

Contrastive Adversarial Domain Adaptation for Machine Remaining Useful Life Prediction

Mohamed Ragab¹, Zhenghua Chen^{2*}, Min Wu², Chuan Sheng Foo², Chee Keong Kwoh¹, Ruqiang Yan³, and Xiaoli Li²

Abstract—Enabling precise forecasting of the remaining useful life (RUL) for machines can reduce maintenance cost, increase availability and prevent catastrophic consequences. Data-driven RUL prediction methods have already achieved acclaimed performance. However, they usually assume that the training and testing data are collected from the same condition (same distribution or domain), which is generally not valid in real industry. Conventional approaches to address domain shift problems attempt to derive domain invariant features, but fail to consider target-specific information, leading to limited performance. To tackle this issue, we propose a contrastive adversarial domain adaptation (CADA) method for cross-domain RUL prediction. The proposed CADA approach is built upon an adversarial domain adaptation architecture with a contrastive loss, such that it is able to take target-specific information into consideration when learning domain invariant features. To validate the superiority of the proposed approach, comprehensive experiments have been conducted to predict the RULs of aero-engines across 12 cross-domain scenarios. The experimental results show that the proposed method significantly outperforms state-of-the-arts with over 21% and 38% improvements in terms of two different evaluation metrics.

Index Terms—Remaining Useful Life, Domain Adaptation, Transfer Learning, Deep Learning

I. INTRODUCTION

Prognostics and health management (PHM) is a milestone technology for realizing predictive maintenance of industrial systems, e.g., manufacturing machines and aerospace engines. The PHM technology can shorten inspection time, reduce costs and enable maintenance scheduling in advance [1]. A key task of the PHM technology is the precise prediction of remaining useful life (RUL) of an industrial system. Numerous approaches have been developed for RUL prediction, which can be divided them into two main categories: model-based and data-driven. Model-based approaches require domain expertise to accurately model the dynamics of a system and estimate

the fault progression [2]. These approaches include physics-based methods, empirical-based methods and Kalman/particle filtering techniques [1]. However, they may fail to model the dynamics of highly complex systems. Recently, data-driven prognosis is becoming more and more attractive with the availability of large amount of data and the less requirement of expert knowledge [3]. Data-driven approaches, including conventional machine learning methods and deep learning models, rely on the available data to extract hidden patterns for accurate RUL prediction.

For conventional machine learning based RUL prediction, the first step is to extract various features from different sensor readings such as vibration, temperature and pressure. Then, traditional learning algorithms, such as support vector machines, random forest and artificial neural networks, can be adopted for RUL prediction [4], [5]. These approaches rely heavily on the features extracted from the sensor readings. However, the extraction and selection of these important features require domain knowledge and human intervention.

Deep learning is also popular for RUL prediction [6]. It is able to automatically learn representative features from raw sensory data. Moreover, it jointly optimizes feature learning and RUL prediction in an end-to-end manner, and thus achieves a better generalization performance. Recently, various deep learning algorithms have been used for machine RUL prediction such as convolutional neural network (CNN) [7], [8], deep belief network (DBN) [9], deep autoencoder (DAE) [10], and Long Short-Term Memory (LSTM) [11], [12].

Data-driven approaches can only work well under two main assumptions: (1) training and testing data are collected under the same operating condition, and (2) rich-labeled data are available for the RUL prediction task [13], [14]. These assumptions can be impractical for many real world applications with the following reasons. First, the collection of labelled data (failures) is expensive. For some complex and critical machines, running to failure can be costly and cause catastrophic consequences [14], [15]. Furthermore, machine deterioration process may prolong up to years, which can also limit the availability of faulty data [16]. Second, the labelled data may only be available under a specific working condition, which can be leveraged to build a model for RUL prediction. However, when the working condition changes, the previously trained model often cannot work well, due to the distinct data distributions for different working conditions [13], [15], [17].

With the aforementioned problems, the RUL prediction for scarce-labeled machines/working conditions can be very challenging. Therefore, there is an urgent need for a prognostic

This work is supported by the A*STAR Industrial Internet of Things Research Program under the RIE2020 IAF-PP Grant A1788a0023, and partially supported by National Natural Science Foundation of China (No. 51835009). In addition, the first author of this work is supported by A*STAR SINGA Scholarship. (Zhenghua Chen is the corresponding author.)

¹ Mohamed Ragab and Chee Keong Kwoh are with School of Computer Science and Engineering, Nanyang Technological University, Singapore (Email: mohamedr002@e.ntu.edu.sg, asckkw@ntu.edu.sg).

² Zhenghua Chen, Min Wu, Chuan Sheng Foo and Xiaoli Li are with the Institute for Infocomm Research, A*STAR, Singapore (Email: chen0832@e.ntu.edu.sg, wumin@i2r.a-star.edu.sg, foo_chuan_sheng@i2r.a-star.edu.sg, xlli@i2r.a-star.edu.sg).

³ Ruqiang Yan is with School of Mechanical Engineering, Xi'an Jiaotong University, China (Email: yanruqiang@xjtu.edu.cn).

model that is able to estimate RUL of new working conditions with no labelled data available. Domain adaptation (DA), which enables knowledge transfer from rich-labeled domain to a different but related scarce-labeled domain [17], provides a good candidate solution for this problem. Most of existing DA algorithms are designed for image-related tasks [18]. Recently, some approaches extended DA for fault diagnosis problems (classification problems) to classify faults among different machines or working conditions [10], [19]. However, less attention has been paid to domain adaptation for the RUL prediction which is a typical time-series regression problem.

To promote the intelligent fault prognosis applications with unlabeled data, we propose a novel contrastive adversarial domain adaptation (CADA) approach for machine RUL prediction across different working conditions. More specifically, CADA aims to transfer the knowledge learnt from one working condition to solve the RUL prediction problem in another working condition. Generally, adversarial adaptation approaches aim to find a feature representation of the target domain that can be invariant from the source domain. Existing deep feature extractors with its large complexity can find arbitrary transformation of the target domain that can be similar to the source. However, only finding domain invariant features does not guarantee good performance on the target domain [20], [21]. Specifically, forcing target domain features to be similar to source domain features with no constraints can remove the target specific information, i.e., the mutual information between the target data and the target extracted features, which could hinder the model performance. To handle this issue, inspired by the noise contrastive estimation (NCE), we propose a novel approach that leverages the InfoNCE loss [22] to preserve the structure of the target domain features during the domain adaptation process. We jointly optimize the target feature extractor to minimize both the domain adaptation loss and the InfoNCE loss. Specifically, the domain adaptation loss guides the target feature extractor to produce source-like features, and the InfoNCE loss preserves the target specific features by maximizing the mutual information between the target input data and the target features. Maximizing the mutual information between the input space and the feature space can preserve intrinsic structure of the target data during domain alignment process, which can boost the performance of domain adaptation. We have performed extensive experiments to verify the performance of the proposed CADA method on machine RUL prediction across different working conditions.

The main contributions of this work are summarized as follows:

- We designed a novel adversarial domain adaptation approach for challenging yet practical machine RUL prediction. This approach successfully transfers knowledge for RUL prediction from one condition (distribution/domain) to another.
- We proposed a novel solution based on the InfoNCE loss to learn the invariant representation and preserve the original structure for the target domain. As such, satisfactory performance for RUL prediction can be achieved.

II. RELATED WORKS

In this section, we highlight the related works in data-driven RUL prediction and domain adaptation.

