Bearing remaining useful life estimation across operating conditions and bearing types based on multi-source domain adaptation

MSc Thesis at the Royal Netherlands Air Force

Maurice de Bruin

Technische Universiteit Delft

This page was intentionally left blank.

Bearing remaining useful life estimation across operating conditions and bearing types based on multi-source domain adaptation

MSc Thesis at the Royal Netherlands Air Force

by

Maurice de Bruin

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on 29 November 2024

Student number:	5082250
Project duration:	February, 2024 – November, 2024
Thesis committee:	
Chair	Ir. P.C. Roling
Examiner	Dr. V. Yaghoubi
Thesis supervisor	Dr. I. de Pater

The cover image has been taken from the Ministry of Defence:
`https://magazines.defensie.nl/vliegendehollander/2018/05/08_hot-blade`.

Preface

With this thesis, I conclude my studies in Aerospace Engineering at Delft University of Technology. I was fortunate to perform this thesis at the Royal Netherlands Air Force (RNLAF), where I could apply the skills I learned during my MSc studies to a challenging and relatively new area in data-driven predictive maintenance: transfer learning for bearing remaining useful life estimation. This project not only expanded my knowledge in the field of predictive maintenance but also gave me an interesting view inside the RNLAF.

I would like to thank my former supervisor, Marcia Baptista, for providing valuable advice during the initial stages of the project. I would also like to thank my current supervisor, Ingeborg de Pater, for her support of my work. The progress meetings were always very valuable and the feedback was always useful. I am also grateful to my supervisor from the RNLAF for providing me the opportunity to conduct my thesis at the RNLAF and his encouragement and support during my work. Additionally, I would like to thank the other interns at Woensdrecht Air Base for the fun working days and our visits to other air bases of the RNLAF. During my stay at Woensdrecht Air Base, I was able to visit every air base of the RNLAF, which was very memorable, interesting and fun.

Maurice de Bruin,
November 2024

Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
BPFO	Ball Pass Frequency Outer Race
BPFI	Ball Pass Frequency Inner Race
BSF	Ball Spin Frequency
IMS	Center for Intelligent Maintenance Sytems
CBM	Condition Based Maintenance
CF	Crest Factor
CI	Condition Indicator
CNN	Convolutional Neural Network
DANN	Domain Adversarial Neural Network
EI	Energy Index
FFT	Fast Fourier Transform
FTF	Fundamental Train Frequency
GRU	Gated Recurrent Unit
HI	Health Indicator
HUMS	Health and Usage Monitoring System
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MDAN	Multi-source Domain Adaptation Network
MMD	Maximum Mean Discrepancy
MSE	Mean Squared Error
NN	Neural Network
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
RKHS	Reproducing Kernel Hilbert Space
RMS	Root Mean Square
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RPM	Revolutions per Minute
RUL	Remaining Useful Life

Introduction

Bearings are an important part of rotating machinery, such as the drive train of a helicopter. The degradation of a bearing can lead to reduced system performance or even system failure. An example of such a failure is the crash of a Super Puma helicopter due to the failure of a bearing in the drive train. In order to prevent such accidents, helicopters are equipped with systems that monitor vibration levels at critical locations within the drive train. These systems enable condition-based maintenance, allowing maintenance activities to be scheduled in a cost and time efficient manner. In order to schedule these activities, accurate estimations of the time until failure of the system are important. This time until failure is also known as the RUL.

For estimating the RUL, data-driven methods are widely used in industry. However, data-driven models require large amounts of run-to-failure data of a bearing subjected to a certain operating condition. Such data is difficult to collect, as running a system to failure may lead to safety and cost concerns. Therefore, in practice, bearing run-to-failure data is only available in certain operating conditions. As a result, traditional data-driven models fail to produce accurate prediction results when applied to different operating conditions where run-to-failure data is unavailable.

To address this problem, the concept of transfer learning was introduced. Transfer learning aims to transfer degradation knowledge from a source domain to a target domain. By using this approach, degradation knowledge learned from run-to-failure data in specific operating conditions can be applied to operating conditions where run-to-failure data is unavailable. As a result, run-to-failure data is no longer required for every operating condition. Previous transfer learning approaches for RUL prediction have demonstrated promising prediction results.

Some systems, such as a helicopter drive train, only collect vibration data in the frequency domain. Therefore, this study aims to develop a prediction model capable of estimating the RUL across various operating conditions and bearing types, solely based on frequency domain data. The model makes use of all available degradation data from different operating conditions and bearing types to learn the relationship between degradation and RUL. Such a model allows for RUL predictions for a bearing type, even if no failures have been recorded for that specific type. This model can be used as a basis for further transfer learning approaches, where the relationship between degradation and RUL is derived from run-to-failure data collected under various operating conditions and bearing types in a laboratory. Next, this relationship is applied to a real-world bearing, such as the helicopter drive train. By using this method, the need for large amounts of run-to-failure data from actual helicopter drive train bearings can be minimised.

This study is divided into two parts. The first part, Part I, is a scientific paper that describes the RUL prediction model for estimating the RUL across various operating conditions and bearing types. The next part, Part II, is the literature review. In literature this review, the elements of transfer learning models are described.

I

Scientific Paper

Highlights

Bearing remaining useful life estimation across operating conditions and bearing types based on multi-source domain adaptation

Maurice de Bruin

- A multi-source domain adaptation model is proposed for bearing remaining useful life prediction across operating conditions and bearing types.
- As some systems only collect vibration data in the frequency domain, the proposed model generates RUL predictions exclusively based on frequency domain data.
- The model employs a common feature extractor to capture domain invariant features and domain specific regressors for RUL predictions.
- The proposed model achieves evaluation scores comparable to other advanced models that are able to use time and time-frequency domain data.

Bearing remaining useful life estimation across operating conditions and bearing types based on multi-source domain adaptation

Maurice de Bruin[a]

[a] Faculty of Aerospace Engineering, Delft University of Technology, Delft, 2629 HS, The Netherlands

ARTICLE INFO

Keywords:
Bearing remaining useful life prediction
Multi-source domain adaptation
Across operating conditions and bearing types
Frequency domain data
Transfer learning

ABSTRACT

For the remaining useful life (RUL) prediction of bearings across varying operating conditions, transfer learning models have demonstrated high accuracy. To make use of the maximum amount of bearing degradation information, multi-source domain adaptation models have been developed to enable predictions across operating conditions and bearing types. These models typically make use of vibration data in the time, frequency, and time-frequency domains. However, in practice, some systems only collect data in the frequency domain, such that prediction models developed for these systems lack access to degradation features from the time or time-frequency domains. To address this, this study proposes a multi-source domain adaptation (MSDA) model for bearing RUL prediction across operating conditions and bearing types using only frequency domain data as input data. The model employs a common feature extractor to capture domain invariant features, and domain specific regressors for RUL prediction. Domain adaptation is done by minimising the maximum mean discrepancy between features from the source and target domains, while the domain specific RUL predictions are aligned by minimising the distance between the RUL predictions. Experimental results across three scenarios demonstrate that the MSDA model achieves accurate RUL predictions. Even with only frequency domain data, the RMSE and Score are comparable to those of other advanced algorithms that use both frequency and time domain data.

1. Introduction

Bearings are an important part of rotating machinery, such as the drive train of a helicopter. Degradation of a bearing may lead to reduced system performance or system breakdown in case of a bearing failure [21]. An example of this is the crash of a Super Puma helicopter in Norway caused by the failure of a bearing in the drive train of this helicopter [1]. In order to prevent such accidents, the helicopter health and usage monitoring system (HUMS) has been introduced in the 1980s. Such systems collect vibration data from the rotating machinery and generate warnings when certain thresholds are exceeded [49]. Collecting this vibration data also contributes to the implementation of condition-based maintenance (CBM) [43].

CBM allows for cost and time efficient maintenance scheduling. By monitoring the degradation of the system, maintenance actions can be planned at the appropriate time. This leads to reduced system downtime and costs and improves safety [12, 16, 37, 38]. One part of CBM is prognostics and health management (PHM). PHM aims to detect and diagnose wear of the system and perform prognostics [11]. An example of a task related to PHM is estimating the time until failure of a component, which is also known as the remaining useful life (RUL) [12]. As RUL predictions are important for scheduling maintenance activities, RUL estimation methods are receiving a significant amount of interest from industry [42].

Bearing RUL prediction is done by collecting vibration data from sensors and feeding this data into a physics based model, a data driven model, or a hybrid model [26]. In a physics based model, the bearing degradation process is modelled by a mathematical model. Such models can deliver accurate results based on a physical phenomenon. However, deriving such models requires a significant amount of expertise and is time consuming [40]. In a data driven model, an algorithm aims to derive relationships between bearing health degradation and time to failure based on historical sensor data [12, 34]. As data driven models are less complex and more general compared to physics based models, they are widely used in industry [6, 12].

Examples of such data driven models are convolutional neural networks (CNN) and recurrent neural networks (RNN). Ding et al. constructed a deep convolutional neural network without a pooling layer to predict the RUL of bearings based on features obtained from the vibration data [7]. Nie et al. extracted similarity features from the time, frequency, and time-frequency domain in order to predict the bearing RUL using a CNN [28]. Han et al. used a long short-term memory (LSTM) model for bearing RUL prediction [15]. These methods used manual feature extraction for RUL prediction, which is time consuming [30]. Therefore, other methods used a CNN for automatic feature extraction from the raw vibration signal or the frequency spectrum of the signal [30, 41].

The aforementioned methods have achieved accurate RUL prediction results. These methods use bearing vibration data collected under a certain operating condition to train a RUL prediction model, which is then tested using data from the same operating condition. This approach ensures that the assumption of training and testing data coming from similar distributions is met [46]. In practical scenarios, however, vibration data is collected under various operating conditions and bearings may fail in different failure modes. In this case, the testing operating condition differs from the training operating condition, or the bearing failure behaviour is different, such that the assumption is not satisfied. As a result, the model may produce inaccurate prediction results for a bearing under a different operating condition, particularly when run-to-failure data for that condition is not available yet [4, 46, 50].

A solution to this problem is to construct more generalised RUL prediction models capable of adapting to different operating conditions. Hence, the concept of transfer learning was introduced. Transfer learning aims to transfer bearing degradation knowledge from a source domain to a target domain [4]. In this case, the source and target domain data are collected from different operating conditions, such that both domains follow different distributions. To align these distributions, Cheng et al., Dong et al., and Rathore et al. extracted features that minimised the distribution discrepancy using the maximum mean discrepancy metric [6, 10, 32]. Other scholars, such as Zeng et al. and Mao et al., instead used an adversarial approach involving a domain classifier to align the source and target domains [25, 44].

However, these transfer learning methods use source domain data from a single operating condition to predict the RUL of a target operating condition. However, in reality, source domain vibration data can be collected from multiple operating conditions. Source domain data from multiple operating conditions cannot be combined into a single source domain dataset, due to the distribution shift between operating conditions [8]. To allow for the use of more information on bearing degradation, Ding et al. developed a multi-source domain adaptation model for predicting the RUL of a bearing in a target operating condition. This approach aligned distributions of the multiple source domains and target domains with each other by minimising the domain discrepancy and aligned the predictions from the source domains by regressor adaptation [8].

In order to use even more bearing degradation information and to achieve more accurate prediction results, some scholars transferred degradation knowledge across different bearing types. For example, the source domain can contain vibration data from multiple bearing types, while the target domain is an operating condition of a certain bearing type. Another example is that the source domain contains data of one of these bearing types, while the target domain is of a different bearing type. In the latter case, it is possible to generate RUL predictions for a specific bearing type and operating condition, even if no failures have been observed for that bearing type yet. Only a few studies have been performed on these scenarios. Liu et al. developed a multi-source adversarial online knowledge distillation method for RUL predictions across different bearing types [22]. Shuang et al. constructed a multi-source multi-target domain adaptation model for RUL prediction across bearing types [31].

These multi-source transfer learning methods for RUL prediction across bearing types used the raw vibration data or features from the time domain and frequency domain as input to the model [22, 31]. However, some systems, such as helicopter drive trains, only provide monitoring data in the frequency domain. In this case, the model must provide RUL predictions based exclusively on frequency domain data. As a result, the model cannot use important degradation information provided in the time domain or time-frequency domain, complicating bearing RUL prediction [28, 31, 45].

Based on the literature review, no studies have been published on bearing RUL prediction across different bearing types using multi-source domain adaptation with only frequency domain data. To address this research gap, this study proposes a transfer learning RUL prediction model based on multi-source domain adaptation. A common feature extractor, which is a combination of a CNN and fully connected layers, is used to automatically extract features from a frequency spectrum. By minimising the maximum mean discrepancy between each pair of source and target domain features, domain invariant features are obtained. Then, each source domain is assigned a regressor for RUL prediction. The predictions of these regressors are aligned as well, by minimising the distance between the RUL predictions. To validate the effectiveness of the proposed model, several experiments on two different data sets are conducted. These experiments show that the proposed model accurately predicts the RUL across operating conditions and bearing types, using only frequency domain data. The main contributions of this paper are as follows:

1. A multi-source domain adaptation model (MSDA) is proposed to predict bearing RUL across different operating conditions and bearing types.
2. The MSDA model predicts bearing RUL exclusively based on frequency domain data.
3. Alignment of the source and target domain distributions, as well as RUL predictions, is done by minimising distribution discrepancies and the distance between predictions respectively.

The remainder of this paper is organised as follows: Section 2 provides a mathematical description of the transfer learning problem definition. Next, Section 3 describes the RUL prediction model. Section 4 presents experimental results. Finally, Section 5 concludes this paper.

2. Problem definition

In industrial scenarios, bearing run-to-failure data is not always available in certain operating conditions. Allowing a bearing to run to failure may lead to system damage or safety concerns. As a result, traditional RUL prediction models trained on a specific operating condition with run-to-failure data may show degraded performance if the prediction model is applied to a different operating condition without run-to-failure data [2].

Transfer learning aims to solve this problem by transferring knowledge from a source domain to a target domain [4]. In practical scenarios, bearing run-to-failure data is only available in certain operating conditions. This data is the source domain data, and provides knowledge about bearing degradation. Bearing RUL prediction is done for the target domain. The target domain does not contain run-to-failure data. Therefore, the target domain data alone is insufficient for constructing a RUL prediction model [4].

In multi-source domain adaptation, there are N source domain data sets, denoted by $D^s = \left\{\left(X^{s_j}, Y^{s_j}\right)\right\}_{j=1}^{N}$. Here, s_j indicates the j^{th} source domain data set. $X^{s_j} = \left\{x_i^{s_j}\right\}_{i=1}^{n_{s_j}}$, where n_{s_j} is the number of samples in source domain data set s_j. $x_i^{s_j}$ is a sample of pre-processed vibration data from data set s_j. The RUL labels are denoted by $Y^{s_j} = \left\{y_i^{s_j}\right\}_{i=1}^{n_{s_j}}$, where $y_i^{s_j}$ is the RUL label belonging to sample $x_i^{s_j}$. The target domain contains unlabelled data and is represented by $D^t = X^t = \left\{x_i^t\right\}_{i=1}^{n_t}$. Here, n_t is the number of samples in target domain data set [8].

The aim of this approach is to predict the RUL corresponding to the samples in the target domain data set D^t using bearing degradation knowledge from N source domain data sets. This approach builds on the assumption that although the bearing vibration data is collected under various operating conditions, the bearing degradation process from the initially healthy stage to failure shows similar characteristics in source and target domains [24].

3. Methodology

This section outlines the methodology to predict the RUL of a bearing in different operating conditions. First, the data pre-processing steps will be given in section 3.1. Next, a general overview of the proposed model will be presented in section 3.2. The model and domain adaption strategy will be described in detail in sections 3.3, 3.4, and 3.5. Finally, an overview of the model testing process is presented in section 3.6.

3.1. Data pre-processing

Before inputting a data sample into the MSDA model, some data pre-processing steps have to be taken. The first step is to convert the time domain data to a frequency domain representation, which is done using a Fast Fourier Transform (FFT) [29]. The FFT is shown in Equation 1.

$$z_i^{s_j}(k) = \sum_{m=0}^{M-1} \tilde{x}_i^{s_j}(m) e^{\frac{-j2\pi km}{M}}, \quad k = 0, 1, \ldots, M-1 \quad (1)$$

In Equation 1, $\tilde{x}_i^{s_j}(m)$ for $m = 0, 1, \ldots, M-1$ is the raw time domain vibration signal belonging to sample $\tilde{x}_i^{s_j}$. Additionally, M is the number of data points in signal $\tilde{x}_i^{s_j}$ and j is the imaginary number. Furthermore, $z_i^{s_j}$ is the frequency domain representation of time domain signal $\tilde{x}_i^{s_j}$ [18].

The second step is to reduce differences in data magnitude. Each frequency spectrum value $z_i^{s_j}(k)$ is standardised to $x_i^{s_j}(k)$ using Equation 2 [6].

$$x_i^{s_j}(k) = \frac{z_i^{s_j}(k) - \mu_i^{s_j}}{\sigma_i^{s_j}}, \quad \forall k \in M \quad (2)$$

In this equation, $x_i^{s_j}(k)$ is the standardised value of $z_i^{s_j}(k)$. $\mu_i^{s_j}$ is the mean of the spectral magnitudes and $\sigma_i^{s_j}$ is the variance of the spectral magnitudes.

Additionally, the MSDA model training process uses labelled source domain training data. In this study, the RUL is scaled between 0 and 1. If the RUL label equals 1, then the bearing is in healthy condition. If the RUL label equals 0, then the bearing has failed. The RUL is labelled by Equation 3 [30].

$$\text{RUL}_i = \frac{n_{\text{time steps}} - i}{n_{\text{time steps}}} \quad (3)$$

In this equation, $n_{\text{time steps}}$ is the total number of time steps for a bearing. i is the number of the i^{th} time step. The total number of time steps is equal to the number of acquisitions for a bearing. For example, if a bearing data set contains 1,000 acquisitions (i.e. vibration measurements at 1,000 moments in time), then $n_{\text{time steps}}$ equals 1,000.

Other approaches, such as [3] and [19], divide the bearing life cycle into a healthy and degraded phase. However, such approaches require additional expertise and may increase computational resources. Therefore, the proposed approach assumes that bearing degradation starts when the first measurements are recorded [30].

3.2. Model overview

In Figure 1, the structure of the MSDA model can be observed. The model consists of a common feature extractor and N domain specific regressors. Hence, N regressors are used to generate a RUL prediction. The feature extractor, a CNN denoted by $H(\cdot)$, extracts features from the frequency domain representation of a signal sample. The regressor, consisting of multiple fully connected layers denoted by $P_j(\cdot)$, predicts the RUL of the bearing based on the output of the feature extractor. In notation $P_j(\cdot)$, j denotes the source domain to which the regressor belongs.

During model training, a data sample $x_i^{s_j}$ from source domain j passes through through the common feature extractor and regressor j. As a result, a feature representation, $H(x_i^{s_j})$, and RUL prediction, $\hat{y}_i^{s_j} = P_j(H(x_i^{s_j}))$, are obtained respectively. A target domain data sample x_i^t passes through the common feature extractor and all the domain specific regressors in order to obtain feature representation $H(x_i^t)$ and RUL predictions $P_1(H(x_i^t)), \ldots, P_N(H(x_i^t))$ respectively.

In the following sections, each element of the MSDA model will be explained in more detail.

3.3. CNN structure for feature extraction

In Figure 1, it can be observed that feature extraction is done by a CNN. Such networks consist of convolutional

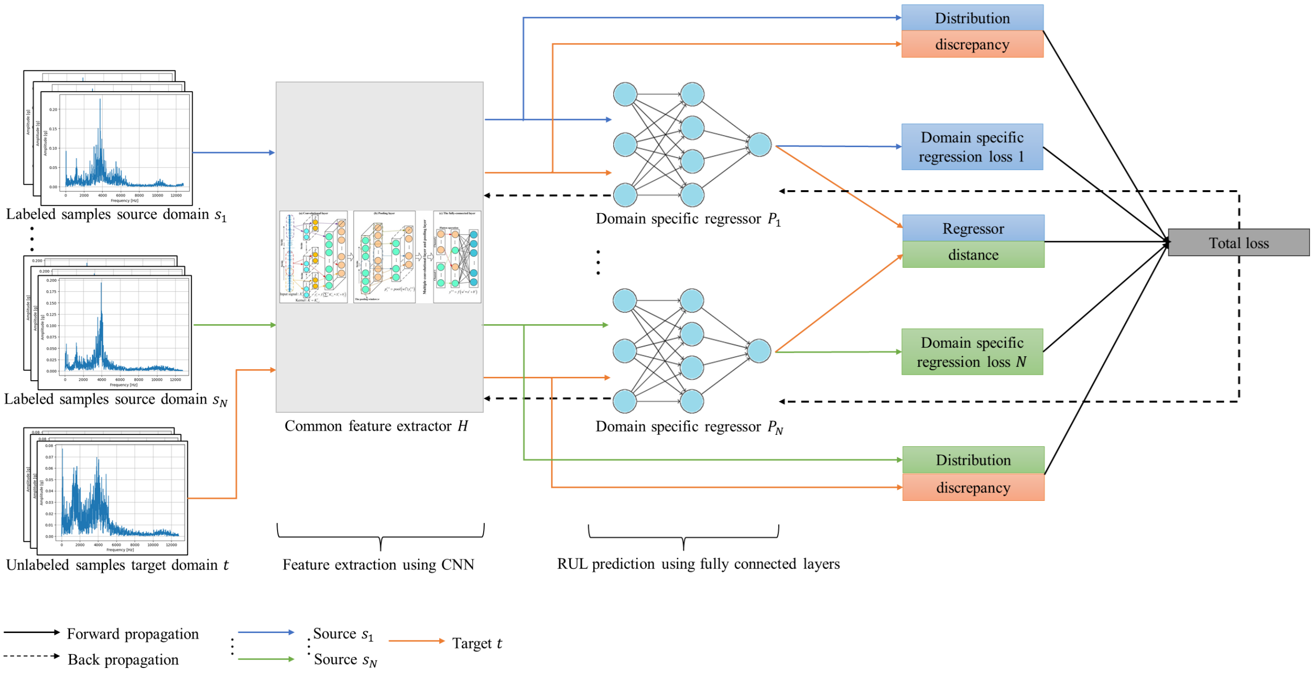

Labeled samples source domain s_1
Labeled samples source domain s_N
Unlabeled samples target domain t
Common feature extractor H
Domain specific regressor P_1
Domain specific regressor P_N
Distribution discrepancy
Domain specific regression loss 1
Regressor distance
Domain specific regression loss N
Distribution discrepancy
Total loss
Feature extraction using CNN
RUL prediction using fully connected layers
Forward propagation
Back propagation
Source s_1
Source s_N
Target t

Figure 1: The structure of the MSDA model.

layers, pooling layers, and fully connected layers. Due to these convolutional layers, a CNN is capable of automatic feature extraction [5]. Extracting features manually from vibration data requires expertise and can be time consuming. Therefore, this study makes use of a CNN for feature extraction. The one dimensional (1D) CNN is widely used for signal processing and has provided accurate results on bearing RUL prediction [7, 28, 47]. Since in a 1D CNN convolutions are only in one direction, training a 1D CNN requires fewer computational resources than a 2D CNN [17]. Therefore, the 1D CNN will be used in this RUL prediction model. A schematic overview of a CNN can be observed in Figure 2.

(1) Convolutional layer

The convolutional layer extracts features from the input data. A kernel slides along the input data and performs a dot product between the region of input data perceived by the kernel and the kernel weights [28]. This convolution operation can be expressed by Equation 4 [47]:

$$y_{l,j}^{\text{ReLU}} = f\left(\sum_{i=1}^{k} w_{i,j}^{l} * y_{l-1,i}^{\text{pool}} + b_j^l\right) \tag{4}$$

In Equation 4, $y_{l,j}^{\text{ReLU}}$ represents j^{th} channel output of convolutional layer l, after the activation function has been applied. $y_{l-1,i}^{\text{pool}}$ is the i^{th} channel output of the previous pooling layer $l-1$, and thus the input to layer l. $w_{i,j}^{l}$ denotes the kernel weights for input channel i and output channel j of layer l. b_j^l represents the bias of the j^{th} channel in layer l. k is the number of channels in layer $l-1$. $*$ denotes the convolution operation. $f(\cdot)$ is the activation function. In this case, a ReLU activation is used. The activation function determines whether a neuron is activated. The ReLU activation function is presented in Equation 5. The activation function accelerates convergence of the model, as the ReLU function reduces the vanishing gradient problem [33].

$$f(x) = max(0, x) \tag{5}$$

(2) Pooling layer

Usually, pooling layers are placed after a convolutional layer. The pooling layer reduces the dimension of the output of the previous layer. Due to this dimensionality reduction, computational effort is reduced [28]. Two types of pooling layers are maximum pooling and average pooling. In maximum pooling, the maximum value of a region of data inside the pooling window is extracted. On the other hand, average pooling extracts the average value. Since maximum pooling only extracts the maximum value of the data, some information may be lost [6]. Therefore, this approach makes use of average pooling. Average pooling is expressed in Equation 6 [47]:

$$y_{l,j}^{\text{pool}} = \text{mean}\left(w \cap y_{l-1,j}^{\text{ReLU}}\right) \tag{6}$$

In this equation, $y_{l-1,j}^{\text{ReLU}}$ represents the output of the previous convolutional layer $l-1$. w is the pooling window size and $\cap$ is the intersection between the pooling window and the input data to the pooling layer.

