer), which may result in representations that are less related to the main task under these settings.

Let's now delve into the impact of reducing the alpha parameter on predictive performance. The results presented in Figure (5) underscores the efficacy of this adjustment across the two data-sets, FD004 (top) and N-CMAPSS (bottum). When considering the impact of weighting the primary loss, a smaller alpha (represented by blue boxes), the RMSE is consistently lower or comparable to the approach where alpha equals 1 (orange boxes), with 80% cases showing better performance with the reduced alpha, underscoring the advantage of a more regularized model. This pattern suggests that a smaller alpha helps to mitigate over-fitting, enhancing the model's ability to generalize and thus reliably predict unseen data. The efficacy of adjusting the alpha parameter is evident across varied sample sizes, emphasizing its critical role in model tuning to achieve optimal balance and robustness in predictive modeling tasks.

The results indicate that Auxiliary Training (AT) is a promising approach for few-shot prognostics. This approach outperforms other methods across various shot settings on both FD004 and N-CMAPSS benchmarks. The absence of directly comparable studies from the literature is not an impediment, but rather a testament to the novelty of our work. We acknowledge these existing works as they explore the broader landscape of transfer learning in the context of the same data-set. Yet, our research uniquely navigates within a specific set of constraints, producing superior results in doing so. Overall, these findings suggest that the proposed approach improves the model's generalization ability by utilizing related auxiliary data.

6 Conclusion

This paper introduces a method that uses auxiliary training to enhance the Remaining Useful Life prediction. By utilizing auxiliary data-sets, while reducing the weight of

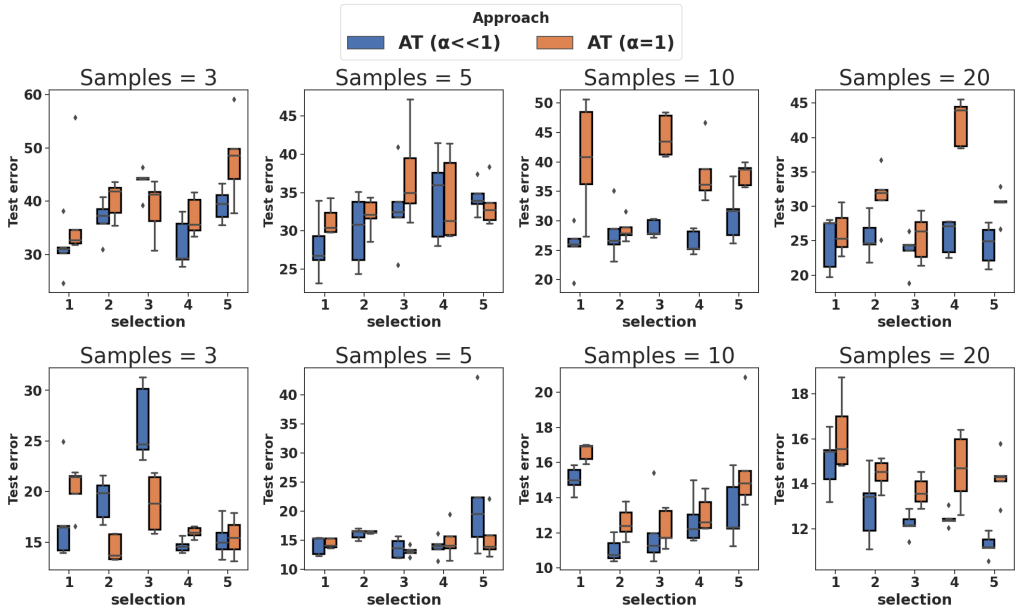

Fig. 5 Boxplots illustrating the RMSE distributions on FD004 (top) and N-CMAPSS (bottom). The results contrast using a main task loss weight of 1 with our proposed approach, where the weight is significantly reduced to a value much less than 1.

the main task loss, the proposed approach effectively extracts and uses knowledge from other data sets. The application of task-specific layers is a notable contribution of this paper. These layers project inputs/outputs from various tasks to pertinent spaces, demonstrating utility in dealing with dissimilar data-sets for auxiliary training. This is also beneficial for other approaches such as pre-training using multiple data sources.

In comparison to existing methodologies, Auxiliary Training demonstrates lower susceptibility to over-fitting under the few-shot learning paradigm. Empirical evidence in this paper shows that it outperforms other baseline approaches from existing literature, such as Pre-Training followed by Fine-Tuning, further solidifying its potential.

