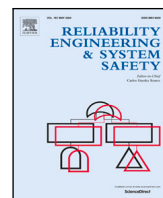




Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Domain adaptation via alignment of operation profile for Remaining Useful Lifetime prediction

Ismail Nejjar^a, Fabian Geissmann^b, Mengjie Zhao^a, Cees Taal^c, Olga Fink^{a,*}^a Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland^b Swiss Federal Institute of Technology Zürich (ETH Zürich), Switzerland^c SKF Group, Netherlands

ARTICLE INFO

Keywords:

Remaining Useful Lifetime
Deep learning
Domain adaptation
Prognostics

ABSTRACT

Effective Prognostics and Health Management (PHM) relies on accurate prediction of the Remaining Useful Life (RUL). Data-driven RUL prediction techniques rely heavily on the representativeness of the available time-to-failure trajectories. Therefore, these methods may not perform well when applied to data from new units of a fleet that follow different operating conditions than those they were trained on. This is also known as domain shifts. Domain adaptation (DA) methods aim to address the domain shift problem by extracting domain invariant features. However, DA methods do not distinguish between the different phases of operation, such as steady states or transient phases. This can result in misalignment due to under- or over-representation of different operation phases. This paper proposes two novel DA approaches for RUL prediction based on an adversarial domain adaptation framework that considers the different phases of the operation profiles separately. The proposed methodologies align the marginal distributions of each phase of the operation profile in the source domain with its counterpart in the target domain. The effectiveness of the proposed methods is evaluated using the New Commercial Modular Aero-Propulsion System (N-CMAPSS) dataset, where sub-fleets of turbofan engines operating in one of the three different flight classes (short, medium, and long) are treated as separate domains. The experimental results show that the proposed methods improve the accuracy of RUL predictions compared to current state-of-the-art DA methods.

1. Introduction

Remaining Useful Life (RUL) prediction, also referred to as prognostics, is a key element of Prognostics and Health Management (PHM). Prognostics aim to estimate how long a system can maintain its specified functionality before reaching its end of life [1]. Accurate RUL predictions enable scheduling maintenance from the perspective of operational and resource availability, avoid costly downtime, and prevent critical failures. In recent years, technological advancements and decreasing sensor costs have led to an increase in the amount of data collected from industrial assets [2]. This increased data availability allows taking advantage of advanced data-driven approaches such as Deep Neural Networks (DNNs), which have recently shown great potential in PHM applications such as fault detection and diagnosis [3].

Traditional supervised machine learning methods assume that, on the one hand, a representative labeled dataset is available and, on the other hand, that the training and test datasets stem from similar distributions. Unfortunately, in real-world scenarios, both assumptions do not hold for many applications [4,5]. While these challenges are

also present in fault detection and diagnostics, they are particularly pronounced in prognostics. On the one hand, collecting a sufficiently representative dataset of run-time failure trajectories is often impossible due to the potentially catastrophic consequences of such an event in reality. On the other hand, deep learning models still face the challenge of domain shift due to the wide variety of operating conditions and limited training samples. These two challenges considerably impact the generalization ability of a model trained on a specific dataset and applied to a different but related dataset.

Unsupervised Domain Adaptation (UDA) has been shown to be effective in addressing the domain gap problem by adapting a model that has been trained on one domain (the source) to a different, unlabeled domain (the target). Previous research on domain adaptation has largely focused on classification tasks [6], and therefore, UDA has been extensively applied to fault diagnostics tasks [7,8]. Recently, several research studies have also proposed applying UDA to prognostics tasks. These approaches typically involve aligning the feature distributions between the source and target domains through adversarial training [9]

* Corresponding author.

E-mail address: olga.fink@epfl.ch (O. Fink).<https://doi.org/10.1016/j.ress.2023.109718>

Received 13 February 2023; Received in revised form 28 August 2023; Accepted 6 October 2023

Available online 27 October 2023

0951-8320/© 2023 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

or reducing the disparity between feature distributions [10,11]. Although adversarial training has been widely used in prognostics, it has not taken into account the distinct phases of cyclical operation profiles, such as those found in flights, for example, take-off, cruise and landing phases. A distinct marginal distribution defines each phase of the operation profile, and adversarial adaptation methods may fail to capture such multi-modal structures in condition monitoring data. Therefore, the alignment of the source and target domains using DANN [12] assumes that the marginal distributions of each phase of the operation profile are also aligned. However, it may lead to misalignment of the phase of the operation profile due to under- or over-representation since the different operation phases may have different durations in the different operation profiles, for example.

To improve the ability to adapt between different domains, we propose utilizing the common sequence of operation profile phases which each system undergoes. We take advantage of the prior knowledge of the operating cycles since the phases always occur in the same order, share specific characteristics, and their existence and sequence are invariant across all domains. For instance, in the aviation domain, a flight can be divided into several phases, that a flight follows, such as take-off, cruise, and descent. The flight phases differ in terms of, e.g., altitude, duration, speed, etc., and have distinct marginal distributions. Within the same flight class, e.g., the cruise phase will have similar characteristics. However, between different flight classes, the characteristics of the flight phases will be more different. Each phase has a different impact on engine stress and degradation and consequently also on the RUL. By incorporating the operation profile into domain adaptation, we aim to improve the target RUL prediction in the case of a domain shift.

In this paper, we consider the scenario where time-to-failure trajectories for complex systems such as aircraft jet engines and power plants are available within one fleet but not for units from a different fleet. To address this challenge, we define domains as sub-fleets of these systems that are operated under similar conditions. Our focus is on the domain adaptation problem, where we aim to transfer knowledge from a labeled source domain to an unlabeled target domain. The units within the same fleet may have a heterogeneous composition of operating conditions during their respective missions. These operating conditions can include differences in operating speed, machine load, working temperature, and environmental noise. These differences can lead to diverse distributions of marginal characteristics, making it difficult to adapt models to new domains. Previous research has shown that when the feature distribution is multimodal, adapting only the feature representation can be challenging for adversarial networks [13,14].

To tackle this issue, we propose to replace the single-domain discriminator from DANN [15] method with a discriminator for each phase of the operation profile. This approach is inspired by the success of multi-task learning [16], and auxiliary tasks [17] in facilitating adaptation between domains. The proposed methodologies align the marginal distributions of each phase of the operation profile in the source domain with its counterpart in the target domain. This work investigates in depth the learning of invariant features by adversarial learning for the alignment of sub-fleets (units operated in a similar way), taking into account their operation profiles. Two methods are proposed to deal with the above scenarios: (1) each part of the operational profiles can be assigned to a specific regime, or (2) the parts of the operational profile are smoothly assigned to all regimes with a defined probability, which is particularly useful for transition phases.

We evaluate the performance of the proposed algorithm on the NASA New Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS) turbofan degradation dataset [18]. The dataset contains three flight classes: short (S), medium (M), and long (L), with a total of 15 units, with five units in each flight class. The proposed domain adaptation methods were evaluated on three adaptation tasks of increasing difficulty. The models were applied between sub-fleets, each consisting of five units, flying in a specific flight class. The adaptation scenarios

consist of transferring from medium to long flights ($M \rightarrow L$), from short to medium flights ($S \rightarrow M$), and finally from short to long flights ($S \rightarrow L$). The adaptation scenarios always involve the adaptation from a flight class where run-to-failure trajectories are available to a flight class without any labels. The results of the three tasks demonstrate that the proposed algorithm, which takes into account the operation profile, can improve over the traditional DANN [15] and outperform other comparative benchmark methods.

2. Related work

2.1. Unsupervised domain adaptation

Unsupervised domain adaptation (UDA) [19] aims to improve the performance of a model on a target domain in the presence of a domain shift between the labeled source domain and unlabeled target domain. Several UDA methods have been proposed to align the feature distributions between the two domains during training using either discrepancy losses or adversarial training.

Prior works mainly relied on distribution alignment. While Deep Adaptation Networks (DAN) [20] minimize the Maximum Mean Discrepancy (MMD) over domain-specific layers, Joint Adaptation networks [21] align the joint distributions of domain-specific layers across different domains based on a Joint Maximum Mean Discrepancy (JMMD). In contrast, adversarial-based methods strive to obtain domain-invariant representations through adversarial training. For instance, Domain-Adversarial Training of Neural Networks (DANN) [15] aims to create a domain-invariant representation by using a domain discriminator. To achieve this, the model is trained with a reversal gradient layer to make the feature space indistinguishable for different domains. Maximum Classification Discrepancy (MCD) [22] is another adversarial-based method that aims to reduce domain divergence and align the distribution of a target domain by considering task-specific decision boundaries through the use of task-specific classifiers in an adversarial training approach. Other approaches focused on the distribution of features for both domains across the batch normalization layers. Such approaches include Adaptive Batch Normalization (AdaBN) [23] and Automatic Domain Alignment Layers (AutoDial) [24], which aligns the distributions via modified batch normalization layers.

Most DA approaches have been developed for classification tasks, and domain adaptation for regression has received little attention. While classification approaches aim at generating decision boundaries to separate data into different classes, regression methods, on the other hand, aim at predicting continuous numerical outputs with a defined ordinal relationship. Early works on domain adaptation for regression problems introduced a weighting scheme that assigns a weight according to the similarity between the training and test samples [25,26]. Recent works in the deep representation learning regime proposed different strategies to close the domain gap. For example, Representation Subspace Distance (RSD) [27] explores the Riemannian geometry of the Grassmann manifold to reduce the domain gap using the orthogonal bases representation of the subspace as opposed to instance representations. More recently, taking inspiration from the closed-form solution of least-square problems, DARE-GRAM [28] proposed a strategy centered on aligning the inverse Gram matrices of source and target features, with the aim of mitigating this issue.

Many of the mentioned methods for classification can be naturally extended to regression problems. Nevertheless, when dealing with complicated regression problems, there are still no straightforward solutions to the fundamental problem of unsupervised domain adaptation for regression.

2.2. Domain adaptation applied to PHM

In recent years, domain adaptation has been increasingly applied to PHM applications [29], particularly in the area of fault diagnostics

classification. Three main domain gaps exist in the PHM context: between varying operating conditions [30,31], between different units of a fleet [32], and between simulations and real data [33–36].