A. Deep Learning for RUL prediction

Deep learning approaches for RUL prediction can be categorized into feed-forward neural networks and recurrent neural networks [6]. For instance, Zhu *et al.*, used CNN to extract features in multiple scales for the detection of the fault growth and the prediction of the machine RUL [8]. Liu *et al.* proposed a CNN network with joint loss to detect fault and predict RUL concurrently [23]. Deutch and He applied a DBN to extract features and a deep neural network to predict the RUL [9].

Recurrent neural network (RNN) with its sequential modeling capability can be more suitable to model dynamic systems. The LSTM is one of the most popular recurrent approaches that can model long-term dependencies and tackle vanishing gradient problems of RNN. In [11], the authors proposed a bidirectional LSTM (BiLSTM) approach with auxiliary features to predict the RUL under multiple operation conditions. Chen *et al.* developed an attention based LSTM approach to adaptively select important features, resulting an accurate prediction of the RUL [12].

B. Domain Adaptation

Most of RUL prediction methods assume: (1) access to enough labeled failure information; (2) training data (source) and testing data (target) are drawn from the same distribution. In reality, labeled data can be scarce and marginal distribution of data can vary according to the variation of working conditions.

A subset of transfer learning named unsupervised domain adaptation (DA) is developed to address distribution shift problem of unlabeled domains. Conventional approaches for DA re-weight source samples according to their similarity with target samples [24]. While other approaches aim to reduce the domain shift problem in the feature space by minimizing the divergence between the source and target features. In [25], the Maximum Mean Discrepancy (MMD) metric was developed to mitigate the domain shift problem. Sun *et al.* aimed to minimize the covariance shift between the source and target features to align the two domains [26]. Recently, adversarial domain adaptation approaches, which intend to find invariant features in both source and target domains, have achieved the state-of-the-art performance. Inspired by Generative Adversarial Networks (GANs), adversarial adaptation entails a domain classifier to discern between the source and target features and a deep network to extract features that can fool the domain classifier. For instance, the authors in [27] proposed a reverse gradient (RevGrad) strategy to adversarially train the domain classifier and the feature extraction network. While in [28], a typical GAN loss was employed with flipped labels to find domain invariant features. Russo *et al.* proposed a generative domain adaptation approach to align the source and target domains. Specifically, they used a bi-directional mapping from source to target and from target to source,

while using self-labeling for the target domain [29]. Satio *et al.* aligned distributions of the source and target domains by designing task-specific decision boundary. To achieve that, they minimized the maximum discrepancy loss between two different classifiers for the same sample [30]. Lee *et al.* proposed a similar approach, which attempts to replace the L1_loss term with a new sliced Wasserstien distance [31]. In [32], a teacher model was employed to generate psoudo labels for the target domain and align the source and target clusters. In [33], the authors proposed a new adversarial loss that aims to align the joint distribution explicitly. Particularly, they introduced a classifier-aware adaptation method, where the classifier has one additional neuron for the domain classification task.

Li *et al.* developed a heterogeneous adaption approach, where the source and target have different feature space. They considered both the sample space and feature space for domain alignment with the MMD. Then, a graph-based sample re-weighting method was used to transfer knowledge on the sample space [34]. In [35], a progressive domain alignment approach has been developed to adapt two heterogeneous domains. Specifically, a shared codebook was employed to align the feature discrepancy while progressively minimizing the domains discrepancies. In [36], the feature space and the sample space were jointly adapted to preserve the local consistency among samples. In [37], the authors designed the maximum density divergence to enforce clustering assumption while adversarially adapting the two domains.

In RUL domain, very few works have tried to address knowledge transfer problem among different domains. Zhang *et al.* proposed a transfer learning approach for the RUL problem, where they trained the model on the source dataset and fine-tuned the model on target working condition [25]. Yet, they assumed accessibility to labeled data for the target domain, which cannot hold for real-world scenarios. Very recently, Costa *et al.* proposed a deep domain adaptation (DDA) method for the RUL prediction problem using unlabeled target domain data. The DDA applied the LSTM network to extract features and the reverse gradient approach to alleviate the domain shift problem [26]. Most of these approaches aim to find a domain invariant features between the source and target domains. Yet, simply enforcing the target features to be similar to the source with no constrains may remove useful target-specific information in target domain, i.e., the mutual information between the target data and the target extracted feature. This would limit the performance of domain adaptation for the RUL prediction task.

Differently, in our method, we develop a robust adversarial domain adaptation approach that can find domain invariant features while preserving the target-specific features. To achieve that, we propose a novel contrastive loss based approach to maximize the mutual information between the input space and the latent space of the target domain data during domain alignment. To the best of our knowledge, the proposed CADA is the first approach that realizes adversarial domain adaptation while preserving the target-specific features for RUL prediction. Specifically, the CADA can find new feature representation of the target domain data that can be similar to the source and have maximum mutual information

with the target where no labelled data are available.

III. METHODOLOGY

A. Problem Formulation and Notations

To clearly formulate the problem, we introduce the basic standard notations of domain adaptation [17]. Let a domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, where \mathcal{X} is the feature space, $X \in \mathcal{X}$, and $P(X)$ is the marginal distribution of data in this feature space. Given a labeled source domain $\mathcal{D}_S = \{\mathcal{X}_S, P_S(X)\}$ and unlabeled target domain $\mathcal{D}_T = \{\mathcal{X}_T, P_T(X)\}$, the unsupervised domain adaptation problem aims to transfer knowledge from the labeled source to improve the performance on the unlabeled target. In our problem, \mathcal{D}_S and \mathcal{D}_T are both multivariate time-series data of aircraft engines under different working/fault conditions. Particularly, we have labeled data from aircraft engines with a specific working/fault condition, and we aim to improve the RUL prediction of unlabeled data with different working/fault conditions. We denote the source domain $\mathcal{D}_S = \{X_S^i, y_S^i\}_{i=1}^{n_S}$, with n_S the total number of samples, where $X_S^i \in \mathbb{R}^{M \times K}$ is the input source sample with M sensors and K time steps, $y_S^i \in \mathbb{R}$ is the corresponding RUL label. Similarly, the unlabeled target domain $\mathcal{D}_T = \{X_T^j\}_{j=1}^{n_T}$, where $X_T^j \in \mathbb{R}^{M \times K}$ and n_T is the number of target domain samples. Table I summarizes the notations used in this paper.

TABLE I: Notations

Notation	Definition
$\mathcal{D}_S/\mathcal{D}_T$	source/target domain
$\mathcal{X}_S/\mathcal{X}_T$	source/target input space
n_S/n_T	number of source/target samples
P_S/P_T	source/target marginal distribution
$\mathbf{f}_S/\mathbf{f}_T$	source/target latent features
E_S/E_T	source/target encoder
D	domain discriminator
R	RUL regressor
M	number of sensors
K	sequence length

B. Overview

Domain adaptation for multivariate time-series regression can be a very challenging task. Therefore, only few works have been presented for RUL estimation problems across domains [14]. In this paper, we develop a novel contrastive adversarial domain adaptation (CADA) approach for machine RUL prediction. Specifically, it is able to transfer the knowledge learned from the data under one condition (labeled source domain) to the data from another condition (unlabeled target domain). The proposed CADA can find domain invariant representations of the target domain data while preserving their intrinsic structure which is crucial to achieve satisfactory performance in the target domain.

