Unsupervised Domain Adaptation based Remaining Useful Life Prediction of Rolling Element Bearings

Chenyu Liu[1,2] and Konstantinos Gryllias[1,2]

[1] *Division Mecha(tro)nic System Dynamics, Department of Mechanical Engineering, KU Leuven*
[2] *Dynamics of Mechanical and Mechatronic Systems, Flanders Make*
Celestijnenlaan 300, BOX 2420, 3001 Leuven, Belgium
chenyu.liu@kuleuven.be, konstantinos.gryllias@kuleuven.be

Abstract

With the rise of Artificial Intelligence (AI), machine learning techniques are now conquering the research field of Prognostics and Health Management (PHM). Classic deployable prognostic models manipulate large amount of machinery historical data to map the degradation process based on inherent features. Nowadays one of the major challenges in prognostics research is the data deficit problem when historical data is not available or accessible, in enough quantity and variety. In the frames of Transfer Learning, the domain adaptation technique aims to build a model with strong generalization ability which can be transferred to datasets with different distributions. In this paper, a Domain Adversarial Neural Network (DANN) model is combined with a Bidirectional Long Short-Term Memory (Bi-LSTM) neural network for the estimation of the Remaining Useful Life (RUL) of rolling element bearings. The unsupervised domain adaptation is fulfilled using a labelled bearing degradation dataset as the source domain data and an unlabelled dataset captured under different operation conditions as the target domain data for the Bi-LSTM DANN. The proposed method achieves promising results, applied on real bearing vibration data captured on run-to-failure tests, with high prediction accuracy of the bearing RUL compared to un-adapted methods.

1. Introduction

Modern condition monitoring techniques leverage digitalized sensory connectivity to collect a vast amount of data reflecting the health status of industrial assets. Sophisticated technologies of industrial informatics, including microelectromechanical sensors, Internet-of-Things communication, and edge-cloud computing platforms, enable the processing of Big Data at a low cost. In the meantime, data-driven based condition monitoring methods have been proposed to align the data efforts to the practical industrial needs, especially for the predictive maintenance scheduling based on the prediction of the components' RUL. Currently, the common deployable RUL prediction approach manipulates the historical data from the target mechanical setup in order to map the prognostic model. Considering the long operating time and the varied degradation modes of the run-to-failure tests, the data collection, and the model construction process can be extremely expensive in practical industrial applications. One possible solution to this challenge would be to take advantage of the model generalization ability by transferring the prognostic model from one machine to others.

Machine learning models have been considered as indicative tools of this solution. In the frame of machine learning, an intuitive implementation of this scenario is to apply a model, trained with the data and the labels from other machinery, directly on the target dataset for RUL estimation. The performance of this method is strongly related to the data amount and the variety of operating conditions during the training process. Current research has shown that the statistical distribution of the sensory dataset can behave very differently even for an identical mechanical setup due to unseen factors, such as the lubrication conditions, the temperature, and the assembly errors, which demonstrate the stifling fact that the constructed fine-tuned machine learning models are case-specified.

As the new frontier of machine learning, Transfer Learning (TL) is a promising approach to improve the model generalization ability. By transferring knowledge between models, TL leverages a model, trained on source domain data, to be used directly on target domain data to facilitate the model developing process. According to the model transfer tasks between the two domains, TL models can be categorized into three types, i.e. the inductive, the transductive and the unsupervised model (Pan & Yang, 2010). The transferred knowledge can be either the model parameters or feature representations. Some instances have shown the high performance of

TL in rotating machinery condition monitoring. A transferable Convolutional Neural Network (CNN) was proposed for bearing and gearbox diagnostic tasks by Chen et al. (2019). By freezing the weights of several layers and fine-tuning the rest, the model was expected to extract more discriminative features compared to the un-transferred ones. Targeting at the simulation of aeroengine degradation datasets, Fan et al. (2019) transferred the features learned from consensus self-organized models and utilized a random forest regressor to predict the RUL. Shao et al. (2018) employed the weights of a pre-trained CNN based on the ImageNet dataset on a classification model using a 2d spectrogram as inputs for gearbox fault diagnostics. Through these intriguing findings, TL models are proved effective in supervised learning problems with known label information for both target and source domain data during the model fitting process.

In practical prognostic scenarios, the RUL labels for the target domain data are not accessible, which makes the supervised TL models infeasible to deploy. Considering the distribution mismatch between the source and the target datasets, the domain adaptation method of TL employs a shared feature space that can bypass the labelling issue by training the model in a semi-supervised or unsupervised way. One of the domain adaptation techniques is using the Maximum Mean Discrepancy (MMD) as the metric to evaluate the discrepancy between the two domains. Wen et al. (2017) explored the combination of MMD with a deep auto-encoder to classify the unlabelled target domain data for the bearing fault diagnostic task.

Another domain adaptation technique, the Domain Adversarial Neural Network (DANN), has gained much interest from machine learning researchers in the directions of Natural Language Processing (Fu et al., 2017), machine translation (Britz et al., 2017), and semantic segmentation (Tsai et al., 2018). DANN employs a domain classifier to discriminate the features from either the source or the target domain. By making use of adversarial training, the joint feature representations are extracted when the classifier can no longer identify which domain these features are coming from. On the other hand, the features are simultaneously fed to a label predictor to learn the labels of the target domain data as the outputs.

Some attempts have been made using DANN for machinery condition monitoring purposes. Han et al. (2019) fed the vibration signals to a DANN model using a 1d CNN based feature extractor for the fault diagnostics of wind turbines. Q. Wang et al. (2019) introduced a semi-supervised way of using DANN for bearing fault diagnosis and meanwhile compared the results with other domain adaptive methods. Beyond the classification tasks, da Costa et al. (2020) applied the DANN framework on aeroengine degradation simulation datasets to predict the RUL, which shows the effectiveness of this method in dealing with prognostic problems.