Most research in this area has focused on the transfer between discrete operating conditions. Classic domain adaptation methods, such as distribution alignment [37,38] and adversarial alignment [39,40], have been widely used in this context and have been found to be beneficial [8]. Proposed approaches include discrepancy-based domain adaptation, using various criteria such as Maximum Classification Discrepancy (MCD) [40] and Maximum Mean Discrepancy (MMD) [41–43]. Deep domain adversarial frameworks such as DANN [44–46] have been developed and applied for fault diagnosis of machines under varying working conditions. Some works have sought to improve upon adversarial alignment through the use of conditional discriminators [47, 48].

Remaining useful life predictions have received increasing attention. [49] proposes a novel RUL prediction approach that combines knowledge of the sensor relationship and deep learning models. Recently, [50] proposed an adaptive framework that can select the correct trained model according to failure modes. DSCN-DTAM was proposed in [51] and employs multiple regularization strategies and a double transferable attention mechanism to improve feature transferability in the presence of shift. Furthermore, transfer learning has also been recently increasingly applied to RUL prediction to increase the transferability of the learned feature, [51–53]. For example, a transfer learning algorithm based on Bidirectional Long Short-Term Memory (BLSTM) recurrent neural networks was proposed in [52] for RUL estimation. The algorithm fine-tunes a model trained on a large amount of data from a source task with a small amount of data from a target task, usually a different but related task. The results showed that transfer learning is effective, except when transferring from a dataset of multiple operating conditions to a dataset of a single operating condition, leading to negative transfer learning. In addition to cases where labels are available for the target, also unsupervised domain adaptation approaches have been applied for RUL prediction. Different adversarial methods have been proposed to align the source and target domains and improve the performance of RUL prediction [9]. For example, [9] used adversarial training to extract domain-invariant features from a Long Short-Term Memory (LSTM) model. Recent work [54] introduced a novel Wasserstein distance-based weighted domain adversarial neural network (WD-WDANN) for RUL prediction under different operating conditions by measuring the similarity of source samples to the target domain to determine the sample quality. Multiple-kernel maximum mean discrepancies (MK-MMD) were proposed and shown to be more robust than single-kernel methods [11,55] and help to minimize the distribution discrepancy between different failure behaviors in the feature space.

Previous approaches proposed for RUL prediction in the context of PHM have mainly focused on applying domain adaptation on the entire operating cycle without considering the distinct phases of the operation profile for each domain. Such an approach aligning source and target domains using domain invariant feature learning implicitly assumes that the marginal distributions of different operating conditions are also aligned. We propose replacing the single-domain discriminator with multiple-domain discriminators for each operating condition. In this way, we ensure that the different marginal distributions are aligned with their counterparts in the other domain.

3. Methodology

3.1. Terminology

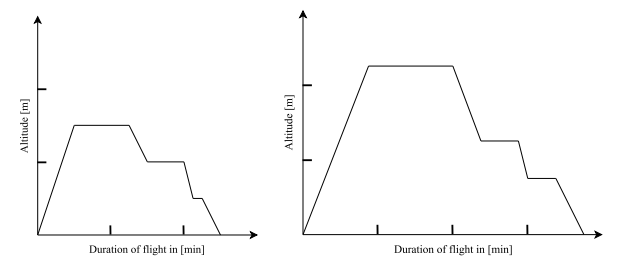
Before delving into the details of the proposed algorithm, it is important to define some key terms that will be used throughout the

text. In this study, we make use of the terms *operation profile* and *operation condition*.

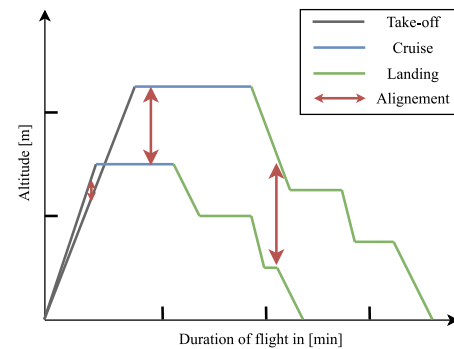
An operation condition: refers to the specific state or set of conditions under which a system is operating and can be thought of as different domains.

An operation profile: is a detailed and specific way of describing how industrial assets are operated and are controlled under different conditions while sharing similar characteristics in the sequences in which the operating conditions occur. An operation profile is typically composed of **distinct and discrete phases**, which are characterized by definite control and condition parameter ranges defining the states of the system that are retained for specific periods of time and are characterized by a unique marginal distribution of its characteristics.

Together, the operation profile (with different phases) and operation conditions provide a comprehensive understanding of how industrial assets function and perform over time.



(a) Operating Condition: Short flight (b) Operating Condition: Long flight Altitude profile



(c) Altitude level at the different phases of the flight Take-off, cruise, and Landing.

Fig. 1. The Figs. 1(a) and 1(b) show two flights that were operated differently and have different profiles. Fig. 1(c) illustrates the presence of similar phases in both flights, which were used for the alignment process.

3.2. General problem definition

In this paper, we study the unsupervised domain adaptation problem in the context of regression. During training, we are given access to a set of n_s labeled samples from a source domain $D_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ and n_t unlabeled samples from a target domain $D_t = \{(x_t^i)\}_{i=1}^{n_t}$.

Each observation $x^i \in \mathcal{X} \subset \mathbb{R}^{K \times T}$ is a multivariate time sequence of K raw measurements of length T , where T is the length of the observation window. Finally, given a sample i from the set of all trajectories in the source domain, $y^i \in \mathcal{Y} \subset \mathbb{R}$ denotes the Remaining Useful Lifetime (RUL) value of the observed sequence x^i . Labels of target data are not available during training. The goal of the task is to train a model using labeled set D_s and unlabeled set D_t multivariate time-series data of aircraft engines operated under different operating conditions.

3.3. Operation profile

Industrial assets may operate under different operating conditions but share similar phases of the operation profile. For example, in aviation, a flight is usually composed of different phases, such as take-off, cruise, and descent, requiring different thrust levels to propel the aircraft forward (Fig. 1), thus applying different types and levels of stress on the jet engine.

More generally, the operation profile can be determined by discretizing a particular measured or control parameter based on domain knowledge or using an unsupervised clustering algorithm on the control parameters or multivariate observations. The final objective is to obtain a set of n_p discrete and common phases of the operation profile for each of the two domains.

This research assumes that a finite number of discrete operation profile labels can either be provided based on domain knowledge or identified using the measured or control parameters. We argue that taking into account the different phases of the operation profile can improve the alignment process.

The following section presents how these operation profile labels can be exploited to extend upon the existing DANN methods to support the alignment process.

3.4. Domain-adversarial neural networks

As the operating conditions of the source and target sub-fleets differ, a model trained on the source data would be able to easily distinguish the feature vectors of the target domain from the source domain due to the domain shift, leading to poor performance when applied to the target domain. Adversarial distribution alignment methods aim to address this problem by ensuring that the feature extractor is unbiased with respect to the characteristics of the source and target domains.

DANN [12] is a broadly applied domain adaptation method that aligns the distributions of source and target features by adding a domain discriminator and introducing adversarial training.

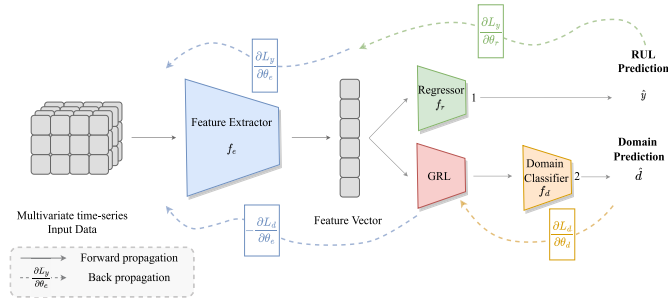


Fig. 2. Standard architecture of DANN approaches. The gradient Reversal Layer (GRL) ensures that the feature distributions over the two domains are indistinguishable for the domain classifier.

As illustrated in Fig. 2, the DANN model is composed of a feature extractor f_e , a regressor f_r , and a domain classifier f_d parameterized by $\theta_e, \theta_r, \theta_d$ respectively.

The feature extractor f_e takes the input data x and learns a feature representation denoted by $f_e(x)$.

The regressor predicts the RUL label for each input sample $\hat{y} = f_r(f_e(x))$. The RMSE loss is used for minimizing the error between the true and the predicted RUL on the source data (for the RUL prediction task):

$$\mathcal{L}_{y,s}(\theta_e, \theta_r) = \sqrt{(\hat{y}_s - y_s)^2} \quad (1)$$

The second part of the model includes a domain discriminator, which is responsible for aligning the two domains. The domain classifier predicts the domain label for each input sample $\hat{d} = f_d(f_e(x))$,

where $d \in \{0, 1\}$ is a domain label assigned to each training example to indicate its origin. The domain classifier uses the binary cross entropy loss function shown in Eq. (2).

$$\mathcal{L}_d(\theta_e, \theta_d) = - (d \log(\hat{d}) + (1 - d) \log(1 - \hat{d})) \quad (2)$$

While the domain discriminator weights θ_d are trained to minimize the binary cross-entropy loss, the feature extractor weights θ_e are updated to maximize the binary cross-entropy loss. The feature extractor learns to extract domain-invariant features, rendering the domain discriminators incapable of predicting the true domain label. These two contradicting objectives are trained in an adversarial procedure utilizing a gradient reversal layer (GRL). GRL is an operation that acts as an identity function during the forward pass and reverses the sign of the gradient during the backward pass. The GRL allows us to implement this optimization problem in practice easily [12].