Fig. 1 shows the overall framework that presents the detailed steps of learning procedure of the CADA model. The first stage

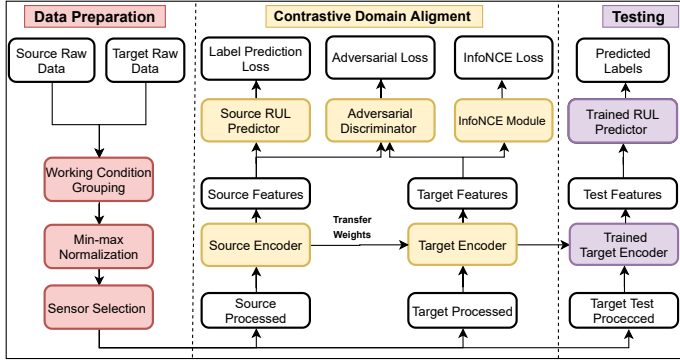


Fig. 1: Flowchart of the proposed approach

involves data preparation for both source and target domains. In the second stage, the source and target features are extracted by the source and target encoders respectively. Given the target features, the target encoder E_T is updated to optimize both the adversarial loss and the InfoNCE loss. In the last stage, the trained target feature extractor and the trained source RUL predictor are combined to predict the RULs for the target domain data. We will provide a detailed explanation of each module in the following subsections.

C. Supervised Pre-training on the Source Domain

In this section, we will present our approach that models the dynamics of multivariate time series and automatically extracts salient features. In addition, we will provide details about the RUL prediction network that maps from the latent features to the RUL.

1) *Recurrent Multivariate Modeling*: Recurrent based approaches are widely adopted for modeling temporal dependencies of time-series data. But RNNs often suffer from the problem of vanishing gradient with long-term sequences [38]. Alternatively, the LSTM which is a strong variant of RNN can handle long-term dependencies and tackle vanishing gradient problem. In this work, we design a very deep bi-directional LSTM network with 5 successive layers for automatic and representative feature extraction. The LSTM feature extractor represents the multivariate time series to a single-vector hidden representation as shown in Fig. 2. Specifically, the LSTM network can be represented as multiple sequential feed-forward layers. The transition function between these layers is a key function to model the temporal dependency along the data, which can be formulated as follows:

$$\mathbf{h}_k, \mathbf{c}_k = H_{cell}(\mathbf{x}_k, \mathbf{h}_{k-1}, \mathbf{c}_{k-1}) \quad (1)$$

where H_{cell} receives the current input \mathbf{x}_k , the previous hidden \mathbf{h}_{k-1} , and the previous memory cell \mathbf{c}_{k-1} . The output will be the updated hidden \mathbf{h}_k and cell \mathbf{c}_k at the current time step as show in Fig. 3. The following equations formalize the transition function of the LSTM cell at time step k :

$$\mathbf{i}_k = \sigma(V_i \mathbf{x}_k + W_i \mathbf{h}_{k-1} + \mathbf{b}_i), \quad (2)$$

$$\mathbf{e}_k = \sigma(V_e \mathbf{x}_k + W_e \mathbf{h}_{k-1} + \mathbf{b}_e), \quad (3)$$

$$\mathbf{f}_k = \sigma(V_f \mathbf{x}_k + W_f \mathbf{h}_{k-1} + \mathbf{b}_f), \quad (4)$$

$$\mathbf{g}_k = \delta(V_g \mathbf{x}_k + W_g \mathbf{h}_{k-1} + \mathbf{b}_g), \quad (5)$$

$$\mathbf{c}_k = \mathbf{e}_k \odot \mathbf{c}_{k-1} + \mathbf{i}_k \odot \mathbf{g}_k, \quad (6)$$

$$\mathbf{h}_k = \mathbf{f}_k \odot \delta(\mathbf{c}_k), \quad (7)$$

where σ and δ represent nonlinear activation functions of logistic sigmoid and hyperbolic tangent respectively, $\mathbf{x}_k \in \mathbb{R}^M$, the $V_* \in \mathbb{R}^{M \times d}$ and $W_* \in \mathbb{R}^{d \times d}$ are shared model weights. The operator \odot represents the element-wise multiplication.

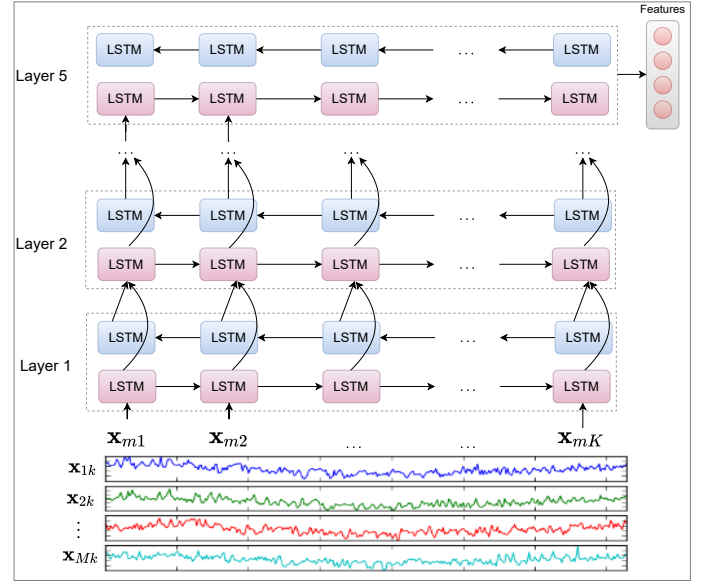


Fig. 2: Deep BiLSTM feature extractor

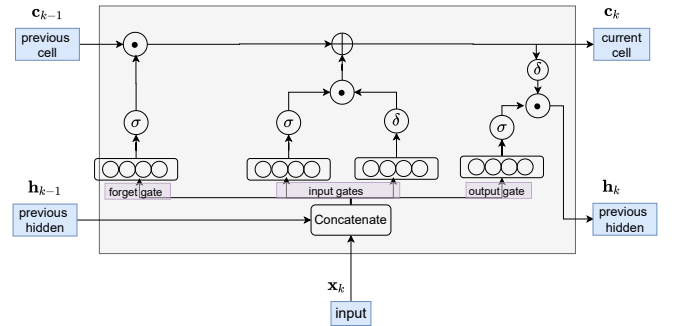


Fig. 3: Structure of LSTM cell

2) *RUL Prediction Network*: Given the extracted features from the LSTM feature extractor $\mathbf{f}_S = E_S(X_S)$. The RUL predictor is a multi-layer network $R: \mathbb{R}^d \rightarrow \mathbb{R}$ that maps the latent features into the corresponding RUL value. The RUL predictor R and the feature extractor E_S are trained in an end-to-end manner using the mean square error loss between the predicted RULs and the true RULs, which can be formalized as follows:

$$\mathcal{L}_{mse} = \frac{1}{n_S} \sum_{i=1}^{n_S} (\hat{y}_S^{(i)} - y_S^{(i)})^2 \quad (8)$$

where $\hat{y}_S = R(E_S(X_S))$ is the predicted RUL label, y_S is the ground-truth RUL values, and n_S is the number of source samples.