Under the context of bearing prognostics, this paper utilizes a Bidirectional Long Short-Term Memory neural network (Bi-LSTM) as a feature extractor associated with the DANN model. Raw vibration signals from bearing datasets captured under different operating conditions are used as source and target domain data. The model is applied in an unsupervised way without any priori information about the RUL of the target domain data. Compared to the un-adapted models, the proposed Bi-LSTM DANN shows better performance in the prediction of bearings RUL. The rest of the paper is organized as follows: the fundamental theories on DANN and Bi-LSTM are introduced in Section 2, and the proposed model is discussed in Section 3. The run-to-failure experimental setup and the datasets are illustrated in Section 4. The experimental results, as well as the comparative analysis, are presented in Section 5. The paper closes with some conclusions in the last section.

2. THEORETICAL PART

2.1. Domain Adversarial Neural Network (DANN)

DANN can be traced back to the research from Ganin & Lempitsky (2014) and Ganin et al. (2016). It is designed to encounter the domain shift problem, i.e. the distribution mismatch of the training and the testing datasets. Inspired by the Generative Adversarial Networks (GANs), DANN uses adversarial training to construct a domain-invariant feature space for both the source and the target domain data. These features are simultaneously sent to a domain classifier and a label predictor. By learning from the source domain data and the associated labels, the goal of DANN is to map a function that can precisely label the target domain data.

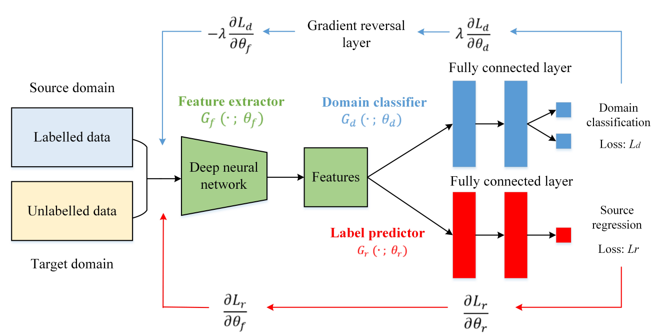


Figure 1. The architecture of DANN for regression.

As shown in Figure 1, the DANN architecture includes three parts: the feature extractor $G_f(\cdot;\theta_f)$, the domain classifier $G_d(\cdot;\theta_d)$ and the label predictor $G_r(\cdot;\theta_r)$, where θ_f, θ_d and θ_r are respectively their hyper-parameters. G_f is used to create the feature space needed for the following networks based on the mixture of source and target domain samples. In the forward propagation, the features are sent to G_d to classify whether they come from the source or the target domain. The loss of G_d can be described as:

$$L_d(\theta_f, \theta_d) = L_d(G_d(G_f(x; \theta_f); \theta_d)) \tag{1}$$

where x is the input. According to Ganin et al. (2016), the training target of G_d is to reduce the $\mathcal{H}$-divergence between the two domains, which can be fulfilled using a Gradient Reversal Layer (GRL). During the back propagation process, the GRL is inserted between the domain classifier and the feature extractor to make the reverse training targets, i.e. G_f is optimized to extract domain-invariant features from the two domains, but G_d is optimized to discriminate as much as possible their belonging domain. GRL can be simply implemented by multiplying the gradient by -1 without introducing other hyper-parameters. When the classifier can no longer identify the exact domain of a sample, the domain-invariant feature space aligning the two distributions is considered to be successfully constructed.

Since the labels are available for the source domain data, the features are simultaneously sent to G_r, which is trained in a supervised way. The loss function of G_r is described as:

$$L_r(\theta_f, \theta_r) = L_r(G_r(G_f(x; \theta_f); \theta_r)) \tag{2}$$

Combining the domain classifier and the label predictor, the total loss of DANN can be defined as follows according to Ganin et al. (2016):

$$\begin{aligned} L(\theta_f, \theta_d, \theta_r) = & \frac{1}{n}\sum_{i=1}^{n} L_r^i(\theta_f, \theta_r) - \\ & \lambda\left(\frac{1}{n}\sum_{i=1}^{n} L_d^i(\theta_f, \theta_d) + \frac{1}{n'}\sum_{i=1}^{n'} L_d^i(\theta_f, \theta_d)\right) \end{aligned} \tag{3}$$

where n and n' are respectively the numbers of samples from the source and the target domain. λ is the weight scalar of the loss. By implementing optimizers like SGD, Adam, or RMSProp, Equation 3 is expected to converge to a saddle point, and then the labels of the target domain can be predicted. It is noticeable that the DANN model can be used in either a semi-supervised or unsupervised way. The proposed method in this paper is leveraging the unsupervised DANN with the Bi-LSTM feature extractor.

2.2. Bi-LSTM feature extractor

As one of the most popular algorithms for sequence related problems, LSTM is an all-rounder tool which has been implemented in image caption, text translation, handwriting generation, etc. (Brownlee, 2017). Generally, vibration-based rotating machinery degradation datasets embed the degradation information in the time domain observations, measurement by measurement till the end of the life. The temporal connections of the signals should be taken into account when dealing with sequential learning. Compared to other types of neural networks, LSTM has unparalleled advantages to satisfy this requirement with the ability to learn and manipulate the memory or state from the previous time steps.

Classic LSTM network uses the input time sequence in one direction, i.e. from the past to the future. When the entire input sequence is available, a bidirectional training of the LSTM from both forward and backward directions gives access to the information of both the past and the future time stamps. Bi-LSTM has been proved an effective method to lift the performance of the sequence prediction results.

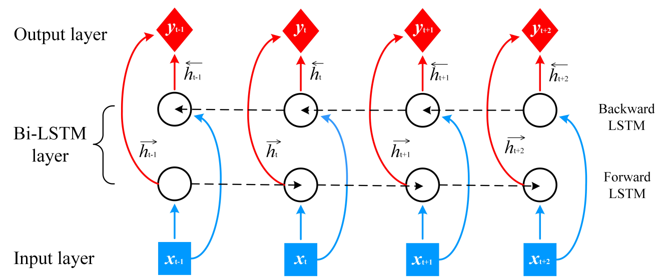


Figure 2. The schematic of the Bi-LSTM neural network.