Finally, the regular stochastic gradient solvers (SGD) can be adjusted to optimize the model as suggested by Eqs. (3) and (4).

$$(\hat{\theta}_e, \hat{\theta}_r) = \arg \min_{\theta_e, \theta_r} \mathcal{L}(\theta_e, \theta_r, \hat{\theta}_d) \quad (3)$$

$$\hat{\theta}_d = \arg \max_{\theta_d} \mathcal{L}(\hat{\theta}_e, \hat{\theta}_r, \theta_d) \quad (4)$$

3.5. Operation profile-specific alignment

In this research, we propose the Operation Profile-Specific (OPS) alignment framework, which aims to align the marginal distributions of the specific phases of the operation profile between different domains. These phases are often overlooked by other domain-invariant feature-learning techniques. Our approach extends the alignment component of DANN to consider each operating phase individually when aligning the source and target domains. In previous research, it has been demonstrated that when the feature distribution is multimodal, adapting only the feature representation can be challenging for adversarial networks [13,14]. Therefore, to improve DANN's performance in these scenarios, we chose to use the DANN method, as it has a strong track record in domain adaptation tasks and has previously been successful in regression tasks for PHM [9]. We propose two different approaches for operation profile-specific alignment:

- Hard assignment : OPS-DANN (hard)
- Soft assignment: OPS-DANN (soft)

For hard assignment, each sample is strictly assigned to its corresponding phase within the operational profile. In contrast, soft assignment involves assigning samples to multiple discriminators, which can be advantageous when operation phases are uncertain. This approach proves particularly useful during transitional periods between phases, where samples could be associated with more than one operation phase.

3.5.1. OPS-DANN (hard)

Our first proposed extension to implement OPS alignment involves using dedicated domain discriminators for each operation profile. A hard assignment is made for each sample to its respective phase of the operation profile. The goal of OPS alignment is to separately align the distinct marginal distributions of each regime of the operation profile across the two domains.

This can be achieved by replacing the single domain discriminator task that discriminates between the source and the target domain with as many individual discriminators as there are discrete operating phases. Each individual domain discriminator still aims to differentiate between the source and the target domain. However, it only does so for samples belonging to one operating phase. This ensures that the marginal distribution of each operating phase is aligned with its counterpart in the other domain. The proposed framework is visualized in Fig. 3.

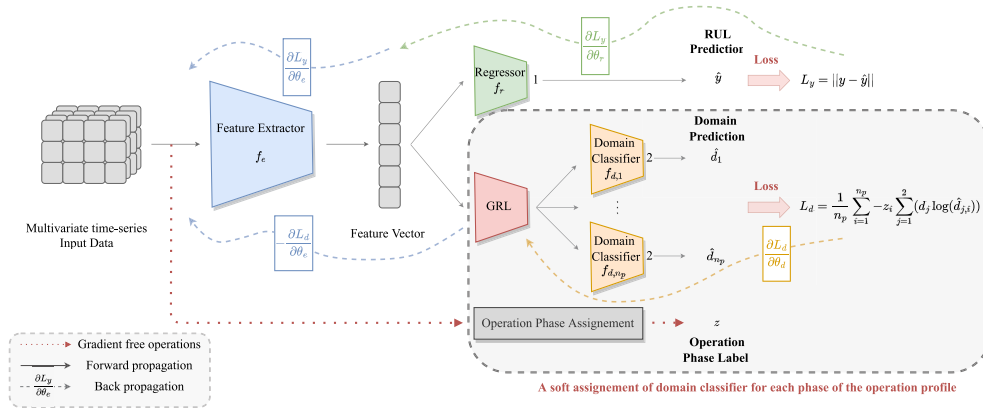


Fig. 3. Proposed OPS-DANN (hard) approach. The source and target samples of each operation profile are aligned with a dedicated domain discriminator.

The domain discriminators are separated for each n_p operating phases. Each domain discriminator is parameterized with its own set of weights θ_{d_j} with $j \in \{1, \dots, n_p\}$ and aims to predict the domain label $\hat{d}_{j,i} = f_{d_j}(f_e(x))$ for a sample that is assigned to the i th domain discriminator. The operating phase label z^i assigns each sample to one of the n_p discriminators. The binary cross entropy loss function shown in Eq. (5) is utilized to determine the loss of each domain discriminator, where z^i is a binary variable indicating whether a sample belongs to operating phase i or zero otherwise. The new domain discriminator loss is presented in Eq. (5):

$$\mathcal{L}_{d,i}(\theta_e, \theta_d) = -z_i (d \log(\hat{d}) + (1 - d) \log(1 - \hat{d})) \quad (5)$$

3.5.2. OPS-DANN (soft)

A soft assignment of samples to multiple discriminators can be useful when the operation phase labels are not certain. This can happen, for example, when a system is transitioning between two phases, and samples can be considered to belong to both operation phases. Instead of using a hard assignment of each sample to a single phase of the operation profile, we propose using a probabilistic assignment (soft assignment). To achieve this, we add an additional classifier after the feature extractor to classify the operation phase. This classifier is trained in a supervised manner using the available operating phase labels and outputs a probability distribution over the n_p phases. The soft assignment allows samples to be assigned to multiple domain discriminators when the operating phase classifier is uncertain. The predicted probabilities are then used to weigh each sample's contribution to each of the n_p domain discriminators. This model is depicted in Fig. 4.

For OPS-DANN (soft), the previously introduced equations change slightly due to the additional operating phase classifier parameterized by θ_z . For each source and target sample, it predicts a probability of a sample belonging to each of the n_p operating phases $\hat{z} = f_z(f_e(x))$. The prediction \hat{z} can be written as a n_p -dimensional vector $\hat{z} = [\hat{z}_1, \dots, \hat{z}_{n_p}]$. The cross-entropy loss function compares the model's predictions to the true operating phase labels.

$$\mathcal{L}_z^i(\theta_f, \theta_z) = -z^i \log(\hat{z}^i) \quad (6)$$

The final loss for the OPS-DANN (soft) can be written as :

$$\mathcal{L}_{d,i}(\theta_e, \theta_d) = -\hat{z}_i (d \log(\hat{d}) + (1 - d) \log(1 - \hat{d})) \quad (7)$$

4. Case study

4.1. N-CMAPSS dataset

We evaluate our proposed methods for Domain adaptation on the new Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS) dataset, which contains run-to-failure trajectories of large

turbofan engines [18]. We compare the performance to previously proposed DA approaches. The dataset was created using NASA's high-fidelity simulation model [56], which allows the simulation of flight data over a wide range of flight conditions. Concretely, the flight data covers take-off, cruise, and descent flight conditions corresponding to different commercial flight routes. Each engine unit is assigned to one of the three flight classes based on the duration of the individual flights. Short flights are defined as flights that last between one and three hours, medium flights are between three and five hours long, and long flights last longer than five hours.

The degradation behavior of each engine unit is modeled as the combination of three contributors: an initial degradation, a normal degradation, and an abnormal degradation due to a fault. In the first phase, the engine degrades due to the normal degradation until the onset of a fault, whereby different fault types can occur. *Normal degradation* is modeled linearly. Once a fault is initiated, the engine enters an *abnormal degradation* phase until it ultimately reaches its end-of-life (EOL). During the abnormal degradation phase, the engine health decays exponentially. Fig. 5 shows sample degradation trajectories of six engine units.

The degradation trajectories are given as a multivariate time series of the sensor readings, containing measurements from 14 sensors from the turbofan engine, as well as scenario descriptors containing 4 parameters characterizing the operating condition of the flight, also referred to as scenario descriptors. The signals and the scenario descriptors are sampled at a frequency of 1 Hz. The different sensors include various temperature and pressure measurements, fan speeds, and fuel flow. The scenario descriptors include the altitude, speed, throttle-resolved angle, and total temperature at the fan inlet.

Each turbofan engine consists of five main components: the fan, the low and high-pressure compressor, and the low-and high-pressure turbine. During the simulated lifetime of each unit, one or a combination of several sub-components fails, leading to the onset of one of seven possible fault types. Fault severity increases over time. The N-CMAPSS dataset introduced several improvements compared to the popular original CMAPSS dataset [57], frequently used by researchers working on RUL-prediction tasks [9,58–60]. The N-CMAPSS differs with respect to two main aspects. First, it considers actual flight conditions recorded on board of a commercial aircraft. Secondly, it extends the modeling of degradation by linking the degradation process to its operational history. The RUL prediction aims to indicate how many flights (i.e., cycles) of a particular engine (unit) are left before its EOL.

The N-CMAPSS dataset [18] contains eight datasets, each with different fault types and flight classes. In this study, we consider the different flight classes as distinct domains and focus on dataset three (DS03) of the N-CMAPSS dataset, which is characterized by a single fault type (HPT-efficiency failure combined with LPT flow and efficiency failure) and equally distributed flight classes. We define the

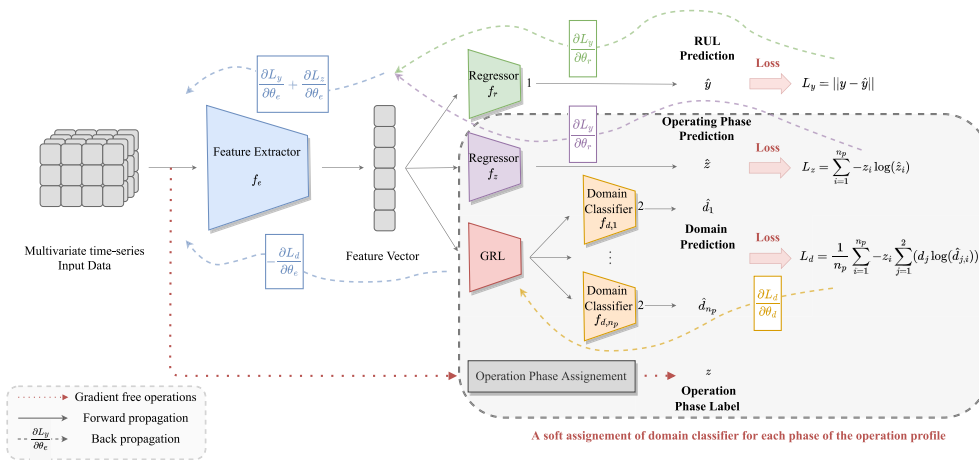


Fig. 4. Proposed OPS-DANN (soft) approach. An additional operation profile classifier is added to allow for the assignment of each sample to multiple domain discriminators depending on its probability of belonging to the respective operating regime.