(3) Fully connected layer

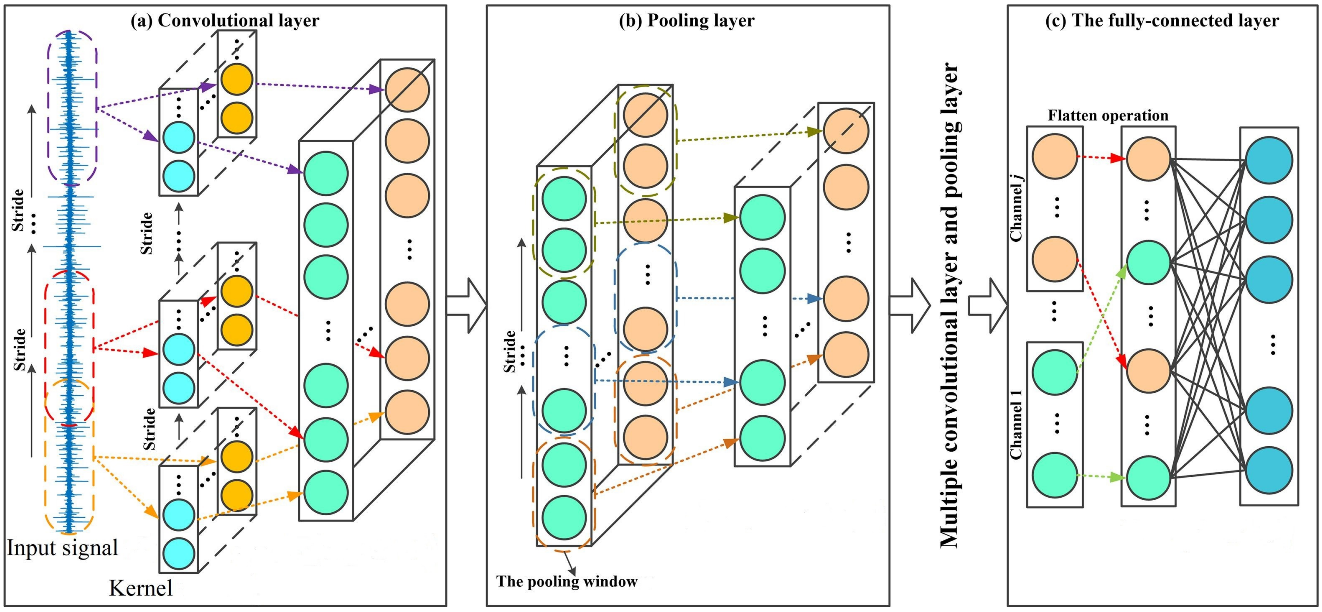


Figure 2: A schematic overview of a 1D CNN [6].

The output of the convolutional layers and pooling layers is flattened into a column vector. The elements of this flattening layer are fully interconnected to further fully connected layers. The interconnection is presented in Equation 7 [47]:

$$y^l = f\left(w^l * x^{l-1} + b^l\right) \tag{7}$$

Here, x^{l-1} and y^l are the input and output to fully connected layer l respectively. w^l and b^l are the weight matrix and the bias term of fully connected layer l. $f(\cdot)$ is the activation function, which is a ReLU activation function.

3.4. Domain adaptation

As the source domain and target domain data are recorded in different operating conditions, the distributions of source and target domain data may be different. If a RUL prediction model is trained using data from a certain operating condition and is used to generate RUL predictions for a different operating condition, the prediction results may be of poor accuracy [2]. In order to mitigate this problem, the distribution discrepancy between data samples from the source and target domain can be minimised. By minimising this discrepancy for each pair of source and target domains, the model generates domain invariant features that capture bearing degradation, enabling more accurate RUL predictions for the target domain.

Maximum mean discrepancy (MMD) is a measure for the distribution discrepancy between data samples from two different domains [35]. The MMD can be obtained by mapping data points from distributions p and q into a Reproducing Kernel Hilbert Space (RKHS) and calculating the distance between the means of these embeddings of p and q in RKHS [8]. The domain discrepancy (d_{MMD_f}) between a feature representation of a data sample from source domain s_d and target domain t, using a specific kernel f, is expressed in Equation 8 [13, 23, 32, 35].

$$\begin{aligned} d_{\mathrm{MMD}_f}\left(H^{s_d}, H^t\right) &= \left\| \frac{1}{n_{s_d}} \sum_{i=1}^{n_{s_d}} \phi\left(h_i^{s_d}\right) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi\left(h_j^t\right) \right\|_{\mathcal{H}_f}^2 \\ &= \frac{1}{n_{s_d}^2} \sum_{i=1}^{n_{s_d}} \sum_{j=1}^{n_{s_d}} k_f\left(h_i^{s_d}, h_j^{s_d}\right) + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k_f\left(h_i^t, h_j^t\right) \\ &\quad - \frac{2}{n_{s_d} n_t} \sum_{i=1}^{n_{s_d}} \sum_{j=1}^{n_t} k_f\left(h_i^{s_d}, h_j^t\right) \end{aligned} \tag{8}$$

In this equation, $H^{s_d} = \left\{h_i^{s_d}\right\}_{i=1}^{n_{s_d}} = \left\{H\left(x_i^{s_d}\right)\right\}_{i=1}^{n_{s_d}}$ and $H^t = \left\{h_j^t\right\}_{j=1}^{n_t} = \left\{H\left(x_j^t\right)\right\}_{j=1}^{n_t}$ respectively. n_{s_d} and n_t are the number of samples in the source and target domain (domains s_d and t) respectively. Function $\phi(\cdot)$ maps the data (h) into the RKHS, denoted by $\mathcal{H}$ [35]. Equation 8 essentially calculates the Euclidean Distance between two points, namely the mean embeddings of the distributions.

In Equation 8, the notation $k_f(\cdot,\cdot)$ can be observed. This is also known as the kernel trick and simplifies the calculation of $\phi(\cdot)$. The kernel trick is calculated as follows [6]:

$$k_f\left(x_i, x_j\right) = \left\langle \phi\left(x_i\right), \phi\left(x_j\right)\right\rangle = \phi\left(x_i\right)^T \phi\left(x_j\right) \tag{9}$$

By making use of the kernel trick, one can calculate the inner product $\left\langle \phi\left(x_i\right), \phi\left(x_j\right)\right\rangle$ without calculating the coordinates of the data in RKHS. Using the kernel trick, fewer computational resources are required.

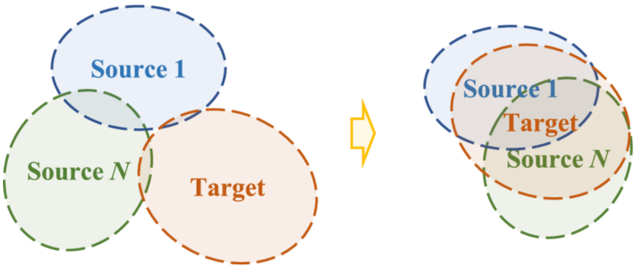


Figure 3: An example of minimising the distribution discrepancy between each pair of source and target domains [8].

As using a single kernel may be insufficient to fully capture the discrepancies between two data distributions, this approach makes use of multiple kernels to increase the accuracy of the MMD calculation [14]. Each specific kernel is denoted by f in $k_f(\cdot,\cdot)$. Five Gaussian kernels are commonly used for calculating MMD, showing accurate RUL prediction results [6, 9, 20, 32]. Therefore, this approach makes use of five Gaussian kernels with bandwidth parameter values (σ_f) of 1, 2, 4, 8, and 16 [32]. The Gaussian kernel allows to rewrite Equation 9 as follows:

$$k_f\left(x_i, x_j\right) = e^{-\frac{(x_i - x_j)^2}{2\sigma_f^2}} \tag{10}$$

In the proposed approach, the aim is to minimise the MMD between each pair of source and target domains. First, a sample from the source domain j and target domain passes through the common feature extractor H in order to obtain feature representations $H\left(x_i^{s_j}\right)$ and $H\left(x_i^t\right)$. Then, the distribution discrepancy between a batch of these two representations (H^{s_j} and H^t respectively) is calculated using Equation 8 [6]. Equation 11 shows the MMD loss between a sample from source domain j and target domain t.

$$L_{\text{MMD}_j} = \sum_{f=1}^{K} d_{\text{MMD}_f}\left(H^{s_j}, H^t\right) \tag{11}$$

In this equation, K represents the set of kernels used for computing the MMD and f is a specific kernel. The total MMD loss, L_{MMD}, is obtained by averaging L_{MMD_j}:

$$L_{\text{MMD}} = \frac{1}{N}\sum_{j=1}^{N} L_{\text{MMD}_j} \tag{12}$$

Using this loss term, each source domain data distribution is aligned with the target domain data distribution. This is also displayed in Figure 3.

3.5. *Model optimisation objective*

During model training, a sample $x_i^{s_j}$ from source domain s_j passes through the common feature extractor ($H(\cdot)$) and regressor ($P_j(\cdot)$) to obtain a domain specific RUL prediction for that sample, $\hat{y}_i^{s_j}$. It is desired that $H(\cdot)$ and $P_j(\cdot)$ learn an accurate relationship between bearing degradation and RUL label. As RUL labels in the source domain are available, the source domain prediction error can be minimised [6]. In this approach, the mean squared error (MSE) is used as the regression loss term [8]. The domain specific MSE is defined as follows:

$$L_{\text{reg}_j} = \frac{1}{n_{s_j}}\sum_{i=1}^{n_{s_j}}\left(y_i^{s_j} - \hat{y}_i^{s_j}\right)^2 \tag{13}$$

In this equation, n_{s_j} is the number of samples in source domain s_j. $y_i^{s_j}$ is the true RUL label assigned to sample $x_i^{s_j}$. L_{reg} is obtained from all N regressors by averaging L_{reg_j}:

$$L_{\text{reg}} = \frac{1}{N}\sum_{j=1}^{N} L_{\text{reg}_j} \tag{14}$$

Additionally, a sample from the target domain, x_i^t passes through the common feature extractor and N domain specific regressors to obtain N RUL predictions for that sample. These RUL predictions are denoted as $P_1(H(x_i^t)), \ldots, P_N(H(x_i^t))$. It is possible that these RUL predictions differ significantly from each other. However, as the same target domain sample is passed through the network, the N RUL predictions should also be aligned with each other. Therefore, another loss term (L_{dist}) is used in this approach in order to align these predictions. This loss term is presented in Equation 15 [8].

$$L_{\text{dist}} = \frac{1}{n_{\text{comb}} \cdot n_t}\sum_{j=1}^{N-1}\sum_{k=j+1}^{N}\sum_{i=1}^{n_t}\left(P_j\left(H\left(x_i^t\right)\right) - P_k\left(H\left(x_i^t\right)\right)\right)^2 \tag{15}$$

n_{comb} denotes the number of possible source domain combinations, which equals $N(N-1)/2$. n_t is the number of target domain samples.

The three loss components, represented by Equations 12, 14, and 15, are combined into an objective function for training the MSDA network. In the training procedure, the following loss function is minimised:

$$L_{\text{total}} = L_{\text{reg}} + \lambda L_{\text{MMD}} + \mu L_{\text{dist}} \tag{16}$$

In this equation, λ and μ are trade-off hyperparameters. An overview of the training procedure of the MSDA model is presented in algorithm 1.

3.6. *RUL prediction*

Once the MSDA model has been trained, the model can be used for generating predictions for a bearing test data set from the target domain operating condition. This sample can be denoted as $X^{\text{test}} = \left\{x_i^{\text{test}}\right\}_{i=1}^{n_{\text{test}}}$, where n_{test} is the number

Algorithm 1: The training procedure of the MSDA model, adapted from [8].

```
Input : $N$ labelled source domain data sets $\left\{\left(X^{s_j}, Y^{s_j}\right)\right\}_{j=1}^{N}$
        and an unlabelled target domain data set $X^t$, batch
        size $n_b$, maximum number of epochs $n_{epochs}$
Initialize parameters of source domain models $1, \ldots, N$
Determine number of batches $num_{batches} =$
  $\min\left(length\left(\left(X^{s_1}, Y^{s_1}\right)\right), \ldots, length\left(\left(X^{s_N}, Y^{s_N}\right)\right), length\left(X^t\right)\right)$
while $epoch \leq n_{epochs}$ do
    for $1, 2, \ldots, num_{batches}$ do
        Randomly get samples
          $\left\{x_i^{s_1}, y_i^{s_1}\right\}_{i=1}^{n_b}, \ldots, \left\{x_i^{s_N}, y_i^{s_N}\right\}_{i=1}^{n_b}$ from source
          domains $s_1, \ldots, s_N$ ;
        Randomly get $n_b$ samples from $\left\{x_i^t\right\}_{i=1}^{n_b}$ from target
          domain $t$ ;
        Obtain feature representations $H\left(x_i^{s_1}\right), \ldots, H\left(x_i^{s_N}\right)$
          using the common feature extractor $H(\cdot)$ ;
        Obtain feature representation $H\left(x_i^t\right)$ of the target
          domain sample using common feature extractor $H(\cdot)$ ;
        Obtain domain specific RUL prediction
          $P_1\left(H\left(x_i^{s_1}\right)\right), \ldots, P_N\left(H\left(x_i^{s_N}\right)\right)$ using domain
          specific regressor $P_j(\cdot)$ ;
        Obtain $N$ RUL predictions
          $P_1\left(H\left(x_i^t\right)\right), \ldots, P_N\left(H\left(x_i^t\right)\right)$ for target domain
          sample ;
        Calculate domain specific regression error $L_{\mathrm{reg}_j}$
          between $y_i^{s_j}$ and $P_j\left(H\left(x_i^{s_j}\right)\right)$ for each source
          domain model $j$ using Equation 13 and obtain $L_{\mathrm{reg}}$ by
          Equation 14 ;
        Calculate domain specific $L_{\mathrm{MMD}_j}$ between domain
          pairs $\left(H\left(x_i^{s_1}\right), H\left(x_i^t\right)\right), \ldots, \left(H\left(x_i^{s_N}\right), H\left(x_i^t\right)\right)$
          using Equation 11 and obtain $L_{\mathrm{MMD}}$ by Equation 12 ;
        Calculate distance between $N$ RUL predictions
          $P_1\left(H\left(x_i^t\right)\right), \ldots, P_N\left(H\left(x_i^t\right)\right)$ for target domain
          sample using Equation 15 ;
        Calculate the total loss $L_{\mathrm{total}}$ using Equation 16 ;
        Backpropagate $L_{\mathrm{total}}$ tot the model ;
        Update the parameters of the common feature extractor
          and domain specific regressors ;
    end
end
Output : The trained MSDA model
```

of samples in the test data set [8]. This testing process is displayed in Figure 4. In this figure, it can be observed that a sample from the target domain dataset passes through the common feature extractor and N domain-specific regressors. The final RUL prediction output of the MSDA model is the average of the predictions obtained by N regressors. The prediction of the RUL label is expressed in Equation 17 [8].

$$\hat{y}_i^{\text{test}} = \frac{1}{N} \sum_{j=1}^{N} P_j\left(H\left(x_i^{\text{test}}\right)\right) \tag{17}$$

To summarise, the MSDA model is able to generate RUL predictions for bearings under different operating conditions as follows:

1. Obtain N run-to-failure data sets from bearings under multiple operating conditions and of multiple bearing types.
2. Obtain bearing vibration data from the desired target condition and bearing type.
3. Preprocess the $N + 1$ data sets by converting the time domain signal to a frequency domain representation using the FFT.
4. Train the MSDA model using the procedure described in Figure 1 and algorithm 1.
5. Test the model with test data from the target domain using the procedure presented in Figure 4.

4. Experimental results and discussion

In order to validate the effectiveness of the proposed approach on RUL prediction across operating conditions and different machines, some experiments are performed. First, the bearing datasets used for these experiments will be described in section 4.1. Next, the transfer tasks and model structure will be given in sections 4.2 and 4.3 respectively. Finally, the RUL prediction results will be presented and discussed in sections 4.4 and 4.5.

4.1. *Description of the datasets*

The experiments are conducted using two bearing datasets, namely the IEEE PHM 2012 prognostic challenge dataset [27] and the XJTU-SY bearing dataset [36] (hereafter referred to as PHM and XJTU respectively). Bearings from different datasets are referred to as different bearing types.

The PHM dataset is provided by the FEMTO-ST Institute in France. This dataset contains run-to-failure data of 17 bearings under three different operating conditions. These laboratory experiments were carried out on the PRONOSTIA platform, visualised in Figure 5. Vibration measurements were recorded in horizontal and vertical directions using accelerometers. The sampling frequency was 25.6 kHz, the sampling interval was 10 s, and the sampling time was 0.1 s. Information on the operating conditions and bearing datasets can be observed in Table 1.

The XJTU dataset is provided by Xi'an Jiaotong University and Changxing Sumyoung Technology. This dataset contains 15 run-to-failure experiments under three different operating conditions. The testbed is shown in Figure 6. Also here, accelerometers were installed in horizontal and vertical directions. For the XJTU dataset, the sampling frequency was also 25.6 kHz, but the sampling interval was 1 min and the sampling time was 1.28 s. Information on operating conditions of this dataset is listed in Table 2.

In both datasets, vibration measurements were recorded in horizontal and vertical direction. However, previous research has found that the vertical direction does not show bearing degradation clearly [30]. Therefore, this study only uses the horizontal direction for RUL prediction.

4.2. *Transfer tasks*

Eight transfer tasks have been set up for the MSDA model. These tasks include three scenarios: RUL prediction across operating conditions using the same bearing type (i.e. PHM or XJTU only), RUL prediction across operating conditions

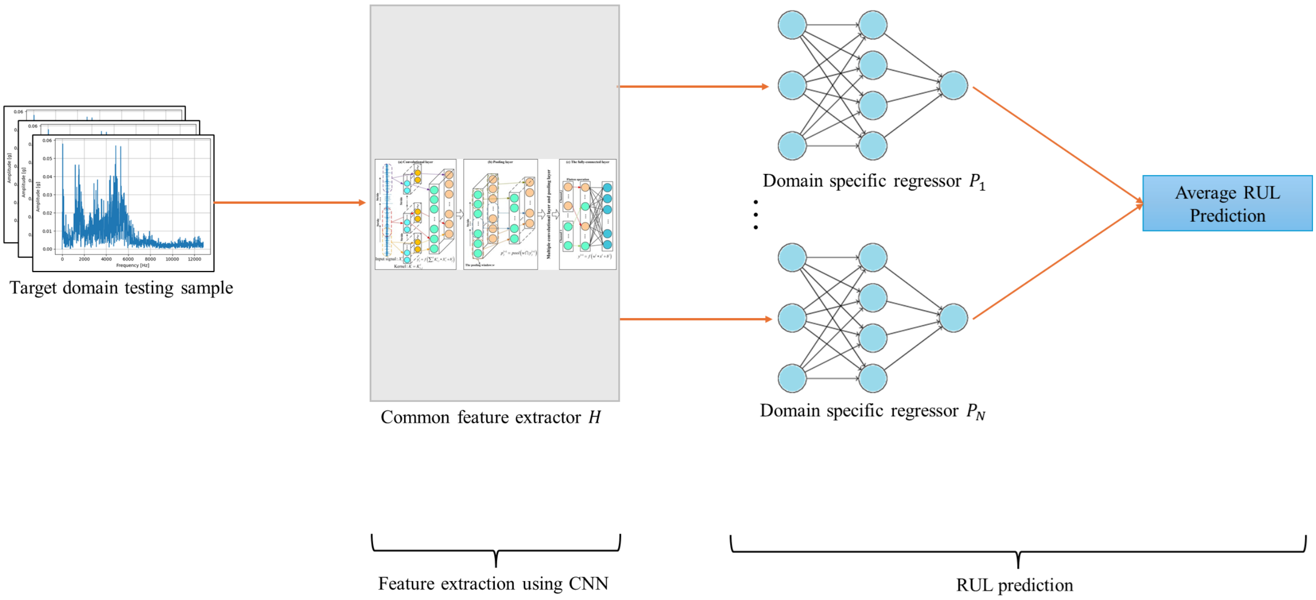

Target domain testing sample
Common feature extractor H
Domain specific regressor P_1
Domain specific regressor P_N
Average RUL Prediction
Feature extraction using CNN
RUL prediction

Figure 4: The testing procedure for the MSDA model.

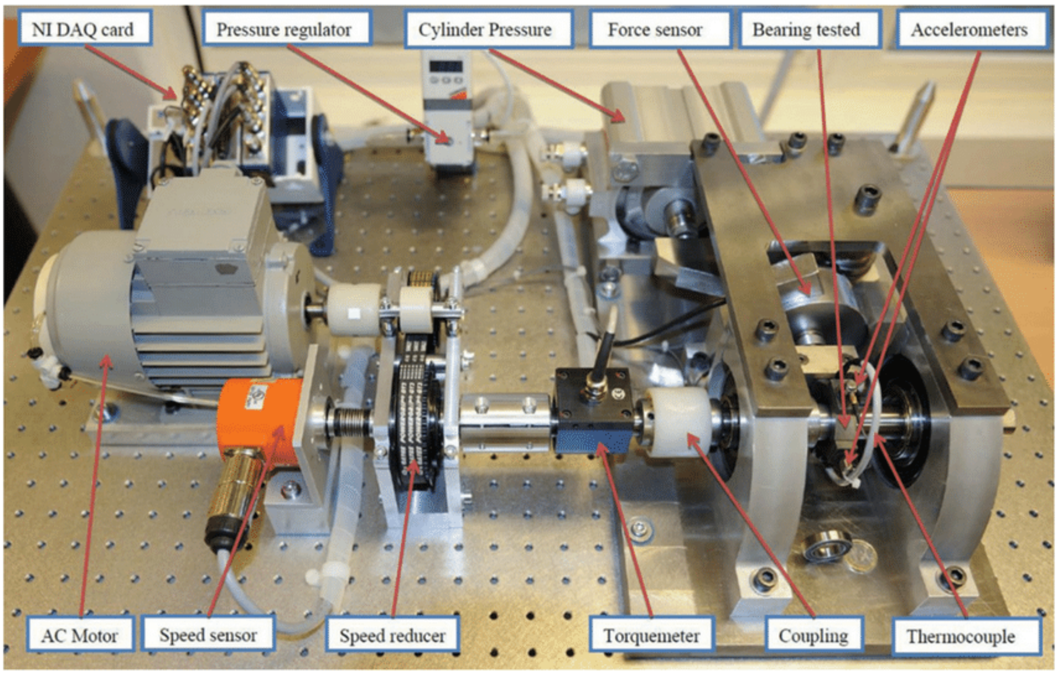


Figure 5: The PRONOSTIA platform [27].

where the source domain contains data from two different bearing types (i.e. PHM and XJTU mixed), and RUL prediction across operating conditions and bearing types. In the latter scenario, the source domain data contains bearing data of only one bearing type and the target domain contains data from a different bearing type. All tasks are listed in Table 3.

The first scenario assumes that there is sufficient labelled operating data for a bearing in given operating conditions, but a RUL prediction is required for this bearing type in a different operating condition without labelled operating data. In the second scenario, there is not a sufficient amount of labelled operating data for a certain bearing type. To address this, labelled operating data from a different bearing type is incorporated in the dataset. Using this enlarged dataset, RUL prediction can be performed for the target bearing type in a given operating condition. The third scenario assumes that a RUL indication is required for a certain bearing type in a given operating condition. However, no run-to-failure data has yet been collected for this bearing. In this case, the RUL

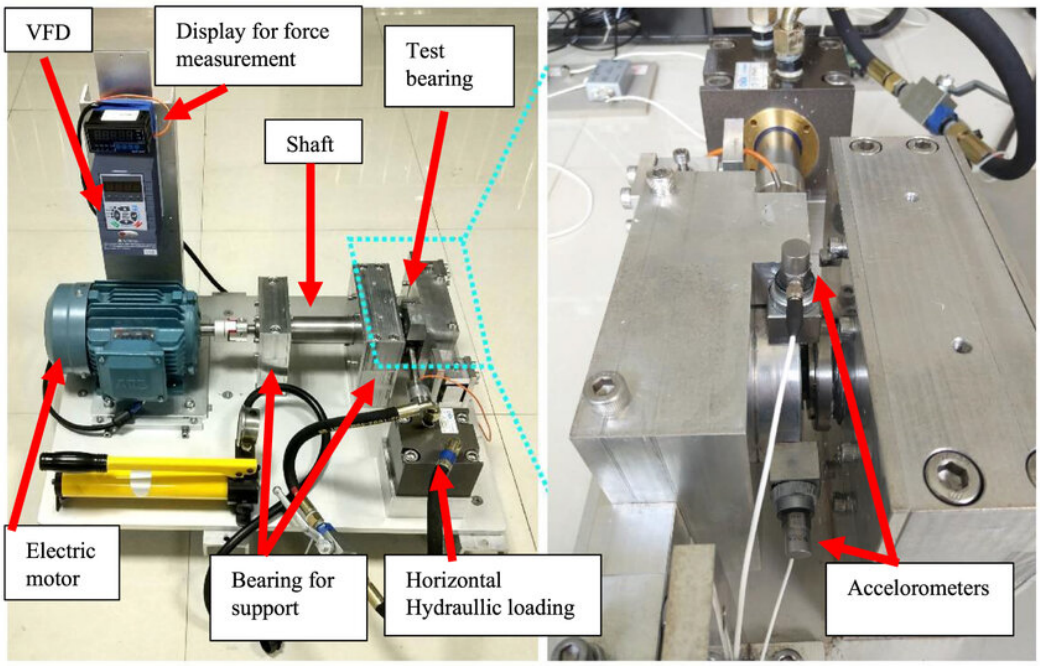


Figure 6: The platform of the XJTU bearing dataset [36].

