Cross-Condition and Cross-Platform Remaining Useful Life Estimation Via Adversarial-Based Domain Adaptation

Dongdong Zhao (ddzhao@bjtu.edu.cn)
Beijing Jiaotong University
Feng Liu
Beijing Jiaotong University

Research Article

Keywords: Cross-condition, cross-platform, life estimation, domain adaptation, machine learning, knowledge, architecture, industry

Posted Date: July 9th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-669444/v1

Cross-condition and cross-platform remaining useful life estimation via adversarial-based domain adaptation

Dongdong Zhao[1,*] **and Feng Liu**[2]

[1]Beijing Jiaotong University, School of Computer and Information Technology, Beijing, 100089, China
[*]ddzhao@bjtu.edu.cn

ABSTRACT

Supervised machine learning is a traditionally remaining useful life (RUL) estimation tool, which requires a lot of prior knowledge. For the situation lacking labeled data, supervised methods are invalid for the issue of domain shift in data distribution. In this paper, a adversarial-based domain adaptation (ADA) architecture with convolution neural networks (CNN) for RUL estimation of bearings under different conditions and platforms, referred to as ADACNN, is proposed. Specifically, ADACNN is trained in source labeled data and fine-tunes to similar target unlabeled data via an adversarial training and parameters shared mechanism. Besides a feature extractor and source domain regressive predictor, ADACNN also includes a domain classifier that tries to guide feature extractor find some domain-invariant features, which differents with traditional methods and belongs to a unsupervised learning in target domain, which has potential application value and far-reaching significance in academia. In addition, according to different first predictive time (FPT) detection mechanisms, we also explores the impact of different FPT detection mechanisms on RUL estimation performance. Finally, according to extensive experiments, the results of RUL estimation of bearing in cross-condition and cross-platform prove that ADACNN architecture has satisfactory generalization performance and great practical value in industry.

Introduction

Remaining useful life (RUL) estimation, which is one facet of prognostics and health management (PHM) aiming to provide users with an integrated view about the health state of a machine or an whole system[1], defined as “the length from the current time to the end of useful life”[2]. With the development of sensing technology, deep learning (DL) technology with nonlinear deep representation capabilities has gradually matured. A large amount of studies[3–9] on RUL estimation of bearing in same operating condition have obtained good results.

Although every healthy mechanical component will has a rated working time after it is created, varying operating conditions and platforms will affect the service life of mechanical component to varying degrees. In the case where the training data and the test data come from different operating conditions and platforms, the generalization ability on the test data is usually poor.

In the case of varying operating conditions and platforms, there are many factors influencing on performance of RUL estimation, such as feature extractor and regressive predictor, but in the final analysis, it is caused by the number of observation samples, first predictive time (FPT) detection method, etc. For example, when migrating from a source bearing entity with observation samples $N1$ to a target bearing entity with observation samples $N2$, if $N1$ is much smaller than $N2$, the RUL predicted by the target domain usually fluctuates more strongly. We assume that the bearing entity degrades linearly after the failure prediction time point. The source bearing entity 1 has N1 samples, target bearing entity has N2 samples, and their FPT are respectively FPT1 and FPT2. It is a challenge for feature extractors and regressive predictors to accurately predict RUL. Particularly, the feature extractor needs to obtain discriminative features, and the regressive predictor needs to map the discriminative features to actual RUL value as accurately as possible.

Though there are many effective domain adaptation (DA) methods for classification, but we practically found that few regressive DA methods.[10] suggested that the regression space is usually continuous on the contrary, for example, there is no clear decision boundary. In fact, the observation samples have limited, the RUL estimation is still in a large discrete space.[10] firstly proposed regressive DA for keypoint detection, which proved the effectiveness of DA in regressive prediction tasks. For RUL estimation of bearing under varying conditions or platforms, the key factor is how to assign cross-invariant features of same health status built by neural network a equal RUL, which is vital for unsupervised RUL estimation in cross-domain.

Therefore, we found that in fact, the idea of regressive DA is still the same as the classification algorithm, such as SVM trying to find margins of boundaries between different classes. RUL estimation under varying operating conditions or varying platforms aims to find a boundary to distinguish the different operating states of bearing entity. In whole, there are three key points worth paying attention to:

1. Domain-invariant features. $F(D_s^i(RUL=0.5)) \approx F(D_t^j(RUL=0.5))$, $F(x)$ is the feature representation of sample x, $D_s^i(RUL=0.5)$ denotes the samples data, where RUL equals to 0.5, of the i-th source bearing entity. $D_t^j(RUL=0.5)$ denotes the samples, where RUL equals 0.5, of the j-th target bearing entity.

2. Discriminative features. $F(D_s^i(RUL=0.5))$ is not equal to or far from $F(D_s^i(RUL=0.6))$, and $F(D_t^j(RUL=0.5))$ is not equal to or far from $F(D_t^j(RUL=0.6))$.

3. If the input of regressive predictor are some similar domain-invariant features respectively come from source domain and target domain, and the predicted results of regressive predictor should be approximately equal with each other. We use the formula to express as: $Y(F(D_s^i(RUL=0.5))) \approx Y(F(D_t^j(RUL=0.5)))$, where $Y()$ is the function of regressive predictor.

Motivated by domain adaptation neural networks (DANN)[11], in this paper, we introduce the original intention proposed by the DANN architecture into machinery RUL estimation. Through combining with the superiority of CNN in vibration signals process, adversarial-based domain adaptation (ADA), refers to ADACNN, consists of three parts: feature extractor, regressive predictor and domain classifier. Feature extractor firstly acquires domain-invariant and discriminative representation from raw vibration signals. The domain classifier, as an important auxiliary tool, forces feature extractor finding common space where samples have corresponding domain-invariant representation. The regressive predictor outputs a estimated RUL close to actual RUL value from domain-invariant feature.

Overall, in this paper, we proposes a novel neural networks framework with ADA for RUL estimation of bearings in different condition and platform. The main contributions of this paper are as follows:

1. For scenarios lacking labels or no label, ADACNN can be simply switched for maximizing known labels' value in real application scenarios. At the same time, it ensures excellent estimation accuracy in source domain and generalization ability in target domain. To our best knowledge, this framework with ADA is firstly introduced for RUL estimation of bearing in varying conditions and platform. In addition, in this paper, we just pay attention to a harder unsupervised case, that is, there are all unlabeled data in target domain.

2. The ADACNN was verified on two public datasets: cross-condition experimental scenarios (FEMTO and XJTU-SY dataset, introduced afterthis) between cross-condition and cross-platform experimental scenarios.

3. The proposed methodology compares with two non-adapted models respectively training with only source data and only target data to verify the generalization ability of proposed ADACNN.