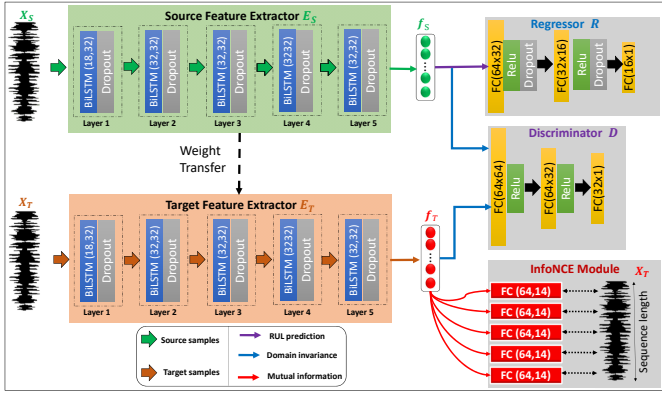


Fig. 4: Proposed CADA Approach

D. Contrastive Adversarial Domain Alignment

The contrastive adversarial adaption module consists of a domain discriminator D and the InfoNCE module as shown in Fig. 4. Firstly, the weights of the trained source feature extractor are adopted to initialize the target feature extractor. The output features from both the source and target domains are fed into an adversarial discriminator network to minimize the discrepancy. Concurrently, the target features are fed into the InfoNCE loss module to preserve the target specific features during the alignment process. In particular, the InfoNCE loss will maximize the mutual information between the learned target domain features and the input target domain data to preserve the task specific features during adversarial alignment process. Detailed procedures are presented in the following paragraphs.

1) *Adversarial Adaptation Module*: Let E_S and R_S be the source-trained LSTM feature extractor and the RUL predictor respectively. To predict the RUL labels of the unlabeled target domain data, we can naively initialize our target model (i.e., E_T and R_T) with pre-trained source models. However, due to the large discrepancy among the data from different working/fault conditions, the model can fail to predict RUL accurately. To tackle this domain discrepancy problem, we adversarially train the LSTM feature extractor against a domain discriminator network to minimize the distribution differences between the source features and the target features. Specifically, the domain discriminator network D is trained to

Algorithm 1: Contrastive Adversarial Domain Adaptation

Input: Source domain: $\mathcal{D}_S = \{X_S^i, y_S^i\}_{i=1}^{n_S}$
Target domain: $\mathcal{D}_T = \{X_T^i\}_{i=1}^{n_T}$
Output: Trained target encoder E_T
 $E_S \leftarrow$ Trained source encoder
 $E_T \leftarrow$ Initialize with E_S parameters
 $D \leftarrow$ Domain Discriminator
for number of iterations **do**
1. Sample mini-batch of m source samples $X_S \sim P_S$
2. Sample mini-batch of m target samples $X_T \sim P_T$
3. Extract source features: $\mathbf{f}_S = E_S(X_S)$
4. Extract target features: $\mathbf{f}_T = E_T(X_T)$
5. Feed \mathbf{f}_S and \mathbf{f}_T to D
6. Compute adversarial loss \mathcal{L}_{adv} by Eq. 9
7. Update D by \mathcal{L}_{adv}
8. Compute InfoNCE loss $\mathcal{L}_{\text{InfoNCE}}$ based on Algorithm 2
9. Update E_T by $\mathcal{L} = \mathcal{L}_E + \lambda \mathcal{L}_{\text{InfoNCE}}$
end

discern between the source and target features. Concurrently, we train the target feature extractor E_T to produce target features such that the domain discriminator network cannot distinguish them from the source features. The adversarial training between the discriminator network D and the target E_T can be expressed as follows:

$$\min_{E_T} \max_D \mathcal{L}_{adv} = \mathbb{E}_{X_S \sim P_S} [\log D(E_S(X_S))] + \mathbb{E}_{X_T \sim P_T} [\log(1 - D(E_T(X_T)))] \quad (9)$$

where X_S and X_T are the source and target samples respectively. The target feature extractor E_T is updated to minimize \mathcal{L}_{adv} , and the discriminator network D is adversarially trained to maximize \mathcal{L}_{adv} . Eventually, the trained target feature extractor E_T will be able to extract features \mathbf{f}_T that have minimum discrepancy from the source features.

Algorithm 2: Contrastive Loss

Input: $X_T = \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$, $\mathbf{f}_T = E_T(X_T)$
Output: Contrastive Loss $\mathcal{L}_{\text{InfoNCE}}$
 $\Theta^k \leftarrow$ Linear layer at timestep k
for K timesteps **do**
1. $\mathbf{q}_k \leftarrow \Theta^k \mathbf{f}_T$
2. Apply $\phi_k(\mathbf{x}_k, \mathbf{q}_k^T)$ as in Eq. 11
3. Compute $\mathcal{L}_{\text{InfoNCE}}$ using Eq. 12
end
return $\mathcal{L}_{\text{InfoNCE}}$

2) *Contrastive Estimation Module*: Adversarial domain adaptation can successfully find target domain features that are invariant from the source features. However, it can remove task-specific information from the target features to minimize the adversarial loss, which can deteriorate the performance on the target domain – even with perfect domain alignment. Hence, it is required to preserve target-specific features during the domain alignment task. To achieve that, we rely on InfoNCE loss [22] to maximize the mutual information between the encoded representations of the target domain and the original inputs, as shown in Algorithm 2. Given a sample

$X_T \sim \mathcal{X}_T$, where $X_T \in \mathcal{R}^{M \times K}$, we apply the target encoder E_T on X_T to obtain its corresponding feature representation $\mathbf{f}_T = E_T(X_T)$. To model the mutual information between \mathbf{x}_k , and \mathbf{f}_T , following the previous studies [39], we define a density ratio function ϕ_k at each time step, which formalized as follows.

$$\phi_k(\mathbf{x}_k; \mathbf{f}_T) \propto \frac{p(\mathbf{x}_k | \mathbf{f}_T)}{p(\mathbf{x}_k)} \quad (10)$$

By maximizing the mutual between the latent target features \mathbf{f}_T and the input \mathbf{x}_k , we can preserve the common latent variables between the target features \mathbf{f}_T and the input \mathbf{x}_k . To compute ϕ_k , the latent features \mathbf{f}_T and the input \mathbf{x}_k should be mapped to the same dimension. To achieve that, we use a fully connected network $\Theta : \mathbb{R}^d \rightarrow \mathbb{R}^M$ that maps feature dimension d to input dimension M . Thereafter, the density ratio ϕ_k is estimated by a dot product between the transformed features $\mathbf{q}_k = \Theta^k(\mathbf{f}_T)$ and the the original input \mathbf{x}_k , which can be compactly represented as follows:

$$\phi_k(\mathbf{x}_k, \mathbf{f}_T) = \mathbf{x}_k^\top \mathbf{q}_k \quad (11)$$

where $\Theta^k = \{\theta_1, \dots, \theta_M\}$ are the weights of a fully connected layer at time step k . Note that Θ^k is different among the time steps. To maximize the density ratio function, we jointly op-

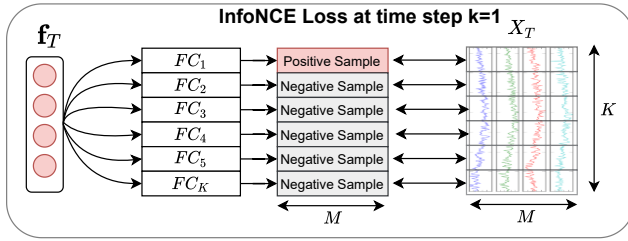


Fig. 5: Computation of InfoNCE loss at time step k=1

imize the target feature extractor E_T and the fully connected layers Θ using the contrastive estimation loss. The InfoNCE loss maximizes the mutual information by contrasting between the positive and negative samples. Fig. 5 illustrates the positive and negative samples for time step $k = 1$. The overall InfoNCE loss can be formulated as

$$\min_{E_T, \Theta} \mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}_{X_T} \left[\log \frac{e^{\phi_k(\mathbf{x}_k, \mathbf{f}_T)}}{\sum_{\mathbf{x}_j \in X_T} e^{\phi_k(\mathbf{x}_j, \mathbf{f}_T)}} \right] \quad (12)$$