Table 1
Operating conditions of the PHM dataset.

Working condition	Bearing dataset	Bearing lifetime
Load 4.0 kN, Speed 1,800 rpm	PHM1_1	7 h 47 m 0 s
	PHM1_2	2 h 25 m 0 s
	PHM1_3	5 h 0 m 10 s
	PHM1_4	3 h 9 m 40 s
	PHM1_5	6 h 23 m 30 s
	PHM1_6	6 h 23 m 29 s
	PHM1_7	4 h 10 m 11 s
Load 4.2 kN, Speed 1,650 rpm	PHM2_1	2 h 31 m 40 s
	PHM2_2	2 h 12 m 40 s
	PHM2_3	3 h 20 m 10 s
	PHM2_4	1 h 41 m 50 s
	PHM2_5	5 h 33 m 30 s
	PHM2_6	1 h 35 m 10 s
	PHM2_7	0 h 28 m 30 s
Load 5.0 kN, Speed 1,500 rpm	PHM3_1	1 h 25 m 40 s
	PHM3_2	4 h 32 m 40 s
	PHM3_3	0 h 58 m 30 s

Table 2
Operating conditions of the XJTU dataset.

Working condition	Bearing dataset	Bearing lifetime
Load 12 kN, Speed 2,100 rpm	XJTU1_1	2 h 3 m
	XJTU1_2	2 h 41 m
	XJTU1_3	2 h 38 m
	XJTU1_4	2 h 2 m
	XJTU1_5	0 h 52 m
Load 11 kN, Speed 2,250 rpm	XJTU2_1	8 h 11 m
	XJTU2_2	2 h 41 m
	XJTU2_3	8 h 53 m
	XJTU2_4	0 h 42 m
	XJTU2_5	5 h 39 m
Load 10 kN, Speed 2,400 rpm	XJTU3_1	42 h 18 m
	XJTU3_2	41 h 36 m
	XJTU3_3	6 h 11 m
	XJTU3_4	25 h 15 m
	XJTU3_5	1 h 54 m

prediction is based on labelled operating data of a different bearing type.

In order to evaluate the RUL prediction results, several metrics will be used. These are the mean absolute error (MAE) and root mean squared error (RMSE). The MAE and RMSE are given in Equations 18 and 19 respectively.

$$\mathrm{MAE} = \frac{1}{O}\sum_{i=1}^{O}|y_i - \hat{y}_i| \tag{18}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{O}\sum_{i=1}^{O}\left(y_i - \hat{y}_i\right)^2} \tag{19}$$

Here, O is the number of samples for the test bearing. y_i and $\hat{y}_i$ are the actual RUL and predicted RUL respectively. Additionally, the Score metric is used. In this metric, early predictions of the RUL (where the RUL prediction is lower than the actual RUL) contribute more to the score than late RUL predictions (where the RUL prediction is higher than the actual RUL). A higher score metric indicates better model performance [27]. The score metric is calculated using Equations 20 - 22.

Table 3
The transfer tasks for multi-source domain adaptation.

Scenario	Task	Training bearings		Test bearing
		Labelled source domain	Unlabelled target domain	
1	1	S1: PHM1_1-PHM1_7 S2: PHM3_1-PHM3_3	PHM2_3, PHM2_5	PHM2_6
	2	S1: PHM1_1-PHM1_7 S2: PHM2_1-PHM2_7	PHM3_1, PHM3_2	PHM3_3
2	3	S1: PHM1_1-PHM1_7 S2: XJTU1_1, XJTU1_2, XJTU1_3, XJTU1_5	PHM2_3, PHM2_5	PHM2_6
	4	S1: PHM1_1-PHM1_7 S2: XJTU2_1, XJTU2_2, XJTU2_3, XJTU2_5	PHM2_1	PHM2_2
	5	S1: PHM1_1-PHM1_7 S2: XJTU2_1, XJTU2_2, XJTU2_3, XJTU2_5	XJTU1_1	XJTU1_3
3	6	S1: PHM1_1-PHM1_7 S2: PHM2_1-PHM2_7	XJTU1_1	XJTU1_3
	7	S1: PHM1_1-PHM1_7 S2: PHM2_1-PHM2_7	XJTU2_1	XJTU2_2
	8	S1: PHM1_1-PHM1_7 S2: PHM2_1-PHM2_7	XJTU2_1, XJTU2_5	XJTU2_2

Table 4
Structure of the MSDA model.

Module	Layer name	Parameters	Activation function	Output size
Common feature extractor	Convolutional1	Kernel number: 8 Kernel size: 4 Stride: 2	ReLU	640×8
	Pooling1	Kernel size: 2 Stride: 2	-	320×8
	Convolutional2	Kernel number: 16 Kernel size: 4 Stride: 2	ReLU	160×16
	Pooling2	Kernel size: 2 Stride: 2	-	80×16
	Convolutional3	Kernel number: 32 Kernel size: 4 Stride: 2	ReLU	40×32
	Pooling3	Kernel size: 2 Stride: 2	-	20×32
	Flattening	-	-	640
	FC1	(640, 160)	ReLU	160
	Dropout1	0.4	-	160
	FC2	(160, 40)	ReLU	40
	Dropout2	0.4	-	40
	FC3	(40, 10)	ReLU	10
	Dropout3	0.4	-	10
Regressor	FC4	(10, 1)	Sigmoid	1

$$\mathrm{ER}_i = \frac{y_i - \hat{y}_i}{y_i} \times 100 \tag{20}$$

$$A_i = \begin{cases} e^{-\ln(0.5)\cdot(\mathrm{ER}_i/5)} & \text{if } \mathrm{ER}_i \leq 0 \\ e^{+\ln(0.5)\cdot(\mathrm{ER}_i/20)} & \text{if } \mathrm{ER}_i > 0 \end{cases} \tag{21}$$

$$\mathrm{Score} = \frac{1}{O}\sum_{i=1}^{O} A_i \tag{22}$$

4.3. Model structure

As described in section 3.1, an acquisition is first converted to a frequency spectrum by FFT and standardised. This frequency spectrum is then used as input to the MSDA model. The PHM data contains 2,560 data points per acquisition. After FFT, a one sided frequency spectrum containing 1,280 data points remains. Therefore, the input shape of the data is (1280, 1), where 1 denotes that the data is associated to the horizontal vibration signal. Each acquisition in the XJTU dataset contains 32,768 data points. Therefore, the first 2,560 data points of each acquisition are selected for the generation of a frequency spectrum in order to align the shape of the data with the input shape required by the MSDA model [48].

The model structure is shown in Table 4. This model structure is based on the single-source domain adaptation model developed by Cheng et al. [6]. In MSDA model, the single-source model has been adapted for usage in a multi-source domain adaption scenario according to the method described in sections 3.4-3.6. Additionally, since the RUL labels are scaled between 0 and 1, the sigmoid activation function is used for the final layer [30]. This function scales its input to values between 0 and 1.

In the MSDA model, the learning rate is 0.0005, the batch size is 32, the Adam optimizer is used, and the model is trained for 3,000 epochs. Additionally, the trade-off parameters in Equation 4, λ and μ, are set to 0.5 and 0.1 respectively.

4.4. RUL prediction results

After collecting the vibration measurements in the frequency domain, the MSDA model can be trained according to the transfer tasks reported in Table 3.

The RUL prediction results related to these tasks can be observed in Figure 7. The MAE, RMSE, and Score results are presented in Table 5. This table shows the average error values after 10 repetitions of each task. Based on these 10 repetitions, 95% confidence intervals are calculated for the metrics. Additionally, RUL predictions from deep learning models can show undesirable fluctuations. Therefore, the RUL prediction results are smoothed with a weighted moving average with window size 10 [30].

The model may not be able to fully capture the degradation trend of a bearing. Hence, only providing point predictions of the RUL is unreliable [39]. Therefore, the model uncertainty is represented by the the 95% confidence interval. The confidence interval is obtained by making use of the Monte Carlo Dropout technique. During inference, the dropout layers of the model remain activated. If the same data sample is fed into the model, different RUL predictions can be obtained for that sample [39]. From these different Monte Carlo predictions, the confidence interval is calculated.

The RUL predictions presented in Figure 7 show a decreasing trend. This indicates that the MSDA model is learning the degradation trend of a bearing, regardless of the operating condition or bearing type. In the initial phase of the bearing, the RUL predictions remain approximately horizontal. This indicates that the bearing is in healthy condition. As the bearing has not shown significant degradation yet, the

Table 5
The evaluation metrics, including their 95% confidence intervals, for the RUL predictions.

Task	MAE	RMSE	Score
1	0.0625 ± 0.0052	0.0815 ± 0.0072	0.5303 ± 0.0180
2	0.1600 ± 0.0119	0.1871 ± 0.0154	0.3336 ± 0.0185
3	0.1334 ± 0.0117	0.1599 ± 0.0150	0.3827 ± 0.0263
4	0.0941 ± 0.0089	0.1165 ± 0.0106	0.4365 ± 0.0191
5	0.1137 ± 0.0091	0.1389 ± 0.0082	0.3285 ± 0.0374
6	0.1559 ± 0.0202	0.1935 ± 0.0229	0.3666 ± 0.0425
7	0.1472 ± 0.0103	0.1738 ± 0.0078	0.3555 ± 0.0244
8	0.1392 ± 0.0206	0.1765 ± 0.0212	0.3863 ± 0.0512

predictions are relatively horizontal [39]. Once degradation progresses, the RUL predictions are also decreasing. However, near the failure point of the bearing, the RUL predictions tend to deviate from the actual RUL. This indicates that the model may be unable to learn the fast degradation process of the bearing in the final stage.

When the RUL of test bearing PHM2_6 is predicted using only PHM data (task 1) and using less PHM data, mixed with XJTU data (task 3), the MAE and RMSE increase, while the Score decreases. This indicates that prediction accuracy improves when the source and target domain data come from the same bearing. However, if there is an insufficient amount of labelled source domain data available for a specific bearing type, then a RUL prediction can still be achieved, as shown by the results of task 3. Additionally, the results of tasks 5 and 6, where the RUL of bearing XJTU1_3 is predicted, suggest that if the source domain contains some data from the same bearing type as the target domain, prediction accuracy can be improved. However, if the source domain contains data from one bearing type, and the target domain from another type, a RUL prediction is still possible, particularly when the target domain training dataset includes more data. For example, when the RUL of bearing XJTU2_2 is predicted using source domain data from PHM only, where the target domain contains data from one bearing (task 7) and data from two bearings (task 8), the MAE and Score metrics improve when more data is added to the target domain.

4.5. Discussion

4.5.1. Comparison with other algorithms

In order to get an insight into the performance of the MSDA model, some comparative experiments have been performed for transfer tasks 1, 3, and 7. These experiments included a deep learning method, single-source domain adaptation methods, and multi-source domain adaptation methods. For the deep learning method, the CNN presented in Table 4 without transfer learning is used. For the single-source domain adaptation methods, the following methods have been used: the transferable CNN [6], the transfer learning-based bidirectional LSTM model (TBiLSTM) [32], and the multi-channel transferable bidirectional LSTM model with attention mechanism (TMLAN) [10]. For the multi-source domain adaptation models, the MDAN model [8] and the multi-source adversarial knowledge distillation (AdvKD) model [22] have been used.

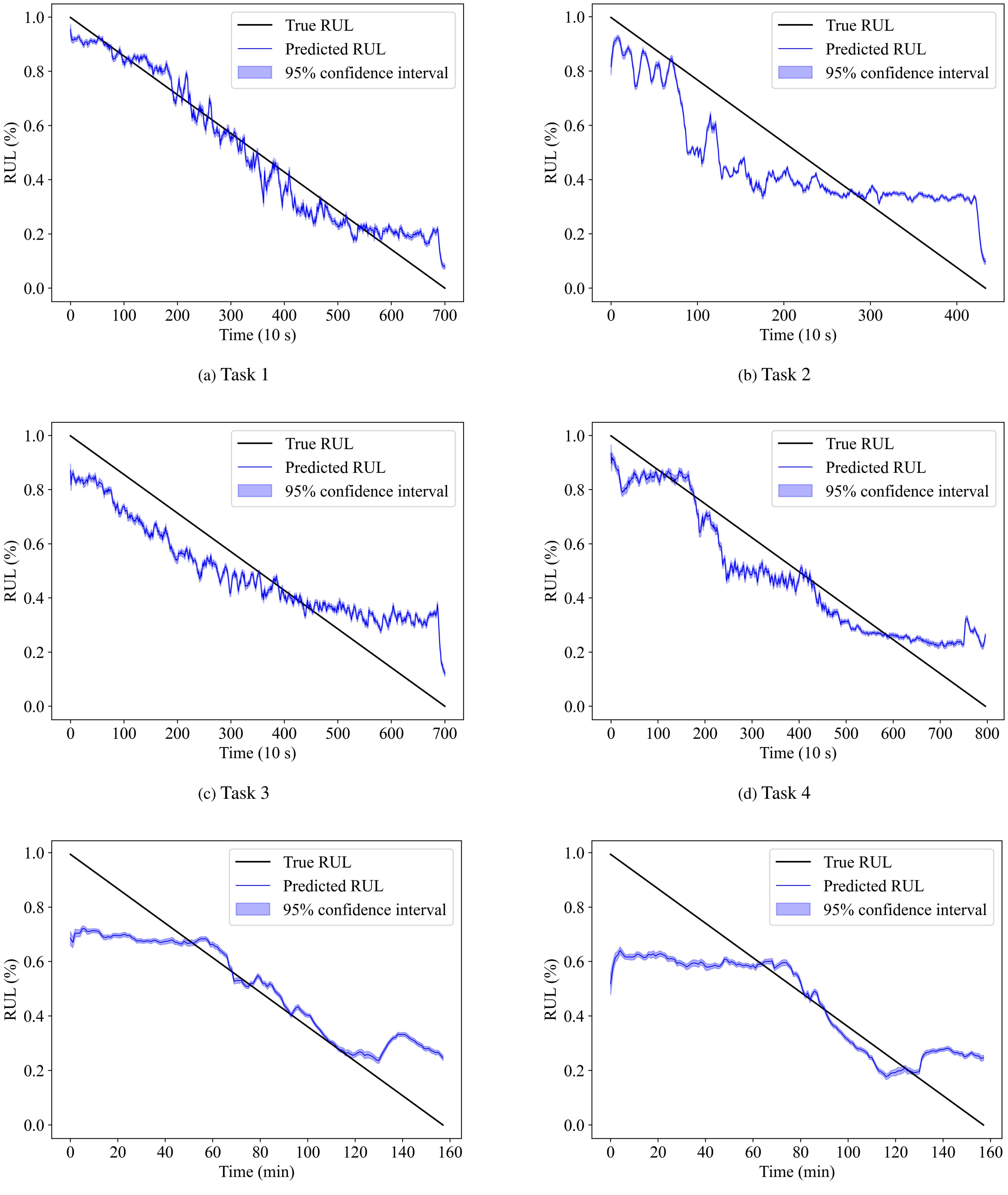
The deep learning model was trained on a single operating condition and then applied to the target operating condition. The operating condition yielding the best evaluation metrics was used for this analysis. For the single-source domain adaptation methods, data from a single operating condition was used as the source domain training data. The results of these comparative experiments are reported in Table 6. Note that the evaluation metrics for TMLAN, MDAN, and AdvKD have been obtained from [31]. In this study, the authors conducted experiments using a similar setup to scenarios 2 and 3, and the same testing datasets were used as in transfer tasks 3 and 7 listed in Table 3.

From Table 6, it can be observed that the transfer learning based models outperform the deep learning based model, highlighting the relevance of domain adaptation. Additionally, the multi-source domain adaptation models provide better results than the single-source domain adaptation models. This may be because a multi-source model has access to more bearing degradation information, as the source domain may include data from different operating conditions and bearing types. Additionally, it can be observed that the proposed MSDA model achieves evaluation scores comparable to other advanced multi-source domain adaptation methods such as MDAN and AdvKD. These MDAN and AdvKD models use data from the time, frequency, and time-frequency domains, allowing them to make use of a large amount of degradation information. However, in practice, only frequency domain data may be available for some systems. Therefore, the MSDA model only uses frequency domain data and it does not have access to degradation information available in the time and time-frequency domains. The MSDA model still achieves results comparable to the MDAN and AdvKD models, which is promising.

4.5.2. Adding additional source domain data

Although the RUL predictions in Figure 7b show a decreasing trend in the middle stage of bearing degradation, they exhibit large deviations from the actual RUL. Therefore, the frequency domain data of the two source domains (operating conditions PHM1 and PHM2) might not contain a sufficient amount of information on bearing degradation.

Adding additional source domain data could increase the amount of information on bearing degradation. Since all source domain data from PHM has already been used, data from the first operating condition of the XJTU dataset has been added to the source domain. Now, the source domain contains the first and second operating conditions of PHM and the first operating condition of XJTU. Figure 8 shows the RUL prediction results using the extra source domain data. From this figure, it can be observed that the RUL predictions in the initial and middle stages of bearing degradation follow the actual RUL more closely compared to Figure 7b. Additionally, the evaluation metrics MAE,



(a) Task 1

(b) Task 2

(c) Task 3

(d) Task 4

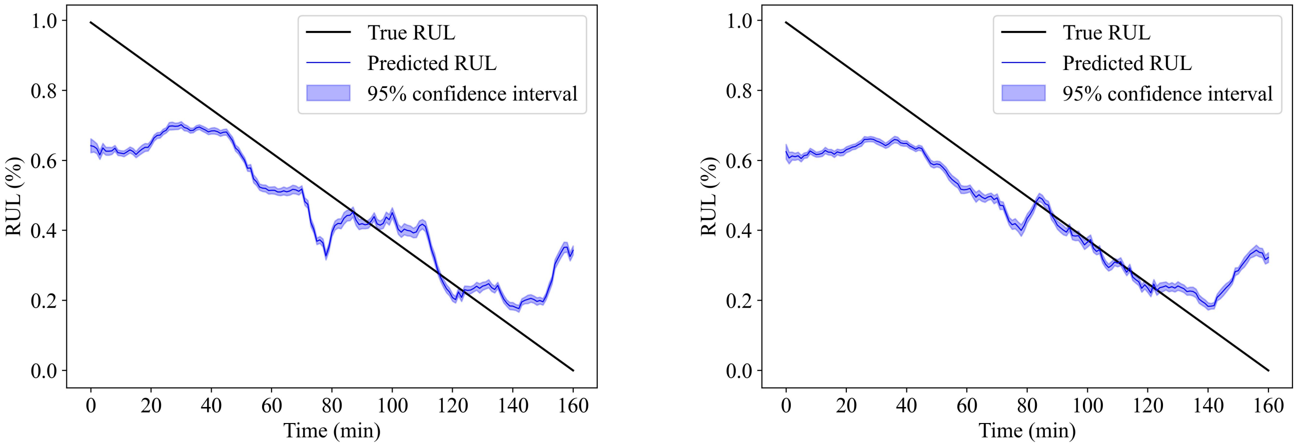
(e) Task 5

(f) Task 6

Figure 7: The RUL prediction results for all transfer tasks.

RMSE, and Score show improvements of 36.7%, 34.4%, and 33.1% respectively.

This further shows that adding additional source domain data helps to improve RUL predictions by increasing the amount of information on bearing degradation, even if this data comes from a different bearing type or experiment.



(g) Task 7

(h) Task 8

Figure 7: The RUL prediction results for all transfer tasks continued.

Table 6
Comparison between the MSDA model and other algorithms, the best results are denoted in bold.

Learning strategy	Method	Task 1		Task 3		Task 7	
		RMSE	Score	RMSE	Score	RMSE	Score
Deep learning	CNN	0.2367	0.1520	0.3647	0.1460	0.2936	0.2553
single-source domain adaptation	TCNN [6]	0.0930	0.3977	0.2969	0.2425	0.2128	0.3506
	TBiLSTM [32]	0.0898	-	-	-	-	-
	TMLAN [10]	-	-	0.2069	0.2188	0.1949	0.2531
Multi-source domain adaptation	MDAN [8]	-	-	0.1664	0.2544	0.1782	0.2650
	AdvKD [22]	-	-	**0.1589**	0.2764	**0.1635**	0.2700
	Proposed MSDA	**0.0815**	**0.5303**	0.1599	**0.3827**	0.1738	**0.3555**

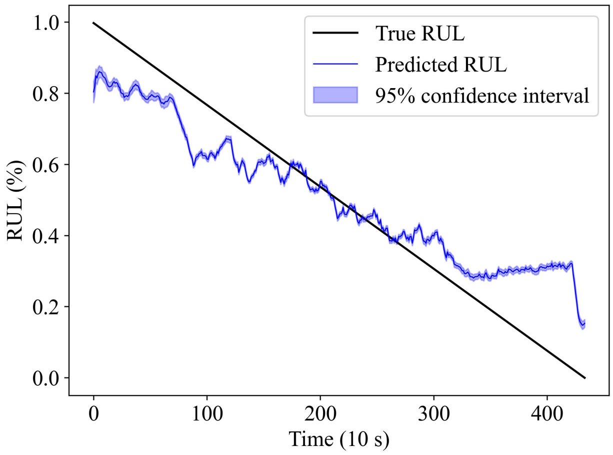


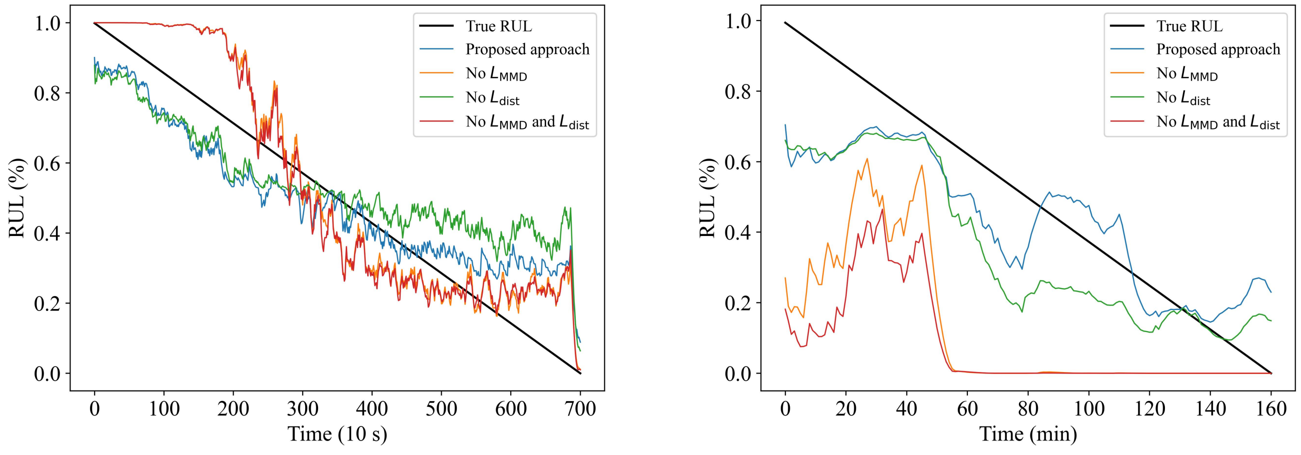
Figure 8: The RUL prediction results for task 2 if an extra source domain is added.

4.5.3. Ablation experiments

The loss function of this model, Equation 16, consists of several components. These are the regression loss, L_{reg}, the distribution discrepancy loss, L_{MMD}, and the output distance loss, L_{dist}. L_{reg} ensures that the model learns a relationship between bearing degradation and the RUL. L_{MMD} helps the MSDA model to learn domain invariant features by minimising the distribution discrepancy between a sample from the source and target domain. The third loss term, L_{dist}, aligns the RUL predictions of the domain specific regressors.