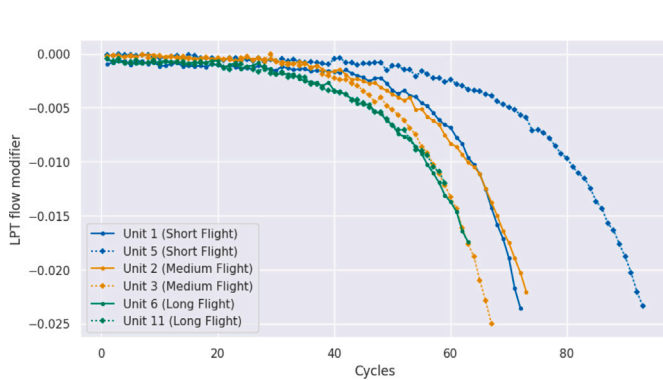


Fig. 5. Example of degradation trajectories of engine units over time for engines from different domains. Vertical lines denote the fault initialization of each engine unit.

Table 1

Number of units and samples of the three considered domains.

Domain flights	Short (S)	Medium (M)	Long (L)
Number of units	5	5	5
Number of samples	104k	202k	362k

sub-fleets operated under different flight classes as a domain. The subset used for this paper contains an equal number of five units per flight class. Despite the identical number of units, the total number of samples of the three domains varies considerably due to the different lengths of the flights, as shown in Table 1.

In aviation, engines are subject to high stress during take-off. Therefore, in this dataset, short-haul flights have more take-offs and landings compared to long-haul flights, resulting in a different type of degradation. In addition, the sensor measurements of long-haul flights are significantly different from those of short-haul flights, as they reach higher altitudes and faster speeds, as can be seen in Fig. 6.

Given the three domains, there are six possible domain adaptation tasks, of which only the three presumably most difficult tasks are considered in this research. The three tasks comprise the transfer from a sub-fleet of short-range flights to the sub-fleets of mid-range and long-range flights, as well as from mid-range to long-range flights. These three tasks are particularly difficult because the marginal distribution of the characteristics of shorter flights does not cover those of longer flights, as visualized using the four scenario descriptors in Fig. 6. The

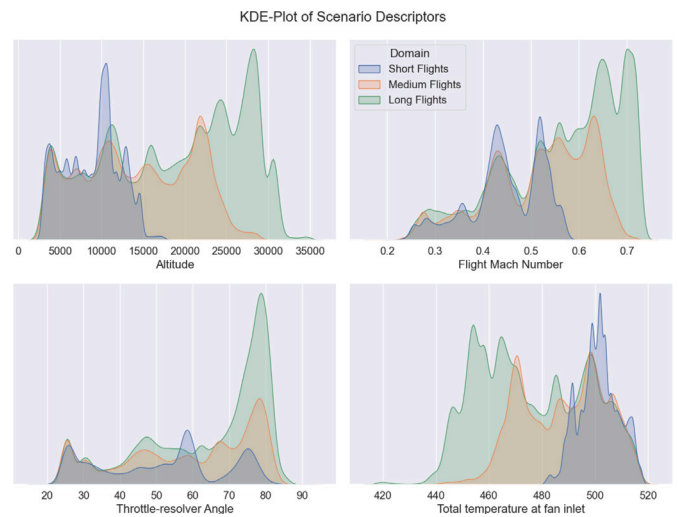


Fig. 6. The kernel density estimate (KDE) shows the probability distribution of the four scenario descriptors for each domain separately. In three out of four cases, it can be seen that the long-haul flights span the widest range of feature values, while the short-haul flights cover a noticeably smaller part.

corresponding considered adaptation tasks are abbreviated as $S \rightarrow M$ for short to medium flight length adaptation tasks by $S \rightarrow L$ and $M \rightarrow L$ for the other two tasks, respectively.

In general, it seems intuitive to subdivide a flight into take-off, cruise, and landing operating conditions. However, in this work, we found a better suitable division of the operation profile: in ascending, steady, and descending operating conditions. This division allows a more fine-grained separation and is more closely related to how the airplane is operated. For example, the operating conditions of an airplane that changes its flight altitude during a flight would only be assigned to the cruise condition with the more intuitive partitioning approach. In contrast, the partitioning proposed in this research allows separating the two steady flight sequences from the descending one in between (see Fig. 7).

Ascending, steady, and descending flight conditions can be identified using the first-order derivative of the altitude measurement. To that end, the change in altitude between neighboring sampling points was considered and grouped into the three operating conditions using a threshold value. The threshold was experimentally set to $T = 0.5 \frac{ft}{s}$. The Operation Phase label z can be defined based on the following

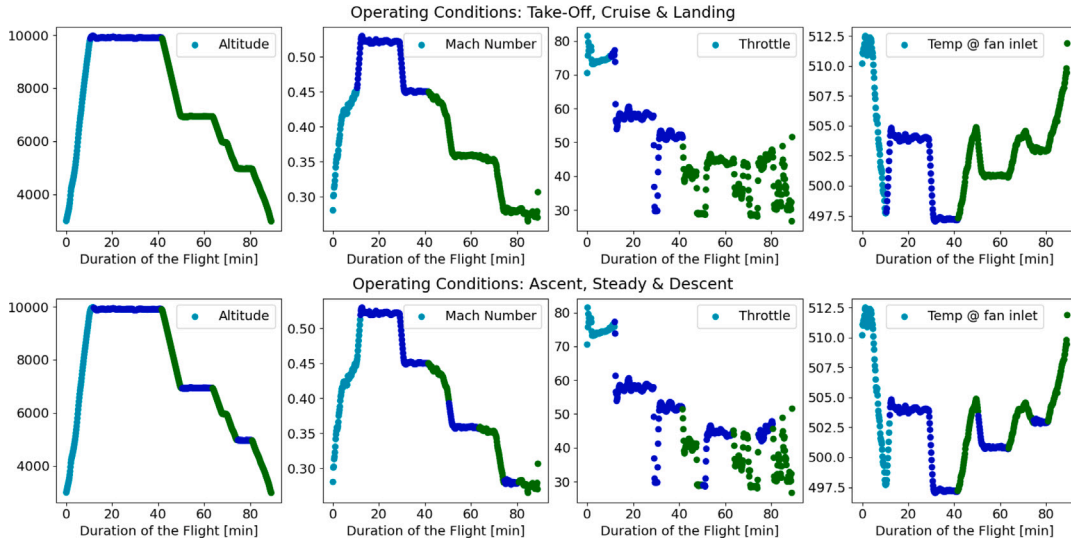


Fig. 7. Direct comparison of two possible separations of the operating conditions. Ascent and take-off are colored in turquoise, respectively; cruise and steady flight in dark black; and descent and landing in green. The short periods of steady flying are differentiated from the rest using the change in altitude.

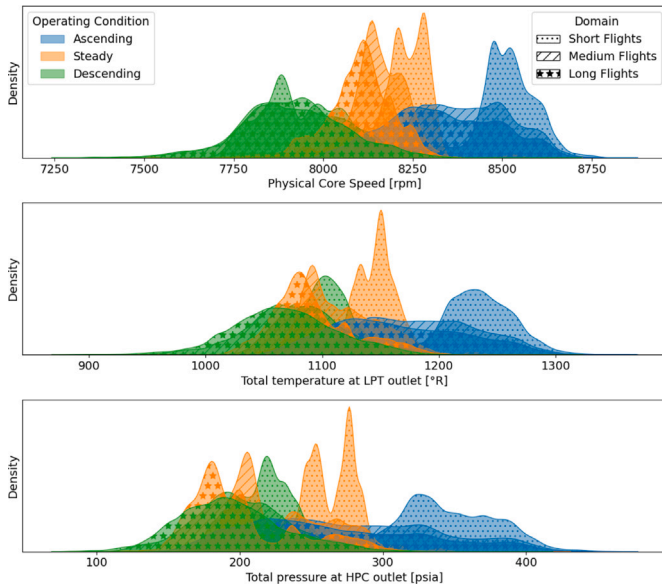


Fig. 8. Kernel density estimate for marginal distributions of each operation profile for the three domains. Three exemplary sensor values were selected: Physical Core speed, Total Temperature at the LPT outlet, and total pressure at the HPC outlet.

inequalities:

$$z = \begin{cases} \text{Ascending} & \text{if } \frac{x^{i+1}_{alt} - x^i_{alt}}{\Delta t} \geq T \\ \text{Descending} & \frac{x^{i+1}_{alt} - x^i_{alt}}{\Delta t} \leq -T \\ \text{Steady} & \text{otherwise} \end{cases} \quad (8)$$

A median filter with a length of 51 was applied to smoothen the predictions.

The key idea behind the proposed OPS alignment is to consider each operation phase separately during the alignment to ensure that its marginal distribution is matched with its counterpart in the other domain. Fig. 8 shows the marginal distributions of three exemplary sensors for the short, medium, and long flight domains. Most of the other sensor values follow similar patterns.

4.2. Preprocessing

Before a fault initiates, the normal degradation process is slow, and it can be assumed that the RUL is very large. Thus, for this work, we only start predicting the RUL after the onset of the fault. This results in a two-step process in which fault detection is performed first, and the RUL prediction is only initiated after the fault has been detected. In this paper, we are only focusing on the second step and assume that the fault onset detection is given. Therefore, the RUL prediction task aims to predict the remaining cycles after a fault has been initiated and detected.

Sensor measurements and scenario descriptors are taken as inputs to ensure a realistic usage scenario, resulting in 18 sensor values. Other quantities, such as virtual sensors and model health parameters, which require internal parameters of a simulator, are not considered in this research.

The dataset was downsampled by a factor of ten using a Chebyshev filter with order eight, which applies an anti-aliasing filter before the down-sampling process. This ensured the same 0.1 Hz sampling frequency that was used in [59].

Each signal measurement was scaled to a range $[-1, 1]$ using min-max normalization. The scaler was fitted on the source domain data for each adaptation task and subsequently applied to the target domain data.