Figure 2 illustrates the schematic of the Bi-LSTM with the input x feeding into the LSTM neurons from two directions at different time step t. For regular one-directional LSTM, the hidden state of the neuron is denoted with the following equation:

$$h_t = f(x_t, h_{t-1}; \theta_{LSTM}) \tag{4}$$

where θ_{LSTM} represents the hyper-parameters, and f is the mapping function. The internal computation process of an LSTM memory unit is specified below according to Jozefowicz et al. (2015):

$$\begin{cases} i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i) \\ f_t = \sigma(W_f x_t + R_f h_{t-1} + b_f) \\ z_t = tanh(W_z x_t + R_z h_{t-1} + b_z) \\ c_t = z_t * i_t + c_{t-1} * f_t \\ o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o) \\ h_t = tanh(c_t) * o_t \end{cases} \tag{5}$$

where the variables of W and R are respectively the weights related to the current states and the previous states. The variables of b represent the biases. σ and $tanh$ denote respectively the logistic sigmoid and the hyperbolic tangent function. The symbol $*$ indicates the element-wise multiplication.

In the structure of Bi-LSTM, the variables mentioned above are calculated separately for the forward and backward directions denoted, respectively, with $\rightarrow$ and $\leftarrow$ on top. Thus the hidden state, as well as the output of the time step t of Bi-LSTM, can be described as:

$$y_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{6}$$

where $\oplus$ is the element-wise sum. In this way, the Bi-LSTM can effectively capture the long-term dependencies through the input time sequences. By considering the vibration signals as inputs, a Bi-LSTM network is used in this paper as the feature extraction part in the frame of DANN. The detailed implementation is discussed in the following section.

3. PROPOSED METHOD

Vibration signals have been proved efficient to reflect and monitor the degradation of components, which are collected as input data. The flow chart of the proposed method is depicted in Figure 3, which is composed as follows. The source domain data includes the vibration signals and the RUL labels from the bearing under one certain operating condition. The target domain data contains only the signals from the bearing under another operating condition. These two types of information are passed to the proposed neural network.

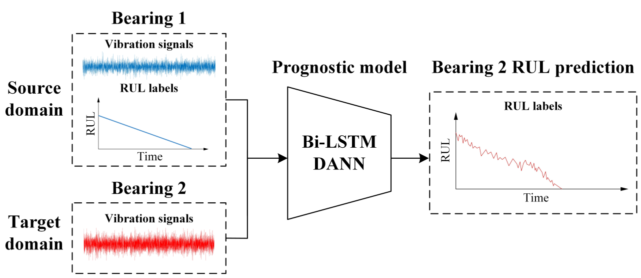


Figure 3. Flow chart of the proposed prognostic method.

In the DANN model, the Bi-LSTM feature extractor is formed with Bi-LSTM and Fully Connected (FC) layers. The FC layers are also used in the domain classifier and the label predictor with shared weights from the Bi-LSTM. A softmax and a linear activation function are applied respectively in the last layers of the domain classifier and the label predictor. In this implementation, the cross-entropy loss function is used for the domain classifier. Since the label predictor is dealing with the regression problem, the root mean square error loss function is selected for the optimization of the label predictor. Based on the work of Elsheikh et al. (2019), the architecture and the parameters of the proposed Bi-LSTM DANN can be found in Table 1.

Besides the layers mentioned above, the GRL is added between the last FC layer of the feature extractor and the first FC layer of the domain classifier, as discussed by Ganin et al. (2016). The weight of loss λ between the domain classifier and the label predictor is set to 1.0 in this work.

4. EXPERIMENTAL PART

4.1. XJTU-SY bearing datasets

The proposed model is evaluated using the XJTU-SY bearing datasets (B. Wang et al., 2018). The testing setup is shown in Figure 4. The vibration signals of 15 type LDK UER204 rolling element bearings are collected, during accelerated run-to-failure tests, by two accelerometers with a sampling frequency of 25.6 kHz. The sampling duration of each signal is 1.28s (32,768 points), and the signal collection interval is 1 minute during the whole measurement campaign.

Table 1. Architecture of the Bi-LSTM DANN.

Feature extractor		
1	Input layer	Size: length of the signal
2	Bi-LSTM layer	Unit: 500
3	Dropout layer	Rate: 0.5
4	FC layer	Unit: 500, activation: Relu
5	Dropout layer	Rate: 0.5
6	Feature output	Unit: 200, activation: Relu
Domain classifier		
7	FC layer	Unit: 100, activation: Relu
8	Dropout layer	Rate: 0.5
9	Domain classification	Unit: 2, activation: Softmax
Label predictor		
10	FC layer	Unit: 100, activation: Relu
11	Dropout layer	Rate: 0.5
12	Source regression	Unit: 1, activation: Linear

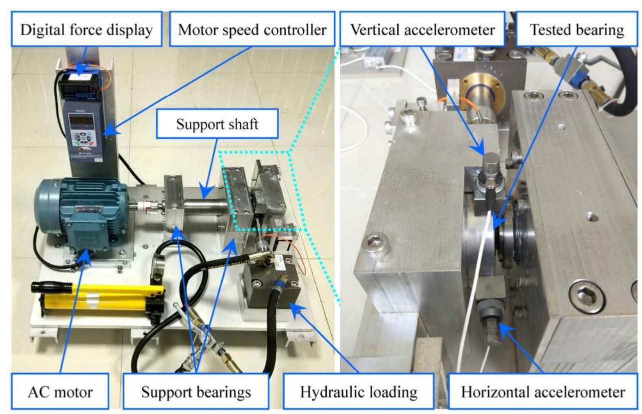


Figure 4. Experimental setup of XJTU-SY bearing datasets.

The accelerated run-to-failure tests are realised under 3 operating conditions with different load and speed, as presented in Table 2. The data collection automatically stops when the maximum amplitude of the vibration signals exceeds 10 times the normal status. The failure types of the 15 bearings have been inspected after the experiment and are listed also in Table 2.