Theoretical analysis

Problem Formulation

In this paper, we proposed a framework with ADA for RUL estimation, which constitutes feature extractor, domain classifier, and a regressive predictor. It should be pointed out that we assume the source domain training are run-to-failure vibration data. Let $X^S = \{x_i^j\}_{i=1}^{n^j} \in R^{D_{source}}$, $j = 1, 2, ..., N_{source}$, denote the i-th sample of the j-th bearing entity in source domain, where n^j is the number of samples for the j-th bearing entity in source domain. By analogy, $X^T = \{x_i^j\}_{i=1}^{n^j} \in R^{D_{target}}$ denotes the i-th sample of the j-th bearing entity in target domain.

Source data with fully labeled and target data with unlabeled are used as input to train DANN. For the target data, it is an unsupervised learning process. We record source data as (X^S, Y^S) and target data as (X^T), and $D_S \neq D_T$ which means the distribution of source domain is different from the distribution of target domain.

The methodology proposed in this paper mainly includes four steps as follows:

1. Data preparation: Calculate RUL percentage label $Y^S = \{x_i^j\}_{i=1}^{n^j}$ corresponding to the i-th sample of the j-th bearing entity in source domain.

2. In addition to X^S and Y^S in source domain as input of ADACNN, unlabeled data X^T in target domain are also used as input when training ADACNN, which includes three parts shown in Fig1:

   (a) A regressive predictor θ_y is introduced to accomplish main regression task through a supervised learning in source domain.

   (b) A feature extractor θ_f finds a common space with domain-variant feature from source and target domain data.

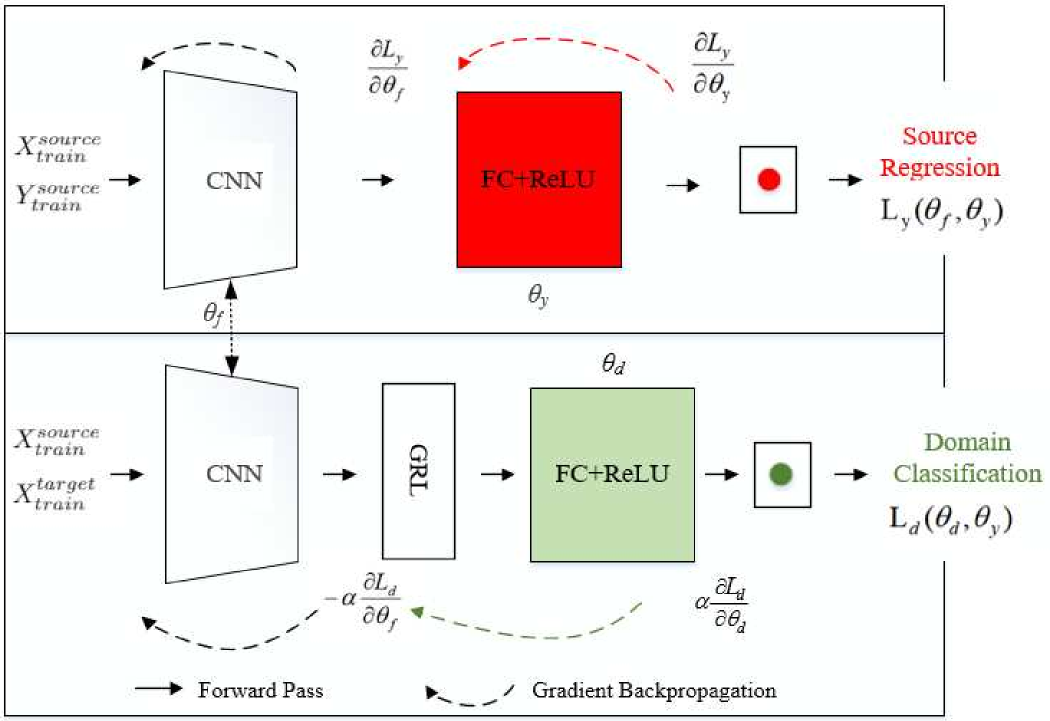

Figure 1. The architecture of ADACNN.

(c) A domain classifier θ_d combined with gradient reversal layer (GRL) can not makes the data output by the feature extractor be distinguished as impossible as possible.

Experimental design

Data preparation

The source domain data are divided into training data X_{train}^{source} and training labels $Y_{train}^{source}$and test data X_{test}^{source} and test labels Y_{test}^{source}. The target domain data are also divided into training data X_{train}^{target} and test data X_{test}^{target} and test labels Y_{test}^{target}. X_{train}^{source} are used in the supervised training of ADACNN. X_{test}^{source} are used as input of the test part of the training process to observe the maturity of model training. X_{train}^{target} participates in unsupervised training ADACNN to improve its generalization ability, X_{test}^{target} involves in the evaluation of ADACNN. The training data are run-to-failure data, and the test data are truncated data.

The training label generation: Taking FPT as the boundary, the actual RUL values of data samples before FPT point are equal to 1, and the actual RUL value corresponding to the j-th sample after FPT point drops from 1 to 0 successively, which denoted as $\frac{x_i^j}{Total_i - FPT_i}$, where x_i^j represents the j-th sample after FPT point of i-th bearing entity, $Total_i$ is equal to the number of whole run-to-failure samples of i-th bearing entity, and FPT_i represents the number of samples before FPT point of i-th bearing entity.

The test label generation: $RUL_i^j = 1 - \frac{K}{Total_i^j - FPT_i^j}$. FPT_i^j represents the FPT of j-th bearing entity. We assume that known truncation points are always after FPT point, and K denotes the number of samples between truncation point and FPT point is K.

Usually, before FPT point, it is not necessary or difficult to predict RUL because there is no obvious sign of degradation, and after this point, signs of degradation began to appear. Therefore, a FPT detection mechanism is very important to capture real-time changes. Kurtosis[3,12,13] was used to detect FPT. Many feature-fused methods combine time domain with frequency domain vibration characteristics.[14,15] fused some features into one index descripting degradation process, and it is worth mentioning that these features are obtained by calculating the mahalanobis distance (MD) from original healthy state, which is a relative feature and suitable for some scenarios of vibration signal processing. In this paper, we will explore the impact of different FPT detection methods on RUL estimation.

Data normalization

In order to speed up the training speed and align the test data with the training data during the test, we have some pre-processing for the vibration data. Data normalization mainly includes four parts:

$norm(X_{test}^{source}, X_{train}^{source})$, $norm(X_{train}^{source}, X_{train}^{source})$

and

$norm(X_{test}^{target}, X_{train}^{target})$, $norm(X_{train}^{target}, X_{train}^{target})$.

Where $norm(a,b)$ represents the normalization function and consists of two steps: firstly to calculate the mean m and variance d of b, and secondly to normalize a by m and d.

The ADACNN model construction

1. Initialize feature extractor: The input parameters of feature extractor include input data, number of CNN layers, number of filters per layer, and dropout rate. The kernel size of first layer equals to 25 by default (Has been proven its effectiveness in[16].), and the remaining layers are initialized according to input parameters. The feature extractor mainly includes one dimensional convolution layer (Conv1D), activation layer, Dropout, and MaxPooling1D. The output of feature extractor are latent features, its dimension depends on the initialization parameters f.