The RUL was normalized for each engine unit to decrease from one to zero **between the point in time when the fault occurs and the end of life**. To that end, all samples belonging to one engine unit were divided by the maximum number of cycles of the respective unit.

4.3. Model architecture

In this research, we compare the proposed methodology to other commonly used DA approaches. All of the DA methods use the same basic architecture to ensure a fair comparison. In particular, we used a one-dimensional-convolutional neural network (1DCNN) with the architecture inspired by that proposed in [59]. The feature extractor consists of three 1D-convolutional layers ($L = 3$). The first two layers have ten channels each, while the third layer condenses the feature representation into a single channel. Filters of size ten are used along with a stride of one, and zero-padding is added to ensure that inputs keep the same length when passing through the network. The network uses *ReLU* as the activation function. The feature extractor has a total of 3k trainable model parameters.

The RUL-regressor contains two fully connected layers ($L = 2$). The first one takes the flattened output from the feature extractor and passes it through a fully-connected layer with 50 neurons. Then, after another *ReLU* activation function, the last layer predicts a single output value, the RUL, which is subsequently normalized to a range between 0 and 1 using a *Sigmoid* activation function. In total, there are approximately 3k parameters in the RUL-regressor.

The domain discriminator has a similar architecture as the RUL regressor but contains an additional layer. A first fully-connected layer with 50 neurons is followed by a second one with 30 neurons. The last layer is also fully-connected and ends with a single output neuron. Again, a *Sigmoid* activation function is applied to the final output, and *ReLU* activation functions are used in between fully connected layers. The number of layers and neurons is loosely inspired by an earlier work, which also used a DANN architecture on a similar dataset [9]. The resulting domain discriminator has roughly 4k parameters.

OPS-DANN (hard): This model utilizes three domain discriminators, as there are three different phases for the operation profile in the dataset used in this work. The architecture of each discriminator is identical to the one described above, leading to a total number of 18k parameters.

OPS-DANN (soft): Similarly to the OCS-DANN (hard) version, this model utilizes three distinct domain discriminators. Additionally, it uses an operating phase classifier, which is added after the feature extractor. This classifier uses the same architecture as the domain discriminator, except for the last activation function. A *Softmax* activation function is used in the model instead of a *Sigmoid* activation function, and this is because there are multiple classes that the model needs to predict and not just two classes since *Softmax* activation is designed to handle multi-class classification. OPS-DANN (soft) has the largest number of trainable model parameters, with a total of 22k.

Multi-Class OPS-DANN: We propose to explore the impact of the domain alignment of the operation phases in an ablation analysis. Compared to the architecture introduced above, this model only requires changes in the architecture of the domain discriminator of DANN. The domain discriminator predicts the domain of each sample and its operating phases. As there are three operating conditions, the discriminator requires six output neurons, one for each domain and operating phase pair. Consequently, the *Sigmoid* function must be exchanged by the *Softmax* activation to handle the multi-class output. This model has slightly more parameters than the original DANN, with roughly 10k parameters.

4.4. Comparison methods

To fairly evaluate the performance of our proposed methods, we first establish a baseline using a feature extractor with a regressor that is trained only on the source data. We then compare the results of our Operation Profile-specific Domain Adaptation Network (OPS-DANN) and its variants to established domain adaptation techniques such as AdaBN, MK-MMD, and DANN. These methods have been previously applied to similar tasks in the field of prognostics and health management and have been shown to have strong performance, as reported in previous research studies such as [8]. **MK-MMD:** This DA method is a domain-invariant feature learning technique. However, instead of using adversarial training, it aims to minimize a divergence measure between the two domains. To that end, the feature representation f , found by a feature extractor, is used to compute the MMD measure as shown in Eq. (9). Multiplications of the feature transformation $\phi(\cdot)$ can be readily computed by taking advantage of the kernel trick as demonstrated in Eq. (10). For the multiple kernel version of MMD, K Gaussian kernels with different bandwidth parameters γ are combined to enhance model performance by adding them together, as shown in Eq. (11).

$$MMD(F_S, F_T) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(f_S^i) - \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(f_T^j) \right\|^2 \quad (9)$$

$$k(f_S^i, f_T^j) = \langle \phi(f_S^i), \phi(f_T^j) \rangle \quad (10)$$

$$k(f_S^i, f_T^j) = \sum_{k=1}^K k_k(f_S^i, f_T^j) \quad (11)$$

For this work, five kernels were selected, similar to previous applications as reported in [10]. The bandwidths parameter was set to 0.01, 0.1, 1, 10, and 100.

AdaBN: Unlike the other baseline methods introduced above, AdaBN [61] is not a domain-invariant feature learning technique. Instead, it aims to replace the normalization statistics of batch-norm layers computed on the source domain with those of the target domain. Notably, the target domain data is only used to update the normalization statistics, while all the other model parameters are trained using source data only. A necessary condition to utilize AdaBN is an architecture with batch-norm layers; consequently, the feature extractor's standard architecture had to be adapted. A 1D-batch-norm layer was placed following each of the three convolutional layers.

In summary, except for AdaBN, which uses additional batch-norm layers, all other baseline methods use the same architecture for the feature extractor and the RUL regressor to ensure a fair comparison.

4.5. Training procedure

Before training, all models were initialized using Xavier normal initialization [62]. All the data from each domain are considered during training for each adaptation task, and no test-training split is used.

In this research, several consecutive measurement points are combined into one input sample as input to the model using sequences of length 50 from the multivariate time series with a step size of one.

Model updates were performed using batch gradient descent with batches of size 256. One batch of source and target domain data was processed for each training step.

All DA methods were trained for the same number of epochs for each of the adaptation tasks to ensure a fair comparison. For the $S \rightarrow L$ and $M \rightarrow L$ adaptation tasks, the models were trained for 15 epochs. However, in the $S \rightarrow M$ adaptation task, the models were trained for 25 epochs due to the lower number of model updates per epoch.

The baseline was trained for 40 epochs in case the source domain was short flights S , and for 20 epochs in case the source domain was medium flights M .

All DA methods based on DANN additionally required the definition of the reverse gradient factor ρ . For this work, the same update rule for ρ is used as the one proposed in the original DANN [15], which gradually updates ρ from 0 to 1 according to Eq. (12).

$$\rho = \frac{2}{1 + e^{-10 l_p}} - 1 \quad (12)$$

The learning rate is reduced after each epoch to ensure a smooth convergence using a learning rate schedule, likewise adopted from the original DANN paper [15]. The proposed schedule is shown in Equation:

$$\alpha = \frac{\alpha_0}{(1 + 10 l_p)^{0.75}} \quad (13)$$

where l_p once again describes the linear training progress. Unlike the gradient reversal factor, which only applies to DANN-based models, the learning rate schedule is used for all models. The initial learning rate α_0 is found using a hyperparameter search.

The hyperparameters and training specifications mentioned above were primarily selected based on prior work and applied to all models without further refinement. Other hyperparameters were explicitly tuned for each domain adaptation method, such as the learning rate, momentum, and the trade-off between multiple loss objectives. Table 2 summarizes the performed grid searches for each method and indicates the optimal hyperparameters found for each method. It is important to

Table 2

Range of the hyperparameter search for all considered DA-methods. The optimal hyperparameters found on the $S \rightarrow L$ adaptation task are emphasized.

Model name	Learning rate	Momentum	Tradeoff alignment loss	Others
Baseline	{ 0.01 , 0.05, 0.25}	{0.5, 0.9 }	–	–
AdaBN	{0.01, 0.05, 0.25 }	{0.5, 0.9}	–	–
MK-MMD	{0.01, 0.05 , 0.25}	{ 0.5 , 0.9}	{0.003, 0.01 , 0.03}	{2, 5, 10 } ¹
DANN	{0.01, 0.05 , 0.25}	{ 0.5 , 0.9}	{0.1, 0.3 , 1}	–
OPS-DANN Hard	{ 0.05 }	{ 0.5 }	{0.03, 0.1, 0.3, 1 , 3}	–
OPS-DANN Soft	{ 0.05 }	{ 0.5 }	{0.1, 1 , 3}	{0.03, 0.1 , 0.3} ²
Multi-class OPS-DANN	{ 0.05 }	{ 0.5 }	{0.1, 0.3 , 1}	–

emphasize that for all methods, the grid search was only performed on the $S \rightarrow L$ adaptation task. The resulting optimal hyperparameters were then used for the other two adaptation tasks. The $S \rightarrow L$ task was selected to perform the grid search because it has the largest domain gap and is considered as the most challenging task. For AdaBN, the learning rate and momentum found on the $S \rightarrow L$ adaptation task were likely too big for the $M \rightarrow L$ adaptation task and did not lead to convergence. Only in this case, the second best set of hyperparameters found on the $S \rightarrow L$ was used for the $M \rightarrow L$ task, which is also indicated in Table 2.

4.6. Evaluation metrics

In this work, all experiments were evaluated using the two common evaluation metrics used for RUL prediction [57]: the root mean square error (RMSE) and the NASA scoring function.

The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{n_T} \sum_{i=1}^{n_T} (\hat{y}^i - y^i)^2} \tag{14}$$

NASA’s scoring metric is not symmetric and penalizes over-estimation more than under-estimation and is defined as follows:

$$s = \sum_{i=1}^{n_T} \exp(\alpha |\hat{y}^i - y^i|) \tag{15}$$

$$\alpha = \begin{cases} 1/10 & \text{if } \hat{y} - y \geq 0 \\ 1/13 & \text{otherwise} \end{cases}$$

4.7. Representation transferability

The Proxy A-distance (PAD) proposed in [63] is a suitable metric to measure the divergence for domain adaptation tasks. Thus, we can use it to evaluate the transferability of representations. Using the PAD, the divergence between two domains is computed by evaluating how well a classifier can separate the source from the target domain. If separation is easy, the two domains are likely dissimilar, and their discrepancy is large. On the other hand, if samples can hardly be discriminated between the two domains, they will likely have a small discrepancy. Using the classification error ϵ , the PAD \hat{d}_A can be found according to Eq. (16).

$$\hat{d}_A = 2(1 - 2\epsilon) \tag{16}$$

In general, all DA methods that contain a domain-invariant feature learning part aim somewhat to minimize the divergence between the source and the target domain. For example, DANN uses the PAD as a discrepancy measure and aims to minimize it during training in the alignment component [15].