4.2. Data preprocessing and hyper-parameter selection

The measured raw vibration signals are used as inputs to the feature extraction part of the proposed model. To fit in the Bi-LSTM layer of the network, the input is reshaped as a raw (N, L_{sig}, N_f) where N is the number of measurements

Table 2. Operating conditions and bearing failure types.

Condition 1 2100 rpm/12 kN		**Condition 2** 2250 rpm/11 kN		**Condition 3** 2400 rpm/10 kN	
Bearing	Fault	Bearing	Fault	Bearing	Fault
1_1	Outer race	2_1	Inner race	3_1	Outer race
1_2	Outer race	2_2	Outer race	3_2	Inner race Outer race Ball Cage
1_3	Outer race	2_3	Cage	3_3	Inner race
1_4	Cage	2_4	Outer race	3_4	Inner race
1_5	Inner race Outer race	2_5	Outer race	3_5	Outer race

which will be loaded and is equal to the batch number, L_{sig} is the length of the signal and N_f is the number of features which in this work is set to 1. To label the datasets with the actual RUL values, the bearing degradation is presumed as a linear decreasing process in the entire life duration, which indicates a range from 1 to 0 for the normalized RUL. The predicted RUL can be calculated based on the normalized value and the inspecting time stamp T as follows:

$$RUL_{pred} = \frac{RUL_{norm}}{1 - RUL_{norm}} T \tag{7}$$

For the hyper-parameters of the proposed model, the batch size of training is selected from the set $\{64, 128, 256\}$ based on grid search. Adam is used as the optimizer for both the domain classifier and the label predictor with the learning rates set respectively equal to 10^{-3} and 10^{-4}.

4.3. Comparative methods

Three different types of neural networks, including Multilayer Perceptron (MLP), 1D CNN, and classic one-directional LSTM, are selected as comparative feature extractors to assess the performance in the frame of DANN. The structure and the parameters of these networks follow respectively the work of Heimes (2008), Liu et al. (2019) and Zhang et al. (2019). For the MLP model, the raw vibration signal is truncated to the length of 2,000 as input for computation efficiency. The kernel sizes of the 1D CNN model are modified as $\{64, 64, 32, 32, 16, 16\}$ for the 6 layer CNN architecture. Adam optimizers are employed for all these models. The architecture and the parameters of the domain classifier and the label predictor remain the same as the proposed model.

The comparison between the un-adapted and the domain adaptation models is conducted based on the Bi-LSTM DANN and three regression models. Two intuitive regression models are selected as baseline models, i.e. the Support Vector Regressor (SVR) and the Random Forest Regressor (RFR), with the parameters from Soualhi et al. (2014) and Patil et al. (2018) as references. Additionally, a source regression Bi-LSTM model using the label predictor part of the proposed DANN architecture is also adopted to examine the influence of the domain shift.

4.4. Evaluation metrics

The performance of the prognostic model can be measured via several metrics (Saxena et al., 2010). The Root Mean Squared Error (RSME) and the Mean Absolute Error (MAE), as denoted in Equation 8 and 9, appear to be the most commonly used ones in the literature and therefore are adopted in this work for performance evaluation.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(RUL_{act}^{i} - RUL_{pred}^{i})^2}{N}} \tag{8}$$

$$MAE = \frac{\sum_{i=1}^{N}\left|RUL_{act}^{i} - RUL_{pred}^{i}\right|}{N} \tag{9}$$

RUL_{act} represents the actual RUL label, and RUL_{pred} is the predicted RUL. i is the sequence number of the time stamp, and N is the total number of measurements.

5. RESULTS

All model computation tests are performed in this paper on an Intel Xeon Gold 6140 (2.3 GHz) with 192 GB RAM and NVIDIA Tesla P100 GPU. The proposed prognostic model is built with `Python 3.6` and `TensorFlow 1.9`. Due to the existence of the domain shift between different operating conditions, the implementation of the proposed model focuses in the cross-condition adaptation. Thus the source and the target domain datasets are selected from different operating conditions including three transfer scenarios: Condition 1 $\rightarrow$ Condition 2, Condition 2 $\rightarrow$ Condition 3 and Condition 3 $\rightarrow$ Condition 1. One of the bearings in each scenario is used for training and the rest are used for testing. Each test is repeated 10 times, and the averaged metrics are recorded to reduce the influence of randomness.

The bearing fault type is considered related to the degradation mode. Therefore it is also necessary to investigate if the proposed model could achieve cross-mode adaptation. In the following subsections, the comparative analysis of the feature extractors and the un-adapted models are performed based on the bearings with the same type of fault (outer race defect). Moreover, the evaluation of cross-mode adaptation is analyzed by using bearings with different faults in the source and the target domain.

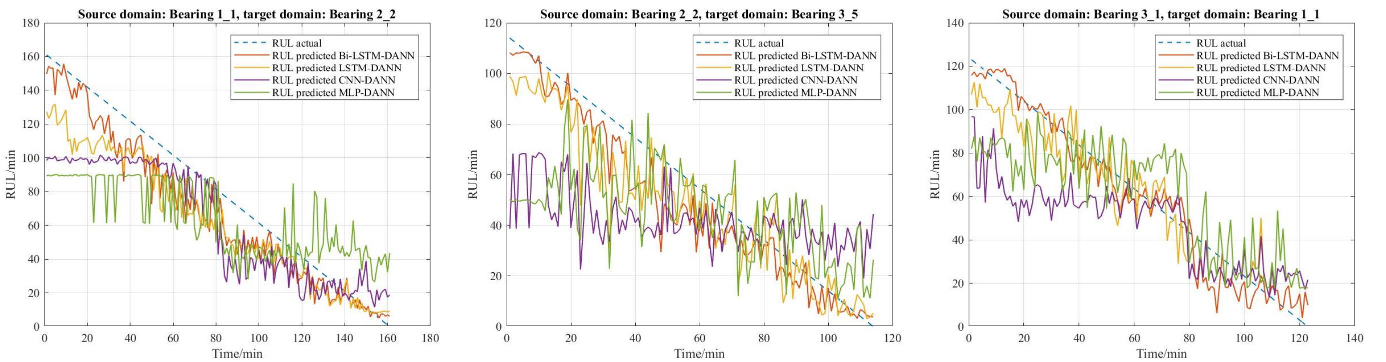
5.1. Feature extractor comparison

As illustrated in the previous section, three feature extractors are proposed within the same architecture of DANN to be compared with the Bi-LSTM network. The batch size is set to 128 in this test. The detailed results of the performance are listed in Table 3, where the RMSE and the MAE of each architecture are presented with their standard deviation.