2. Initialize regressive predictor: The input parameters of regressive predictor include input data, number of convolution network layers, and number of nodes per layer. The architecture of regressive predictor mainly consist of FCN, activation layer, and dropout layer. Finally, the output predicted value is between 0 and 1, indicating RUL percentage. The closer the predicted value is to 1, the healthier it is, and the closer it is to 0, the closer it is to a fault.

3. Initialize classifier: The classifier includes FCN, Activation layer, Dropout layer.

4. Construct regression model: Source regression model consists of feature extractor and regressive predictor. The output latent features of the feature extractor is fed to the regressive predictor after passing through a flatten layer.

5. Construct domain classification model: Through the parameter sharing mechanism of feature extractor, domain classification model mainly consists of feature extractor, GRL, and domain classifier.

Training the ADACNN

1. Initialization: Start iterative training with iteration i=0. Patience value M=0. In order to reduce the memory pressure, data are read in batches in each iteration.

2. Training source regressive predictor and domain classifier: As shown in Fig.1, through the forward propagation mechanism, the source regression model takes X_{train}^{source} and Y_{train}^{source} as input and RUL prediction value as output, according to the known Y_{train}^{test} to calculate the prediction loss. The domain classifier model takes X_{train}^{source} and X_{train}^{target} as input, outputs binary classification and calculate the loss of domain classification, and then update the parameters of the regressive predictor and the feature extractor and classifier through the backward propagation method of gradient learning as shown in Fig1. The i-th updation formula are defined as Equation (1):

$$\begin{aligned}\theta_f &= \theta_f - \mu\left(\frac{\partial L_y^i}{\partial \theta_f} - \alpha \frac{\partial L_d^i}{\partial \theta_f}\right) \\ \theta_y &= \theta_y - \mu\left(\frac{\partial L_y^i}{\partial \theta_y}\right) \\ \theta_d &= \theta_d - \mu\left(\frac{\partial L_d^i}{\partial \theta_d}\right)\end{aligned} \tag{1}$$

.

3. Evaluate the ADACNN model by calculating the loss of RUL estimation: Calculate the accuracy acc of current model using X_{test}^{source} and Y_{test}^{source} by root mean square error (RMSE) evaluation metrics. If $RMSE_i$ is less best accuracy $RMSE_{best}$, then $RMSE_i$ will be assigned to $RMSE_{best}$, otherwise, the patience value M is increased by 1.

4. Judgement 1: If M is greater than the preset value, stop iterative training and save the model parameters of the i-th iteration.

5. Judgement 2: If iteration i increases to threshold, stop iterative training and save the result of the i-th iteration.

Table 1. The information of FEMTO dataset.

Operating conditions	training data	test data
A1(1800rpm and 4000N)	A1-1 ∼ A1-2	A1-3 ∼ A1-7
A2(1650rpm and 4200N)	A2-1 ∼ A2-2	A2-3 ∼ A2-7
A3(1500rpm and 5000N)	A3-1 ∼ A3-2	A3-3

Table 2. The information of XJTU-SY dataset.

Operating conditions	training data	test data
B1(2100rpm and 12kN)	B1-1 ∼ B1-2	B1-3 ∼ B1-5
B2(2250rpm and 11kN)	B2-1 ∼ B2-2	B2-3 ∼ B2-5

6. Start a new iteration: The entire experimental flow chart is shown in Fig. 1, the value of *i* plus one is re-assigned to *i*, and continue going to the step 2.

7. Testing the ADACNN: Use target test data X_{test}^{target} to evaluate the accuracy of the trained model.

Experimental Setup

Dataset description

1. FEMTO dataset. The FEMTO dataset comes from an experimental platform called PRONOSTIA, where bearings' degradation experiments are allowed to conduct in only few hours, this platform can obtain true bearing degradation data by accelerating bearing degradation under different operating conditions so that some data-driven techniques are studied further. PRONOSTIA includes three main parts: a rotating part, a degradation generation part, and a measurement part. For more rotating part and degradation generation, please refer to[17]. For the measurement part, there are two types of signals: temperature and vibration with horizontal and vertical respectively from their own acceleration sensor and temperature sensor. The algorithm proposed in this paper only uses vibration signals, and the sampling frequency of the acceleration sensor is 25.6 KHZ. As tabulated in Table 1. , FEMTO data set includes three different operating conditions, we use A to represent the FEMTO data set, and Ai-j to represent the *j*-th bearing of *i*-th conditions in FEMTO. Six bearings data are run-to-failure data, which we use as training data. The 11 bearing data are truncated to predict the remaining life, which we use as training data.

2. XJTU-SY dataset. The XJTU-SY dataset was collected by the Xi'an Jiaotong university and the Changxing Sumyoung Techonology Company[18]. 32768 data points are collected on 1.28 s of every minute with sampling rate of 25.6 kHZ. The tested processes of bearing are stopped when the amplitude of the vibration signal is higher than 20 g for protecting the test bed. There are two PCB 352C33 accelerometers are placed on the housing of the tested bearings, which are respectively on the vertical and horizontal axis. The information of XJTU-SY Dataset are shown in Table 2.

Data preprocessing

For FEMTO, 2560 data points are collected on 0.1 s of every 10 s with sampling frequency of 25.6 kHZ. For XJTU-SY, 32768 data points are collected on 1.28 s of every minute with sampling rate of 25.6 kHZ. In this study, we reduced the dimensions of the sample data of the two datasets to 1280 by way of dimensionality reduction. For FEMTO, we extract 0.05s of data from every 0.1s sample data. In other words, reduce the dimensionality by half. For XJTU-SY, we extract 0.05s of data from every 1.28s of sample data. In other words, we reduce the dimensionality of sample data by 25.6 times.

Comparative baselines

1. Comparison against methods with different FPT mechanisms. For FPT detection mechanism influences the performance on RUL estimation, we choose three FPT detection methods: MD, kurtosis and no FPT. MD and Kurtosis sensitive to early failure has been widely used in FPT detection. No FPT detection mechanism means that the bearing will degenerates from the initial state. We denotes these three methods as MD-ADACNN, Kur-ADACNN and NoFPT-ADACNN respectively.

2. Comparison against non-adapted methods. In order to verify whether the proposed ADA method works, we use the following two baselines for comparison. The Source-Only method is trained on the source domain data and tested in the target domain. The Target-Only method is trained on the target domain data and tested on the target domain (There is no intersection between training data and test data). To be fair, the parameters of feature extractor and regressive predictor of the Source-Only and Target-Only methods are consistent with the parameters of ADACNN.

Table 3. Hyperparameter evaluated in the proposed method.

Hyper-parameter	Range
Layers:(θ_f, θ_y, θ_d)	$\{1, 2\}$
Units: (θ_f, f, θ_y, θ_d)	$\{16, 32, 64, 128, 256, 512\}$
Learning Rate: ($\theta_y(\lambda_y)$, $\theta_d(\lambda_d)$)	$\{0.0001, 0.001, 0.01\}$
Dropout Rate	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$
α	$\{0.8, 1.0, 2.0\}$
Batch Size	$\{64, 128, 256\}$
Threshold of patience M	$\{20\}$
Max iteration N	$\{200\}$

Implementation details.