The optimal probability of the NCE loss $p(d = k | X_T, \mathbf{f}_T)$ can be formulated as:

$$p(d = k | X_T, \mathbf{f}_T) = \frac{p(\mathbf{x}_k | \mathbf{f}_T) \prod_{l \neq k} p(\mathbf{x}_l)}{\sum_{j=1}^K p(\mathbf{x}_j | \mathbf{f}_T) \prod_{l \neq j} p(\mathbf{x}_l)} \quad (13)$$

$$= \frac{\frac{p(\mathbf{x}_k | \mathbf{f}_T)}{p(\mathbf{x}_k)}}{\sum_{j=1}^K \frac{p(\mathbf{x}_j | \mathbf{f}_T)}{p(\mathbf{x}_j)}} \quad (14)$$

By substituting Eq. 12 into the above equations, we can formalize the mutual information in terms of the InfoNCE loss $\mathcal{L}_{\text{InfoNCE}}$, detailed derivation can be found in [39]. The resulting formula can be written as:

$$I(\mathbf{x}_k, \mathbf{f}_T) = \log(K) - \mathcal{L}_{\text{InfoNCE}} \quad (15)$$

where $I(\cdot)$ represents the mutual information between \mathbf{x}_k and \mathbf{f}_T . It can be seen that minimizing InfoNCE loss is maximizing the lower bound of $I(\mathbf{x}_k, \mathbf{f}_T)$, which in turn maximizing the mutual information.

3) *Overall Loss Function:* In this work, the adversarial adaptation loss and contrastive estimation loss are jointly optimized in an end-to-end manner. The total domain alignment loss can be summarized as follows:

$$\begin{aligned} \min_{E_T, \Theta} \max_D V(D, E_T, \Theta) \\ = \mathcal{L}_{adv} + \lambda \mathcal{L}_{\text{InfoNCE}} \\ = \mathbb{E}_{X_S \sim p_S} [\log D(\mathbf{f}_S)] + \\ \mathbb{E}_{X_T \sim p_T} \left[\log(1 - D(\mathbf{f}_T)) - \lambda \log \frac{e^{\phi_k(\mathbf{x}_k, \mathbf{f}_T)}}{\sum_{\mathbf{x}_j \in X_T} e^{\phi_k(\mathbf{x}_j, \mathbf{f}_T)}} \right] \end{aligned} \quad (16)$$

where \mathcal{L}_{adv} is the adversarial loss, \mathcal{L}_{NCE} is the contrastive estimation loss, and λ is a weight parameter that controls the proportion of learning domain invariant features and preserving task-specific information.

IV. EXPERIMENTS AND RESULTS

A. Preparation of Data

To evaluate the performance of our approach, we employ the popular C-MAPSS [40] benchmark dataset which describes the run-to-fail experiments of aero-engines. It contains four different subsets, namely FD001, FD002, FD003 and FD004, which differ in terms of working conditions, fault modes, life spans, and number of engines, as shown in Table II. Particularly, “# Training engines” represents the number of available engines to train the model, while “# Testing engines” represents the number engines available for testing. “# Training samples” is the total number of training samples per data subset. “# Testing samples” is the total number of testing samples per data subset. “# Max life span” is the maximal number of cycles that an engine takes to go from healthy to the failure condition. “# Operating conditions” represents the number of operating conditions. “# Fault types” represents the number of failure modes occurblue. Particularly, we take the scenario FD001 \rightarrow FD002 as an example. We use both the training samples of FD001 (17731 samples with labels) and FD002 (48558 samples without labels) to train our CADA model.

Different types of sensors have been used to monitor rotating components of each engine. Here, we briefly introduce our procedure for data processing. First, we select sensors that are informative for RUL prediction, following the previous studies [11], [12]. The informative sensors are those sensors which can show clear degradation trend from run to failure. Here, we visualize the sensor readings of randomly selected engines. Fig. 6 and Fig. 7 show the sensor readings from FD001 and FD002 subsets respectively. Clearly, some sensors are almost constant during the whole degradation, which can hinder the model from correctly modeling the deterioration process. In the cross-domain problem, we intend to transfer the

knowledge from a source data subset (e.g., FD001) to a target data subset (e.g., FD002). Thus, we only select the common sensors among source and target domains which are the most informative ones. Following this strategy, we have selected the following sensors, i.e., S2, S3, S4, S7, S8, S9, S11, S12, S13, S14, S15, S17, S20 and S21.

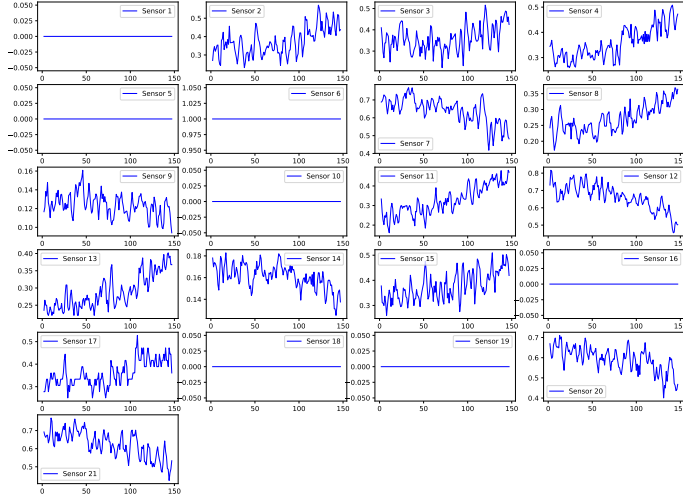


Fig. 6: The readings of 21 sensors for a randomly selected engine in FD001 dataset

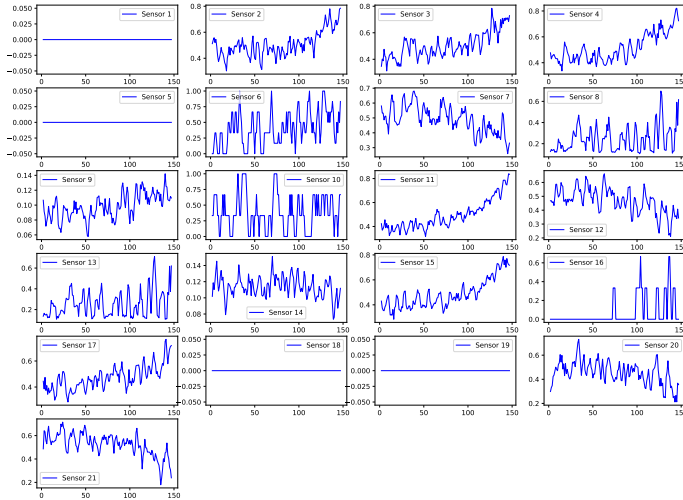


Fig. 7: The readings of 21 sensors for a randomly selected engine in FD002 dataset

Second, the same type of sensors may have quite different readings under different working conditions. To reduce the effect of working conditions, we apply the Min-Max normalization with respect to each working condition. As such, the data under different working conditions is normalized into the range of $[0, 1]$. Third, we apply sliding windows to generate data samples from run-to-fail cycles. Following previous studies [11], [12], we set the window size and the step size as 30 and 1, respectively. Moreover, a piece-wise linear RUL [2] is adopted instead of the true RUL, i.e., if the true RUL is larger than the maximal RUL, then it is set to the maximal RUL.