The results show that the Bi-LSTM and LSTM architectures

Table 3. Results of the feature extraction networks with DANN (Unit: min).

Source domain	Target domain	MLP		CNN		LSTM		Bi-LSTM	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1_1	2_2	31.72±2.82	26.38±4.22	29.87±7.21	24.72±3.23	18.28±2.02	14.57±3.01	15.11±2.09	12.19±2.12
1_1	2_4	30.57±1.25	25.10±2.20	29.63±2.35	24.60±4.82	19.45±1.01	15.37±1.72	16.79±2.08	13.43±1.02
1_1	2_5	26.88±3.02	23.02±2.26	24.75±2.20	20.55±2.42	22.51±2.73	18.60±3.02	13.23±1.52	10.17±0.96
1_1	3_1	37.69±4.52	33.10±3.20	36.38±4.62	32.02±1.24	28.36±3.25	25.50±2.47	22.21±1.01	19.72±1.20
2_2	1_2	20.89±1.28	16.93±2.37	18.80±1.22	15.49±2.06	16.61±0.82	13.20±1.03	16.22±1.06	12.87±1.83
2_2	1_3	22.19±0.99	17.21±1.20	19.00±1.63	14.87±1.03	16.07±0.77	12.96±1.31	15.87±0.93	12.03±1.09
2_2	3_1	36.17±2.08	32.01±1.73	34.80±1.92	30.07±2.00	28.23±1.14	24.94±1.04	26.09±1.87	22.21±1.79
2_2	3_5	26.66±1.29	20.13±1.03	26.60±1.35	21.37±0.83	18.95±1.23	15.83±0.87	14.69±1.32	11.78±1.28
3_1	1_1	18.61±1.97	14.56±1.52	17.72±1.33	14.22±2.10	12.26±1.35	9.92±1.43	11.77±1.08	8.02±1.29
3_1	1_2	19.76±2.26	15.99±1.92	17.80±1.77	14.59±1.26	13.77±1.87	10.09±1.32	11.83±1.05	8.13±1.35
3_1	1_3	20.32±1.57	16.81±2.02	19.96±2.06	16.01±1.66	13.92±1.27	10.01±2.42	12.20±1.62	9.63±1.51
3_1	2_2	19.29±2.81	16.02±1.87	20.03±1.78	17.72±1.97	15.39±1.86	12.04±1.20	16.10±1.01	13.02±1.16
3_1	2_4	20.22±2.20	16.76±1.98	19.93±1.27	16.52±1.28	16.62±1.09	12.79±1.53	14.77±1.73	11.62±1.04



(a) (b) (c)

Figure 5. RUL predictions of DANN with different feature extractors. (a) Source domain bearing in Condition 1. (b) Source domain bearing in Condition 2. (c) Source domain bearing in Condition 3.

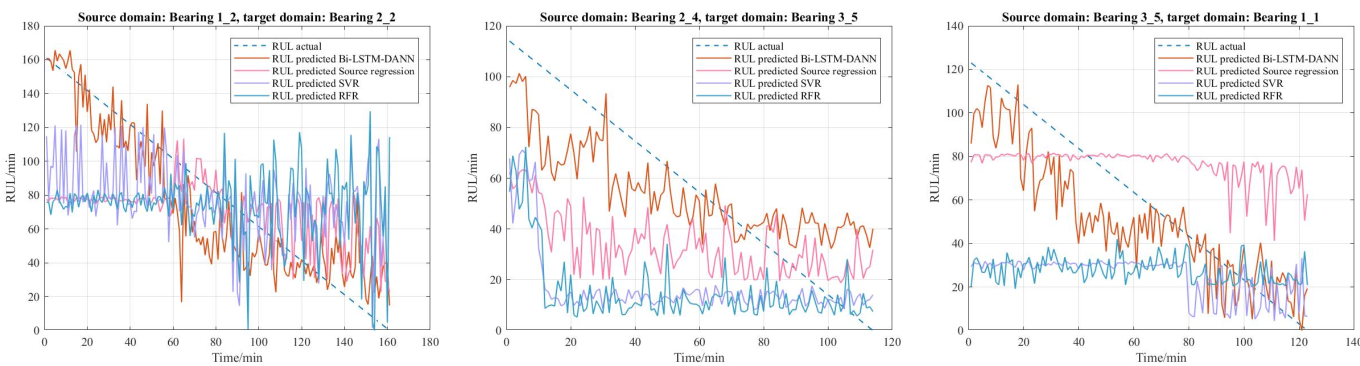
outperform the MLP and the 1D CNN feature extractors. Although MLP and 1D CNN can map a function between the extracted features from the time sequences and the RUL, these models cannot ensure the domain-invariability of the feature space since the long-term temporal information is not taken into account. On the other hand, the memory cell of LSTM gives access to the recurrent connections embedded in the input signal sequences thus can map the invariant features from the temporal flow.