Evaluation metrics

RMSE, Root mean square error, and Score are used as performance metrics to evaluate the error between the predicted RUL and the true RUL. RMSE has been used in many publications[19–21], and its definition formula is the Equation (2).

$$RMSE = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(y_i - \hat{y}_i)^2} \tag{2}$$

, where $\hat{y}_i$ and y_i is the predicted RUL and actual RUL of i-th test sample, K denotes the total number of test samples. A larger RMSE value means a larger prediction error.

Score, defined as Equation (3), used in this study was first proposed in[17] and has been used in many studies[3,20,21]. The predicted RUL is greater than or less than the actual RUL should be treated differently. In other words, in the case of the same absolute value, the penalty for a positive value is less than the penalty for a negative value.

$$Score = \sum_{i=1}^{K} s_i \tag{3a}$$

$$s_i = \begin{cases} e^{-\frac{err_i}{13}} - 1, & err_i \geq 0 \\ e^{\frac{err_i}{10}} - 1, & err_i < 0 \end{cases} \tag{3b}$$

$$err_i = y - \hat{y} \tag{3c}$$

, where K is the total number of test samples. y_i, $\hat{y}_i$, and err_i respectively represent actual RUL, predicted RUL and the difference between actual RUL and predicted RUL for the i-th testing data sample.

Hyper-parameter selection

In adaptation training processes, the learning rate of source domain regression and domain classification and the parameter of CNN largely determine the experimental performance. Therefore, we use the grid-search method to find the optimal learning rate (λ_y,λ_d,[Layer,units,Dropout]), and then manually fine-tune other parameters presented in Table 3. Overall, we did 6 experiments (E1-E6) on the FEMTO dataset, 2 experiments (E7-E8) on the XJTU-SY dataset, and 12 experiments (E9-E20) on the FEMTO and XJTU-SY. Their parameter pairs are tabulated in Table 4.

Discussion

Cross-condition.

In Fig.2, the horizontal axis represents every test units of same operating condition in target domain (5, 5 and 1 on FEMTO, 3 and 3 on XJTU-SY), and the vertical axis represents RUL percentage. The thick histogram and the thin histogram respectively represent the predicted value and label (or actual value) of the same method. The closer the highest points of the thick histogram and the thin histogram are, the higher the accuracy. As described in Fig.2, it can be seen that no matter which data set it is verified on, the predicted value of MD-ADACNN method is closer to its actual label. In addition, the MD-ADACNN method usually gives a predicted value slightly smaller than the true value, which will provide constructive warnings for engineering operation and maintenance engineers. However, for the other two FPT detection mechanisms, the prediction accuracy of the

Table 4. Selected hyperparemeter for each source-target experiment pair.

No.	From to	CNN: Layer, (units), [Dropout]	f	Source Regression: Layers, (units), [Dropout]	Domain Classification: Layers, (units), [Dropout]	α	Batch Size	(λ_y,λ_d)
E1	A1 $\rightarrow$A2	2, (128,64), 0.9	512	1, (64), 0.1	2, (256,128), 0.9	2	256	0.01,0.01
E2	A1 $\rightarrow$A3	2, (128,32), 0.5	64	2, (32,16), 0.3	2, (32,16), 0.3	0.8	256	0.0001,0.0001
E3	A2 $\rightarrow$A1	2, (64,32), 0.1	64	2, (32,32), 0.1	2, (16,16), 0.1	1	256	0.001,0.001
E4	A2 $\rightarrow$A3	2, (128,64), 0.1	512	2, (64,32), 0.1	2, (64,32), 0.1	2	256	0.001,0.001
E5	A3 $\rightarrow$A1	2, (64,32), 0.3	128	2, (64,64), 0.1	2, (64,64), 0.1	2	256	0.001,0.001
E6	A3 $\rightarrow$A2	2, (128,32), 0.9	512	2, (128,128), 0.1	2, (128,64), 0.1	2	256	0.001,0.001
E7	B1 $\rightarrow$B2	2, (128,32), 0.1	64	2, (256,128), 0.1	2, (128,64), 0.5	0.8	128	0.001,0.001
E8	B2 $\rightarrow$B1	2, (128,32), 0.9	64	2, (256,128), 0.1	2, (128,64), 0.9	0.8	128	0.001,0.001
E9	A1 $\rightarrow$B1	2, (32,32), 0.1	64	1, (64), 0.1	1, (64), 0.5	0.8	64	0.001,0.001
E10	A1 $\rightarrow$B2	2, (128,32), 0.1	64	1, (64), 0.1	2, (64,64), 0.5	0.8	64	0.001,0.001
E11	A2 $\rightarrow$B1	2, (32,32), 0.1	64	1, (64), 0.1	1, (64), 0.5	0.8	64	0.001,0.001
E12	A2 $\rightarrow$B2	2, (32,32), 0.1	64	1, (64), 0.1	1, (64), 0.5	0.8	64	0.001,0.001
E13	A3 $\rightarrow$B1	2, (32,32), 0.1	64	1, (64), 0.1	1, (64), 0.5	0.8	64	0.001,0.001
E14	A3 $\rightarrow$B2	2, (32,32), 0.1	64	1, (64), 0.1	1, (64), 0.5	0.8	64	0.001,0.001
E15	B1 $\rightarrow$A1	2, (32,32), 0.9	64	1, (64), 0.9	2, (64,64), 0.5	0.8	64	0.001,0.001
E16	B1 $\rightarrow$A2	2, (128,32), 0.9	64	1, (64), 0.9	2, (64,64), 0.5	0.8	64	0.001,0.001
E17	B1 $\rightarrow$A3	2, (128,32), 0.9	64	1, (64), 0.9	2, (64,64), 0.5	0.8	64	0.001,0.001
E18	B2 $\rightarrow$A1	2, (64,64), 0.1	64	1, (32), 0.9	2, (64,64), 0.5	1	64	0.001,0.001
E19	B2 $\rightarrow$A2	2, (128,32), 0.9	64	1, (64), 0.9	2, (64,64), 0.5	2	64	0.001,0.001
E20	B2 $\rightarrow$A3	2, (128,32), 0.9	64	1, (64), 0.1	2, (64,64), 0.5	0.8	64	0.001,0.001

Kur-ADACNN method is obviously the lowest. The kurtosis-based FPT detection mechanism is an indicator in the vibration data, and the MD-based FPT detection mechanism is a relatively joint indicator of multiple indicators in the vibration data after dimensionality reduction in a relatively healthy state. Therefore, it can be seen from the results that the MD-based method is a more suitable FPT detection mechanism which is is closer to the degradation trend of the bearing.