5. Results

5.1. Experimental results

In this section, we compare the prediction accuracy of the considered models in terms of the RMSE and the NASA score. The results are shown in Table 3. Additionally, the upper bound RMSE for

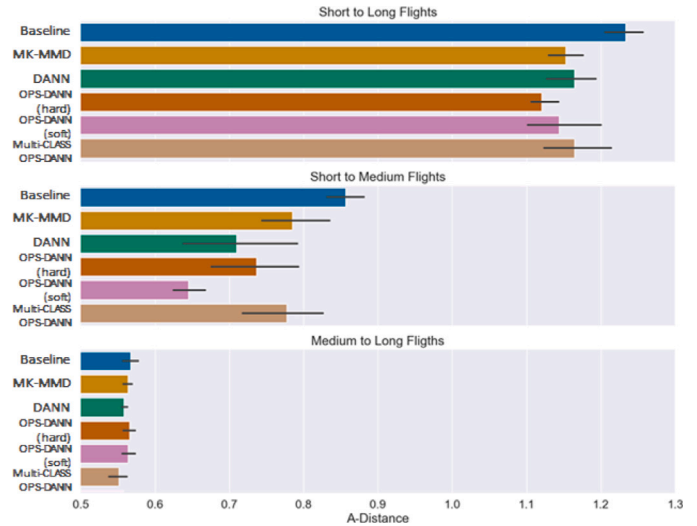


Fig. 9. A-distance for each method and task respectively.

the medium-haul and long-haul flight domains are 3.16 and 2.17, respectively.

In addition, to evaluating the performance, we also evaluate the PAD (qualitatively measuring the divergence between the domains), calculated from the learned feature embeddings. As described in Section 4.7, the PAD is used to evaluate the ability of the applied DA methods to extract domain-invariant features. In Fig. 9, the A-distance varies from one task to another, showing a larger domain gap for the task $S \rightarrow L$, a smaller one for the task $S \rightarrow M$ and almost no domain gap for the task $M \rightarrow L$.

5.1.1. $S \rightarrow L$ task

Performance evaluation: The adaptation task from the short to the long flight domain is the most challenging because it has the largest domain gap. The baseline model trained solely on source data for this task (without any adaptation) exhibits an RMSE value that is 240% larger than the one of the models trained on the target domain. As seen in Table 3, the MK-MMD and DANN methods improved the baseline methods in terms of RMSE by 35 and 40%, respectively. The improvements by DA methods are also visible in the NASA score. AdaBN is a simple method without any additional parameters. It improves the model performance with respect to RMSE only slightly. However, it leads to a higher NASA score compared to the baseline due to the over-estimations of the RUL predictions. For the $S \rightarrow L$ task, all three proposed OPS-DANN variants improve substantially upon the baseline by at least 45% in terms of RMSE and by 20% in terms of S-score.

It is similarly important to compare the performance of a novel DA method to other state-of-the-art methods. Even though DANN has the lowest RMSE out of the three considered baseline methods, the proposed OPS-DANN methods outperform DANN by at least 9% in RMSE and 2% in S-score. Thus, all models benefit from unlabeled target domain data, as expected in this setup. However, comparisons with the

Table 3
The results for the three considered domain adaptation tasks.

Name	$S \rightarrow L$		$S \rightarrow M$		$M \rightarrow L$	
	RMSE	s-score	RMSE	s-score	RMSE	s-score
	Mean \pm Std	Mean \pm Std	Mean \pm Std	Mean \pm Std	Mean \pm Std	Mean \pm Std
Baseline	7.34 \pm 2.12	1.74 \pm 0.24	4.59 \pm 0.51	1.38 \pm 0.07	2.63 \pm 0.19	1.24 \pm 0.03
AdaBN	7.13 \pm 0.72	1.82 \pm 0.18	5.03 \pm 0.66	1.49 \pm 0.09	3.23 \pm 0.26	1.26 \pm 0.02
MK-MMD	4.76 \pm 0.57	1.47 \pm 0.04	4.40 \pm 0.42	1.36 \pm 0.04	2.64 \pm 0.20	1.22 \pm 0.04
DANN	4.38 \pm 0.56	1.43 \pm 0.11	4.07 \pm 0.23	1.35 \pm 0.04	2.75 \pm 0.35	1.20 \pm 0.01
OPS-DANN Hard	3.90 \pm 0.56	1.39 \pm 0.10	3.71 \pm 0.20	1.31 \pm 0.04	2.66 \pm 0.23	1.20 \pm 0.00
OPS-DANN Soft	4.00 \pm 0.33	1.40 \pm 0.11	3.86 \pm 0.18	1.33 \pm 0.03	2.72 \pm 0.20	1.21 \pm 0.02
Multi-Class						
OPS-DANN	3.97 \pm 0.65	1.40 \pm 0.11	3.93 \pm 0.46	1.33 \pm 0.03	2.58 \pm 0.20	1.20 \pm 0.01

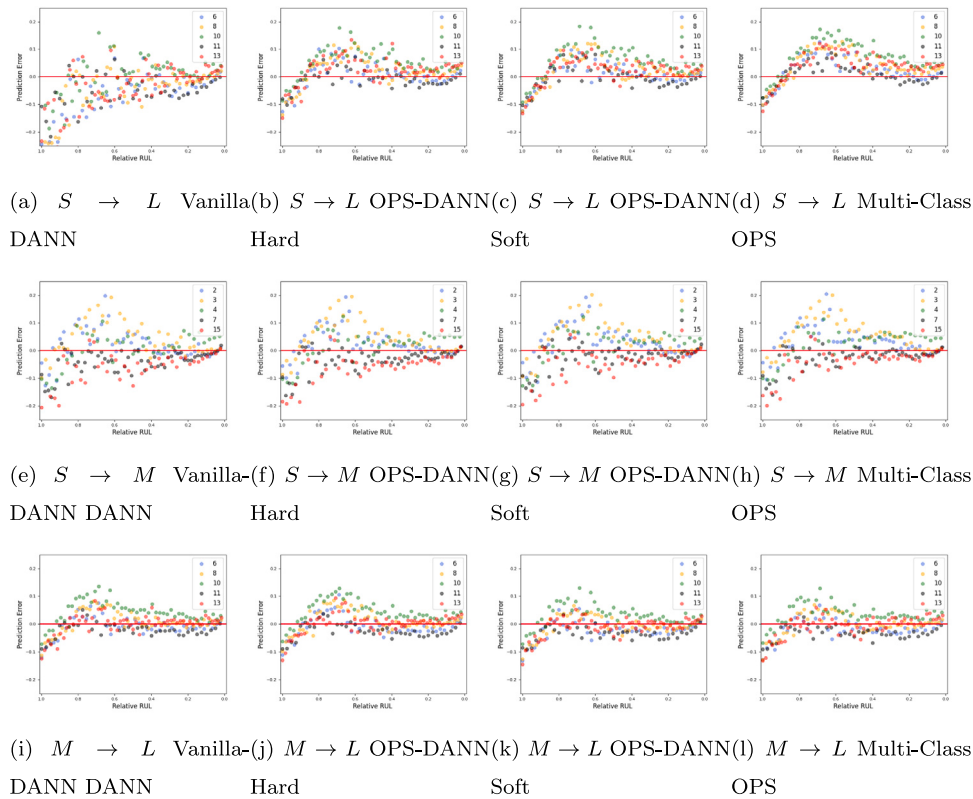


Fig. 10. Prediction error of the different models as a function of the relative lifetime.

upper bound RMSE of 2.17 show that by using labeled short-haul flight data for adaptation, there is still a considerable performance gap to models trained with the labeled long-haul flight data.

Evaluation of the alignment: Fig. 9, shows that all three proposed OPS-DANN methods can extract features that contain less discriminative information about the two domains than the baseline on the $S \rightarrow M$ DA task. The two methods using individual domain discriminators (OPS-DANN soft and hard) reach the lowest PAD values for this task, indicating that the usage of a dedicated domain discriminator per operating phase can support the extraction of features that are more domain-invariant compared to domain adaptation techniques aligning the entire marginal distributions of the two domains.

5.1.2. $S \rightarrow M$ Task

Performance evaluation: Compared to the $S \rightarrow L$ task, the $S \rightarrow M$ task has a smaller domain gap. The baseline model trained without any adaptation solely based on source data for this task results in an RMSE that is 45% larger than the one from the model trained on the target domain. Due to the smaller domain gap in the $S \rightarrow M$ task compared to the $S \rightarrow L$ task, the source domain is more similar to the target domain

and, therefore, it allows learning models with a similar performance level as models trained on the target data. As seen in Table 3, the MK-MMD and DANN methods are able to improve the baseline methods slightly in terms of RMSE by 4 and 11%, respectively. However, in terms of the s-score, the results decreased slightly from 1.38 to 1.36 and 1.35, respectively. AdaBN approach performed worse than the baseline in this case. Also, in this task, DANN is the best-performing comparison model. Nevertheless, all three proposed OPS-DANN methods outperform the original DANN. While for the Multi-Class OPS-DANN, the improvement is 3% compared to the DANN performance, the OPS-DANN (soft) and OPS-DANN (hard) reduce the RMSE by 5% and 9%, respectively, and the s-score by 1% and 3%

Evaluation of the alignment: On the $S \rightarrow M$ task, all methods reach considerably lower PAD values, even the lower bound, without any adaptation. This makes it more challenging to differentiate between these two domains. However, it is worth noting that all the proposed methods in the Operation Profile-specific Domain Adaptation Network (OPS-DANN) have considerably lower PAD values compared to the baseline. However, only the OPS-DANN (soft) method can extract more domain-invariant features than DANN, while the other two proposed

methods have slightly higher PAD values. Even though OPS-DANN (hard) has a higher PAD than both DANN and OPS-DANN (soft), it reaches a lower RMSE demonstrating that the sole ability to extract domain-invariant features does not guarantee a superior performance on the regression task.