Further research would be beneficial to explore the adaptability of this approach across a broader range of data sets and equipment types. Investigating how different data impact RUL prediction will enhance our understanding of the method's real-world applicability. Additionally, developing a methodology for the intelligent and automatic selection of task weights during training could further refine the model's performance. Our ongoing objective is to develop a reliable and adaptable method, capable of accurately predicting the RUL for a wide array of equipment and systems within the few-shot learning paradigm.

7 Data Availability Statement

The data sets used in this study are publicly available in the Prognostics Data Repository: PCOE data set repository

Specifically, this research utilizes:

- The C-MAPSS data set is accessible at : CMAPSS Jet Engine Simulated Data
- The N-CMAPSS data set is available for download at : Turbofan Engine Degradation Data Set 2

8 Ethics declarations

8.1 Conflict of Interest Statement

The authors declare that there is no conflict of interest.

References

Al-Dulaimi A, Zabihi S, Asif A, et al (2019) A multimodal and hybrid deep neural network model for remaining useful life estimation. Computers in Industry 108:186–196. https://doi.org/10.1016/j.compind.2019.02.004

Arias Chao M, Kulkarni C, Goebel K, et al (2021) Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. Data 6(1). https://doi.org/10.3390/data6010005, [dataset]

Behera S, Misra R (2023) A multi-model data-fusion based deep transfer learning for improved remaining useful life estimation for iiot based systems. Engineering Applications of Artificial Intelligence 119:105712

Chaoub A, Voisin A, Cerisara C, et al (2021) Learning representations with end-to-end models for improved remaining useful life prognostics. CoRR abs/2104.05049. 2104.05049

Chaoub A, Cerisara C, Voisin A, et al (2022) Deep learning representation pre-training for industry 4.0. In: PHM Society European Conference, pp 571–573

Chatterjee S, Keprate A (2021) Exploratory data analysis of the n-cmapss dataset for prognostics. In: 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp 1114–1121, https://doi.org/10.1109/IEEM50564.2021.9673064

Couture J, Lin X (2022) Image-and health indicator-based transfer learning hybridization for battery rul prediction. Engineering Applications of Artificial Intelligence 114:105120

Deng Y, Huang D, Du S, et al (2021) A double-layer attention based adversarial network for partial transfer learning in machinery fault diagnosis. Computers in

Industry 127:103399. https://doi.org/10.1016/j.compind.2021.103399

Eker OF, Camci F, Jennions IK (2012) Major challenges in prognostics: Study on benchmarking prognostics datasets. In: PHM Society European Conference

Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. CoRR abs/1703.03400. URL http://arxiv.org/abs/1703.03400, 1703.03400

Hu Y, Liu R, Li X, et al (2022) Task-sequencing meta learning for intelligent few-shot fault diagnosis with limited data. IEEE Transactions on Industrial Informatics 18(6):3894–3904. https://doi.org/10.1109/TII.2021.3112504

Lin T, Wang H, Song L, et al (2021) Multi-task learning based classified-assisted prediction network for remaining useful life prediction. In: 2021 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD), pp 1–6, https://doi.org/10.1109/ICSMD53520.2021.9670776

Liu S, Davison AJ, Johns E (2019a) Self-supervised generalisation with meta auxiliary learning. CoRR abs/1901.08933. 1901.08933

Liu X, He P, Chen W, et al (2019b) Multi-task deep neural networks for natural language understanding. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp 4487–4496, https://doi.org/10.18653/v1/P19-1441

Luo Q, Chang Y, Chen J, et al (2020) Multiple degradation mode analysis via gated recurrent unit mode recognizer and life predictors for complex equipment. Computers in Industry 123:103332. https://doi.org/10.1016/j.compind.2020.103332

Nie L, Xu S, Zhang L, et al (2022) Remaining useful life prediction of aeroengines based on multi-head attention mechanism. Machines 10(7). https://doi.org/10.3390/machines10070552

Ochella S, Shafiee M, Dinmohammadi F (2022) Artificial intelligence in prognostics and health management of engineering systems. Engineering Applications of Artificial Intelligence 108:104552

Palazuelos ART, Droguett EL, Pascual R (2020) A novel deep capsule neural network for remaining useful life estimation. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 234(1):151–167. https://doi.org/10.1177/1748006X19866546

Ragab M, Chen Z, Wu M, et al (2020) Adversarial transfer learning for machine remaining useful life prediction. In: 2020 IEEE international conference on prognostics and health management (ICPHM), IEEE, pp 1–7

Saxena A, Goebel K, Simon D, et al (2008) Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 International Conference on Prognostics and Health Management, pp 1–9, https://doi.org/10.1109/PHM.2008.4711414, [dataset]