From Fig.3, we find that the RUL estimation results of MD-ADACNN method are between the results of Source-Only and Target-Only methods, and even more closer to the actual RUL value than Target-Only method (Fig.3(b)) , which proves effectiveness of ADA.

From the perspective of the FPT detection mechanisms, as shown in the cross-condition results of the three methods (MD-ADACNN, NoFPT-ADACNN and Kur-ADACNN) on the FEMTO dataset listed from E1 to E6 in Table 5., MD-ADACNN has obtained the four best RMSE and Score accuracy (E1:A1 $\rightarrow$A2, E3: A2 $\rightarrow$A1, E5: A3 $\rightarrow$A1 and E6: A3 $\rightarrow$A2). In the other two cross-condition (A1 $\rightarrow$A3 and A2 $\rightarrow$A3), although MD-ADACNN has a larger error than the other two methods, from the 6 cross-condition experiments as a whole, MD-ADACNN predicts RUL more stable, and the two methods NoFPT-ADACNN and Kur-ADACNN, especially Kur-ADACNN, are greatly affected by the total number of cycle of units under varying conditions. In the FEMTO data set, the cycle of units under condition A3 is shorter, and the cycle of units under condition A1 and A2 is longer. Therefore, these two methods have better results in RUL estimation of cross-condition bearings from longer to shorter cycles, but very poor in the opposite case.

From the perspective of whether to use domain adaption technology in Table 5., in most experiment with cross-condition (E1: A1 $\rightarrow$A2, E2: A1 $\rightarrow$A3, E3: A2 $\rightarrow$A1, E5: A3 $\rightarrow$A1 and E6: A3 $\rightarrow$A2), RMSE(Source-Only) $>$ RMSE(MD-ADACNN) $>$ RMSE(Target-Only). In these cross-condition (E2: A1 $\rightarrow$A3 and E4: A2 $\rightarrow$A3), RMSE(MD-ADACNN) $<$ RMSE(Target-Only), which shows that the generalization ability of ADA technology from condition A2 with a long cycle time to condition A3 with a short cycle time exceeds supervised algorithms that only use target data, that is, to some extent, the data in target domain are benefit to guide the whole ADA model fine-tune to target data.

Judging from the results of the five methods (Source-Only, Target-Only, MD-ADACNN, NoFPT-ADACNN and Kur-ADACNN in Table 5.) verified on the FEMTO data set, basically, we found the effects of these three methods are superior to that of Source-Only, so these results prove that the validity of domain adaption technology in the research field of cross-condition bearing RUL estimation. Observing the results listed from E7 to E8 in Table 5. on the XJTU-SY data set from the same perspective, the same conclusion was confirmed again.

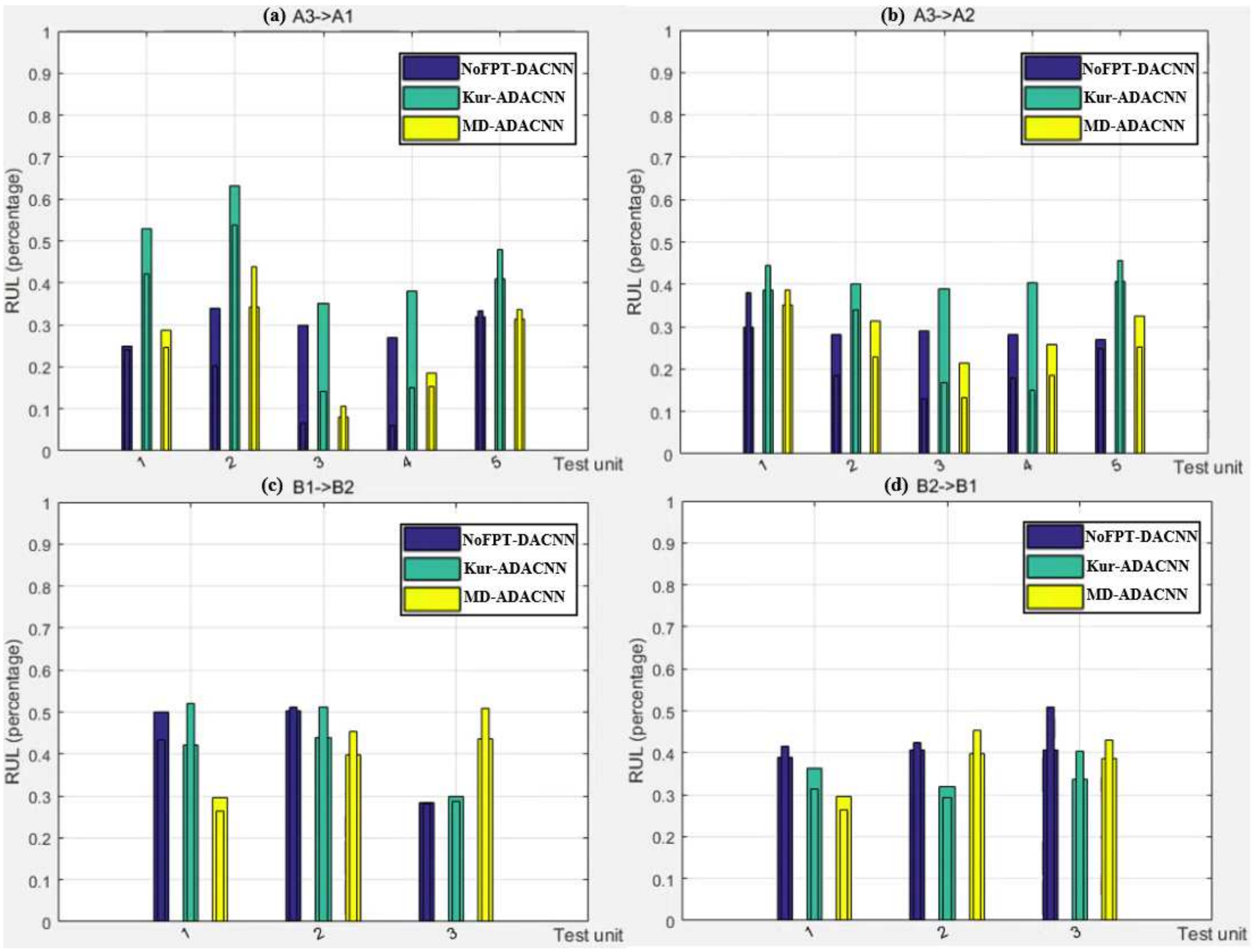

Figure 2. RUL estimation comparisons with different FPT detection mechanisms on FEMTO and XJTU-SY datasets:(a)A3 →A1, (b)A3 →A2, (c)B1 →B2, (d)B2 →B1

Cross-platform.

In order to further verify the performance of the proposed ADACNN method in the inter-platform domain for RUL estimation, we choose the three conditions in the FEMTO dataset as the source domain or target domain data, and at the same time, 2 conditions in the XJTU-SY dataset are used as target domain or source domain data, so there are a total of 12 experiments (E9-E20 tabulated in Table 6 between two platforms. Different from the cross-condition, the cross-platform experiment part only explores the research on the remaining life of different FPT establishment mechanisms under the same DA technology, because the superiority of the DA technology has been clearly proved in the previous experiments.