Meanwhile, the Bi-LSTM DANN architecture shows better performance compared to the one-directional LSTM in most of the cases, which depicts the efficiency of the bidirectional training process and ends up with more accurate mapping of the RUL. The lowest prediction error is found in the case Bearing 3_1 as source and Bearing 1_1 as target where all the models achieve relatively low prediction errors. Two main reasons can be deducted: i) Massive source domain data is available. The long degradation duration of Bearing 3_1 provides a large number of measurements, including 2,538 signals. With the abundant input data, the dataset could contribute to a large feature space, which makes it easier to map domain-invariant features. This can be proved by the fact that most of the experiments using Bearing 3_1 as source get lower errors compared to other cases. ii) The degradation modes of the two bearings could be similar. Compared to other target bearings, all the feature extractors achieve relatively good results for this case. Therefore it can be concluded that it is easier to construct a joint feature space for the two bearings. This indicates that the dataset of Bearing 3_1 and 1_1 could have a close statistical distribution caused by similar degradation processes.

Figure 5 depicts the prediction results from 3 cases using the domain adaptation model with different feature extractors. It is evident that for all the plotted cases, the predictions of

Table 4. Comparison of the un-adapted models and the domain adaption DANN (Unit: min).

Source domain	Target domain	SVR		RFR		Source regression		Bi-LSTM DANN	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1_2	2_2	42.81±3.25	33.67±2.54	43.23±3.09	34.67±1.98	35.41±2.09	30.72±1.90	17.21±1.05	13.72±1.72
1_2	2_4	40.05±5.23	32.85±4.22	38.57±5.12	32.15±2.09	33.09±1.34	28.32±2.02	16.82±2.33	12.04±1.05
1_2	2_5	36.12±1.77	30.79±5.02	40.67±2.34	33.74±4.09	36.39±2.40	31.18±1.93	19.91±2.43	15.28±1.92
1_2	3_1	45.20±4.25	34.48±3.92	48.10±2.93	39.64±4.24	44.91±1.95	35.21±2.08	25.82±2.25	21.87±2.08
2_4	1_2	65.63±7.02	53.52±10.32	60.02±7.21	51.03±4.35	38.09±2.12	32.43±6.72	22.01±3.05	17.65±2.11
2_4	1_3	62.63±6.59	53.52±6.24	47.07±6.39	40.63±8.02	37.64±2.45	32.05±1.59	18.89±4.09	15.17±3.06
2_4	3_1	66.73±6.60	54.69±5.72	60.02±10.20	52.44±5.92	55.97±2.69	45.32±1.35	27.43±4.09	24.28±3.37
2_4	3_5	53.69±5.20	44.77±7.23	54.08±8.44	45.02±6.92	35.79±2.83	30.79±3.24	19.55±4.02	15.52±2.09
3_5	1_1	38.31±4.20	32.54±2.23	40.58±3.22	34.76±1.46	28.93±5.22	24.01±4.27	17.46±2.43	13.70±1.09
3_5	1_2	50.80±2.56	41.42±5.29	43.32±4.66	50.62±3.64	36.55±3.35	31.25±2.36	15.23±2.60	11.20±2.51
3_5	1_3	39.70±3.56	33.22±4.25	37.10±1.03	31.73±2.57	38.64±1.48	32.68±2.58	13.71±2.72	10.05±1.53
3_5	2_2	41.38±2.95	34.47±1.02	47.02±2.26	38.52±3.25	35.86±1.29	31.03±3.25	14.26±1.66	10.35±1.74
3_5	2_4	38.18±6.24	32.26±3.25	40.51±2.47	33.90±4.20	26.43±3.02	22.73±2.53	15.02±3.24	11.53±2.53



(a) (b) (c)

Figure 6. Comparison of the un-adapted models and domain adaption DANN. (a) Source domain bearing in Condition 1. (b) Source domain bearing in Condition 2. (c) Source domain bearing in Condition 3.

LSTM and Bi-LSTM follow the linear degradation process, but the MLP and the 1D CNN fail to capture the decreasing trend. Notably, in the case Bearing 2_2 as source and Bearing 3_5 as target, as shown in Figure 5b, the predictions made by the CNN-DANN and the MLP-DANN fluctuate around 50 min as noise, but the results of the Bi-LSTM and the LSTM models perform very close to the actual RUL curve.

5.2. Un-adapted methods

The results for 3 un-adapted regression models and the proposed model using batch size 256 are presented in Table 4. It is obvious that the un-adapted models, especially the shallow methods like the SVR and the RFR, present high prediction errors dealing with the domain shift. Since these methods can only obtain the superficial characteristics of the input data, the inherent temporal patterns of the vibration signals become obscure and irrelevant to map the RUL regression function.

The influence of domain shift can also be found from the source regression only model. Although the deep structure can extract more information related to the degradation process than the shallow methods, the model still suffers from the distribution mismatch and fails in cross-domain knowledge transfer. Bi-LSTM DANN presents better prediction results which proves the effectiveness of domain adaptation. Figure 6 shows the results from 3 cases of the un-adapted models and the Bi-LSTM DANN. The curves indicate that the SVR and the RFR can barely learn the decreasing trend from the inputs and result in strong randomness. In contract, the adapted DANN could obtain the information from source domain and applied on the target dataset with improved predictions.

5.3. Influence of the source data proportion

Supervised learning models prefer a large training dataset to extend the feature space, meanwhile avoiding overfitting. In prognostic cases, classic machine learning models mine the features from historical data for training, which might be 1/2 or even 2/3 of the entire degradation dataset. It is common knowledge that with more training data, the model performs better in the prediction. Although unsupervised domain adaptation does not specify a large training set, the scales of datasets for the source and target domain also influence the performance of the DANN model based on the results in the previous section. It is necessary to understand the impact considering the data proportion of the source and the target domain for the DANN model.

The source domain data proportion, denoted as ε, could be defined as:

$$\varepsilon = \frac{N_{source}}{N_{source} + N_{target}} \tag{10}$$

where N_{source} and N_{target} represent the number of samples respectively in the source and the target domain. In this study, each bearing dataset owns the different amounts of signals depending on the degradation duration, which results in different ε. Therefore the impact of data amount could be inspected in the form of prediction results vs. ϵ. Based on the 16 groups of tests from the test in Section 5.2, the $RMSE$ with the standard deviation for the Bi-LSTM DANN model and the un-adapted source-only Bi-LSTM model can be plotted as shown in Figure 7. The calculation for the ϵ of the un-adapted model also adopts Equation 10 regarding the training set as source and the testing set as target.