5.1.3. $M \rightarrow L$ Task

Performance evaluation: The third considered task uses medium-haul flights as the source domain and aims to adapt a model to perform well on long-haul flights. Compared to the two previous tasks, the discrepancy between these two domains is much smaller. This observation is also confirmed by the performance of the baseline method that has a value RMSE that is only 21% larger than the one of a model trained on the target domain. All domain adaptation methods performed slightly worse than the baseline model. Using the MK-MMD and DANN methods appears to result in under-estimated predictions, thus achieving a lower s-score than the baseline. Out of the three proposed OPS-DANN models, only one of them (Multi-Class OPS-DANN) has a slightly lower RMSE value compared to the baseline (2.58 compared to 2.63). The other two approaches are at a similar performance level as the baseline model, however, achieving a lower s-score (1.20 compared to 1.24). The observed slight differences between the models are likely insignificant and are potentially caused by the randomness of the training procedure. Overall, none of the comparison methods DA methods could improve this task's target-free baseline significantly. None of the methods are able to reach the upper bound RMSE value of 2.17, which is achieved when the model is trained on the labeled target dataset.

Visualizations: We provide additional visualizations, in Fig. 10, by showing the test RUL evolution over time for the three adaptation approaches ($S \rightarrow L$, $S \rightarrow M$, $M \rightarrow L$) using DANN and the proposed three methods. The figures confirm our initial expectations. As the units approach the end of their lifetime, the predictions tend to converge with the ground truth RUL value.

Evaluation of the alignment: Comparing the PAD values of all applied methods in Fig. 9, it becomes apparent that none of the methods are able to reduce substantially the domain distinctiveness compared to the models trained without adaptation.

6. Discussion

The findings of this study demonstrate that, on two out of the three tasks, the proposed three variants of the operation phase alignment methods perform similarly well and are able to achieve better results compared to both the DANN model as well as the other domain adaptation methods. This demonstrates the effectiveness of the proposed methods in adapting to different domains, particularly in addressing the unique challenges posed by multimodal feature distributions and the variability of operating conditions and system configurations. Furthermore, it should be noted that DANN is already an effective method for domain adaptation. However, the proposed operation phase alignment methods are able to improve upon the performance of DANN and other traditional domain adaptation methods in the context of PHM.

In addition to evaluating the performance of the proposed methods for RUL prediction, we also investigate the impact of each method on the embeddings generated by the alignment process. To enable 2D visualizations, we applied a principal component analysis (PCA) and present the first two principal components in Figs. 11 and 12. This visualization provides insights into how the different phases of the operation profile are represented in the embeddings and how the proposed methods affect this representation.

As shown in Fig. 11 DANN and our proposed approaches are able to effectively align the source and target embeddings, as demonstrated by the overlapping of the source and target data points in the figures. However, it is worth noting that there is a distinct difference between

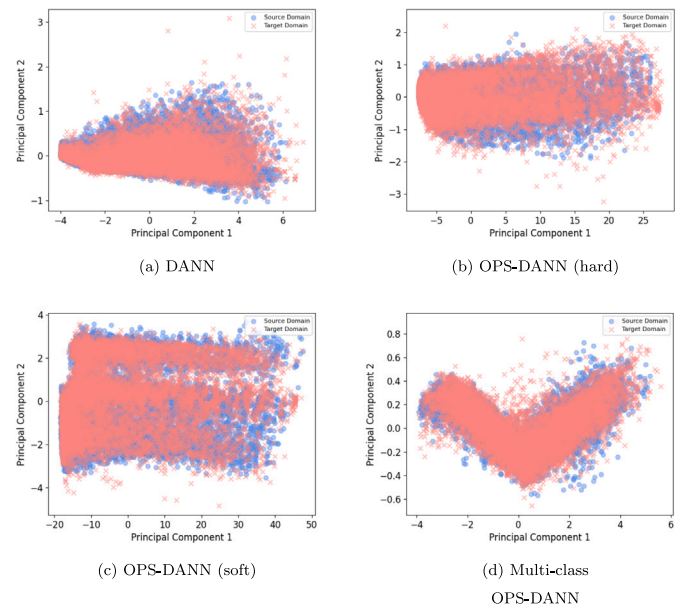


Fig. 11. Visualizing the impact of the operation profile alignment on Domain alignment: A Comparison of DANN, OPS-DANN (soft), OPS-DANN (soft), and Multi-Class OPS-DANN models using PCA on the embedding space.

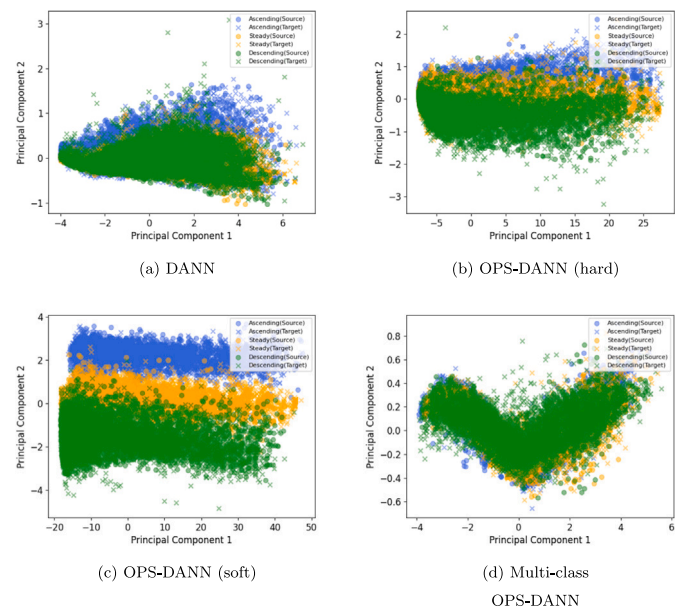


Fig. 12. Visualizing the impact of the operation profile alignment on the Phase alignment: A Comparison of DANN, OPS-DANN (soft), OPS-DANN (soft), and Multi-Class OPS-DANN models using PCA on the embedding space.

the methods with respect to how distinguishable the operation phases are in the embedding space.

In Figs. 11 and 12, the OPS-DANN soft and hard approaches have phases that can be distinguished but that are more pronounced for the soft approach. On the contrary, the Multi-Class OPS-DANN approach has only one cluster that is similar to the one obtained with the DANN approach which is not able to distinguish between the operating phases. However, the difference between DANN and Multi-Class OPS-DANN is that they have different shapes in the embedding space. Moreover, DANN aims to learn domain invariant features, and Multi-Class OPS-DANN aims to learn domain and operating-condition invariant features. This difference in the ability to distinguish between the operation

phases highlights the unique characteristics of each proposed approach and their ability to align the operating characteristics and phases of the different domains.

It should be noted that the Multi-Class OPS-DANN uses the operation phase labels in a distinctly different way than the other two proposed methods. Instead of aligning the marginal distributions of all three operation phases separately, it aims to extract features invariant to both the domain and the operation phase of the input data simultaneously. Fig. 12 shows that Multi-Class OCS-DANN successfully learns such a feature representation that is invariant to the operation phases.

Contrary to the Multi-Class OPS-DANN, the OCS-DANN (soft) models contain an additional classifier that aims to learn how to distinguish between different operating phases while simultaneously learning to extract features that are domain invariant. Fig. 12 illustrates this behavior: each operating phase forms a separate cluster while the source and target domains overlap to a large extent. This behavior is similar to the OCS-DANN (hard).

This observation highlights the effectiveness of the proposed approach in aligning the operating characteristics and conditions of the different domains, leading to improved performance for RUL prediction

7. Conclusion

In this paper, we propose a novel approach that utilizes domain adaptation techniques to align the operating phases to improve the accuracy of RUL predictions.

The main novelty of our proposed approach is the integration of the information on the different phases of the operation profile into the alignment process. The proposed approaches align the marginal distributions of each phase of the operation profile in the labeled source domain with its counterpart in the unlabeled target domain. Two novel domain adaptation approaches are proposed based on an adversarial domain adaptation by considering the different phases of the operation profile separately.

The proposed methods have shown to be effective in improving the performance of deep learning models for RUL prediction by effectively transferring the models between sub-fleets that are operated under different conditions. The results of this study demonstrate the potential of these methods to improve the accuracy and reliability of prognostics and health management in real-world applications. Furthermore, the proposed methods have a better performance compared to state-of-the-art domain adaptation methods such as DANN, MK-MMD, and AdaBN.

There are three interesting potential future research directions resulting from this research. First, the proposed methods can be extended to enable them to tackle challenges arising from imbalanced operating conditions, which is a very common case in practical applications. Second, it would be interesting to investigate whether learning a soft assignment before performing adaptation may be beneficial by training an operating phase classifier separately from the DA model. Thirdly, while this research focused on regression, the proposed methodology can be easily extended to classification tasks such as for example for fault diagnostics problems. We leave the aforementioned open research directions for future work.

CRedit authorship contribution statement

Ismail Nejjar: Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Fabian Geissmann:** Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization. **Mengjie Zhao:** Writing – review & editing, Supervision, Conceptualization. **Cees Taal:** Writing – review & editing, Supervision, Conceptualization. **Olga Fink:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ismail Nejjar reports financial support was provided by Swiss National Science Foundation.

Data availability

The data are available online.