It can be seen from Table 6. that compared with NoFPT-ADACNN and Kur-ADACNN, when the evaluation metric is RMSE, MD-ADACNN obtained the optimal value in 9 out of 12 cross-platform experiments (E11: A2 →B1, E12: A2 →B2, E13: A3 →B1, E14: A3 →B2, E15: B1 →A1, E16: B1 →A2, E18: B2 →A1, E19: B2 →A2, and E20: B2 →A3), when the evaluation metric is Score, MD-ADACNN obtained the optimal value in 8 of the 12 cross-platform experiments (E3: A2→B1, E12: A2 →B2, E13: A3 →B1, E14: A3 →B2, E15: B1 →A1, E16: B1 →A2, E19: B2 →A2, and E20: B2 →A3). We still find that the Kurtosis-based FPT detection method is unstable.

It can be seen from Fig.4 that in Fig.4(b) and Fig.4(d), the Kur-based FPT detection mechanism considers the RUL label at the prediction point to be 1, while RUL label of the prediction point determined by the No FPT and MD-based FPT detection mechanism are different not big. From the principle that the majority obeys the minority, for test unit=1 in Fig.4(b) and test unit=4 in Fig.4(d), we can think that Kur-based does not perform well. On the whole, whether it is A →B or B→A, the MD-ADACNN predicted value is not far from the corresponding label, and even more often slightly smaller than the corresponding label value.

Table 5. RMSE/Score±Standard Deviation comparison between Source-Only, Target-Only, NoFPT-ADACNN, Kur-ADACNN and MD-ADACNN on FEMTO dataset and XJTU-SY dataset.

No.	Source-Only		Target-Only		MD-ADACNN		NoFPT-ADACNN		Kur-ADACNN	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
E1	53.2	101.3	32.6	64.6	36.2±1.6	74.0±46.4	41.5±2.7	91.9±66.7	36.5±2.5	88.3±88.2
E2	26.0	12.5	8.7	1.0	8.5±14.4	5.1±14.0	3.4±4.0	0.3±0.4	2.9±0.5	0.3±0.1
E3	39.4	239.8	22.5	15.3	26.8±5.9	14.4±10.9	44.2±7.4	328.0±700.0	64.3±8.9	8542.5±9330.2
E4	30.5	20.2	24.6	10.7	19.8±18.8	10.3±4.5	9.9±9.1	1.9±2.8	3.8±1.3	0.4±0.1
E5	67.0	2084.7	33.5	930.4	31.4±3.2	7.1±4.7	40.0±7.4	136.6±215.9	52.0±4.0	544.4±322.3
E6	48.8	1752.6	30.7	35.9	35.4±0.8	37.2±9.1	43.6±14.6	1088.6±2835.6	37.3±2.0	89.4±43.4
E7	18.3	10.1	13.0	1.0	13.3±0.3	1.2±0.6	14.0±2.9	2.1±1.1	17.6±3.4	9.6±1.0
E8	5.6	0.7	3.1	0.3	3.6±0.7	0.4±0.4	5.3±2.2	0.6±0.3	4.1±0.8	0.7±0.1

Table 6. RMSE/Score±Standard Deviation comparison under cross-platform between FEMTO and XJTU-SY

No.	From To	MD-ADACNN(proposed)		NoFPT-ADACNN		Kur-ADACNN	
		RMSE	Score	RMSE	Score	RMSE	Score
E9	A1 $\rightarrow$B1	6.1±2.6	0.9±0.5	6.6±2.1	0.8±0.4	2.3±2.0	0.2±0.1
E10	A1 $\rightarrow$B2	60.9±4.6	4027.6±1904.8	54.1±3.4	2531.6±1341.1	65.2±3.9	22933.6±10923.1
E11	A2 $\rightarrow$B1	1.7±1.6	0.1±0.2	4.5±3.7	0.4±0.5	4.6±3.5	0.7±0.9
E12	A2 $\rightarrow$B2	8.5±12.7	10.4±30.1	18.7±15.4	18.3±29.3	63.1±15.5	97874.2±125079.4
E13	A3 $\rightarrow$B1	4.4±2.1	0.5±0.3	6.9±2.6	0.9±0.6	4.4±5.2	0.5±0.7
E14	A3 $\rightarrow$B2	13.7±9.0	7.8±11.6	23.9±10.3	71.1±141.4	44.2±6.4	117.3±85.6
E15	B1 $\rightarrow$A1	32.7±5.1	59.3±86.9	41.5±8.4	81.6±70.7	45.8±13.3	101.3±86.4
E16	B1 $\rightarrow$A2	32.6±1.7	26.5±7.8	36.3±3.3	44.8±23.0	36.2±2.0	64.1±21.5
E17	B1 $\rightarrow$A3	21.6±6.8	10.2±8.6	7.7±14.0	3.0±6.2	5.2±1.8	0.7±0.2
E18	B2 $\rightarrow$A1	30.2±5.8	25.1±16.0	31.9±7.1	14.9±6.0	34.8±10.0	39.6±18.9
E19	B2 $\rightarrow$A2	35.5±1.9	43.4±13.4	35.8±3.2	122.1±179.7	47.0±9.0	1510.6±2003.2
E20	B2 $\rightarrow$A3	1.5±2.6	0.1±0.3	10.7±14.9	4.4±8.0	6.1±0.5	0.8±0.1