TABLE II: Properties of C-MAPSS Dataset

Dataset	FD001	FD002	FD003	FD004
# Training engines	100	260	100	249
# Testing engines	100	259	100	248
# Training samples	17731	48558	21220	56815
# Testing samples	100	259	100	248
# Max life spans (cycles)	362	378	512	128
# Operating conditions	1	6	1	6
# Fault types	1	1	2	2

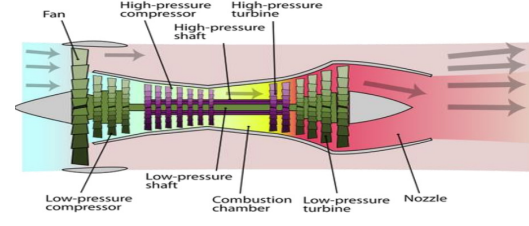


Fig. 8: Diagram of the engines in C-MAPSS dataset [40]

B. Experimental Settings

Our CADA approach consists of five main models: Source feature extractor (E_S), target feature extractor (E_T), RUL predictor (R), domain discriminator (D), and InfoNCE module. Detailed structure of each model has been shown in Fig. 4. Specifically, the source and target feature extractors are deep BiLSTM networks with 5 layers, where each layer has 32 neurons. The Discriminator is composed of three fully connected (FC) layers with 64, 32 and 1 hidden neurons. The RUL predictor also consists of three FC layers, i.e., hidden layer 1 with 32 neurons, hidden layer 2 with 16 neurons, and output layer with a single neuron. Each layer is followed by nonlinear activation function called rectified linear unit (ReLU) and the dropout regularization technique to relieve the over-fitting problem. The detailed architecture of the RUL predictor is shown in Fig. 9.

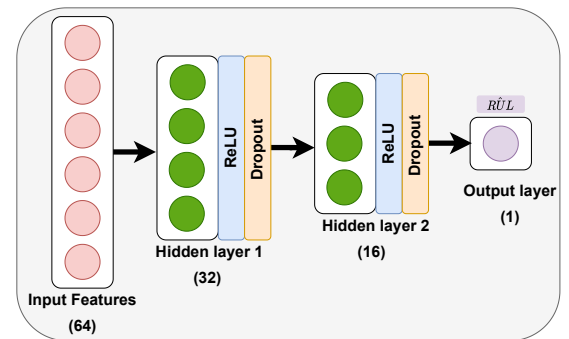


Fig. 9: Detailed architecture of the RUL predictor network.

To train our model, we adopt the mini-batch training with a batch size of 256. To reduce overfitting, dropout regularization is adopted across the whole structure and the dropout ratio is set to be 0.5. We use Adam optimizer to minimize the joint loss

with the learning rate of $0.5e-4$ for the feature extractor and the domain discriminator. As the InfoNCE module is trained from scratch during the alignment process, we apply larger learning rate of $1e-2$. The training epochs range from 20 to 150 epochs. The weight of the InfoNCE loss λ can vary across different cross-domain scenarios and later we will show its effect on the prediction performance through a sensitivity analysis.

To quantify the performance of models, we adopt two evaluation metrics, i.e., root mean square error (RMSE) and score metric, as in [11], [14]. The RMSE metric is defined as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}, \quad (17)$$

where \hat{y}_i and y_i represent the predicted RUL and ground-truth RUL respectively.

The RMSE metric treats the early and late RUL predictions equally. For prognostics applications, late RUL prediction can be more harmful to the systems. To handle this issue, the score metric is used to impose bitter penalty for late RUL predictions. It can be formalized as follows:

$$Score = \begin{cases} \frac{1}{N} \sum_{i=1}^N (e^{\frac{\hat{y}_i - y_i}{13} - 1}), & \text{if } (\hat{y}_i < y_i) \\ \frac{1}{N} \sum_{i=1}^N (e^{\frac{\hat{y}_i - y_i}{10} - 1}), & \text{if } (\hat{y}_i > y_i) \end{cases} \quad (18)$$

C. Comparison with State-of-the-art Methods

To evaluate our approach in cross-domain scenarios, we train the model using a labeled source domain (e.g., FD001) and evaluate on an unlabeled target domain (e.g., FD002, FD003, or FD004). As we have 4 sub-datasets (i.e., domains), we thus have 12 cross-domain scenarios. In this paper, we implement five state-of-the-art approaches as follows. In addition, we report the average performance (i.e., RMSE and Score) over 5 consecutive runs with different random seeds.

- Correlation alignment (CORAL) [26]: CORAL minimizes the covariance shift between the source and target features to align the distribution.
- Deep domain confusion (DDC) [25]: DDC employs a distance metric called MMD to confuse the source and target features.
- Wasserstein distance guided representation learning (WDGRL) [41]: WDGRL employs a neural network to measure the empirical Wasserstein distance, while utilizing the feature extractor network to minimize this distance between the source and target domain.
- Adversarial discriminative domain adaption (ADDA) [28]: ADDA uses a typical GAN loss to find target domain features that can be similar to the source features.
- Deep domain adaptation (DDARUL) [14]: In DDARUL, an LSTM feature extractor is trained to confuse the source and target domains, while a domain classifier network is trained to classify between the source and target features.

Table III shows the experimental results. the CADA outperforms all the competing approaches across the 12 cross-domain scenarios in terms of both RMSE and Score. In

addition, we observe that knowledge transfer between simple and complex datasets is challenging due to the large domain shift, yet our CADA can successfully align the two distant domains. For example, FD001 and FD004 are the simplest and most complex data subsets respectively. As shown in Table III, simply forcing the features to be similar among these two datasets can significantly harm the performance. Overall, we achieve significant improvement over the second best approach (underlined) in each scenario with an average of more than 21% and 38% for RMSE and Score respectively. Our domain adaptation strategy can preserve task-specific information and our proposed deep feature extractor has large generalization capability, leading to the superior performance of our proposed CADA.

D. Model Ablation Study

Here, we perform our ablation study to verify the contribution of individual components in our CADA approach. We derive two variants of CADA, namely, ‘‘Source-Only’’ and ‘‘w/o InfoNCE’’. In particular, ‘‘Source-Only’’ refers to the non-adapted version of our model, whereas the ‘‘w/o InfoNCE’’ is our adversarial adaptation approach without using the contrastive estimation loss.

Table IV shows the comparison between the CADA and its two variants. We observe that the ‘‘Source-Only’’ has the worst performance, indicating that the big gap between the source and target domain data distributions. The proposed CADA method outperforms the one without the InfoNCE loss in most of cases, which signifies the effectiveness of the InfoNCE loss on domain adaptation based RUL prediction.

E. Sensitivity Analysis

1) *Coefficient of the InfoNCE loss λ* : In this section, we investigate the sensitivity of the proposed CADA with respect to the coefficient of the InfoNCE loss λ . We have conducted experiments with λ varying from 0.001 to 1.0 for the 12 cross-domain scenarios. The results are shown in Fig. 10. It can be found that different scenarios may require different λ to boost the performance. Table V summarizes the selected λ values for the 12 cross-domain scenarios in experiments.