Su X, Liu H, Tao L, et al (2021) An end-to-end framework for remaining useful life prediction of rolling bearing based on feature pre-extraction mechanism and deep adaptive transformer model. Computers & Industrial Engineering 161:107531. https://doi.org/10.1016/j.cie.2021.107531

Wang H, Zhao H, Li B (2021) Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation. CoRR abs/2106.09017. 2106.09017

Wang H, Lin T, Cui L, et al (2022) Multitask learning-based self-attention encoding atrous convolutional neural network for remaining useful life prediction. IEEE Transactions on Instrumentation and Measurement 71:1–8. https://doi.org/10.1109/TIM.2022.3185312

Wang Y, Yao Q (2019) Few-shot learning: A survey. CoRR abs/1904.05046. 1904.05046

Wang Y, Zhao Y, Addepalli S (2020) Remaining useful life prediction using deep learning approaches: A review. Procedia Manufacturing 49:81–88. https://doi.org/10.1016/j.promfg.2020.06.015, proceedings of the 8th International Conference on Through-Life Engineering Services – TESConf 2019

Watanabe T, Ichikawa T, Tamura A, et al (2022) Auxiliary learning for named entity recognition with multiple auxiliary biomedical training data. In: Proceedings of the 21st Workshop on Biomedical Language Processing. Association for Computational Linguistics, Dublin, Ireland, pp 130–139, https://doi.org/10.18653/v1/2022.bionlp-1.13

Weller O, Seppi K, Gardner M (2022) When to use multi-task learning vs intermediate fine-tuning for pre-trained encoder transfer learning. arXiv preprint arXiv:220508124

Xia T, Song Y, Zheng Y, et al (2020) An ensemble framework based on convolutional bi-directional lstm with multiple time windows for remaining useful life estimation. Computers in Industry 115:103182. https://doi.org/10.1016/j.compind.2019.103182

Xiang S, Qin Y, Zhu C, et al (2020) Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction. Engineering Applications of Artificial Intelligence 91:103587

Xu L, Ouyang W, Bennamoun M, et al (2021) Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. CoRR abs/2107.11787. 2107.11787

Yan J, He Z, He S (2023) Multitask learning of health state assessment and remaining useful life prediction for sensor-equipped machines. Reliability Engineering & System Safety 234:109141. https://doi.org/10.1016/j.ress.2023.109141

Yang B, Liu R, Zio E (2019) Remaining useful life prediction based on a double-convolutional neural network architecture. IEEE Transactions on Industrial Electronics 66(12):9521–9530. https://doi.org/10.1109/TIE.2019.2924605

Yao S, Kang Q, Zhou M, et al (2023) A survey of transfer learning for machinery diagnostics and prognostics. Artificial Intelligence Review 56(4):2871–2922. https://doi.org/10.1007/s10462-022-10230-4

Zhang A, Wang H, Li S, et al (2018a) Transfer learning with deep recurrent neural networks for remaining useful life estimation. Applied Sciences 8(12). https://doi.org/10.3390/app8122416

Zhang A, Wang H, Li S, et al (2018b) Transfer learning with deep recurrent neural networks for remaining useful life estimation. Applied Sciences 8(12):2416

Zhang B, Zhang S, Li W (2019a) Bearing performance degradation assessment using long short-term memory recurrent network. Computers in Industry 106:14–29. https://doi.org/10.1016/j.compind.2018.12.016

Zhang L, Lin J, Liu B, et al (2019b) A review on deep learning applications in prognostics and health management. IEEE Access 7:162415–162438. https://doi.org/10.1109/ACCESS.2019.2950985

Zhang L, Lin J, Liu B, et al (2019c) A review on deep learning applications in prognostics and health management. IEEE Access 7:162415–162438. https://doi.org/10.1109/ACCESS.2019.2950985

Zhang L, Yu M, Chen T, et al (2020a) Auxiliary training: Towards accurate and robust models. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 369–378, https://doi.org/10.1109/CVPR42600.2020.00045

Zhang S, Ye F, Wang B, et al (2020b) Few-shot bearing anomaly detection based on model-agnostic meta-learning. CoRR abs/2007.12851. 2007.12851

Zhao C, Huang X, Li Y, et al (2022) A novel remaining useful life prediction method based on gated attention mechanism capsule neural network. Measurement 189:110637. https://doi.org/10.1016/j.measurement.2021.110637

Zhao K, Jia Z, Jia F, et al (2023) Multi-scale integrated deep self-attention network for predicting remaining useful life of aero-engine. Engineering Applications of Artificial Intelligence 120:105860