Each of these loss components has some influence on the RUL predictions outputted by the MSDA model. In order to visualise these influences, some ablation experiments have been performed. These experiments are denoted as A (no L_{MMD}) , B (no L_{dist}), and C (no L_{MMD} and L_{dist}). For these experiments, transfer tasks 3 and 7 from Table 3 are used. The RUL predictions related to these experiments are shown in Figures 9a and 9b. Additionally, the evaluation metrics related to these experiments are listed in Table 7.

From Figures 9a and 9b, it can be observed that without L_{MMD} and L_{dist}, the RUL prediction results are less accurate. This is also confirmed by the metrics in Table 7. For example, in case of experiments A and C for task 3, the RUL predictions show a downward trend. However, the predicted RUL values deviate more from the actual RUL than in the proposed approach. In experiment B of task 3, the initial RUL predictions seem to align with the predictions of



(a) Ablation experiments for task 3

(b) Ablation experiments for task 7

Figure 9: The RUL prediction results for the ablation experiments.

Table 7
The RUL prediction results for the ablation experiments, the best results are denoted in bold.

	Task 3				Task 7			
	A	B	C	Proposed approach	A	B	C	Proposed approach
MAE	0.11	0.13	0.11	**0.10**	0.37	0.16	0.42	**0.13**
RMSE	0.13	0.16	0.13	**0.12**	0.42	0.19	0.47	**0.16**
Score	0.30	0.38	0.30	**0.45**	0.08	0.36	0.06	**0.35**

the proposed approach. However, near the end of bearing lifetime, the RUL predictions of the proposed approach are closer to the actual RUL than the predictions of ablation experiment B. Similar findings have also been reported in [31], where the omission of L_{dist} also led to worse prediction performance near the bearing failure point. Additionally, in case of experiments A and C for task 7, the model is not able to generate any meaningful RUL predictions, which shows the relevance of loss term L_{MMD}. Additionally, from Table 7 it can be observed that the proposed approach shows superior performance compared to experiments A, B, and C.

Therefore, L_{MMD} and L_{dist} enable the model to learn domain invariant features which enhance the prediction accuracy of the MSDA model.

4.5.4. Distribution discrepancy visualisation

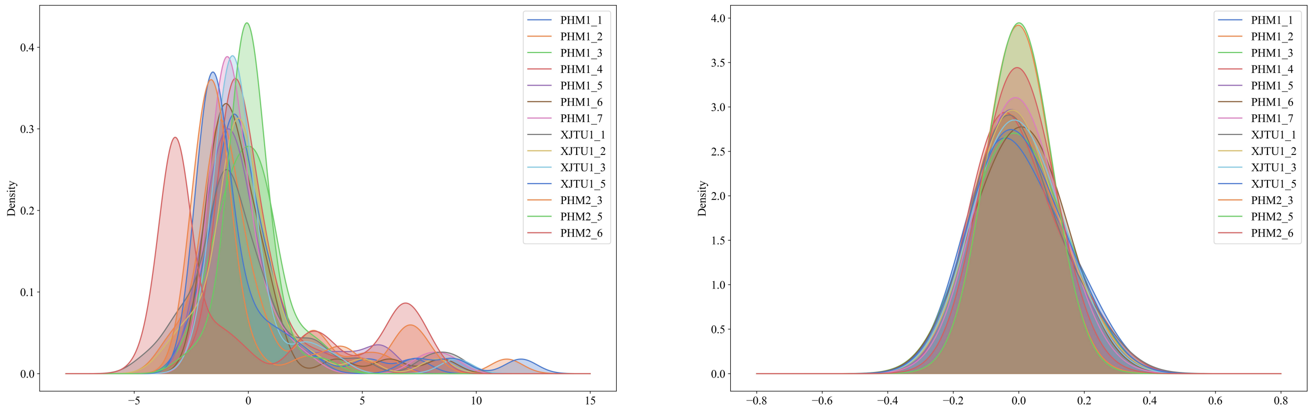
In the MSDA model, distribution discrepancy is minimised by minimising the MMD loss. This allows the model to learn domain invariant features. The effectiveness of the model in aligning the distributions can be visualised by plotting the probability density function (PDF) of the features extracted by the model. First, the model is trained using the proposed approach with and without the L_{MMD} loss term. Next, a feature vector containing 10 features can be obtained from both models by feeding a data sample to the model. Then, a Principal Component Analysis (PCA) is used to reduce the ten dimensional vector to a one dimensional vector. By Gaussian kernel density estimation, a PDF can be obtained for the features related to the data sample. For a more detailed explanation, see [31].

Task 3 from Table 3 has been selected for visualising the distribution discrepancy. The results can be observed in Figure 10. In this figure it can be observed that without using L_{MMD}, the distributions do not overlap. This implies that the MSDA model has failed to learn the generation of domain invariant features. If the L_{MMD} term is used, then the distributions from source and target domains are more aligned with each other. Hence, when using L_{MMD} with 5 kernels, the model is able to reduce the distribution discrepancy between the source and target domains.

5. Conclusion

In order to generate accurate RUL predictions across different operating conditions and bearing types based exclusively on frequency domain data, this paper proposes the transfer learning based multi-source domain adaptation (MSDA) model. This model uses a common feature extractor to extract features from a frequency spectrum. These features are fed into source domain specific regressors in order to provide domain specific RUL predictions. To generate domain invariant features, this model minimises the maximum mean discrepancy between the feature distributions from source and target domains. The outputs of the domain specific regressors are aligned with each other by minimising the distance between the outputs of the regressors.

The effectiveness of the MSDA model was verified by conducting experiments across three scenarios. From these experiments, it can be concluded that the model provides accurate RUL predictions for these scenarios based on domain invariant features obtained from frequency domain data. However, near the bearing failure point, the RUL predictions obtained from the MSDA model tend to deviate significantly from the actual RUL. To address this problem, the model may require additional examples from the fast degradation stage



(a) Feature distribution without using L_{MMD}

(b) Feature distribution using L_{MMD}

Figure 10: Visualisation of the alignment of the feature distributions for task 3.

of the bearings, which could potentially be generated using generative adversarial networks if additional labelled source domain data is unavailable.

Another opportunity is to include more operating conditions in the target domain. Using such an approach, a model can be developed that can be generalised across a larger number of operating conditions. Additionally, in practical scenarios, a bearing may be subjected to varying operating conditions throughout its lifetime, such as different flight regimes in a helicopter. However, such scenarios have not yet been evaluated using the MSDA model. Extending the MSDA model to predict the bearing RUL under diverse operating conditions throughout its lifetime will enhance the practical applicability of the model.

References

[1] AIBN, 2016. Accident at Turoy, near Bergen, Norway on 29 April 2016, involving Airbus helicopters H225, LN-OJF. Technical Report. Accident Investigation Board Norway.

[2] Azari, M.S., Flammini, F., Santini, S., Caporuscio, M., 2023. A Systematic Literature Review on Transfer Learning for Predictive Maintenance in Industry 4.0. IEEE Access 11, 12887–12910. doi:10.1109/ACCESS.2023.3239784.

[3] Cao, L., Zhang, H., Meng, Z., Wang, X., 2023. A parallel GRU with dual-stage attention mechanism model integrating uncertainty quantification for probabilistic RUL prediction of wind turbine bearings. Reliability Engineering & System Safety 235, 109197. doi:https://doi.org/10.1016/j.ress.2023.109197.

[4] Chen, J., Huang, R., Chen, Z., Mao, W., Li, W., 2023. Transfer learning algorithms for bearing remaining useful life prediction: A comprehensive review from an industrial application perspective. Mechanical Systems and Signal Processing 193. doi:10.1016/j.ymssp.2023.110239.

[5] Cheng, C., Ma, G., Zhang, Y., Sun, M., Teng, F., Ding, H., Yuan, Y., 2020. A Deep Learning-Based Remaining Useful Life Prediction Approach for Bearings. IEEE/ASME Transactions on Mechatronics 25, 1243–1254. doi:10.1109/TMECH.2020.2971503.

[6] Cheng, H., Kong, X., Chen, G., Wang, Q., Wang, R., 2021. Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors. Measurement 168, 108286. doi:https://doi.org/10.1016/j.measurement.2020.108286.

[7] Ding, H., Yang, L., Cheng, Z., Yang, Z., 2021a. A remaining useful life prediction method for bearing based on deep neural networks. Measurement: Journal of the International Measurement Confederation 172. doi:10.1016/j.measurement.2020.108878.

[8] Ding, Y., Ding, P., Zhao, X., Cao, Y., Jia, M., 2022. Transfer Learning for Remaining Useful Life Prediction Across Operating Conditions Based on Multisource Domain Adaptation. IEEE/ASME Transactions on Mechatronics 27, 4143–4152. doi:10.1109/TMECH.2022.3147534.

[9] Ding, Y., Jia, M., Miao, Q., Huang, P., 2021b. Remaining useful life estimation using deep metric transfer learning for kernel regression. Reliability Engineering & System Safety 212, 107583. doi:https://doi.org/10.1016/j.ress.2021.107583.

[10] Dong, S., Xiao, J., Hu, X., Fang, N., Liu, L., Yao, J., 2023. Deep transfer learning based on Bi-LSTM and attention for remaining useful life prediction of rolling bearing. Reliability Engineering & System Safety 230, 108914. doi:https://doi.org/10.1016/j.ress.2022.108914.

[11] El-Thalji, I., Jantunen, E., 2015. A summary of fault modelling and predictive health monitoring of rolling element bearings. Mechanical Systems and Signal Processing 60-61, 252–272. doi:https://doi.org/10.1016/j.ymssp.2015.02.008.

[12] Ferreira, C., Gonçalves, G., 2022. Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods. Journal of Manufacturing Systems 63, 550–562. doi:https://doi.org/10.1016/j.jmsy.2022.05.010.

[13] Geng, B., Tao, D., Xu, C., 2011. DAML: Domain Adaptation Metric Learning. IEEE Transactions on Image Processing 20, 2980–2989. doi:10.1109/TIP.2011.2134107.

[14] Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., Sriperumbudur, B.K., 2012. Optimal kernel choice for large-scale two-sample tests, in: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.

[15] Han, T., Pang, J., Tan, A.C., 2021. Remaining useful life prediction of bearing based on stacked autoencoder and recurrent neural network. Journal of Manufacturing Systems 61, 576–591. doi:https://doi.org/10.1016/j.jmsy.2021.10.011.

[16] Kan, M.S., Tan, A.C., Mathew, J., 2015. A review on prognostic techniques for non-stationary and non-linear rotating systems. Mechanical Systems and Signal Processing 62-63, 1–20. doi:https://doi.org/10.1016/j.ymssp.2015.02.016.

[17] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J., 2021. 1D convolutional neural networks and applications: A survey. Mechanical Systems and Signal Processing 151, 107398. doi:https://doi.org/10.1016/j.ymssp.2020.107398.

[18] Li, B., Chow, M.Y., Tipsuwan, Y., Hung, J., 2000. Neural-network-based motor rolling bearing fault diagnosis. IEEE Transactions on Industrial Electronics 47, 1060–1069. doi:10.1109/41.873214.

[19] Li, X., Zhang, W., Ding, Q., 2019a. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. Reliability Engineering & System Safety 182, 208–218. doi:https://doi.org/10.1016/j.ress.2018.11.011.

[20] Li, X., Zhang, W., Ding, Q., Sun, J.Q., 2019b. Multi-Layer domain adaptation method for rolling bearing fault diagnosis. Signal Process. 157, 180–197. doi:10.1016/j.sigpro.2018.12.005.

[21] Liu, H., Mo, Z., Zhang, H., Zeng, X., Wang, J., Miao, Q., 2018. Investigation on Rolling Bearing Remaining Useful Life Prediction: A Review, in: 2018 Prognostics and System Health Management Conference (PHM-Chongqing), pp. 979–984. doi:10.1109/PHM-Chongqing.2018.00175.

[22] Liu, K., Li, Y., 2024. Remaining useful life prediction across machines using multi-source adversarial online knowledge distillation. Engineering Applications of Artificial Intelligence 130, 107726. doi:https://doi.org/10.1016/j.engappai.2023.107726.

[23] Liu, W., Qin, R., 2020. A MultiKernel Domain Adaptation Method for Unsupervised Transfer Learning on Cross-Source and Cross-Region Remote Sensing Data Classification. IEEE Transactions on Geoscience and Remote Sensing 58, 4279–4289. doi:10.1109/TGRS.2019.2962039.

[24] Mao, W., Chen, J., Chen, Y., Afshari, S.S., Liang, X., 2021. Construction of Health Indicators for Rotating Machinery Using Deep Transfer Learning With Multiscale Feature Representation. IEEE Transactions on Instrumentation and Measurement 70, 1–13. doi:10.1109/TIM.2021.3057498.

[25] Mao, W., Chen, J., Liu, J., Liang, X., 2023. Self-Supervised Deep Domain-Adversarial Regression Adaptation for Online Remaining Useful Life Prediction of Rolling Bearing Under Unknown Working Condition. IEEE Transactions on Industrial Informatics 19, 1227–1237. doi:10.1109/TII.2022.3172704.

[26] Medjaher, K., Zerhouni, N., Baklouti, J., 2013. Data-driven prognostics based on health indicator construction: Application to PRONOSTIA's data, in: 2013 European Control Conference (ECC), pp. 1451–1456. doi:10.23919/ECC.2013.6669223.

[27] Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., Varnier, C., 2012. PRONOSTIA : An experimental platform for bearings accelerated degradation tests., in: IEEE International Conference on Prognostics and Health Management, PHM'12., Denver, Colorado, United States. pp. 1–8. doi:hal-00719503.

[28] Nie, L., Zhang, L., Xu, S., Cai, W., Yang, H., 2022. Remaining useful life prediction for rolling bearings based on similarity feature fusion and convolutional neural network. Journal of the Brazilian Society of Mechanical Sciences and Engineering 44. doi:10.1007/s40430-022-03638-0.

[29] Ren, L., Sun, Y., Wang, H., Zhang, L., 2018. Prediction of Bearing Remaining Useful Life With Deep Convolution Neural Network. IEEE Access PP, 1–1. doi:10.1109/ACCESS.2018.2804930.

[30] Shang, Y., Tang, X., Zhao, G., Jiang, P., Ran Lin, T., 2022. A remaining life prediction of rolling element bearings based on a bidirectional gate recurrent unit and convolution neural network. Measurement 202, 111893. doi:https://doi.org/10.1016/j.measurement.2022.111893.

[31] Shuang, L., Shen, X., Zhou, J., Miao, H., Qiao, Y., Lei, G., 2024. Bearings remaining useful life prediction across equipment-operating conditions based on multisource-multitarget domain adaptation. Measurement 236, 115026. doi:https://doi.org/10.1016/j.measurement.2024.115026.

[32] Singh Rathore, M., Harsha, S., 2022. Rolling bearing prognostic analysis for domain adaptation under different operating conditions. Engineering Failure Analysis 139, 106414. doi:https://doi.org/10.1016/j.engfailanal.2022.106414.

[33] Sonoda, S., Murata, N., 2017. Neural network with unbounded activation functions is universal approximator. Applied and Computational Harmonic Analysis 43, 233–268. doi:https://doi.org/10.1016/j.acha.2015.12.005.

[34] Tiddens, W., Braaksma, J., Tinga, T., 2023. Decision Framework for Predictive Maintenance Method Selection. Applied Sciences 13. doi:10.3390/app13032021.

[35] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T., 2014. Deep Domain Confusion: Maximizing for Domain Invariance. arXiv:1412.3474.

[36] 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 PP, 1–12. doi:10.1109/TR.2018.2882682.

[37] Wang, R., Chen, H., Guan, C., 2021. A Bayesian inference-based approach for performance prognostics towards uncertainty quantification and its applications on the marine diesel engine. ISA Transactions 118, 159–173. doi:https://doi.org/10.1016/j.isatra.2021.02.024.

[38] Wu, D., Jennings, C., Terpenny, J., Gao, R.X., Kumara, S., 2017. A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. Journal of Manufacturing Science and Engineering 139, 071018. doi:10.1115/1.4036350.

[39] Yang, J., Peng, Y., Xie, J., Wang, P., 2022. Remaining Useful Life Prediction Method for Bearings Based on LSTM with Uncertainty Quantification. Sensors 22. doi:10.3390/s22124549.

[40] Yang, Z., Baraldi, P., Zio, E., 2016. A comparison between extreme learning machine and artificial neural network for remaining useful life prediction, in: 2016 Prognostics and System Health Management Conference (PHM-Chengdu), pp. 1–7. doi:10.1109/PHM.2016.7819794.

[41] Yao, D., Li, B., Liu, H., Yang, J., Jia, L., 2021. Remaining useful life prediction of roller bearings based on improved 1D-CNN and simple recurrent unit. Measurement 175, 109166. doi:https://doi.org/10.1016/j.measurement.2021.109166.

[42] Yoo, Y., Baek, J.G., 2018. A Novel Image Feature for the Remaining Useful Lifetime Prediction of Bearings Based on Continuous Wavelet Transform and Convolutional Neural Network. Applied Sciences 8. doi:10.3390/app8071102.

[43] Zaman, T., Bayoumi, A., 2014. Analysis of health and usage monitoring system (HUMS) users' perspective towards mission benefits using regression analysis, pp. 1621–1626.

[44] Zeng, F., Li, Y., Jiang, Y., Song, G., 2021. An online transfer learning-based remaining useful life prediction method of ball bearings. Measurement 176, 109201. doi:https://doi.org/10.1016/j.measurement.2021.109201.

[45] Zhang, H.B., Cheng, D.J., Zhou, K.L., Zhang, S.W., 2023. Deep transfer learning-based hierarchical adaptive remaining useful life prediction of bearings considering the correlation of multistage degradation. Knowledge-Based Systems 266, 110391. doi:https://doi.org/10.1016/j.knosys.2023.110391.

[46] Zhang, W., Li, X., Ma, H., Luo, Z., Li, X., 2021. Transfer learning using deep representation regularization in remaining useful life prediction across operating conditions. Reliability Engineering & System Safety 211, 107556. doi:https://doi.org/10.1016/j.ress.2021.107556.

[47] Zhao, B., Zhang, X., Li, H., Yang, Z., 2020. Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions. Knowledge-Based Systems 199, 105971. doi:https://doi.org/10.1016/j.knosys.2020.105971.

[48] Zhao, D., Liu, F., 2022. Cross-condition and cross-platform remaining useful life estimation via adversarial-based domain adaptation. Scientific Reports 12, 878. doi:10.1038/s41598-021-03835-2.

[49] Zhou, L., Duan, F., Corsar, M., Elasha, F., Mba, D., 2019. A study on helicopter main gearbox planetary bearing fault diagnosis. Applied Acoustics 147, 4–14. doi:https://doi.org/10.1016/j.apacoust.2017.12.004. special Issue on Design and Modelling of Mechanical Systems conference CMSM'2017.

[50] Zhu, J., Chen, N., Shen, C., 2020. A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions. Mechanical Systems and Signal Processing 139, 106602. doi:https://doi.org/10.1016/j.ymssp.2019.106602.

II

Literature Study

1
Introduction

Bearings are an important part of rotating machinery, such as the drive train of a helicopter. Degradation of a bearing may lead to reduced system performance or system breakdown in case of a bearing failure [1]. An example of this is the crash of a Super Puma helicopter in Norway caused by the failure of a bearing in the drive train of this helicopter [2]. In order to prevent such accidents, the helicopter Health and Usage Monitoring System (HUMS) was introduced in the 1980s [3]. A number of helicopters, such as the Super Puma helicopter, have been equipped with this system.

The HUMS measures, amongst others, vibration data from several locations in the helicopter drive train, such as the gearboxes and drive shafts. By performing several computations on this data, information can be obtained on potential faults developing within the drive train [4]. Next to using this information for making maintenance decisions, the data also contributes to the implementation of Condition Based Maintenance (CBM) [5], which allows for cost and time efficient maintenance scheduling [6]. By monitoring the degradation of the system, maintenance actions can be planned at the appropriate time, which results in reduced system downtime and less costs, while at the same time improving safety [7]–[9].

Predicting the remaining time until failure is a part of CBM [10]. Estimating this Remaining Useful Life (RUL) is done based on the data collected by the HUMS and past failure data. As RUL prognostics are important for scheduling maintenance activities, RUL estimation has been the focus of a significant number of studies in academia and industry [11]. As a result, several methodologies are already available for RUL prediction, with data-driven approaches being widely used in industry [12]. However, these data-driven methods are tailored to a specific operating condition, such that these methods cannot be used for RUL prediction of a bearing subjected to a different operating condition [13]–[15]. Transfer learning is a potential solution to this challenge associated with RUL prediction. Transfer learning leverages knowledge used to solve a certain problem to solve a new, but similar, problem [16].

A transfer learning method could enable RUL predictions for a drive train bearing, even when no failure data is available for that bearing in a certain operating condition [13]. Developing such a transfer learning model may allow for the implementation of a RUL prediction model that transfers bearing degradation knowledge from a laboratory environment to a real helicopter drive train environment. A benefit of such an approach is that the need for failure data from the helicopter drive train is reduced, which is difficult to collect due to cost and safety concerns [17].

This literature review aims to examine some state-of-the-art transfer learning approaches to bearing RUL prediction. A transfer learning model contains several parts, such as a feature extraction module, a backbone RUL prediction module, and a domain adaptation module. For example, in the feature extraction module several bearing condition indicators may be calculated from the vibration data. A backbone neural network can further extract features from the input data and perform RUL predictions. To gain a comprehensive understanding of transfer learning models, this literature review will also examine their individual components:

- The condition indicators extracted from a vibration signal and how they are used for bearing RUL prediction
- The neural networks used for bearing RUL prediction without transfer learning
- The concept of transfer learning and how transfer learning techniques are employed for RUL prediction of bearings

Chapter 2 will describe the HUMS and how condition indicators can be calculated from the vibration data. Chapter 3 will discuss the three RUL prediction strategies. Next, chapter 4 will describe some common types of neural networks used for RUL prediction of bearings in order to gain some background knowledge on the backbone network of a transfer learning model. This chapter will also explain how condition indicators can be used for RUL prediction. Chapter 5 will introduce the concept of transfer learning and discuss some transfer learning methods for RUL prediction. Finally, chapter 6 provides a summary of the literature review and identifies the research gap.

2

Helicopter Health and Usage Monitoring Systems

In a helicopter, the main rotor and tail rotor are powered by the drive train. In order to monitor the health of the drive train, helicopters are equipped with a HUMS. The HUMS measures, amongst others, vibration data of bearings in the drive train and extracts Condition Indicators (CIs) from this data. This chapter aims to provide background knowledge on this system. Therefore, Section 2.1 introduces a typical helicopter drive train. Section 2.2 will provide some background knowledge on the HUMS system. Finally, Section 2.3 discusses some CIs that can be used to visualise bearing degradation.

2.1. Helicopter Drive Train

The helicopter main and tail rotors are powered by a drive train. Figure 2.1 shows a typical helicopter drive train layout. In this figure, the drive train is from a UH-60 Black Hawk helicopter. Other helicopters, such as the Super Puma helicopter, feature a similar drive train layout. It can be observed that the drive train consists of several components, such as the engines, main gearbox, intermediate gearbox, tail gearbox, and the drive shafts. The output from the engines is transferred to the main gearbox. The main gearbox powers the main rotor and transmits power to the tail rotor through a system of drive shafts and gearboxes. The gearboxes reduce the high revolutions per minute (RPM) generated by the engines to lower levels suitable for the main rotor and tail rotor. The tail drive shaft is supported by hangar bearings [18]. These bearings prevent excessive movement of the tail drive shaft.