Acknowledgments

This work was supported by the Swiss National Science Foundation under Grant PP00P2_176878

References

- [1] Si X-S, Wang W, Hu C-H, Zhou D-H. Remaining useful life estimation—a review on the statistical data driven approaches. *European J Oper Res* 2011;213(1):1–14.
- [2] Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. *Mech Syst Signal Process* 2019;115:213–37.
- [3] Fink O, Wang Q, Svensen M, Dersin P, Lee W-J, Ducoffe M. Potential, challenges and future directions for deep learning in prognostics and health management applications. *Eng Appl Artif Intell* 2020;92:103678.
- [4] Rombach K, Michau G, Fink O. Contrastive learning for fault detection and diagnostics in the context of changing operating conditions and novel fault types. *Sensors* 2021;21(10):3550.
- [5] Rombach K, Michau G, Fink O. Controlled generation of unseen faults for partial and open-partial domain adaptation. *Reliab Eng Syst Saf* 2023;230:108857.
- [6] Wilson G, Cook DJ. A survey of unsupervised deep domain adaptation. *ACM Trans Intell Syst Technol* 2020;11(5):1–46.
- [7] Li C, Zhang S, Qin Y, Estupinan E. A systematic review of deep transfer learning for machinery fault diagnosis. *Neurocomputing* 2020;407:121–35.
- [8] Wang Q, Michau G, Fink O. Domain adaptive transfer learning for fault diagnosis. In: 2019 Prognostics and system health management conference (PHM-Paris). IEEE; 2019, p. 279–85.
- [9] da Costa PRdO, Akçay A, Zhang Y, Kaymak U. Remaining useful lifetime prediction via deep domain adaptation. *Reliab Eng Syst Saf* 2020;195:106682.
- [10] Cheng H, Kong X, Chen G, Wang Q, Wang R. Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors. *Measurement* 2021;168:108286.
- [11] 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.
- [12] Ganin Y, Lempitsky V. Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. PMLR; 2015, p. 1180–9.
- [13] Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks. 2017, arXiv preprint arXiv:1701.04862.
- [14] Long M, Cao Z, Wang J, Jordan MI. Conditional adversarial domain adaptation. *Adv Neural Inf Process Syst* 2018;31.
- [15] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. *J Mach Learn Res* 2016;17.
- [16] Zamir AR, Sax A, Shen W, Guibas LJ, Malik J, Savarese S. Taskonomy: Disentangling task transfer learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 3712–22.
- [17] Wang Q, Dai D, Hoyer L, Van Gool L, Fink O. Domain adaptive semantic segmentation with self-supervised depth estimation. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV). 2021, p. 8515–25.
- [18] Arias Chao M, Kulkarni C, Goebel K, Fink O. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data* 2021;6(1):5.
- [19] Pan SJ, Yang Q. A survey on transfer learning. *IEEE Trans Knowl Data Eng* 2009;22(10):1345–59.
- [20] Long M, Cao Y, Wang J, Jordan M. Learning transferable features with deep adaptation networks. In: International conference on machine learning. PMLR; 2015, p. 97–105.
- [21] Long M, Zhu H, Wang J, Jordan MI. Deep transfer learning with joint adaptation networks. 2017, arXiv:1605.06636.
- [22] Saito K, Watanabe K, Ushiku Y, Harada T. Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 3723–32.
- [23] Li Y, Wang N, Shi J, Liu J, Hou X. Revisiting batch normalization for practical domain adaptation. 2016, arXiv preprint arXiv:1603.04779.

- [24] Carlucci FM, Porzi L, Caputo B, Ricci E, Bulo SR. Autodial: Automatic domain alignment layers. In: 2017 IEEE international conference on computer vision (ICCV). IEEE; 2017, p. 5077–85.
- [25] Cortes C, Mohri M. Domain adaptation in regression. In: International conference on algorithmic learning theory. Springer; 2011, p. 308–23.
- [26] Mansour Y, Mohri M, Rostamizadeh A. Domain adaptation: Learning bounds and algorithms. 2009, arXiv:0902.3430.
- [27] Chen X, Wang S, Wang J, Long M. Representation subspace distance for domain adaptation regression. In: International Conference on Machine Learning. PMLR; 2021, p. 1749–59.
- [28] Nejjar I, Wang Q, Fink O. DARE-GRAM: Unsupervised domain adaptation regression by aligning inverse gram matrices. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 11744–54.
- [29] Li W, Huang R, Li J, Liao Y, Chen Z, He G, Yan R, Gryllias K. A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges. *Mech Syst Signal Process* 2022;167:108487.
- [30] Zhang Z, Chen H, Li S, An Z. Unsupervised domain adaptation via enhanced transfer joint matching for bearing fault diagnosis. *Measurement* 2020;165:108071.
- [31] Pan Y, Mei F, Miao H, Zheng J, Zhu K, Sha H. An approach for HVCB mechanical fault diagnosis based on a deep belief network and a transfer learning strategy. *J Electr Eng Technol* 2019;14(1):407–19.
- [32] Michau G, Fink O. Unsupervised transfer learning for anomaly detection: Application to complementary operating condition transfer. *Knowl-Based Syst* 2021;216:106816.
- [33] Gao Y, Liu X, Huang H, Xiang J. A hybrid of FEM simulations and generative adversarial networks to classify faults in rotor-bearing systems. *ISA Trans* 2021;108:356–66. <http://dx.doi.org/10.1016/j.isatra.2020.08.012>.
- [34] Yu K, Fu Q, Ma H, Lin TR, Li X. Simulation data driven weakly supervised adversarial domain adaptation approach for intelligent cross-machine fault diagnosis. *Struct Health Monit* 2021;20(4):2182–98. <http://dx.doi.org/10.1177/1475921720980718>, arXiv:<https://doi.org/10.1177/1475921720980718>.
- [35] Yang B, Lei Y, Jia F, Xing S. An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. *Mech Syst Signal Process* 2019;122:692–706. <http://dx.doi.org/10.1016/j.ymsp.2018.12.051>.
- [36] Wang Q, Taal C, Fink O. Integrating expert knowledge with domain adaptation for unsupervised fault diagnosis. *IEEE Trans Instrum Meas* 2021;71:1–12.
- [37] Lu W, Liang B, Cheng Y, Meng D, Yang J, Zhang T. Deep model based domain adaptation for fault diagnosis. *IEEE Trans Ind Electron* 2016;64(3):2296–305.
- [38] Li X, Zhang W, Ding Q, Sun J-Q. Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal Process* 2019;157:180–97.
- [39] Zhang B, Li W, Hao J, Li X-L, Zhang M. Adversarial adaptive 1-D convolutional neural networks for bearing fault diagnosis under varying working condition. 2018, arXiv preprint arXiv:1805.00778.
- [40] Bao H, Yan Z, Ji S, Wang J, Jia S, Zhang G, Han B. An enhanced sparse filtering method for transfer fault diagnosis using maximum classifier discrepancy. *Meas Sci Technol* 2021;32(8):085105.
- [41] Tong Z, Li W, Zhang B, Jiang F, Zhou G. Bearing fault diagnosis under variable working conditions based on domain adaptation using feature transfer learning. *IEEE Access* 2018;6:76187–97. <http://dx.doi.org/10.1109/ACCESS.2018.2883078>.
- [42] Han T, Li Y-F, Qian M. A hybrid generalization network for intelligent fault diagnosis of rotating machinery under unseen working conditions. *IEEE Trans Instrum Meas* 2021;70:1–11.
- [43] Li Y, Song Y, Jia L, Gao S, Li Q, Qiu M. Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning. *IEEE Trans Ind Inf* 2021;17(4):2833–41. <http://dx.doi.org/10.1109/THI.2020.3008010>.
- [44] Jiao J, Zhao M, Lin J, Liang K. Residual joint adaptation adversarial network for intelligent transfer fault diagnosis. *Mech Syst Signal Process* 2020;145:106962. <http://dx.doi.org/10.1016/j.ymsp.2020.106962>.
- [45] Li W, Yang W, Jin G, Chen J, Li J, Huang R, Chen Z. Clustering federated learning for bearing fault diagnosis in aerospace applications with a self-attention mechanism. *Aerospace* 2022;9(9). <http://dx.doi.org/10.3390/aerospace9090516>.
- [46] Liu S, Jiang H, Wu Z, Yi Z, Wang R. Intelligent fault diagnosis of rotating machinery using a multi-source domain adaptation network with adversarial discrepancy matching. *Reliab Eng Syst Saf* 2023;231:109036.
- [47] Zhang Q, Zhao Z, Zhang X, Liu Y, Sun C, Li M, Wang S, Chen X. Conditional adversarial domain generalization with a single discriminator for bearing fault diagnosis. *IEEE Trans Instrum Meas* 2021;70:1–15.
- [48] Yu X, Zhao Z, Zhang X, Sun C, Gong B, Yan R, Chen X. Conditional adversarial domain adaptation with discrimination embedding for locomotive fault diagnosis. *IEEE Trans Instrum Meas* 2020;70:1–12.
- [49] Li Y, Chen Y, Hu Z, Zhang H. Remaining useful life prediction of aero-engine enabled by fusing knowledge and deep learning models. *Reliab Eng Syst Saf* 2023;229:108869.
- [50] Xiong J, Zhou J, Ma Y, Zhang F, Lin C. Adaptive deep learning-based remaining useful life prediction framework for systems with multiple failure patterns. *Reliab Eng Syst Saf* 2023;235:109244.
- [51] 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.
- [52] Zhang A, Wang H, Li S, Cui Y, Liu Z, Yang G, Hu J. Transfer learning with deep recurrent neural networks for remaining useful life estimation. *Appl Sci* 2018;8(12):2416.
- [53] 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.
- [54] Hu T, Guo Y, Gu L, Zhou Y, Zhang Z, Zhou Z. Remaining useful life estimation of bearings under different working conditions via wasserstein distance-based weighted domain adaptation. *Reliab Eng Syst Saf* 2022;224:108526.
- [55] Zhu J, Chen N, Shen C. A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions. *Mech Syst Signal Process* 2020;139:106602.
- [56] Frederick DK, DeCastro JA, Litt JS. User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS). Tech. rep., 2007.
- [57] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 International Conference on Prognostics and Health Management. IEEE; 2008, p. 1–9.
- [58] Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab Eng Syst Saf* 2018;172:1–11.
- [59] Chao MA, Kulkarni C, Goebel K, Fink O. Fusing physics-based and deep learning models for prognostics. *Reliab Eng Syst Saf* 2022;217:107961.
- [60] 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.
- [61] Li Y, Wang N, Shi J, Liu J, Hou X. Revisiting batch normalization for practical domain adaptation. 2016, CoRR arXiv:1603.04779.
- [62] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings; 2010, p. 249–56.
- [63] Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW. A theory of learning from different domains. *Mach Learn* 2010;79(1):151–75.