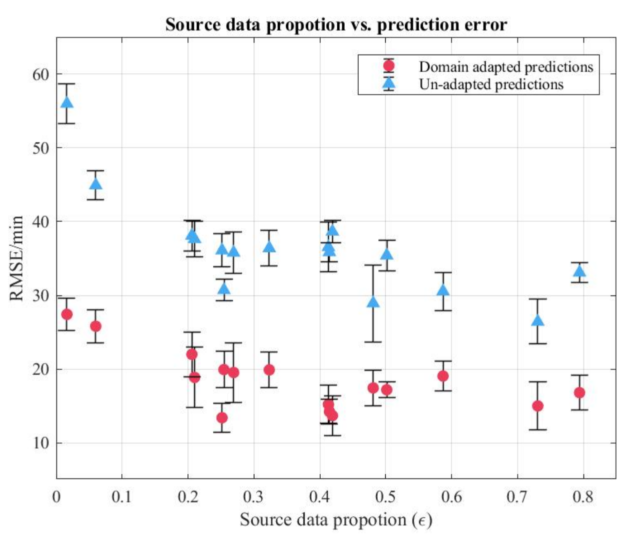


Figure 7. Prediction error under different data proportion.

It can be observed from Figure 7 that generally, a larger ϵ leads to a smaller prediction error for both the adapted and the un-adapted model. The un-adapted model is more vulnerable to the reduction of source data with a broader changing range from the highest error 55.97 min (Bearing 2_4 source, 3_1 target) to the lowest error 26.43 min (Bearing 3_5 source, 2_4 target). Although the changing of source data proportion also affects the DANN model, the results perform more robustly with a smaller fluctuation range from 27.43 min (Bearing 2_4 source, 3_1 target) to 13.42 min (Bearing 3_5 source, 2_5 target).

The influence of the data proportion to the performance of the domain adaptation model could be explained from the feature perspective. Based on the theory of domain adaptation, a domain-invariant feature space could be built with the DANN endeavours. With the changing of the data proportion for the two domains, the feature space might be irreversibly changed during the domain shift and thus leads to a different performance. These findings reveal the fact that the domain adaptation method is influenced by the data amount, especially in the source domain, but with less data dependency compared to un-adapted models.

5.4. Cross-mode adaptation

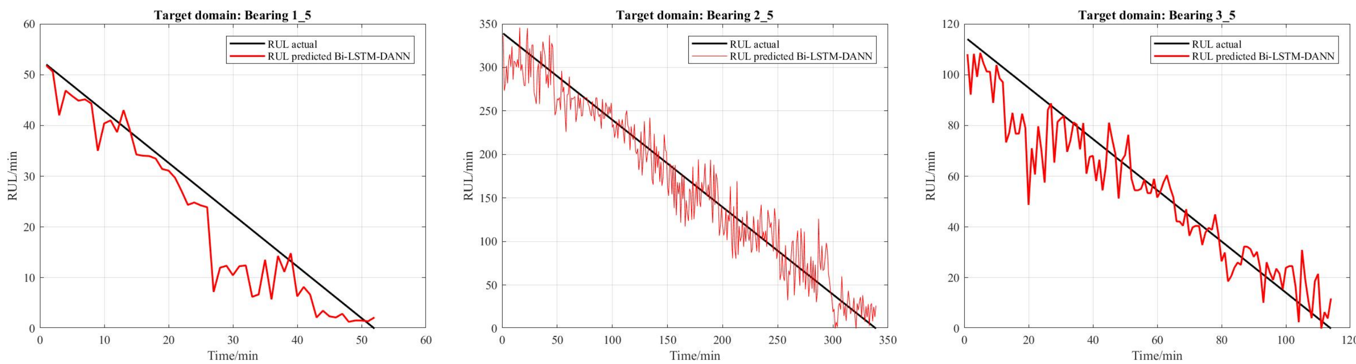
As mentioned in the previous section, domain adaptation aims to map a joint feature space as the mixture with domain-invariant features that can represent the statistical characteristics for both the two domain datasets. In the context of bearing degradation, similar degradation modes are assumed to trigger the same type of bearing faults and therefore should contain the information of invariant features. Nevertheless, it is also possible that such features can be found within different degradation modes as a more general existence. Based on this assumption, a universal cross-mode domain adaptation model could be constructed using multiple bearings with different types of faults as the source and should be valid on target bearings in any degradation mode.

Six groups of bearings with three different fault types are examined in the frame of Bi-LSTM DANN to investigate the cross-mode adaptation. The experimental results are presented in Table 5. Compared to the adaptation within signal degradation mode as addressed before, some cases achieve relatively high accuracy prediction, such as the transfer between Bearing 2_1 and Bearing 1_1. It is therefore indicated that some of the features could be extracted meanwhile adapted cross the inner race and the outer race fault. In other transfer cases, the results present a higher prediction error, especially for the cage fault. The reason could be either that the cage related degradation mode inherently shares limited domain-invariant features with the other two types of faults, or that the current feature extractor is not deep enough to map them.

The universal cross-mode adaptation model utilizes Bearing 1_5, 2_5, and 3_5 as the targets and the remaining 12 as the source which includes all the degradation modes with abundant data for feature space construction. The model leads to good prediction results, as shown in Figure 8, where the predicted RUL fits well with the linear degradation.

Table 5. Cross-mode domain adaptation results (Unit: min).