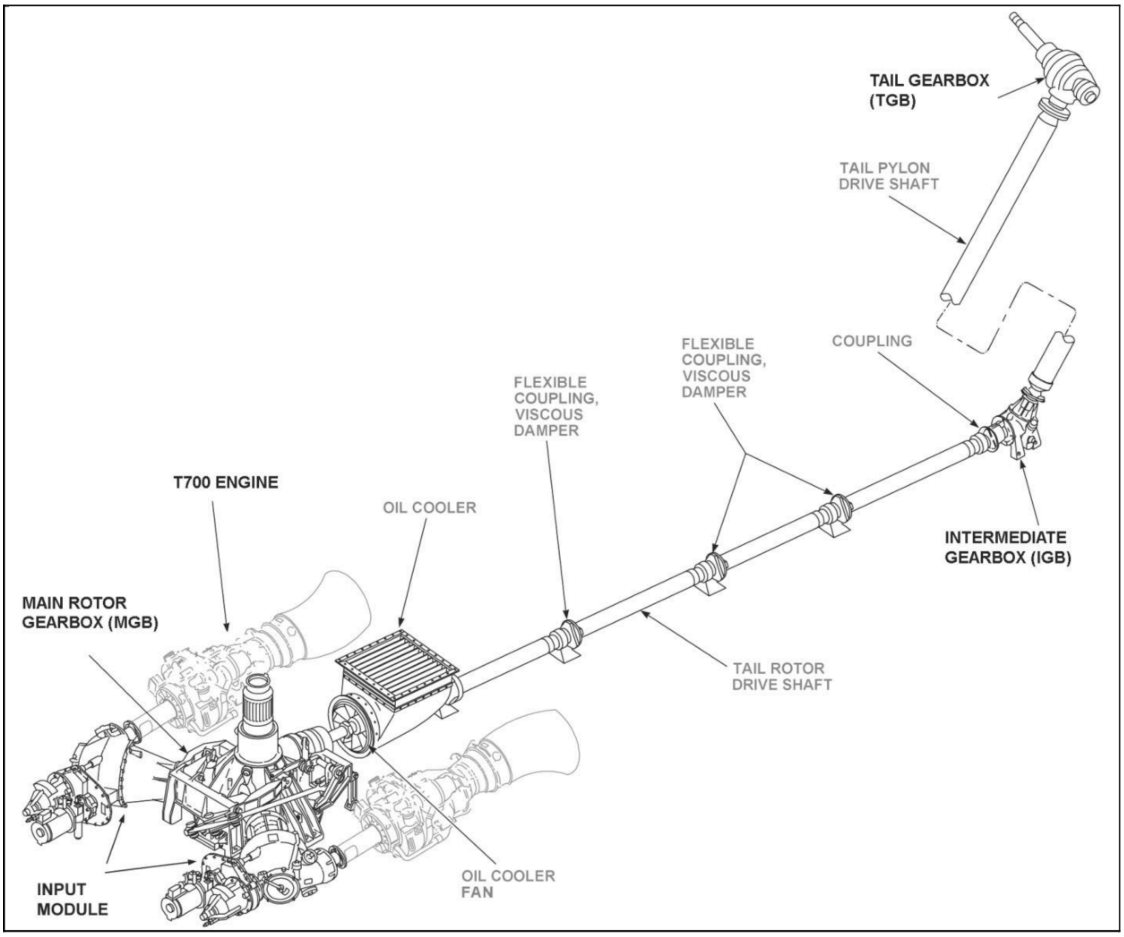


Figure 2.1: The drive train of a UH-60 Black Hawk helicopter [19]

2.2. The Health and Usage Monitoring System

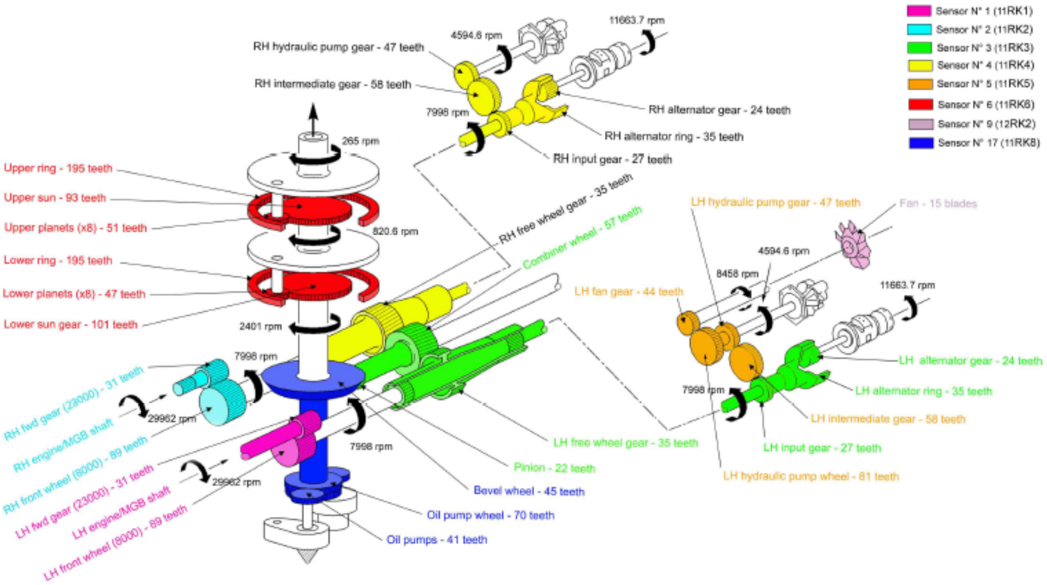
Helicopters such as the Cougar and Super Puma are equipped with a HUMS. Development on such systems started in the 1980s. This was due to a high number of helicopter accidents in the North Sea area [3]. According to the UK Civil Aviation Authority, this accident rate was unacceptable. A solution to reduce the number of accidents was to introduce vibration monitoring to helicopters. The first HUMS systems for helicopters were designed by Steward Hughes Limited. This company also produced a HUMS system for the Super Puma. Next, HUMS systems became a part of helicopter operations [20].

At several locations in the drive train of the helicopters, sensors are placed that measure parameters such as the number of take-offs, landings, engine data, and vibrations during the flight. In figures 2.2 and 2.3, the placement of the sensors in the drive train of the Super Puma helicopter can be observed. Based on the vibration data, the HUMS system calculates several features, also known as CIs [4]. For a gear, this is done by first dividing the raw vibration signal into several segments. Each segment corresponds to one revolution of the shaft. Next, each segment is resampled such that every segments contains the same number of data points. Then, the segments from a certain number of revolutions are summed together and the average values are calculated [21]. This process is also known as computing the time synchronous average. The aim of this process is to reduce the noise of the signal [22]. Based on this averaged signal, the features, or CIs, can be calculated. However, this process cannot be used for bearing vibrations as these vibrations are not synchronous with the shaft speed [23]. In this case, other noise reduction methods have to be implemented in order to calculate the features of the vibration signal.

A damaged component, such as gears, shafts or, bearings, can produce certain fault patterns in a vibration signal. The CIs relate to the vibration signal and thus provide information on the health of the component. This information can then be used for making maintenance decisions [6]. Additionally, chip detectors are used in the engines and gearbox to detect metal particles in the lubrication.

Usage monitoring is also part of the HUMS process [24]. One task of usage monitoring is to report whether parameters are exceeded during the flight. If such a parameter exceedance occurs, then this event is logged such that the information relating to this event can be reviewed by the maintenance crew [20]. An example of a parameter exceedance is overtorque. This occurs when the pilot needs extra lift or thrust such that the torque exceeds the 100% limit. After an overtorque event, parts of the helicopter may need to be inspected. Thus, this part of the system checks whether the helicopter has been used within the specified limits. Another task of the usage monitoring system is to calculate the load cycles of the components, based on torque and rotor speed for example. The system can then warn the maintenance crew when a component is about to reach the limit of its lifetime [20].

Such a HUMS system has several benefits, in terms of safety and costs. Safety related benefits are that HUMS enables to detect faults earlier and reduce the risk of in-flight failures or emergency landings. Additionally, HUMS systems contribute to CBM. This allows to do reduce downtime for unforeseen maintenance events, diagnose problems, and to possibly elongate inspection intervals of components [6].



Sensor N° 1 (11RK1)
Sensor N° 2 (11RK2)
Sensor N° 3 (11RK3)
Sensor N° 4 (11RK4)
Sensor N° 5 (11RK5)
Sensor N° 6 (11RK6)
Sensor N° 9 (12RK2)
Sensor N° 17 (11RK8)
RH hydraulic pump gear - 47 teeth
RH intermediate gear - 58 teeth
4594.6 rpm
11663.7 rpm
7998 rpm
RH alternator gear - 24 teeth
RH alternator ring - 35 teeth
RH input gear - 27 teeth
265 rpm
Upper ring - 195 teeth
Upper sun - 93 teeth
Upper planets (x8) - 51 teeth
820.6 rpm
Lower ring - 195 teeth
Lower planets (x8) - 47 teeth
Lower sun gear - 101 teeth
2401 rpm
RH free wheel gear - 35 teeth
Combiner wheel - 57 teeth
LH hydraulic pump gear - 47 teeth
Fan - 15 blades
4594.6 rpm
8458 rpm
LH fan gear - 44 teeth
11663.7 rpm
LH alternator gear - 24 teeth
LH alternator ring - 35 teeth
LH intermediate gear - 58 teeth
7998 rpm
LH input gear - 27 teeth
LH hydraulic pump wheel - 81 teeth
LH free wheel gear - 35 teeth
Pinion - 22 teeth
Bevel wheel - 45 teeth
Oil pump wheel - 70 teeth
Oil pumps - 41 teeth
7998 rpm
RH fwd gear (23000) - 31 teeth
RH engine/MGB shaft
29962 rpm
RH front wheel (8000) - 89 teeth
7998 rpm
LH fwd gear (23000) - 31 teeth
LH engine/MGB shaft
29962 rpm
LH front wheel (8000) - 89 teeth

Figure 2.2: The placement of sensors in the main gearbox of the Super Puma helicopter [20]

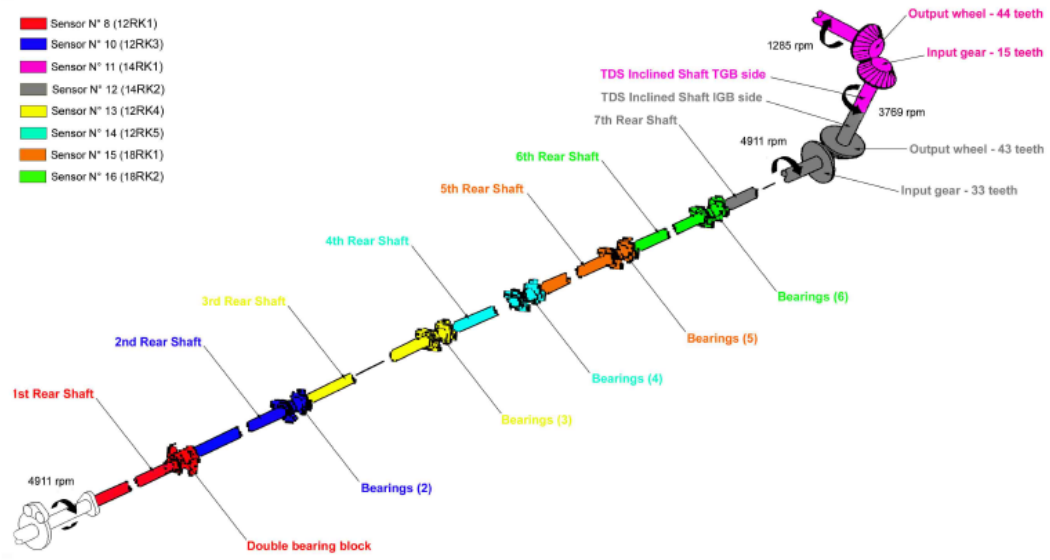


Figure 2.3: The placement of sensors in the tail drive shaft and tail gearbox of the Super Puma helicopter [20]

2.3. Common Condition Indicators

Several different CIs are used for fault diagnosis. Different faulty conditions of a gear, shaft, or bearing can be identified by a CI. In this section, some examples of these indicators will be discussed.

2.3.1. General Condition Indicators

In table 2.1, some commonly used features and their corresponding equations can be observed. In these equations, x_i is the value of the i^{th} data point in signal x. N is the number of data points in signal x. General features are RMS, kurtosis, and Crest Factor (CF). The RMS value is a measure for the energy content of the signal. If there is a defect in a bearing or gear, then the energy content of the vibration will increase. If the damage increases, then the RMS value will also increase. In the early stage of a damage, the RMS will not rise significantly. However, the RMS can be used to track the fault progression.

The kurtosis is a value describing the shape of a distribution. If the distribution is a Gaussian distribution, then the kurtosis value equals three. If the signal contains a number of sharp peaks, then the kurtosis value will be higher than three. If there is a damage to a gear or bearing, then there will be some peaks in the signal. Consequently, the kurtosis value will be bigger than three. The RMS and kurtosis values are widely used in vibration analysis. These values have been used in previous research to predict the RUL of a bearing in a wind turbine [25].

The CF can also indicate faulty machinery. The CF detects changes in the signal due to an impulsive vibration source, due to gear tooth damage for example [26]. The CF value increases if there is a number of peaks with high amplitude [27]. The impulse factor compares the height of a peak to the average vibration level. The skewness factor is a value defining the symmetry of the probability density function of the time series data [28].

Table 2.1: An overview of commonly used features (CIs) for RUL prognostics [26], [28], [29]

Number	Vibration feature	Formula
1	RMS	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2}$
2	Kurtosis	$\frac{N\cdot\sum_{i=1}^{N}(x_i-\bar{x})^4}{\left[\sum_{i=1}^{N}(x_i-\bar{x})^2\right]^2}$
3	Mean	$\frac{1}{N}\sum_{i=1}^{N}(x_i)$
4	Minimum	$\min(x)$
5	Peak-to-Peak	$\max(x)-\min(x)$
6	Crest factor	$\frac{\max(x)}{\mathrm{RMS}(x)}$
7	Energy	$\sum_{i=1}^{N}x_i^2$
8	Skewness	$\frac{\sum_{i=1}^{N}(x_i-\mathrm{mean}(x))^3}{\left(\sqrt{\sum_{i=1}^{N}(x_i-\mathrm{mean}(x))^2}\right)^3}$
9	Standard deviation	$\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mathrm{mean}(x))^2}$
10	Variance	$\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mathrm{mean}(x))^2$
11	Impulse factor	$\frac{\max(x)}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$
12	Shape factor	$\frac{\mathrm{RMS}(x)}{\mathrm{mean}(x)}$

2.3.2. Bearing Condition Indicators

A bearing consists of several components. These are the rolling elements, such as the balls, the inner ring, the outer ring, and the cage [30]. One of the rings does not rotate and one of the rings does rotate. For example, the inner ring can be fixed to a rotating shaft whereas the outer ring remains stationary. The balls roll on the raceways formed by the outer and inner rings. The balls transfer the loads between the inner and outer ring. The cage keeps the balls a certain fixed distance from each other [31]. These elements are depicted in figure 2.4. In these four elements, faults could occur. For example, material may chip off the bearing balls or cracks may develop in the inner or outer ring.

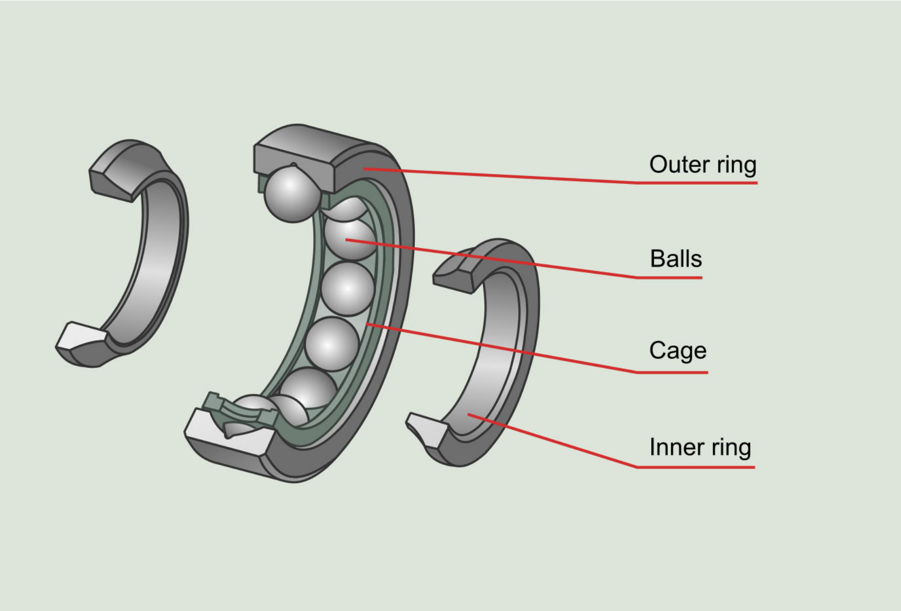


Figure 2.4: The components of a ball bearing [31]

If one the components is damaged, then the vibration signals will show peaks at certain frequencies. These frequencies are excited by the rolling elements, such as the balls, passing the damaged location. The frequencies are the following: [32]:

- Ball Pass Frequency Outer Race (BPFO): this is the frequency at which the balls pass a damaged part of the outer ring.
- Ball Pass Frequency Inner Race (BPFI): this is the frequency at which the balls pass a damaged part of the inner ring.
- Ball Spin Frequency (BSF): this is the frequency at which a damaged ball rotates. This damaged ball impacts both the inner and outer ring during one revolution.
- Fundamental Train Frequency (FTF): this is the frequency at which the damaged cage rotates. Usually, the cage does not carry load and the FTF is not visible in the vibration signal. In the equation for FTF, a minus sign is used for a rotating inner ring. If the outer ring rotates, a plus sign should be used.

The equations for the fault frequencies are shown in table 2.2. In these equations, f is the speed of the shaft in Revolutions per Minute (RPM), Z is the number of rolling elements in the bearing, d is the diameter of the rolling element, D is the pitch of the bearing, and α is the contact angle of the bearing.

Table 2.2: An overview of bearing fault frequencies

Name	**Formula**
BPFO	$f \cdot \frac{Z\left(1+\frac{d}{D}\cos\alpha\right)}{2}$
BPFI	$f \cdot \frac{Z\left(1-\frac{d}{D}\cos\alpha\right)}{2}$
BSF	$f \cdot \frac{1}{2}\frac{D}{d}\left(1-\left(\frac{d}{D}\cos\alpha\right)^2\right)$
FTF	$f \cdot \frac{\left(1-\frac{d}{D}\cos\alpha\right)}{2}$

Depending on which frequencies in the frequency spectrum coincide with the bearing fault frequencies, one can determine whether the outer ring, inner ring, or balls are damaged. An example of an outer ring fault can be observed in figure 2.5 [33]. This frequency spectrum was obtained by performing an envelope analysis of the vibration signal. In this case, the resulting signal contained peaks at the harmonics of the BPFO, which can also be observed in figure 2.5 [33].

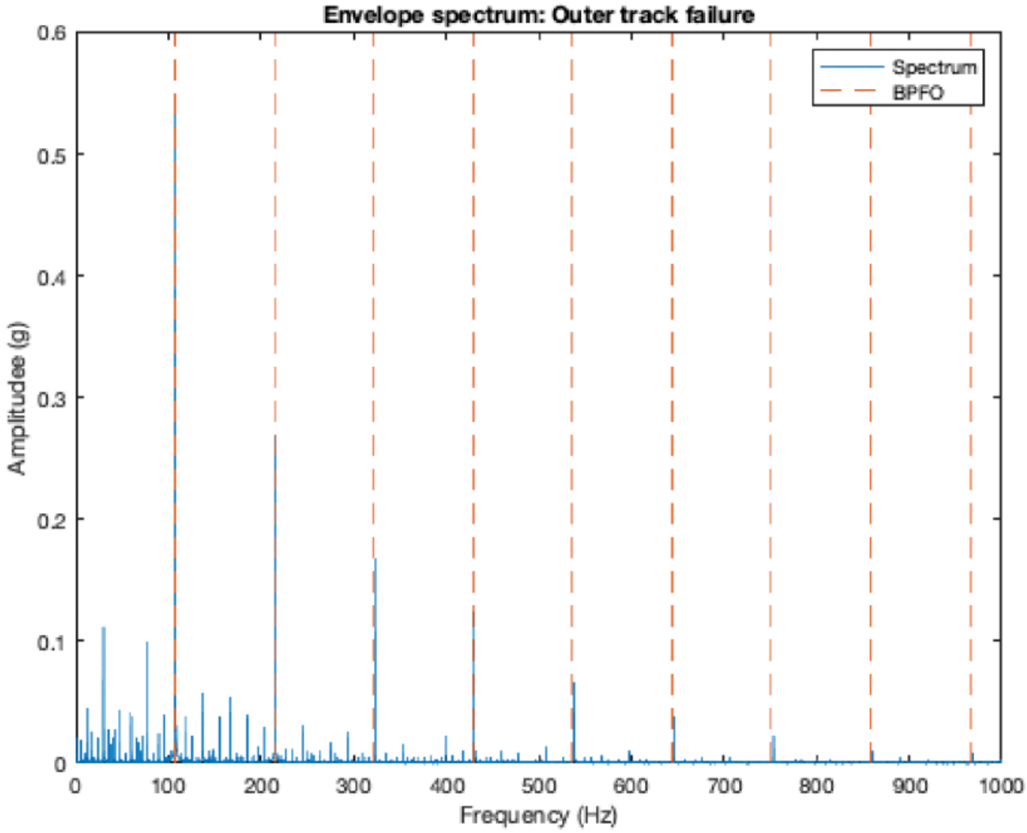


Figure 2.5: A frequency spectrum of a bearing with an outer ring fault [33]

In previous research, researchers also used the Energy Index (EI) of the vibration signal, along with the RMS and kurtosis value to estimate the RUL of a bearing in a wind turbine gearbox [25]. The EI can be calculated by dividing the RMS of a part of a signal by the the RMS of the entire signal and finally squaring this ratio:

$$EI = \left(\frac{RMS_{part}}{RMS_{total}}\right)^2 \tag{2.1}$$

this EI allows to detect damages to bearings in early stages [34]. However, in the wind turbine application, the EI showed worse performance for RUL estimation compared to the RMS and kurtosis [25].

3
Remaining Useful Life Prediction Methods

Prognostics can be defined as estimating the time to failure of an equipment. This estimated time to failure is also known as the RUL. The RUL is also visualised in figure 3.1.

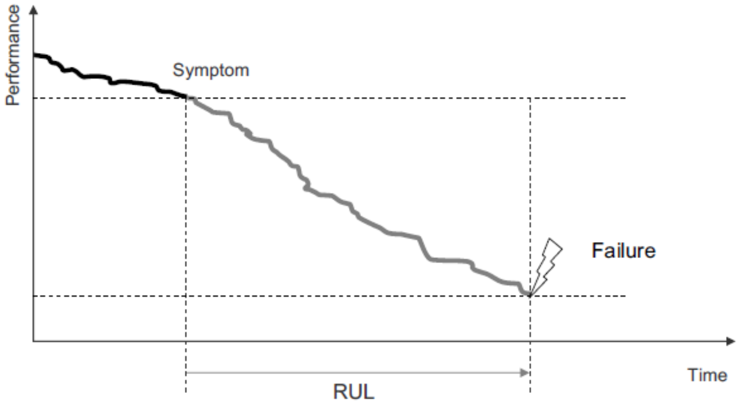


Figure 3.1: A visualisation of the RUL [35]

For RUL prognostics, several approaches are possible. These approaches are a physical model based method, a data-driven method, and a hybrid method [35]. These three approaches are summarised in figure 3.2 [36].

In a physical model based method, a mathematical model is set up to model the degradation of the component. By considering the current degraded state of the component and future usage of the component, the mathematical model can perform RUL estimations. This is also called a physics of failure approach [37]. Such a model can produce accurate results, as the predictions are based on a physical model. Furthermore, the results of such a model are easier to understand, as the results are based on a physical phenomenon. However, deriving these models requires a high amount of expertise on the component and the degradation of the component [35]. Additionally, accurate prediction results require an accurate model of the failure mechanism. In practice, the actual failure mechanisms are usually more complex than the failure mechanism model [38], [39].

In a data-driven method, the RUL is predicted using historical failure data of a machine, or similar machines [25]. An algorithm tries to derive a relationship between bearing health degradation and time to failure based on historical sensor data [12], [37]. The input data, such as usage data or health monitoring data, is then compared with these patterns and relations in order to make a RUL prediction [37]. An advantage of such a data-driven method is that specific expertise on the physics of the system is not needed, as apposed to a physical model method. Examples of algorithms are Neural Networks (NNs), LSTMs, or CNNs. However, such a data-driven method requires large amounts of training data. Additionally, the training data should contain failures in order for the algorithm to learn the correct patterns and relations. However, such data might not always be available [17].

In a hybrid method, both a physical model based approach and a data-driven method is used to predict the RUL. A hybrid model aims to combine the advantages of the physical model method and the data-driven method. An example of such a hybrid method is the estimation of the RUL of wind turbine bearings. In this hybrid method, first the features were extracted from the data. Next, relevant features were selected based on the monotonicity of the feature. Features with greater monotonicity are used as features for the model. A Health Indicator (HI) is constructed from these features using a Principal Component Analysis (PCA). The RUL is predicted by an exponential degradation model. It was shown that this method can accurately predict the RUL [40]. In another study, bearing degradation was also modelled as an exponential function [38]. First, a sparse representation of the features was obtained by relevance vector machine regression. Then, an exponential degradation model was fitted on a sparse representation of features extracted from the vibration data. This model is then used to predict the RUL of a bearing.