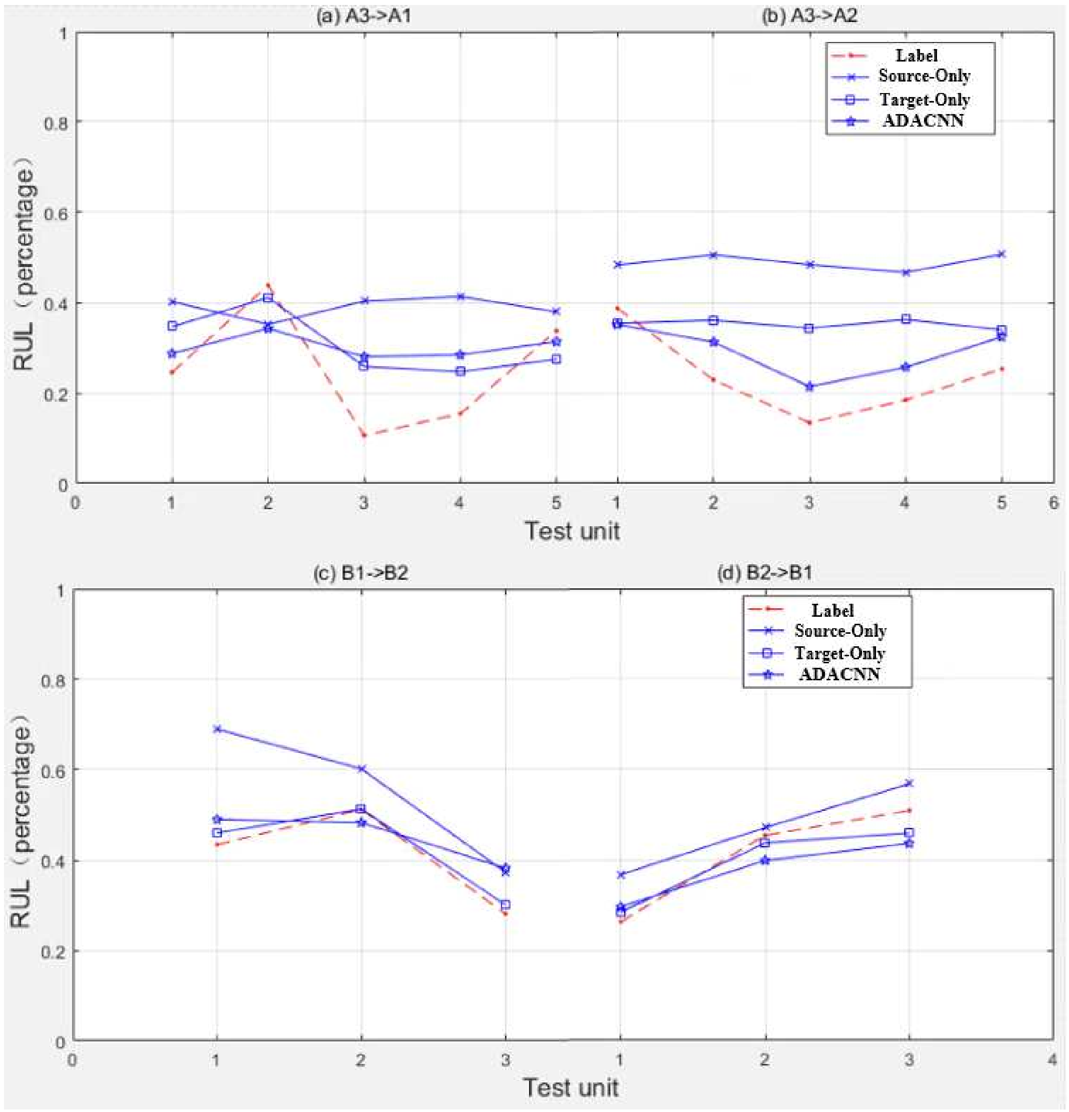

Figure 3. RUL estimation comparisons with Source-Only and Target-Only methods on FEMTO and XJTU-SY datasets. (a)E5: A3 $\rightarrow$A1, (b)E6: A3 $\rightarrow$A2, (c)E7: B1 $\rightarrow$B2, (d)E8: B2 $\rightarrow$B1

Methods

As shown in Fig.1, between source and target data, the θ_f tries to find a common space where same health/degradation state samples are assigned to no discriminative representation, instead of maintaining its own characteristics in different domains. The source domain features extracted by θ_f are fed to θ_y. The target domain data features and source domain data features extracted by θ_f are fed to the θ_d.

[22] uses H-divergence as a measure to calculate the distance between the source samples $S \sim (D_S^X)$ and target domain samples $T \sim (D_T^X)$ during the adversarial training processes, defined as Equation (4):

$$\hat{d}_H(S,T) = 2[1 - \min_{\eta \in H}[\frac{1}{N_S}\sum_{i=1}^{N_S} I[\eta(x_i) = 0] + \frac{1}{N_T}\sum_{i=1}^{N_T} I[\eta(x_i) = 0]]] \tag{4}$$

Where $I[c]$ is the function which is 1 if condition c is true, and 0 otherwise.

The overall key idea of DANN[11] is to estimate the minimum term of (1) through iterative training. In other words, it ensures that latent representation of neural networks doesn't contain the discriminative information about the domain of input data whatever are from source domain or from target domain while having less risk on source examples. Therefore, the composition of the loss function of DANN is divided into two parts: the prediction loss of the main task and the domain classification loss.