2) *The number of LSTM Layers*: Another important hyperparameter for the proposed method is the number of LSTM layers. We have investigated the model performance with different number of LSTM layers, i.e., 1, 3, 5 and 7, in order to find a balance between the model performance and the training time. Fig. 11 shows the experimental results. We can find that the proposed method with 5 layers can achieve the best performance in most of scenarios. However, some scenarios require fewer layers to obtain a better or comparable performance. For example, for the scenario FD004 \rightarrow FD003, the method with 7 LSTM layers performs the best. However, the performance of the method with 1 LSTM layer is comparable to the best performance, but much more efficient. In this case, using a single LSTM layer is more reasonable when considering the balance between the performance and the efficiency of the algorithm. Table V shows the selected number of LSTM layers for each cross-domain scenario.

TABLE III: Comparison of the proposed method against state-of-the-art approaches

Metric	RMSE						Score					
Method	CORAL [26]	WDGRL [41]	DDC [25]	ADDA [28]	RULDDA [14]	CADA	CORAL [26]	WDGRL [41]	DDC [25]	ADDA [28]	RULDDA [14]	CADA
FD001→FD002	22.85	<u>21.46</u>	44.05	31.26	24.08	19.52	2798	33160	5958	4865	<u>2684</u>	2122
FD001→FD003	44.21	71.7	<u>39.62</u>	57.09	43.08	39.58	56991	15936	288061	32472	<u>10259</u>	8415
FD001→FD004	50.03	57.24	<u>44.35</u>	56.66	45.7	31.23	52053	86139	156224	68859	<u>26981</u>	11577
FD002→FD001	24.43	<u>15.24</u>	46.96	19.73	23.91	13.88	3590	157672	<u>640</u>	689	2430	351
FD002→FD003	42.66	41.45	39.87	<u>37.22</u>	47.26	33.53	23071	19053	62823	<u>11029</u>	12756	5213
FD002→FD004	52.12	<u>37.62</u>	43.99	37.64	45.17	33.71	62852	52372	44872	<u>16856</u>	25738	15106
FD003→FD001	40.33	36.05	39.95	40.41	<u>27.15</u>	19.54	4581	18307	25826	32451	<u>2931</u>	1451
FD003→FD002	56.67	40.11	44.07	42.53	<u>30.42</u>	19.33	73026	32112	1012978	459911	<u>6754</u>	5257
FD003→FD004	38.16	<u>29.98</u>	47.46	31.88	31.82	20.61	11407	296061	275665	82520	<u>5775</u>	3219
FD004→FD001	51.44	42.01	41.55	37.81	<u>32.37</u>	20.10	154842	45394	162100	43794	<u>13377</u>	1840
FD004→FD002	31.61	35.88	43.99	36.67	<u>27.54</u>	18.5	38095	38221	179243	23822	<u>4937</u>	4460
FD004→FD003	30.44	<u>18.18</u>	44.47	23.59	23.31	14.49	6919	77977	1623	<u>1117</u>	1679	682

TABLE IV: Ablation study of the proposed approach

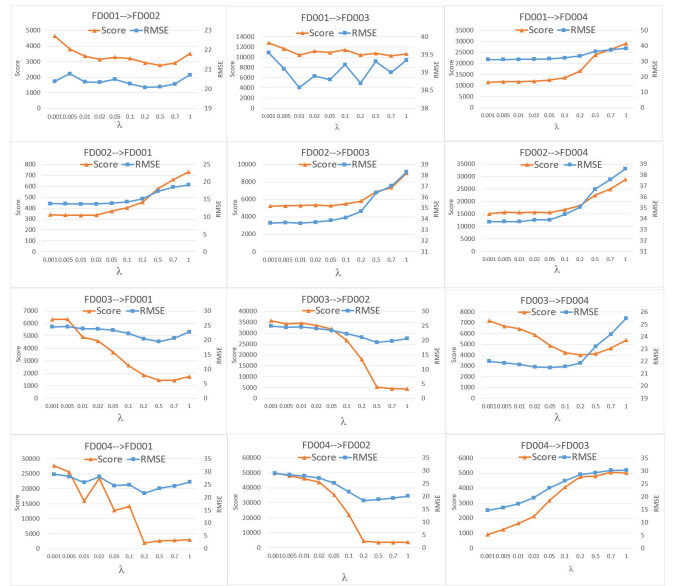
Metric	RMSE			Score		
Method	Source-Only	w/o InfoNCE	CADA	Source-Only	w/o InfoNCE	CADA
FD001→FD002	20.62	20.48	19.52	5448	4600	2122
FD001→FD003	55.09	39.33	39.58	31062	11866	8415
FD001→FD004	36.81	31.19	31.23	20786	11713	11577
FD002→FD001	15.29	13.82	13.88	543	342	351
FD002→FD003	35.46	33.65	33.53	5339	5350	5213
FD002→FD004	37.66	33.82	33.71	19807	15070	15106
FD003→FD001	39.03	24.66	19.54	5700	6469	1451
FD003→FD002	46.11	24.84	19.33	72405	35036	5257
FD003→FD004	31.44	21.94	20.61	40772	8873	3219
FD004→FD001	37.90	26.34	20.10	99597	14985	1840
FD004→FD002	32.98	28.73	18.50	62345	48726	4460
FD004→FD003	19.47	14.38	14.49	2470	793	682

TABLE V: The values of λ and the number of LSTM layers for different scenarios.

Scenario	λ	Number of Layers
FD001→FD002	0.2	5
FD001→FD003	0.2	3
FD001→FD004	0.001	5
FD002→FD001	0.001	5
FD002→FD003	0.001	5
FD002→FD004	0.001	5
FD003→FD001	0.5	5
FD003→FD002	0.5	5
FD003→FD004	0.2	3
FD004→FD001	0.2	3
FD004→FD002	0.2	5
FD004→FD003	0.001	1

V. CONCLUSION

In this paper, we proposed a novel contrastive adversarial domain adaptation (CADA) approach that can automatically find domain invariant features while preserving domain specific information for machine RUL prediction. The proposed CADA method was built upon adversarial domain adaptation architecture with the novel InfoNCE loss. We have performed

Fig. 10: The experimental results with different λ values for 12 cross-domain scenarios

extensive experiments to verify the effectiveness of the CADA method. More specifically, a detailed comparison has been made with five state-of-the-art approaches for domain adaptation in RUL prediction. Our experimental results show that the proposed CADA method significantly outperforms all the state-of-the-arts. Moreover, we also conduct ablation study to show the effectiveness of the InfoNCE loss when performing domain adaptation.

REFERENCES

- [1] G. J. Vachtsevanos and G. J. Vachtsevanos, *Intelligent fault diagnosis and prognosis for engineering systems*. Wiley Hoboken, 2006, vol. 456.
- [2] H. Miao, B. Li, C. Sun, and J. Liu, "Joint learning of degradation assessment and rul prediction for aeroengines via dual-task deep lstm networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5023–5032, 2019.
- [3] Y. Jiang and S. Yin, "Recent advances in key-performance-indicator oriented prognosis and diagnosis with a matlab toolbox: Db-kit," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2849–2858, 2018.