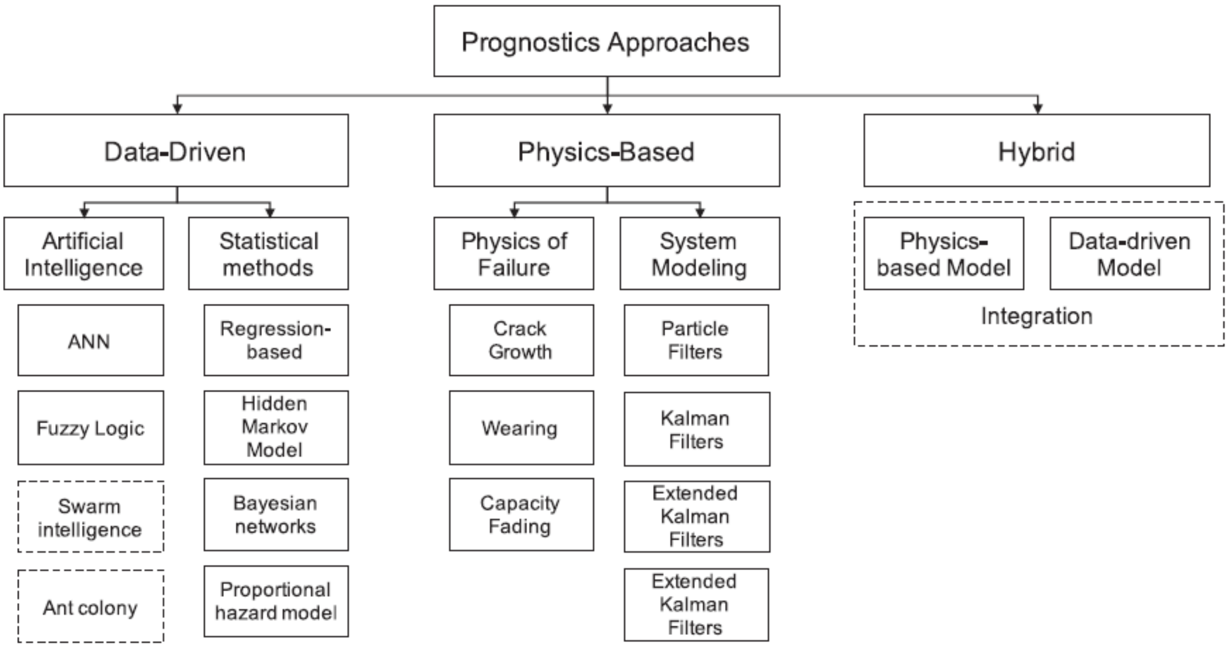


Figure 3.2: An overview of prognostics approaches [36]

Due to their lower expertise requirements and greater generalisability compared to physics-based approaches, data-driven models are widely used in industry [12]. Therefore, the remainder of the literature review will be focused on data-driven bearing RUL prediction methods using neural networks.

4

Data-driven Remaining Useful Life Prediction of Bearings

A general procedure for predicting the RUL of bearing consists of several steps. In the first step, data is to be collected from the bearing. Such data can be a vibration signal for example. The second step is to extract features from this data. Such features can be time domain features, frequency domain features, and time-frequency features. These features do have considerable influence on the prediction performance of the model. The third step is to construct the RUL prediction model. The final step is to generate the RUL predictions for the testing data [41]. There are several types of data-driven models, such as feedforward neural networks (such as a CNN) or recurrent neural networks (such as a LSTM). Feedforward models are able to extract certain patterns from the data due to the model structure [42]. However, such models are not suitable for deriving a relation between time and degradation. Recurrent neural networks can capture these temporal features. However, a feedforward model is superior to a recurrent model in terms of extracting features unrelated to time [43].

This chapter will describe some data-driven RUL prediction methods. Section 4.1 describes an approach involving several CIs for prediction the RUL of a wind turbine bearing. Section 4.2 introduces the LSTM network and describes prediction approaches involving variants of the LSTM. Section 4.3 describes the CNN and presents some methods involving CNNs. Finally, Section 4.4 describes automated RUL prediction approaches where a CNN is used for automatic feature extraction. The methods described in this chapter are selected as they clearly show how LSTM, CNN, or CNN-LSTM models can be used for the RUL prediction of bearings.

4.1. Remaining Useful Life Prediction of a Gearbox Bearing using RMS and Kurtosis

In order to increase wind turbine availability, RUL prognostics of bearings are also relevant in this sector. In Section 2.3, several CIs were described. Some of these condition indicators are used for the prediction of the RUL of a bearing in a wind turbine gearbox [25]. In this work, the RMS and the kurtosis were used for making the RUL predictions.

For their RUL prediction method, the authors use a regression model and an Artificial Neural Network (ANN). The regression model is used to derive a relation between bearing degradation and time. Hence, the authors aim to find a relation between the time and either the RMS or kurtosis. First, the RMS and kurtosis are calculated from the raw data. However, in this data, fluctuations are present. To prevent that the regression model will follow these fluctuations, the authors fit a polynomial regression model and an exponential regression model to the data. How well the regression models fit the data is evaluated by the Root Mean Square Error (RMSE), R^2, and adjusted R^2. The RMSE is calculated by equation 4.1. The R^2 is calculated according to 4.2. In these equations, y_i is the i^{th} calculated RMS or kurtosis. $\hat{y}_i$ is the predicted value. $\bar{y}$ is the mean of the calculated RMS or kurtosis. n is the number of RMS or kurtosis values.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (4.1)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad (4.2)$$

Additionally, the monotonicity and trendability of the CIs is used to evaluate regression performance. These values are shown in equations 4.3 and 4.4 respectively [44]. The monotonicity is a value in the range [0,1] and shows whether the data shows an increasing or decreasing trend. As the bearing is going through a degradation process, it is expected that the monotonicity is close to one. In equation 4.3, M is the number of degradation trajectories. For example, if there are ten bearings, then M equals ten. N_j is the number of data points in trajectory j. $\mathrm{sgn}\left(x_j(k+1) - x_j(k)\right)$ is a function that aims to obtain sign difference between the values of data points $k+1$ and k. The trendability value, a value in the range [0,1], shows whether a CI trajectory shows a similar shape across different bearings. In equation 4.4, the trendability is calculated by the correlation coefficient between trajectories j and k. This is done for all j, k combinations and the minimum value is reported as the trendability for a certain CI.

$$\mathrm{mon} = \frac{1}{M}\sum_{j=1}^{M}\left|\sum_{k=1}^{N_j-1}\frac{\mathrm{sgn}\left(x_j(k+1) - x_j(k)\right)}{N_j - 1}\right| \qquad (4.3)$$

$$\mathrm{tren} = \min_{j,k}|\mathrm{corr}(x_j, x_k)| \quad , \forall j, k \in M \qquad (4.4)$$

The authors of [25] collected wind turbine bearing data for 50 consecutive days. For each day, the authors calculated the RMS and kurtosis. These values were then fitted to an exponential model and polynomial model. Based on the RMSE, R^2, and adjusted R^2, it was concluded that an exponential model represents the degradation process of a bearing the most accurately. The RMS value was the best in terms of monotonicity and trendability. Therefore, the authors used the fitted RMS as outputs for ANN.

A multi-layer feed-forward back-propagation ANN is used for RUL prediction. This network makes use of an input layer, two hidden layers with nine neurons in the first layer and seven neurons in the second layer, and an output layer. Such a network layout resulted in the minimum number of training errors. An example of such a network can be observed in figure 4.1. The input nodes are the nodes x and y is the output node. Additionally, weighting coefficients **w** and bias coefficients **b**. The hidden layers add processing steps in order to provide converging solutions. An formulation describing such an ANN can be observed in equation 4.5 [25].

$$\mathbf{y_i} = \varphi_0 \left[\mathbf{C} \varphi_h \left[\mathbf{B u_i} + \mathbf{b_h} \right] + \mathbf{b_0} \right] \tag{4.5}$$

$\mathbf{y_i}$ is the output of the ANN, $\mathbf{u_i}$ is the input vector. **C** is the weighting matrix between the hidden layer and output layer. **B** is weighting matrix that connects the input layer to the hidden layer. $\mathbf{b_h}$ and $\mathbf{b_0}$ represent the bias vectors of the hidden layers and output layer respectively. The activation functions of the nodes in the the hidden layers and the output are represented by φ_h and φ_0 respectively. The weighting and bias values were obtained by making use of a Levenberg Marquardt learning algorithm. The size of the layers was determined using the mean square errors between the target output and the estimated output.

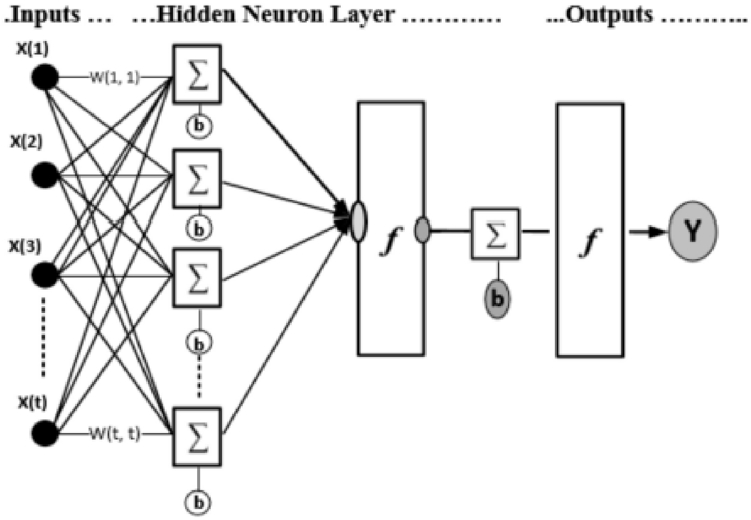


Figure 4.1: A multi-layer feed-forward ANN [25]

This ANN makes use of two input features, namely the raw RMS and kurtosis. The fitted RMS, which was obtained using the exponential degradation model, is the target variable. The RUL is calculated by subtracting the current time from the failure time. For calculating the true RUL, the failure time is the time at which the bearing stopped running. In order to predict the RUL, the failure time must be estimated by extrapolating the fitted RMS curve to a predefined failure threshold. This could be done by making use of an exponential degradation model. An example of the results obtained with this method can be observed in figure 4.2.

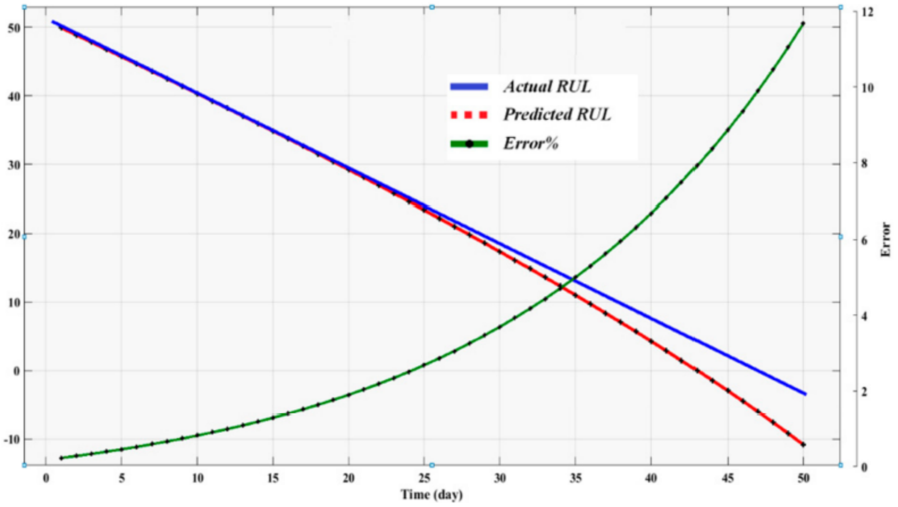


Figure 4.2: The RUL prediction results for an ANN [25]

With this method, the authors obtained a Mean Squared Error (MSE) of 5.62. This shows that the model is accurate. From figure 4.2, it can be observed that the RUL is predicted for a long time span (50 days). This makes it easier to schedule maintenance activities as opposed to a case where the RUL is predicted a few hours before failure. However, it can also be observed that the accuracy of the model deteriorates once the bearing is approaching failure. This is not desired, as the uncertainty regarding the predicted failure time of the bearing increases. This makes it more difficult to schedule maintenance at the optimal moment. Additionally, it should also be noted that this method results in a single value RUL prediction. In order to account for the stochasticity of the degradation process, a probabilistic RUL estimation is recommended by the authors [25].

4.2. Remaining Useful Life Prediction of Bearings by a LSTM Network

A LSTM network is a recently used method for predicting the RUL of bearings. A LSTM is a version of a Recurrent Neural Network (RNN). However, a RNN is difficult to use with large amounts of input data due to the exploding and vanishing gradient problem. The LSTM network aims to alleviate these problems [43]. A LSTM network is build up out of memory, each unit consists of a forget gate, input gate, and output gate. An example of such a memory unit can be observed in figure 4.3 [45].

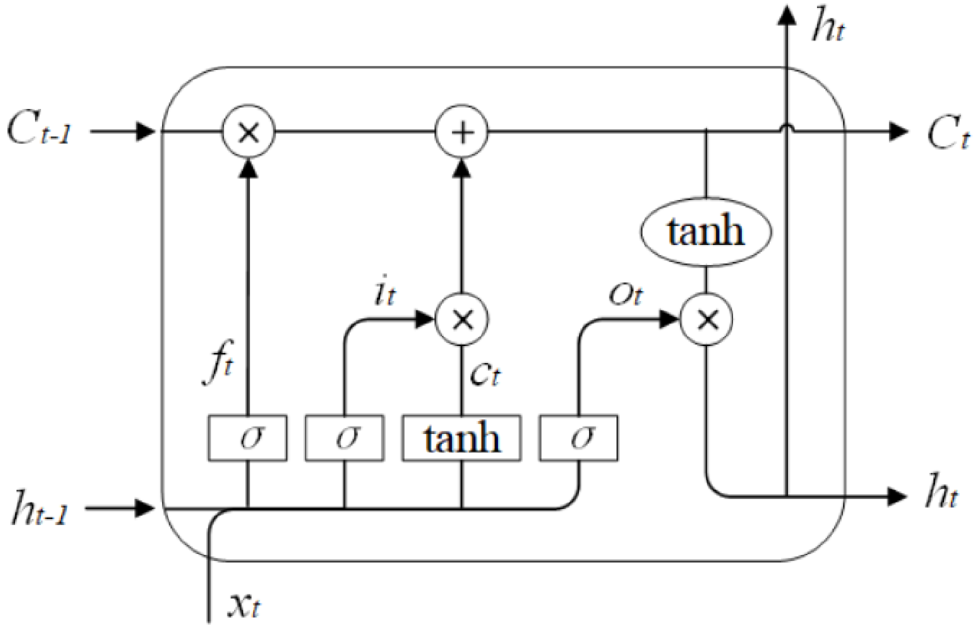


Figure 4.3: A LSTM memory unit structure [45]

In figure 4.3, several variables can be observed. C_{t-1} is the initial long term memory. C_t is the updated long term memory. The black line connecting C_{t-1} and C_t represents the flow of the long term memory through the memory unit. C_t is also the initial long term memory of a neighbouring memory unit. h_{t-1} is the initial short term memory of the unit. h_t is the updated short term memory, which is the output of this memory unit and the initial short term memory of the neighbouring unit. x_t is the input value [43].

The memory unit is governed by several equations. The forget gate is described as follows:

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{4.6}$$

The forget gate determines the amount of long term memory, C_{t-1} to be remembered. b_f is the bias and W_f is the weight matrix of the forget gate. Next, the input gate is expressed by the following two equations:

$$c_t = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{4.7}$$

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{4.8}$$

c_t is the memory that can potentially be added to the long term memory. This value is obtained by combining h_{t-1} and x_t, multiplying the values with the weight matrix of the current input cell state (W_c) and adding the bias of the current input cell state (b_c). Only a fraction of c_t is added to the long term memory. This fraction is the i_t value. This value is obtained in the same way as f_t, however in this case W_i and b_i are used. These are the weight matrix and bias of the input gate respectively. Now that these values are known, C_t can be calculated as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot c_t \tag{4.9}$$

Finally, the output gate is described by the following equations:

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{4.10}$$

$$h_t = o_t \cdot \tanh\left(C_t\right) \tag{4.11}$$

Here, o_t is the output gate. W_o and b_o are the weight and bias terms associated to the output gate respectively [45]. Note that the sigmoid function takes an input value and returns a value between zero and one. The tanh function takes an input value and returns a value between negative one and positive one.

Versions of the LSTM network have been used for the RUL prediction of bearings, in [45]–[47]. A common step in these approaches is that the authors first extract some features from the vibration data. In [47], the authors extract ten time domain features from the data. A number of these features are also presented in Table 2.1. Next to these time domain features, the authors of [46] and [45] extract features from the frequency domain and time-frequency domain as well. Since information on bearing degradation is also reflected in these domains, the authors of [46] and [45] have access to larger amount of information compared to only the frequency domain [29], [48]. After feature extraction, metrics such as the monotonicity (Equation 4.3), trendability (Equation 4.4) and robustness to noise are applied to determine the most useful features [46], [47]. As an additional data preprocessing step, the method of [47] constructs a health indicator based on the four best features using a stacked autoencoder.

The next step is to feed these features into the LSTM networks. The authors of [45] use a different version of the LSTM, namely the Gated Recurrent Unit (GRU). The GRU is a more concise version of the LSTM memory unit depicted in figure 4.3. Additionally, as opposed to [46] and [47], this method also divides the bearing operating data in a healthy and degraded operations phase and train the GRU network on the degraded phase. An advantage of this data division is that the phase in which a relatively small amount of degradation is visible is not used for training the model. All three models fed the extracted features into a (version of) LSTM model. These models contained LSTM layers, dropout layers, and fully connected layers.

These three methods showed competitive results on their test datasets, however the results of [45] and [47] provided only point estimates of the RUL. In order to increase the reliability of their results, the authors of [46] included an uncertainty estimate in their results. This uncertainty quantification is done using the dropout layers. Normally, dropout is used to prevent overfitting of the model. Dropout is then applied during model training and is done by randomly deleting, or reducing, network weights or connections between nodes [46]. Uncertainty quantification can also be done by applying Bayesian theory to neural networks. However, this requires significant adaptations in network structure and increases computational costs [49]. As dropout does not change the network structure, dropout is used to quantify prediction uncertainty. As dropout remains activated during inference, multiple RUL predictions can be obtained for the same data sample if it is passed through the network multiple times. Hence, a distribution of RULs is obtained with the mean of the distribution being the point estimate of the RUL. A nonparametric kernel density estimation is used to obtain the kernel density distribution of the RUL at different time points. An example of the prediction results including uncertainty for a testing bearing are as follows [46]:

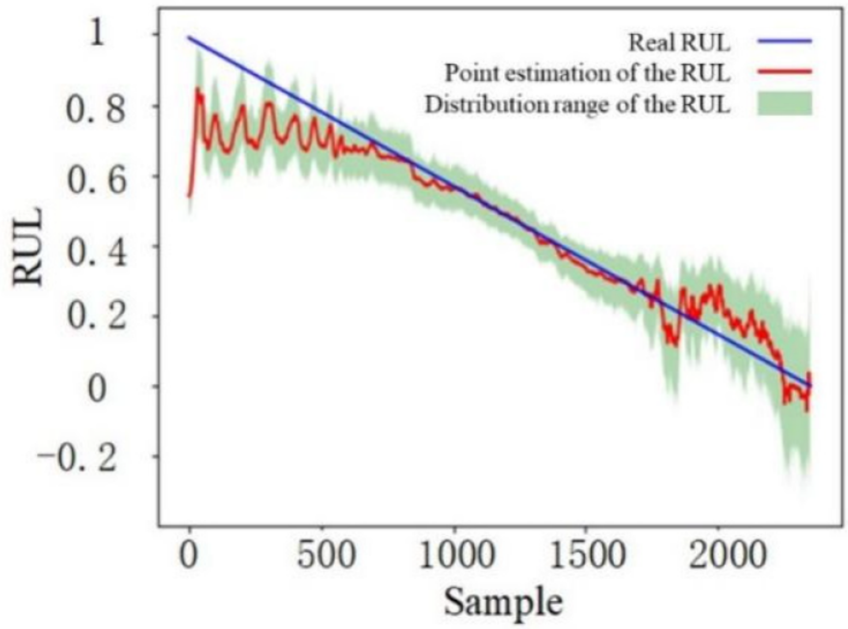


Figure 4.4: RUL prediction results including uncertainty estimates [46]

In figure 4.4, several stages can be observed. For the first 800 sample points, the RUL predictions are horizontal. In this stage, the bearing was still in healthy condition. For sample points 800-1700, it can be observed that the degradation of the bearing can be predicted by the model accurately. In the last stage, the RUL predictions vary widely again. In this stage, bearing degradation accelerates due to a failure occurring in the bearing. As bearing degradation accelerates, some features may also change significantly, such that RUL predictions based on these features will also vary [46]. This can also be observed in the kernel density distributions in figures 4.5 and 4.6 [46]. It can be observed that in the second stage, the predictions are accurate as the distributions are centred on the real RUL. In third stages, it can be observed that the distributions are scattered. This indicates that model performance deteriorates close to the failure.

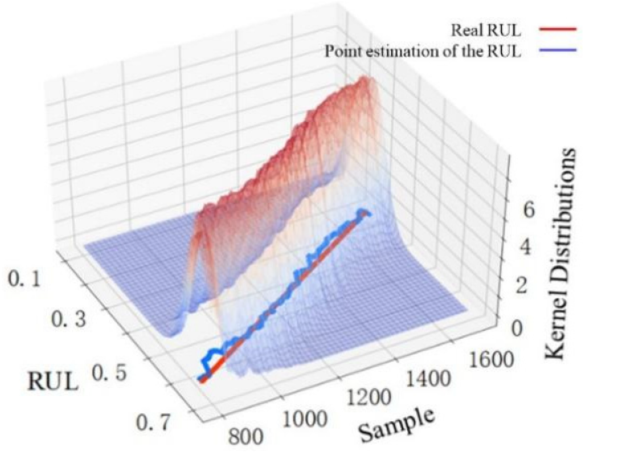


Figure 4.5: RUL point estimation and kernel density distributions in the second stage [46]

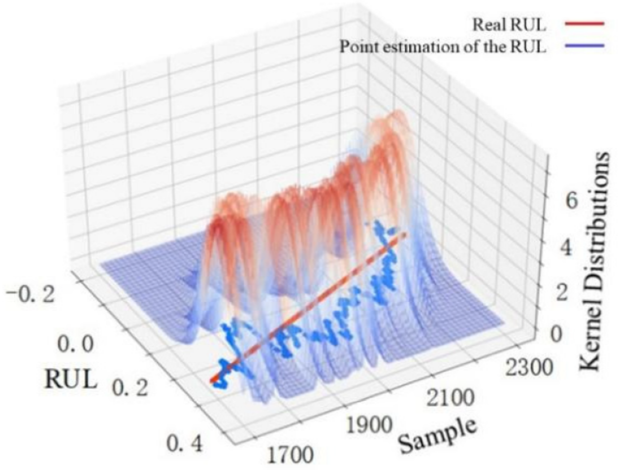


Figure 4.6: RUL point estimation and kernel density distributions in the third stage [46]

From these methods it can be observed that the LSTM is well suited for capturing a relationship between bearing degradation and time, as LSTMs remember information from previous time steps to predict the RUL for the next time step [50].

4.3. Remaining Useful Life Prediction of Bearings by a CNN Network

Another recently used RUL prediction method is to use a (variant of) the CNN. In previous research, a CNN has shown potential for predicting the RUL of a bearing. In earlier research, a CNN combined with a support vector regression model showed promising performance on the Pronostia data set [42]. In this case a benefit of a CNN was exploited, namely that a CNN can be used to identify hidden patterns between the raw vibration data and a HI label [42]. Hence, the CNN can be of great use for feature extraction.

For time series signals, usually a one dimensional CNN is used [29], as opposed to a two dimensional version which is commonly used for image processing. Some elements of a one dimensional CNN are a convolutional layer, a pooling layer, and a fully-connected layer. The function of the convolution layer is to extract features from the data. A filter, or kernel, slides along the input data and performs a dot product between the region of input data perceived by the filter and the weights of the filter. This is represented by equation 4.12 [51]. After the convolution operation, the data passes through the activation function, represented by equation 4.13.

$$y_{l,j}^{\text{ReLU}} = f\left(\sum_{i=1}^{k} w_{i,j}^{l} * y_{l-1,i}^{\text{pool}} + b_j^l\right) \tag{4.12}$$

$$f(x) = \max(0, x) \tag{4.13}$$

In Equation 4.12, $y_{l,j}^{\text{ReLU}}$ represents j^{th} channel output of convolutional layer l, after the activation function has been applied. $y_{l-1,i}^{\text{pool}}$ is the i^{th} channel output of the previous pooling layer $l-1$, and thus the input to layer l. $w_{i,j}^{l}$ denotes the kernel weights for input channel i and output channel j of layer l. b_j^l represents the bias of the j^{th} channel in layer l. k is the number of channels in layer $l-1$. $*$ denotes the convolution operation [51]. $f(\cdot)$ is the activation function and determines whether a neuron should be activated. A commonly used activation function is Rectified Linear Unit (ReLU). If this function receives a negative input, then the function returns zero. If the function receives a positive input, it returns the same value. This is expressed in equation 4.13 [52].