Transfer type	Source domain	Target domain	RSME	MAE
Inner race → Outer race	2_1	1_1	12.32±1.31	9.22±1.43
Outer race → Inner race	1_1	2_1	11.52±0.94	8.45±1.02
Outer race → Cage	1_1	2_3	13.01±1.28	9.82±1.32
Cage → Outer race	2_3	1_1	18.03±1.22	14.47±1.30
Cage → Inner race	2_3	3_3	16.72±2.04	12.03±1.65
Inner race → Cage	3_3	2_3	20.39±1.08	16.45±1.27
Multi-bearing → signal bearing	12 bearings	1_5	6.26±1.24	4.21±1.05
		2_5	18.23±1.20	14.31±1.02
		3_5	10.03±1.37	7.68±1.13



(a) (b) (c)

Figure 8. Universal cross-mode domain adaptation model testing results. (a) Target domain Bearing 1_5. (b) Target domain Bearing 2_5. (c) Target domain bearing 3_5.

6. CONCLUSION

In this paper, a Bi-LSTM DANN prognostic model is proposed based on the unsupervised domain adaptation. The Bi-LSTM neural network is utilized as a feature extractor combined with the DANN to deal with the domain shift problem. Based on the application on real bearing degradation datasets, the proposed method has been proven effective with the high prediction accuracy of the RUL. The comparative analysis is conducted on different types of feature extractors, as well as the un-adapted regression models. The influence of data amount to the prediction performance is discussed according to the evaluation of source domain data proportion. Moreover, the study explores the potential use of the Bi-LSTM model for different degradation modes based on the cross-mode adaptation ability. Due to the less reliance on a large amount of historical data, the proposed model could be used for prognostics in real industrial applications.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the China Scholarship Council, the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme, Flanders Make the strategic research centre for the manufacturing industry in the context of DGTwin Prediction project, the Flemish Supercomputer Center (VSC) for the computation resource, and Mr. Biao Wang and Prof. Yaguo Lei for providing the bearing datasets.

REFERENCES

Britz, D., Le, Q., & Pryzant, R. (2017). Effective domain mixing for neural machine translation. In *Proceedings of the second conference on machine translation* (pp. 118–126).

Brownlee, J. (2017). *Long short-term memory networks with python: Develop sequence prediction models with deep learning*. Machine Learning Mastery.

Chen, Z., Gryllias, K., & Li, W. (2019). Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network. *IEEE Transactions on Industrial Informatics*, *16*(1), 339–349.

da Costa, P. R. d. O., Akçay, A., Zhang, Y., & Kaymak, U. (2020). Remaining useful lifetime prediction via deep domain adaptation. *Reliability Engineering & System Safety*, *195*, 106682.

Elsheikh, A., Yacout, S., & Ouali, M.-S. (2019). Bidirectional handshaking lstm for remaining useful life prediction. *Neurocomputing*, *323*, 148–156.

Fan, Y., Nowaczyk, S., & Rögnvaldsson, T. (2019). Transfer learning for remaining useful life prediction based on consensus self-organizing models. *arXiv preprint arXiv:1909.07053*.

Fu, L., Nguyen, T. H., Min, B., & Grishman, R. (2017). Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of the eighth international joint conference on natural language processing (volume 2: Short papers)* (pp. 425–429).

Ganin, Y., & Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, *17*(1), 2096–2030.

Han, T., Liu, C., Yang, W., & Jiang, D. (2019). A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults. *Knowledge-Based Systems*, *165*, 474–487.

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *2008 international conference on prognostics and health management* (pp. 1–6).

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *International conference on machine learning* (pp. 2342–2350).

Liu, R., Yang, B., & Hauptmann, A. G. (2019). Simultaneous bearing fault recognition and remaining useful life prediction using joint-loss convolutional neural network. *IEEE Transactions on Industrial Informatics*, *16*(1), 87–96.

Pan, S., & Yang, Q. (2010). *A survey on transfer learning. ieee transactions on knowledge and data engineering.* IEEE.

Patil, S., Patil, A., Handikherkar, V., Desai, S., Phalle, V. M., & Kazi, F. S. (2018). Remaining useful life (rul) prediction of rolling element bearing using random forest and gradient boosting technique. In *Asme 2018 international mechanical engineering congress and exposition.*

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and health management*, *1*(1), 4–23.

Shao, S., McAleer, S., Yan, R., & Baldi, P. (2018). Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, *15*(4), 2446–2455.

Soualhi, A., Medjaher, K., & Zerhouni, N. (2014). Bearing health monitoring based on hilbert–huang transform, support vector machine, and regression. *IEEE Transactions on Instrumentation and Measurement*, *64*(1), 52–62.

Tsai, Y.-H., Hung, W.-C., Schulter, S., Sohn, K., Yang, M.-H., & Chandraker, M. (2018). Learning to adapt structured output space for semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 7472–7481).

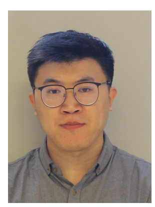
Wang, B., Lei, Y., Li, N., & Li, N. (2018). A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*.

Wang, Q., Michau, G., & Fink, O. (2019). Domain adaptive transfer learning for fault diagnosis. In *2019 prognostics and system health management conference (phm-paris)* (pp. 279–285).

Wen, L., Gao, L., & Li, X. (2017). A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(1), 136–144.

Zhang, B., Zhang, S., & Li, W. (2019). Bearing performance degradation assessment using long short-term memory recurrent network. *Computers in Industry*, *106*, 14–29.

BIOGRAPHIES



Chenyu Liu received the B.S. and M.Sc degree in aircraft manufacturing engineering and solid mechanics both from Northwestern Polytechnical University, China. He joined the Noise and Vibration Research Group in the Department of Mechanical Engineering, KU Leuven, Belgium as a PhD researcher in 2017. His research interests include data-driven based condition monitoring, rotating machinery prognostics, and machine learning and deep learning application.

Konstantinos Gryllias holds a 5 years engineering diploma degree and a PhD degree in Mechanical Engineering from National Technical University of Athens, Greece. Since October 2014 he holds an assistant professor position on vibro-acoustics of machines and transportation systems at the Department of Mechanical Engineering of KU Leuven, Belgium. He is also the manager of the University Core Lab Dynamics in Mechanical & Mechatronic Systems DMMS-M of Flanders Make, Belgium. His research interests lie in the fields of condition monitoring, signal processing, prognostics and health management of mech. & mechatronic systems.