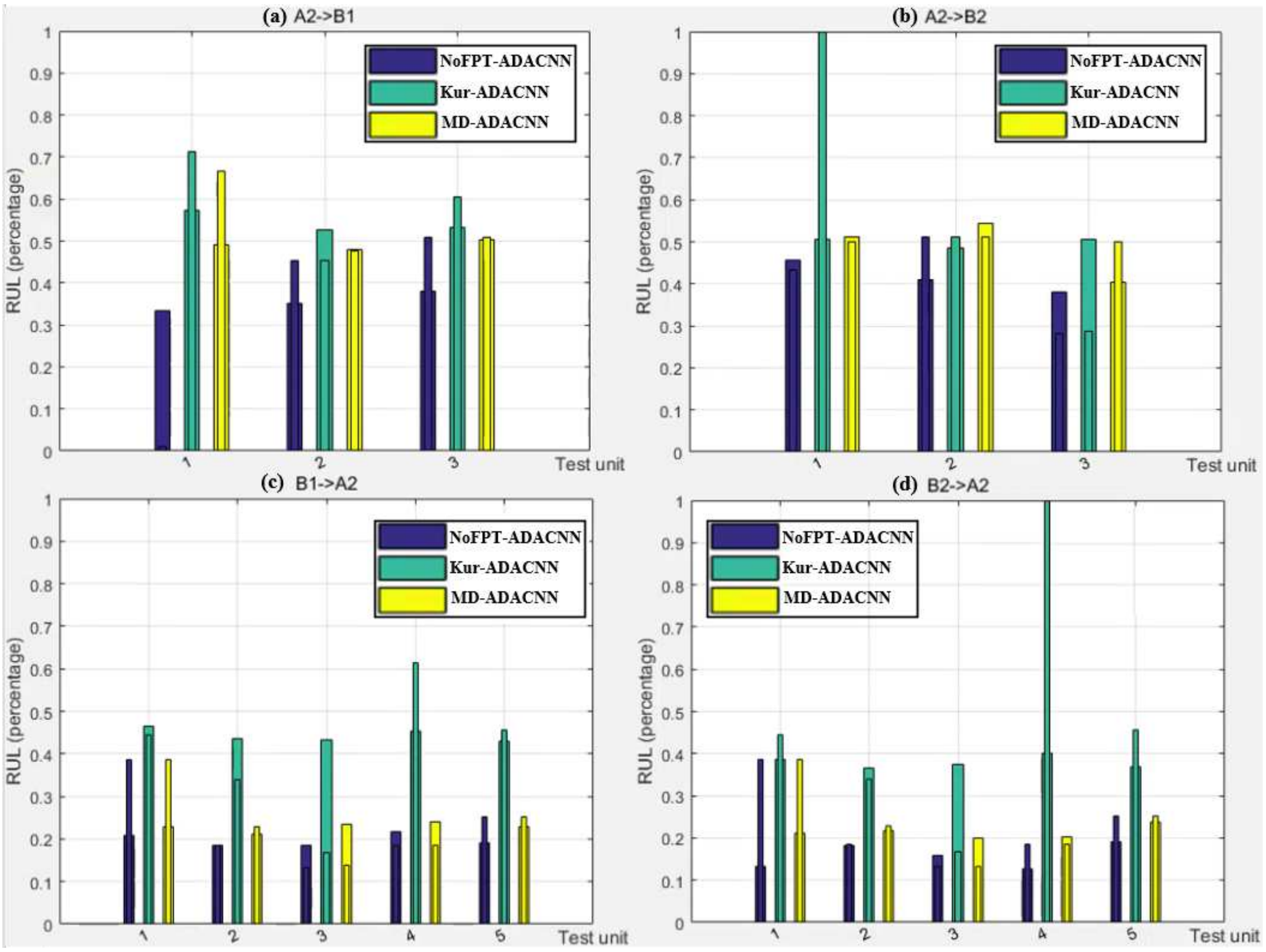

Figure 4. RUL estimation comparisons with different FPT detection mechanisms between two platforms:(a)E11: A2 →B1, (b)E12: A2 →B2, (c)E16: B1 →A2, (d)E19: B2 →A2

The prediction loss L_y^i of i-th batch examples is defined as the Equation (5):

$$L_y^i = \min_{\theta_f \theta_d}[\frac{1}{N_S}\sum_{i=1}^{N_S} L_d^i(\theta_f, \theta_y) + \alpha R(\theta_f)] \tag{5}$$

Where $\alpha R(\theta_f)$ is the regularisation factor, and it's weight is α. Inspired by proxy distance, the optimisation problem of θ_d is denoted as:

$$R(\theta_f) = \max_{\theta_d}[-\frac{1}{N_S}\sum_{i=1}^{N_S} L_d^i(\theta_f, \theta_d) - \frac{1}{N_T}\sum_{i=1}^{N_T} L_d^i(\theta_f, \theta_d)] \tag{6}$$

Equation (5) and Equation (6) are made up of a min-max adversarial optimisation procedure. DANN includes a deep feature extractor and a deep label predictor, which together construct a traditional standard feedforward architecture. The domain classifier is connected to the feature extractor. Last but not least, GRL plays an indispensable role in DANN, which builds a bridge between feature extractor and classifier for guiding feature extractor to acquire domain-invariant features. For domain classifier, it tries to distinguish between samples from source and target domain. However, feature extractor doesn't want to get some discriminative features of samples from source and target domain. So, in the updation processes of model parameters by backward propagation, the gradient is multiplied by a certain negative constant through GRL.

In short, the training processes of DANN are constraint through the min-max formula (Equation (5) and Equation (6)), and stopped training when the optimal balance train-off is found.

References

1. Lee, J. *et al.* Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mech. systems signal processing* **42**, 314–334 (2014).
2. Si, X.-S., Wang, W., Hu, C.-H. & Zhou, D.-H. Remaining useful life estimation–a review on the statistical data driven approaches. *Eur. journal operational research* **213**, 1–14 (2011).
3. Li, X., Zhang, W. & Ding, Q. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliab. Eng. & Syst. Saf.* **182**, 208–218 (2019).
4. Ren, L., Cui, J., Sun, Y. & Cheng, X. Multi-bearing remaining useful life collaborative prediction: A deep learning approach. *J. Manuf. Syst.* **43**, 248–256 (2017).
5. Chen, J., Jing, H., Chang, Y. & Liu, Q. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliab. Eng. & Syst. Saf.* **185**, 372–382 (2019).
6. She, D. & Jia, M. A bigru method for remaining useful life prediction of machinery. *Measurement* **167**, 108277 (2021).
7. Ellefsen, A. L., Bjørlykhaug, E., Æsøy, V., Ushakov, S. & Zhang, H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliab. Eng. & Syst. Saf.* **183**, 240–251 (2019).
8. Cheng, Y., Wu, J., Zhu, H., Or, S. W. & Shao, X. Remaining useful life prognosis based on ensemble long short-term memory neural network. *IEEE Transactions on Instrumentation Meas.* **70**, 1–12 (2020).
9. Guo, L., Li, N., Jia, F., Lei, Y. & Lin, J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing* **240**, 98–109 (2017).
10. Jiang, J. *et al.* Regressive domain adaptation for unsupervised keypoint detection. *arXiv preprint arXiv:2103.06175* (2021).
11. Ganin, Y. *et al.* Domain-adversarial training of neural networks. *The journal machine learning research* **17**, 2096–2030 (2016).
12. Li, N., Lei, Y., Lin, J. & Ding, S. X. An improved exponential model for predicting remaining useful life of rolling element bearings. *IEEE Transactions on Ind. Electron.* **62**, 7762–7773 (2015).
13. Lei, Y., Li, N. & Lin, J. A new method based on stochastic process models for machine remaining useful life prediction. *IEEE Transactions on Instrumentation Meas.* **65**, 2671–2684 (2016).
14. Guo, L., Gao, H., Huang, H., He, X. & Li, S. Multifeatures fusion and nonlinear dimension reduction for intelligent bearing condition monitoring. *Shock. Vib.* **2016** (2016).
15. Wang, Y., Peng, Y., Zi, Y., Jin, X. & Tsui, K.-L. A two-stage data-driven-based prognostic approach for bearing degradation problem. *IEEE Transactions on industrial informatics* **12**, 924–932 (2016).
16. Zhao, D., Liu, F. & Meng, H. Bearing fault diagnosis based on the switchable normalization ssgan with 1-d representation of vibration signals as input. *Sensors* **19**, 2000 (2019).
17. Nectoux, P. *et al.* Pronostia: An experimental platform for bearings accelerated degradation tests. In *IEEE International Conference on Prognostics and Health Management, PHM'12.*, 1–8 (IEEE Catalog Number: CPF12PHM-CDR, 2012).
18. Wang, B., Lei, Y., Li, N. & Li, N. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliab.* **69**, 401–412 (2018).
19. Benkedjouh, T., Medjaher, K., Zerhouni, N. & Rechak, S. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Eng. Appl. Artif. Intell.* **26**, 1751–1760 (2013).
20. Li, X., Ding, Q. & Sun, J.-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab. Eng. & Syst. Saf.* **172**, 1–11 (2018).
21. Farahat, A., Gupta, C. *et al.* Similarity-based feature extraction from vibration data for prognostics. In *Annual Conference of the PHM Society*, vol. 12, 10–10 (2020).
22. Ben-David, S. *et al.* A theory of learning from different domains. *Mach. learning* **79**, 151–175 (2010).

Acknowledgements

This work was supported by the National Key RD Program of China under grant No.2019YFB2102500 and the Science Foundation of Beijing-Shanghai High Speed Railway Co.,Ltd under grant No.2019166.

Author contributions statement

Formal analysis, D.Z.; Software, D.Z.; Writing—original draft, D.Z.; Check, D.Z. and F.L. All authors have read and agreed to the published version of the manuscript.

Additional information

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to D.Z.