The next layer is the pooling layer. The function of the pooling layer is to reduce the dimension of the convolved feature in order to reduce the computational effort. For the pooling layer, there are several types. Examples are average pooling and maximum pooling. Maximum pooling is an efficient pooling method and improves generalisation. The pooling operation has a certain window and the window slides over the data similar to a convolving operation. However, the stride is in this case equal to the window size. In maximum pooling, the maximum is taken from the data in the pooling window obtained by the ReLU operation. This is also shown in equation 4.14 [29]. However, maximum pooling only outputs the maximum value inside the pooling window, such that some information may be lost. Average pooling uses all data points in the pooling window to output the average value, which avoids a potential loss of information [53]. Average pooling is also shown in equation 4.15 [51].

$$y_{l,j}^{\text{pool}} = \max\left(w \cap y_{l-1,j}^{\text{ReLU}}\right) \tag{4.14}$$

$$y_{l,j}^{\text{pool}} = \text{mean}\left(w \cap y_{l-1,j}^{\text{ReLU}}\right) \tag{4.15}$$

In equations 4.14 and 4.15, $y_{l-1,j}^{\text{ReLU}}$ represents the output of the previous convolutional layer $l-1$. w is the pooling window size and $\cap$ is the intersection between the pooling window and the input data to the pooling layer.

The output of the convolutional layers and pooling layers is flattened into a column vector. The elements of this flattening layer are fully interconnected to further fully connected layers. In the case of RUL prediction, the final fully connected layer consists of one neuron, which is the RUL value. The interconnection is presented in Equation 4.16 [51]:

$$y^l = f\left(w^l * x^{l-1} + b^l\right) \tag{4.16}$$

Here, x^{l-1} and y^l are the input and output to fully connected layer l respectively. w^l and b^l are the weight matrix and the bias term of fully connected layer l. $f(\cdot)$ is the activation function, which can be a ReLU activation function. Since the RUL labels are scaled between 0 and 1, the sigmoid activation function can also be used for the final layer [54]. The sigmoid activation function scales the input to the function between 0 and 1, as presented in Equation 4.17.

$$f(x) = \frac{1}{1+e^{-x}} \tag{4.17}$$

A schematic overview of a one dimensional CNN can be observed in figure 4.7 [29].

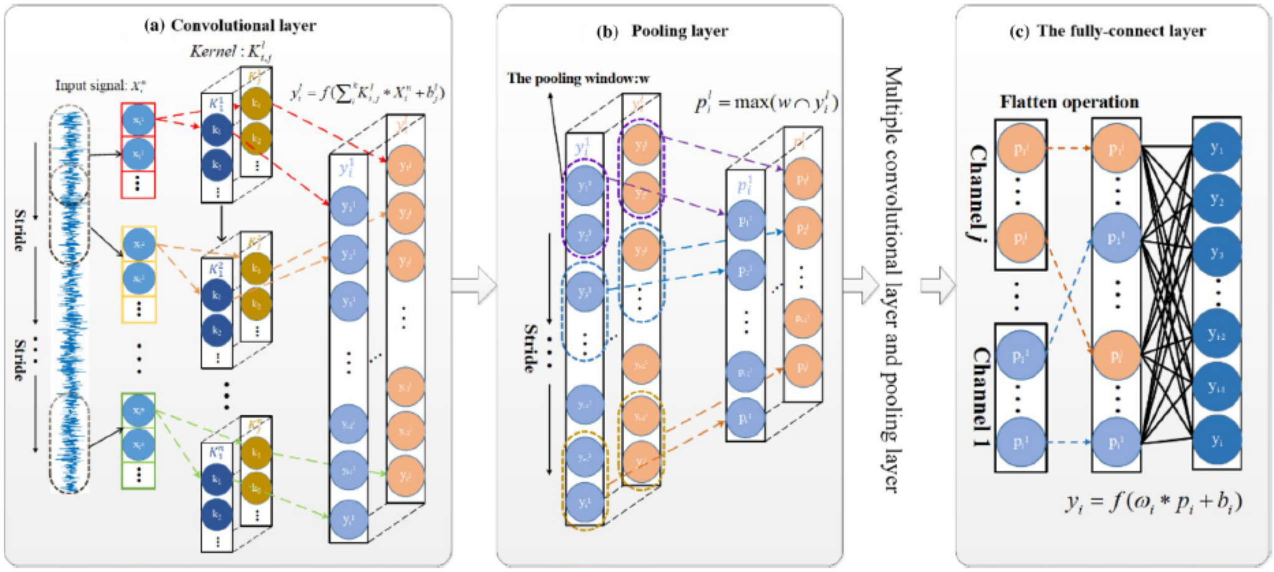

(a) Convolutional layer
Kernel : $K^l_{i,j}$
Input signal: X^n_i
$y^l_i = f(\sum^n_i K^l_{i,j} * X^n_i + b^l_j)$
Stride
Stride
(b) Pooling layer
The pooling window:w
$p^l_i = \max(w \cap y^l_i)$
Stride
Multiple convolutional layer and pooling layer
(c) The fully-connect layer
Flatten operation
Channel j
Channel 1
$y_i = f(\omega_i * p_i + b_i)$

Figure 4.7: A schematic overview of a one dimensional CNN [29]

Versions of the CNN have been successfully used to predict the RUL of a bearing. For example, a CNN without pooling layers was used to predict the RUL of a bearing in the IMS dataset (from the University of Cincinnati) [28]. Another approach was to extract similarity features from the vibration data and predict the RUL using a CNN based on these features [29].

Both approaches first extract and normalise a number of features from the time and frequency domain representations of the data. Normalisation is done according to equation 4.18. The frequency domain representation is obtained by the Fast Fourier Transform (FFT) [28]. The time domain features are the features as depicted in table 2.1. However, the approach of [29] also extracted features from the time-frequency domain, such that more complex nonlinear degradation characteristics could be captured. The time-frequency domain representation was obtained by performing a three level wavelet packet transform by making use of the Haar wavelet. As a result, eight frequency bands were obtained from which the energy ratios of these band could be calculated as time-frequency domain features. Additionally, this approach combines the time domain features into one feature by similarity fusion. Similarity fusion is calculated by taking the absolute value of Pearson's correlation coefficient [29]. The quality of the remaining feature was then evaluated by the Cori-index, which is the average of the monotonicity and trendability of a feature. Only features with an index higher than 0.5 were used for the creation of a health indicator by PCA. Hence, the feature engineering method described in [29] is a more complex procedure compared to the method described in [28].

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{4.18}$$

The approach described in [28] fed their features into a CNN, without pooling layers, in order to predict the RMS and to derive the RUL. The approach described in [29] fed the health indicator into the CNN to predict the RUL directly. Both models showed some accurate RUL prediction results, as shown in figure 4.8. For training their model, the authors of [28] used a different approach, namely stratified sampling. The input data was divided into groups of five, the first four points are used for training and the last data point for testing. Both methods outperformed a LSTM model on the IMS dataset.

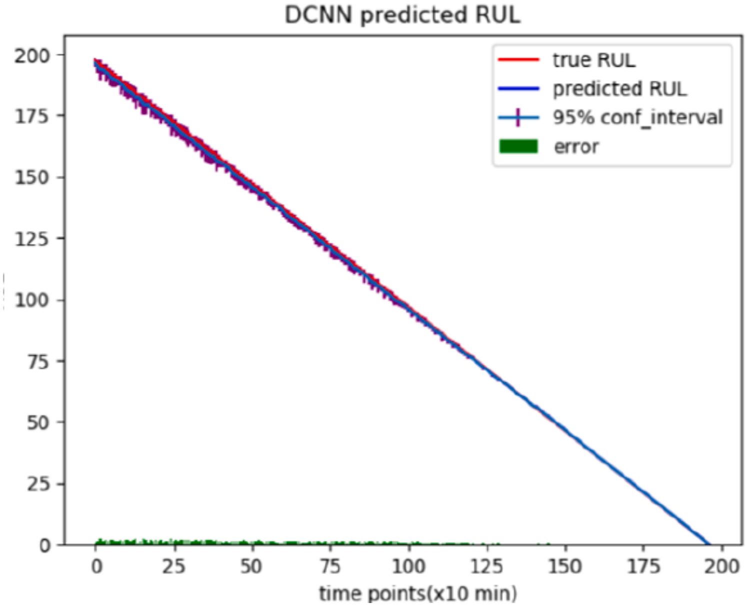


Figure 4.8: An example of the prediction results obtained by a CNN without pooling layers on the IMS data set [28]

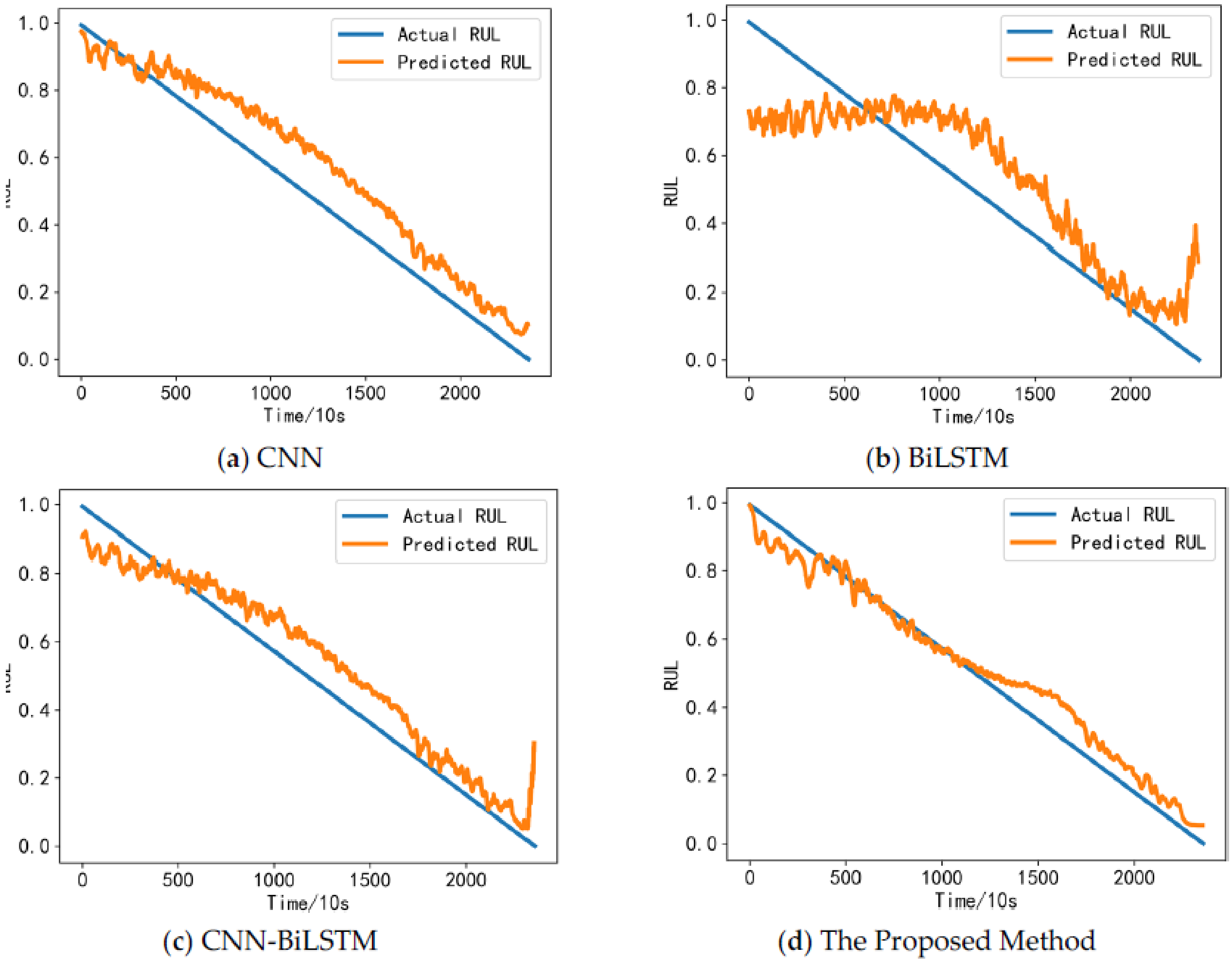
The results of RUL prediction methods using a CNN, as shown in figure 4.8 indicate that CNNs are also capable of accurate RUL prediction. This may be due to good feature extraction capability of the CNN, due convolution operations [28].

4.4. Remaining Useful Life Prediction of Bearings by a CNN-LSTM Network

Combining a CNN for feature extraction with a LSTM for RUL prediction makes use of the strengths of both types of networks. While CNNs are capable of extracting features from data through their convolutional layers, they may neglect temporal information related to bearing degradation [42]. Contrarily, LSTMs are able to establish a relationship between time and degradation since they use information from previous time steps to generate a RUL prediction for the next time step [50]. In the case of bidirectional LSTMs (BiLSTM), even the future degradation status of a bearing is considered for RUL prediction. Compared to a regular LSTM, a BiLSTM may show increased accuracy for predicting bearing degradation. However, a CNN outperforms a LSTM in feature extraction [43].

Another approach to bearing RUL prediction involves the use of an MBCNN-BiLSTM network [43]. This method uses a multi-branch CNN combined with a global attention mechanism for spatial feature extraction. The multi-branch CNN processes both time domain and frequency domain vibration signals as separate inputs. The frequency domain signal is obtained by using the FFT. By employing the attention mechanism, the network can emphasize relevant information while reducing focus on less relevant information. Subsequently, the output of the multi-branch CNN is input into BiLSTM. By making use of a BiLSTM for RUL prediction, the model is also able capture temporal patterns.

To demonstrate their approach, the authors used the Pronostia data set. They compared their results to other models, such as a CNN, BiLSTM, and CNN-BiLSTM. Their results are displayed in figure 4.9. It can be observed that the BiLSTM shows the least accurate results. Although the CNN generally followed the true RUL, there is a risk of overestimating the RUL. The strength of a CNN-LSTM combination can also be observed, with the CNN-BiLSTM outperforming the other models. However, the study only examined one operating condition. In reality, a bearing operates in multiple operating conditions. This complicates feature extraction, as the vibration data distributions will be different. Consequently, the authors recommend using transfer learning to address this problem [43]. An advantage of this approach is that manual feature extraction is not required, as feature extraction is done by a CNN, and this approach is thus less time consuming.



(a) CNN

(b) BiLSTM

(c) CNN-BiLSTM

(d) The Proposed Method

Figure 4.9: A comparison of the results obtained by different models on a bearing in the Pronostia data set. The 'proposed method' is the MBCNN-BiLSTM network [43]

The automated RUL prediction method was also used in [54] and [55]. In both methods, a CNN was used to extract features from the input data. The features were then passed to specific variants of the RNN in order to extract temporal features and to output a RUL prediction. A difference between these methods and the method described in [43] is that in these two methods features are extracted from only one domain. In [54] and [55], features are extracted from the raw time domain vibration signal and the frequency spectrum respectively. Additionally, the method in [54] also smooths the final RUL prediction results by a weighted moving average. Another difference is that in [55], the features extracted by the CNN are first assembled into a dataset, normalised, and split into sequences and then fed into the RNN compared to a direct connection between the CNN and RNN described in [54]. The methods also showed some accurate RUL prediction results, demonstrating the use of automated RUL prediction methods.

5

Remaining Useful Life Prediction using Transfer Learning

In chapter 4, several methods for RUL predictions of bearings were introduced. However, the data used for these methods always included run-to-failure data of bearings in a certain operating condition. In reality, a bearing operates in different operating conditions. Additionally, run-to-failure data is hard to collect from bearings as bearing failure may lead to catastrophic consequences. Transfer learning is a possible solution to these challenges [17]. Therefore, this chapter introduces the transfer learning concept in section 5.1. Additionally, some RUL prediction methods involving transfer learning will be explained in section 5.2. In section 5.2.1, some single-source domain adaptation approaches will be discussed. From the single-source adaptation methods it was observed that if the source domain contains more operating conditions or fault modes, the prediction results may be more accurate. Therefore, multi-source domain adaptation methods have been developed, which will be described in section 5.2.2.

5.1. The Concept of Transfer Learning

The methods described in chapter 4 have achieved accurate RUL prediction results. These methods use bearing vibration data collected under a certain operating condition to train a RUL prediction model, which is then tested using data from the same operating condition. Such an approach ensures that the assumption of training and testing data coming from similar distributions is met [14]. In practical scenarios, vibration data is collected under various operating conditions and bearings may fail in different failure modes. If the testing operating condition is different from the training operating condition or the bearing failure behaviour differs, then the assumption is not satisfied and the RUL prediction model may show inaccurate prediction results [13]–[15]. Hence, a model trained on data from a certain working condition will not perform well on data measured in another working condition. Additionally, in real world systems, most monitoring data is collected during the normal operation of the bearing, as letting a bearing fail on purpose comes with cost and safety concerns. Therefore, bearing run-to-failure operating data in a certain operating condition may be scarce [17].

A solution to this problem is to construct more generalised RUL prediction models capable of adapting to different operating conditions. Hence, the concept of transfer learning was introduced. Transfer learning aims to transfer bearing degradation knowledge from a source domain to a target domain [13]. In this case, the source and target domain data are collected from different operating conditions, such that both domains follow different distributions. Then, these different distributions can be aligned with each other by transfer learning techniques. By aligning the distributions, the model can learn domain invariant features and generate RUL predictions based on these domain invariant features [53]. Such a transfer learning approach is also able to work without labelled target domain samples. In this case, the model learns a relationship between the domain invariant features and degradation based on labelled source domain samples and the unlabelled target domain samples are used to align the distributions of the source and target domain features [13]. This is particularly useful if labelled bearing operating data in a certain operating condition is scarce. The precondition for these approaches is that all bearings show a similar degradation from healthy state to failure. Under different working conditions, the degradation process is similar [13].

For transfer learning, several mathematical notations exist. Firstly, there are two domains: the source domain and the target domain. The source domain provides knowledge about bearing degradation necessary for RUL prediction. The source domain is denoted by D^S. D^S consists of a feature space χ^S and a marginal probability distribution $P(\chi^S)$. Hence, the source domain can be described as $D^S = \{\chi^S, P(\chi^S)\}$. The bearing run-to-failure data can be denoted by $\{x_i^S\}_{i=1}^{n_S}$. Here, n_S is the number of samples and x_i^S is from the feature space χ^S. The RUL labels are contained within the label space γ^S. The target domain can be described in a similar manner. The target domain contains data from the target bearing and does not necessarily include run-to-failure data. The target domain is thus described as $D^T = \{\chi^T, P(\chi^T)\}$. Here, $\{x_i^T\}_{i=1}^{n_T}$ is the monitoring data from the target bearing. n_T is the number of samples in the target domain and x_i^T is from the feature space χ^T. For transfer learning, the working conditions of the source and target domains are usually different, hence $P(\chi^S) \neq P(\chi^T)$ [13]. An example of these different distributions can be observed in figure 5.1. In this figure, a high dimensional feature set was visualised using t-SNE technology [56]. In figure A, no transfer learning has been applied and it can be observed that the feature distributions do not overlap. In figure B, transfer learning has been applied, such that the distributions do overlap.

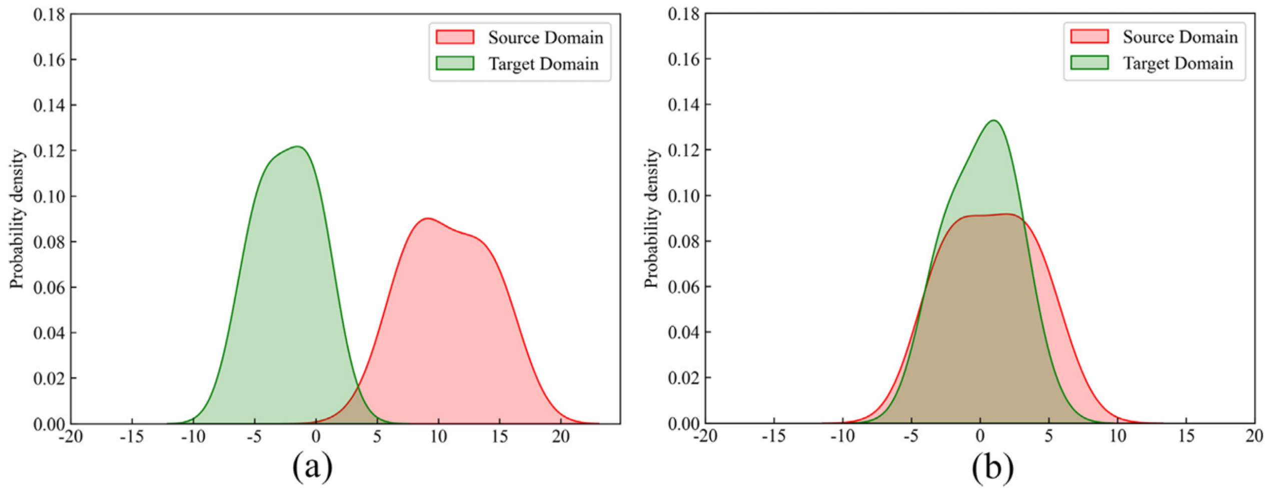


Figure 5.1: A visualisation of the distribution discrepancy before (a) and after (b) domain adaptation [56]

In RUL prediction, a model is constructed that derives a relationship between χ^S and γ^S (the RUL). Transfer learning aims to use the degradation information contained in this model, constructed from source domain data, to improve the performance of the RUL predictions in the target domain [17].

For transfer learning, several techniques are available. The first transfer learning technique is model-based. In this case, it is assumed that the models for the source and target domains have common parameters or hyperparameter distributions [57]. The source and target domains are related, such that a model trained on the source domain can be applied to the target domain as well. This is done by first training a prediction model on the source domain. The weights and biases of the hidden layers of this source domain model are then also used as initial parameters for the target domain hidden layer. The fully connected layer of the target domain model has to learn the weights and biases from scratch. Finally, the model parameters are then updated using the target domain data. This allows to train a model for a certain target domain using a relatively small amount of target domain data. For this transfer learning technique, it is required that the source and target domains are similar [13]. If there is a considerable domain shift, than this technique is not effective [17].

Another technique is to use domain adaptation techniques, the process can be observed in figure 5.2. Domain adaptation is used to minimise the discrepancy between the source and target domain feature distributions [17]. After the two distributions are aligned, the RUL prediction model trained on the source domain can also be used for the target domain. Aligning these distributions can be done by minimising the Maximum Mean Discrepancy (MMD). The MMD is measure for the similarity of two distributions. If the MMD is zero, then the two distributions are the same. The MMD can be obtained by obtained by mapping data points from distributions $P(\chi^S)$ and $P(\chi^T)$ into a Reproducing Kernel Hilbert Space (RKHS) and calculating the distance between the means of these embeddings of $P(\chi^S)$ and $P(\chi^T)$ in RKHS [58]. The domain discrepancy (d_{MMD}) between a data sample from source domain S and the target domain T is expressed in Equation 5.1 [59].

$$d_{\text{MMD}} = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \phi\left(x_i^S\right) - \frac{1}{T} \sum_{j=1}^{n_T} \phi\left(x_j^T\right) \right\|_{\mathscr{H}}^2 \tag{5.1}$$

In this equation, n_S and n_T are the number of data points in the source and target domain (domains S and T) respectively. Function $\phi(\cdot)$ maps the data (x) into the RKHS, denoted by $\mathscr{H}$ [59]. Equation 5.1 essentially calculates the Euclidean Distance between two points, the mean embeddings of the distributions.