Fig. 11: The experimental results with different number of LSTM layers for 12 cross-domain scenarios

- [4] W. Li, S. Zhang, and S. Rakheja, "Feature denoising and nearest-farthest distance preserving projection for machine fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 393–404, 2015.
- [5] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mechanical Systems and Signal Processing*, vol. 107, pp. 241–265, 2018.
- [6] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.
- [7] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.
- [8] J. Zhu, N. Chen, and W. Peng, "Estimation of bearing remaining useful life based on multiscale convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3208–3216, 2018.
- [9] J. Deutsch and D. He, "Using deep learning-based approach to predict remaining useful life of rotating components," *systems man and cybernetics*, vol. 48, no. 1, pp. 11–20, 2018.
- [10] M. Ma, C. Sun, and X. Chen, "Deep coupling autoencoder for fault diagnosis with multimodal sensory data," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1137–1145, 2018.
- [11] C.-G. Huang, H.-Z. Huang, and Y.-F. Li, "A bidirectional lstm prognostics method under multiple operational conditions," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8792–8802, 2019.
- [12] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, "Machine remaining useful life prediction via an attention based deep learning approach," *IEEE Transactions on Industrial Electronics*, 2020.
- [13] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 9, pp. 7316–7325, 2019.
- [14] P. R. d. O. da Costa, A. Akçay, Y. Zhang, and U. Kaymak, "Remaining useful lifetime prediction via deep domain adaptation," *Reliability Engineering & System Safety*, vol. 195, p. 106682, 2020.
- [15] W. Mao, J. He, and M. J. Zuo, "Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1594–1608, April 2020.
- [16] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to reliability prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018.
- [17] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.
- [18] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.
- [19] R. Yan, F. Shen, C. Sun, and X. Chen, "Knowledge transfer for rotary machine fault diagnosis," *IEEE Sensors Journal*, 2019.
- [20] R. Shu, H. Bui, H. Narui, and S. Ermon, "A dirt-t approach to unsupervised domain adaptation," in *ICLR 2018 : International Conference on Learning Representations 2018*, 2018.
- [21] H. Zhao, R. T. des Combes, K. Zhang, and G. Gordon, "On learning invariant representation for domain adaptation," in *ICML 2019 : Thirty-sixth International Conference on Machine Learning*, 2019, pp. 7523–7532.
- [22] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *ICML 2020: 37th International Conference on Machine Learning*, 2020.
- [23] R. Liu, B. Yang, and A. G. Hauptmann, "Simultaneous bearing fault recognition and remaining useful life prediction using joint-loss convolutional neural network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 87–96, Jan 2020.
- [24] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in neural information processing systems*, 2007, pp. 601–608.
- [25] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.
- [26] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 153–171.
- [27] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 1180–1189.
- [28] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.
- [29] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo, "From source to target and back: symmetric bi-directional adaptive gan," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8099–8108.
- [30] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3723–3732.
- [31] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 285–10 295.
- [32] Z. Deng, Y. Luo, and J. Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9944–9953.
- [33] H. Tang and K. Jia, "Discriminative adversarial domain adaptation," in *AAAI*, 2020, pp. 5940–5947.
- [34] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Transfer independently together: A generalized framework for domain adaptation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 49, no. 6, pp. 2144–2155, 2019.
- [35] —, "Heterogeneous domain adaptation through progressive alignment," *IEEE Transactions on Neural Networks*, vol. 30, no. 5, pp. 1381–1391, 2019.
- [36] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, "Locality preserving joint transfer for domain adaptation," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6103–6115, 2019.
- [37] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and H. T. Shen, "Maximum density divergence for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
- [38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [39] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
- [40] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 International Conference on Prognostics and Health Management*. IEEE, 2008, pp. 1–9.
- [41] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.



Mohamed Ragab received the B.Sc. degree (First Class Hons.) and M.Sc. degree from the Department of Electrical Engineering, Aswan University, in 2014 and 2017, respectively. He is currently pursuing his Ph.D. degree from the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. Concurrently, he is with Machine Intelligence (MI) department at the Institute of Infocomm Research (I2R), A*STAR. His research interests include deep learning, transfer learning, and intelligent fault diagnosis and prognosis.



learning, deep learning, transfer learning and related applications.

Zhenghua Chen received the B.Eng. degree in mechatronics engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2011, and Ph.D. degree in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2017. He has been working at NTU as a research fellow. Currently, he is a scientist at Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include sensory data analytics, machine



learning, data mining and bioinformatics.

Min Wu is currently a senior scientist at Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. He received his Ph.D. degree in Computer Science from Nanyang Technological University (NTU), Singapore, in 2011 and B.S. degree in Computer Science from University of Science and Technology of China (USTC) in 2006. He received the best paper awards in InCoB 2016 and DASFAA 2015. He also won the IJCAI competition on repeated buyers prediction in 2015. His current research interests include machine



Chuan Sheng Foo received his BS, MS and PhD degrees in Computer Science from Stanford University, in 2008, 2012 and 2017, respectively. He leads a research group at the Institute for Infocomm Research, A*STAR, Singapore that focuses on developing data-efficient deep learning algorithms that can learn from less labeled data.



soft computing and graph-based inference; application areas include bioinformatics and engineering. He has done significant research work in his research areas and has published many quality international conferences and journal papers. He is a member of the Association for Medical and Bio-Informatics, Imperial College Alumni Association of Singapore. He has provided many services to professional bodies in Singapore and was conferred the Public Service Medal by the president of Singapore in 2008.

Chee Keong Kwoh received the bachelor's degree in electrical engineering (first class) and the master's degree in industrial system engineering from the National University of Singapore, Singapore, in 1987 and 1991, respectively. He received the Ph.D. degree from the Imperial College of Science, Technology, and Medicine, University of London, in 1995. He has been with the School of Computer Engineering, Nanyang Technological University (NTU), since 1993. He is the Deputy Executive Director of PaCE at NTU. His research interests include data mining,



interests include data analytics, machine learning, and energy-efficient sensing and sensor networks for the condition monitoring and health diagnosis of large-scale, complex, dynamical systems. He holds 28 patents, published two books and over 200 papers in technical journals and conference proceedings.

From 2009 to 2018, he was a Professor of the School of Instrument Science and Engineering at the Southeast University, Nanjing, China. He joined the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China, in 2018. His research interests include data analytics, machine learning, and energy-efficient sensing and sensor networks for the condition monitoring and health diagnosis of large-scale, complex, dynamical systems. He holds 28 patents, published two books and over 200 papers in technical journals and conference proceedings.

Dr. Yan is a Fellow of ASME (2019). His honors and awards include IEEE Instrumentation and Measurement Society Technical Award (2019), the New Century Excellent Talents in University Award from the Ministry of Education in China (2009), and multiple Best Paper Awards. He is an Associate Editor-in-Chief for the IEEE Transactions on Instrumentation and Measurement and Associate Editor for the IEEE Systems Journal and IEEE Sensors Journal.

Ruqiang Yan (M'07-SM'11) received the M.S. degree in precision instrument and machinery from the University of Science and Technology of China, Hefei, China, in 2002, and the Ph.D. degree in mechanical engineering from the University of Massachusetts Amherst, MA, USA, in 2007.



paper/benchmark competition awards.

Xiaoli Li is currently a principal scientist at the Institute for Infocomm Research, A*STAR, Singapore. He also holds adjunct professor positions at Nanyang Technological University. His research interests include data mining, machine learning, AI, and bioinformatics. He has been serving as a (senior) PC member/workshop chair/session chair in leading data mining and AI related conferences (including KDD, ICDM, SDM, PKDD/ECML, WWW, IJCAI, AAAI, ACL and CIKM). Xiaoli has published more than 200 high quality papers and won numerous best