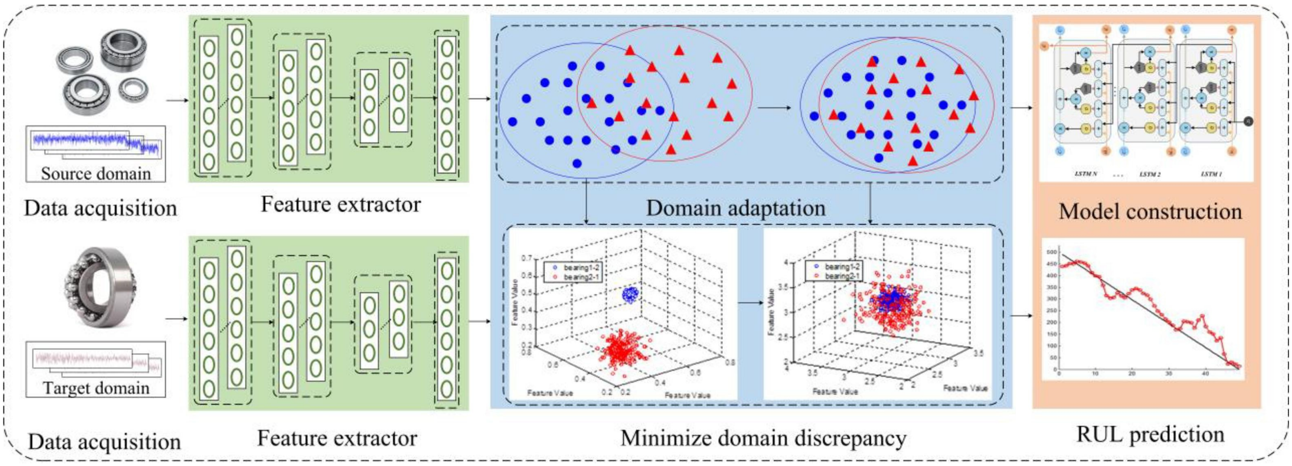


Figure 5.2: A domain adaptation based transfer learning method [13]

A different technique is to use DANNs. The objective of a DANN is to make predictions based on features. However, the feature should not contain any information about the domain (source or target domain) it originated from. This allows the model to make predictions based on domain invariant features. In other words, the domain does not have influence on the model prediction [60]. A schematic overview of a DANN can be observed in figure 5.3 [13]. In this figure, three elements can be observed. These are the feature extractor ($G_f(\cdot;\theta_f)$, the label predictor ($G_y(\cdot;\theta_y)$, and the domain classifier ($G_d(\cdot;\theta_d)$. The feature extractor is used for extracting features from the source and target domains. The label predictor uses the extracted features for RUL predictions. The domain classifier identifies the domain the feature originates from. The DANN is trained by optimising parameters θ_f, θ_y, and θ_d. The domain classifier parameters are trained such that the classification error is minimal. Hence, the domain classifier should classify data from the source domain as source domain data and classify data coming from the target domain as target domain data. The feature extractor is optimised such that the label predictor loss is minimised and the domain classifier loss is maximised. Hence, the domain classifier tries to make correct domain classifications, whereas the feature extractor tries to extract features that 'fool' the domain classifier [61].

The objective function for training a DANN is as follows [60]:

$$E\left(\theta_f,\theta_y,\theta_d\right) = \sum_{x_i\in D^S} L_y\left(G_y\left(G_f\left(x_i\right)\right),y_i\right) - \lambda \sum_{x_i\in D^S\cup D^T} L_d\left(G_d\left(G_f\left(x_i\right)\right),d_i\right) \tag{5.2}$$

Here, L_y is the loss of the label predictor $G_y(\cdot;\theta)$ and L_d is the loss of the domain classifier $G_d(\cdot;\theta_d$. λ represents the trade-off factor between the predictor loss and the domain classifier loss [62]. d_i is the domain label. This label equals zero if the domain is the source domain and one if the feature is coming from the target domain. The objective function is optimised by finding a $\hat{\theta}_f$, $\hat{\theta}_y$, and $\hat{\theta}_d$ such that:

$$\left(\hat{\theta}_d\right) = \underset{\theta_d}{\operatorname{argmax}} E\left(\hat{\theta}_f,\hat{\theta}_y,\theta_d\right) \tag{5.3}$$

$$\left(\widehat{\theta}_f,\widehat{\theta}_y\right) = \underset{\theta_f,\theta_y}{\operatorname{argmin}} E\left(\theta_f,\theta_y,\hat{\theta}_d\right) \tag{5.4}$$

An optimised $\hat{\theta}_d$ minimises the domain classifier loss. An optimal $\hat{\theta}_y$ minimises the predictor loss. If $\hat{\theta}_f$ is optimised, then the predictor loss is minimised and the domain classifier loss is maximised [60]. A benefit of the DANN technique is that this method also works with unlabelled target domain data.

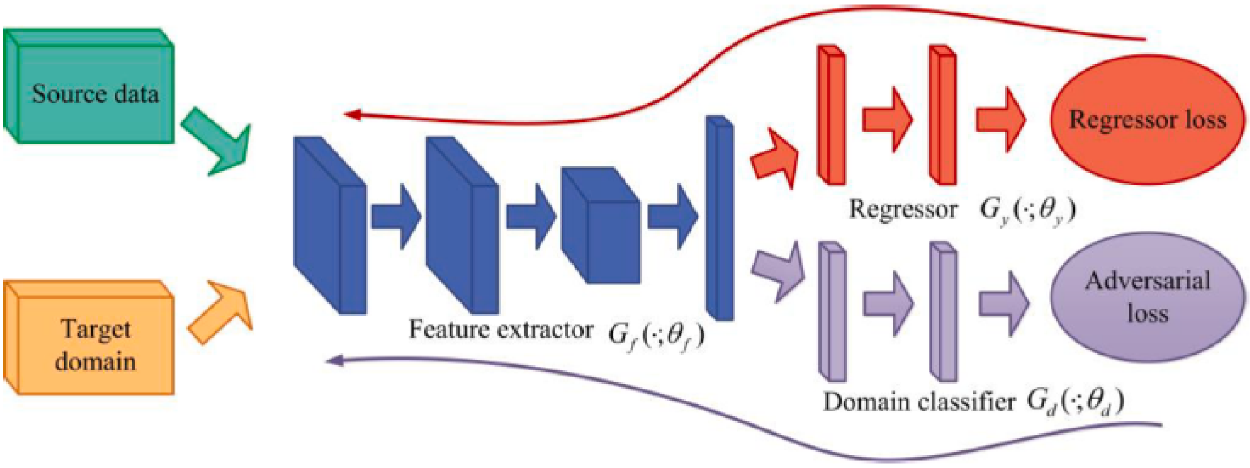

Source data
Target domain
Feature extractor $G_f(\cdot;\theta_f)$
Regressor loss
Regressor $G_y(\cdot;\theta_y)$
Adversarial loss
Domain classifier $G_d(\cdot;\theta_d)$

Figure 5.3: A schematic overview of a DANN [13]

In practice, there are a few possible scenarios for transfer learning. These are as follows [13]:

1. RUL estimation with run-to-failure data in the target domain. Here, the target domain contains labelled data. This data can be used to construct a prediction model for the target bearing. This way, the generalisation of the prediction model is improved.
2. RUL estimation without run-to-failure data in the target domain. In this case, the target domain does not contain labelled data. Degradation information of the bearings is obtained from the source domain and applied to the target domain.
3. RUL prediction with fault information. In this case, there should be information about the health status or fault types of the bearings. This will improve the accuracy of the RUL prediction.

5.2. Remaining Useful Life Prediction using Transfer Learning

The first transfer learning methods for RUL prediction were done using single source domain adaptation, i.e. one source domain and one target domain. More advanced transfer learning methods used multiple source domains and one target domain. These two approaches will be described in this section. Additionally, this part of the literature review will focus on RUL prediction without run-to-failure data in the target domain, as such data is not available in this study.

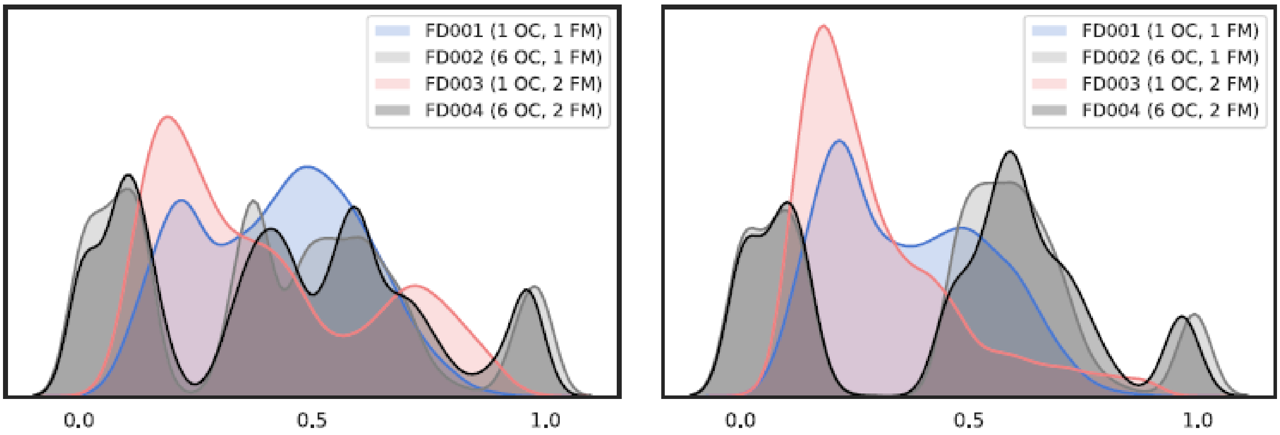
5.2.1. Remaining Useful Life Prediction by Single-Source Domain Adaptation

This subsection will focus on RUL prediction by transfer learning using adversarial and discrepancy-based methods.

Adversarial Methods for Remaining Useful Life Prediction

Predicting the RUL using a DANN was initially introduced by Costa et al. [63]. In this work, a LSTM-DANN RUL prediction model was constructed for the NASA CMAPSS data set. This data set contains four subsets. Each subset contains data from a number of sensors. The sensors recorded values in three different operating conditions. The sensor recordings relate to the degradation of initially healthy engines. In this work, the authors conducted experiments on a case where there was run-to-failure data in a set of fault modes and operating conditions and no run-to-failure data in a different domain.

As the sensor values are recorded in different operating conditions and in different fault modes, the data subsets do not follow the same distribution. Figure 5.4 shows the different distributions of data. Note that FD001 - FD004 denote the different subsets of data. It can be observed that indeed, there exists a domain shift between the subsets. However, Between pairs FD001-FD003 and FD002-FD004 there exists a smaller domain shift. The data belonging to these pairs are recorded in the same operating conditions. The existing discrepancies are due to the different fault modes. Due to this domain shift, a RUL prediction model trained on one subset of data will not perform well on another subset of data [17].



(a) Sensor 15

(b) Sensor 7

Figure 5.4: Distributions of data one time step before a failure recorded by two different sensors [63]

An overview of the LSTM-DANN network can be observed in figure 5.5. It can be observed that the feature extractor is an LSTM network. An LSTM is selected as such a network is suitable for capturing long-term temporal dependencies in the data [63]. The feature extractor is connected to a fully connected layer. This common part of the network is then connected to the domain classifier and the RUL label predictor. The model also contains two learning rates, namely λ_d and λ_y. These values are selected by performing a grid search. The source domain, which includes RUL labels, is used for training the label predictor. Source and target domain data is used for training the domain classifier. In this model, the feature extractor weights, θ_f, and label predictor weights, θ_y, are optimised towards minimising the regression loss. This loss is the difference between the true RUL and the predicted RUL. For the domain classification network, the gradient reversal layer is used between the fully connected layers of the LSTM network and domain classifier. This reversal layer is needed for the feature extractor to learn domain invariant features. The feature extractor weights, θ_f, are optimised such that the prediction loss for the source domain is minimised in one pass. Then, in a second pass, the feature extractor weights are updated such that the domain classification loss is maximised. This is done iteratively in a dual optimisation procedure. This procedure stops once no further improvement is observed. Before training the LSTM-DANN model, the input data is normalised by min-max normalisation in order to aid the optimisation procedure.

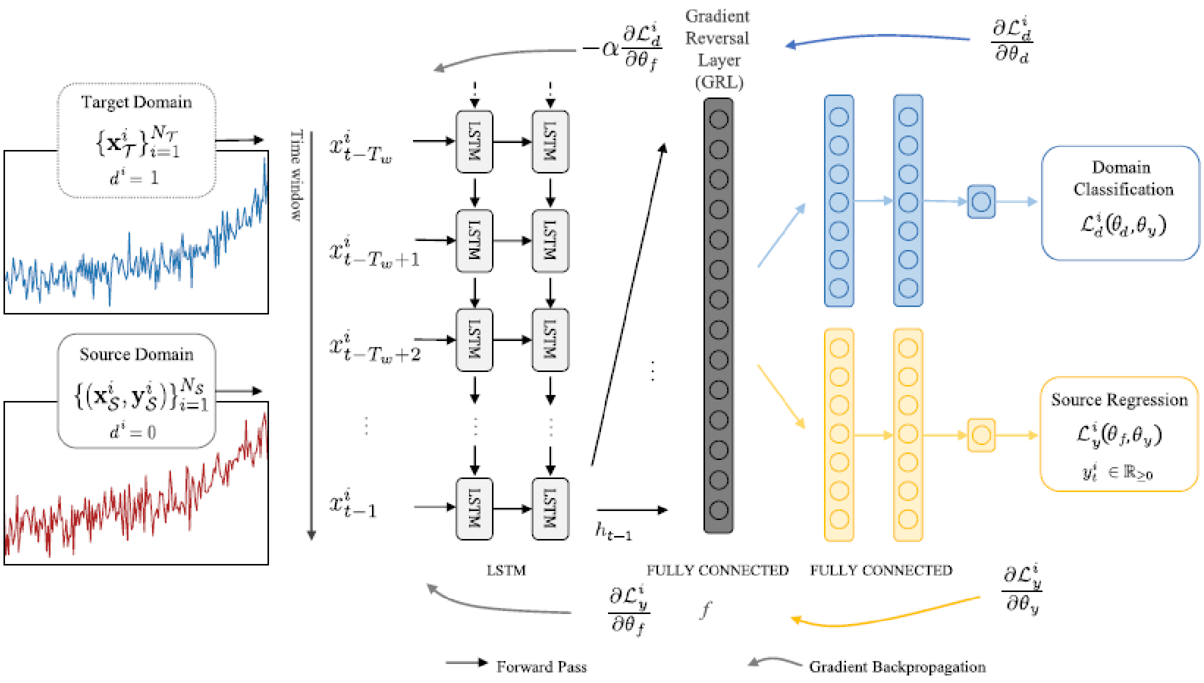
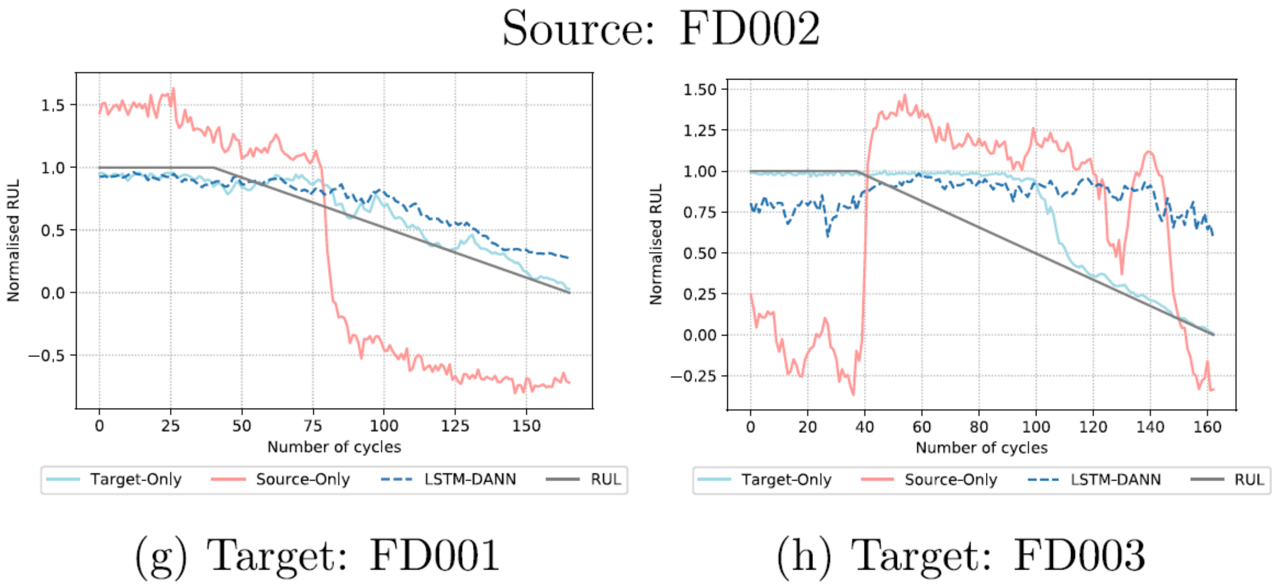

$-\alpha \frac{\partial \mathcal{L}_d^i}{\partial \theta_f}$
Gradient Reversal Layer (GRL)
$\frac{\partial \mathcal{L}_d^i}{\partial \theta_d}$
Target Domain
$\{\mathbf{x}_T^i\}_{i=1}^{N_T}$
$d^i = 1$
Source Domain
$\{(\mathbf{x}_S^i, \mathbf{y}_S^i)\}_{i=1}^{N_S}$
$d^i = 0$
Time window
$x_{t-T_w}^i$
$x_{t-T_w+1}^i$
$x_{t-T_w+2}^i$
x_{t-1}^i
LSTM
h_{t-1}
Domain Classification
$\mathcal{L}_d^i(\theta_d, \theta_y)$
Source Regression
$\mathcal{L}_y^i(\theta_f, \theta_y)$
$y_t^i \in \mathbb{R}_{\geq 0}$
LSTM
FULLY CONNECTED
FULLY CONNECTED
$\frac{\partial \mathcal{L}_y^i}{\partial \theta_f}$
f
$\frac{\partial \mathcal{L}_y^i}{\partial \theta_y}$
Forward Pass
Gradient Backpropagation

Figure 5.5: The LSTM-DANN network for RUL prediction [63]

The authors compare the performance of the LSTM-DANN model to a source-only model and a target-only model.

In the source-only model, a LSTM model is trained on source domain data. Then, this non-adapted model is directly applied to a different data subset, the target. The target-only model is trained directly on the target data. In the latter case, RUL labels are known. An example of the results can be observed in figure 5.6.



(g) Target: FD001

(h) Target: FD003

Figure 5.6: An example of the RUL prediction results obtained by the LSTM-DANN model [63]

In this figure, it can be observed that a DANN model can perform well if the source and target domain share characteristics, such as the same working conditions or similar fault modes. Additionally, if the source domain contains data from multiple operating conditions or fault modes, then the predictions for the target domain will be more accurate as well. In figure 5.6, the source domain contained one fault mode and six operating conditions. As FD001 also contains one fault mode and one operating condition, the prediction results of the model are accurate. On the other hand, FD003 contains two fault modes. Hence, the prediction results on FD003 are less accurate [63]. The authors also showed that the DANN network for domain adaptation performs better than using metrics such as the MMD in their case.

Siahpour et al. build upon this DANN approach with their two-stage RUL prediction model [62]. In their approach, the authors assume that the target domain does not contain severe degradation data, unlike the source domain. In the first stage, a RUL prediction model is trained based on labelled source domain data by minimising the prediction loss, without any transfer learning. The objective of this stage was to establish a relation between the input data and the RUL label. The features extracted by this pretrained model are used as a reference for the second stage. In the second stage, a DANN model is used for achieving cross operating condition RUL predictions. In this approach, the target domain does not contain run-to-failure data, leading to a negative transfer effect that reduces model accuracy [13]. To mitigate this negative transfer effect, the consistency based regularisation term is used. This term ensures that the source domain features extracted by the DANN are similar to the pretrained features, extracted in the first stage of the model. For generating the results, the authors used the bearing data set from Xi'an Jiaotong University and the CMAPSS data set. Results obtained from the two stage model were also compared with a baseline model, a DANN without consistency-regularisation, and a model involving the MMD metric. The baseline model is similar to the feature extractor and label predictor of the two stage network. However, in this case the objective is to minimise the prediction loss only. Examples of the results can be observed in figure 5.7. Note that task B13 means the transfer of operating condition one to operating condition three for example [62].

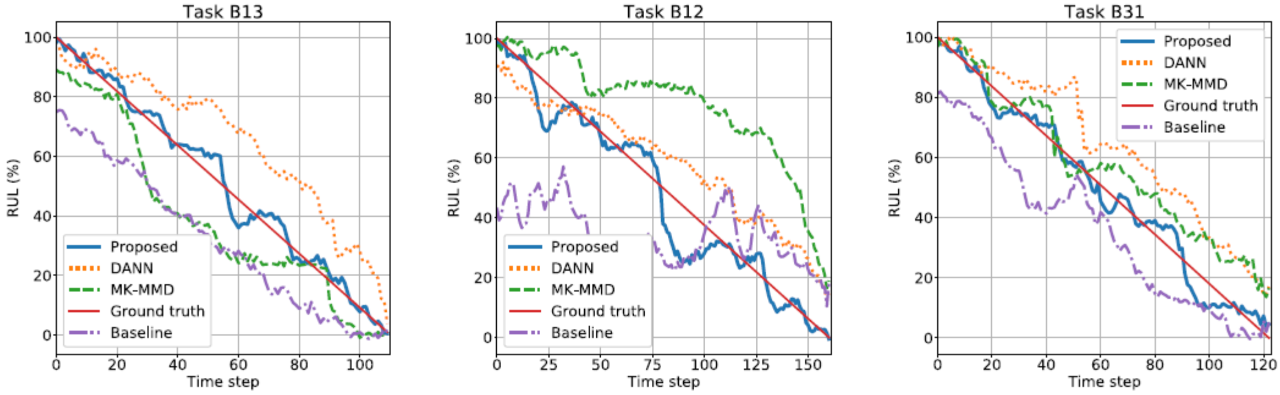


Figure 5.7: An example of the results obtained by Siahpour et al. on the Xi'an Jiaotong University bearing data set [62]

On both the bearing data set and the CMAPSS data set, the proposed DANN with consistency-based regularisation produced results superior to the baseline model, regular DANN model, and the MMD metric model. Therefore, this regularisation term helps to reduce the negative effect of incomplete data in the target domain and to increase the accuracy of the model [13]. The regularisation aligns the missing data from the severe degradation stage in the target domain to the pretrained source domain features [62]. Additionally, the method involving the regularisation term performs more accurately on the CMAPSS data set than the LSTM-DANN model proposed by Costa et al. [63], which shows the benefits of using the consistency-based regularisation term.

Discrepancy based Methods for Remaining Useful Life Prediction

Minimising the MMD has also demonstrated its use in achieving RUL predictions across operating conditions and failure modes. This subsection will highlight some of the methods.

One method is the transferable convolutional neural network (TCNN), which was used for generating RUL predictions across bearing failure modes [53]. If a bearing fails in different failure modes, then the data distributions may still differ from each other, even if the operating condition is the same. Therefore, in this method, the source domain and target domain are different bearing failure behaviours. This TCNN consisted of a CNN and multiple fully-connected layers for feature extraction from a sample in the source and target domain and a domain adaptation module for aligning the feature distributions from the source and target domain, the model architecture is shown in figure 5.8 [53].

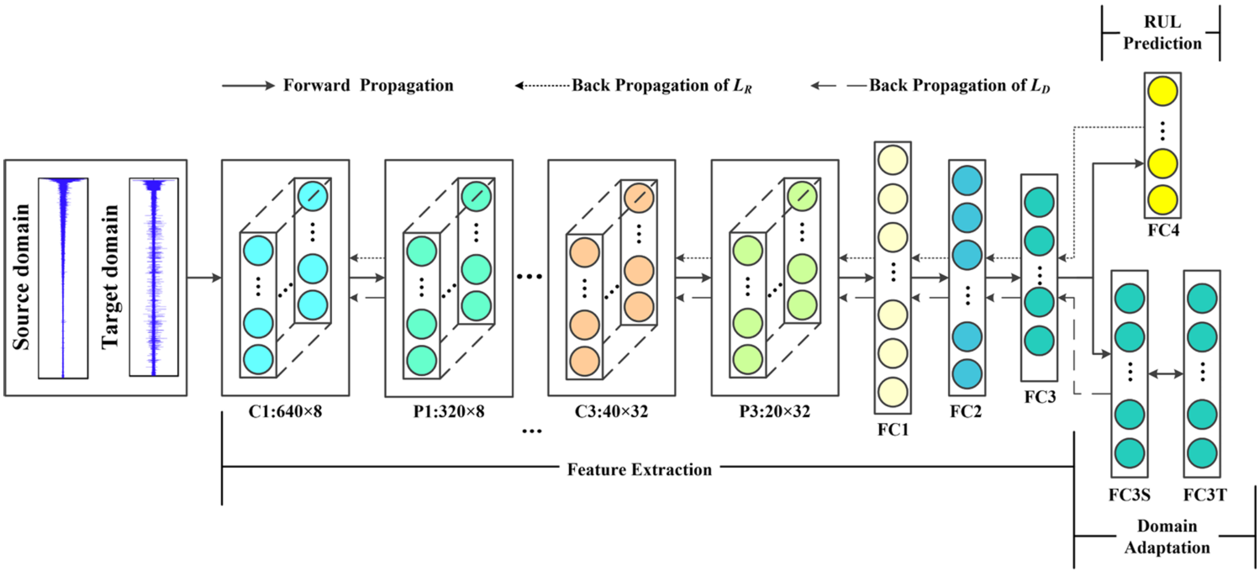


Figure 5.8: The TCNN proposed by Cheng et al. for achieving RUL predictions across different failure behaviours [53]

The TCNN method consists of multiple steps. First, the time domain samples are converted into frequency spectra using the FFT. Next, frequency domain samples from the source and target domains pass through the feature extractor simultaneously. As a result, feature representations of the source and target domain samples are obtained. The source domain features pass through the RUL predictor, another fully-connected layer, in order to generate a RUL prediction for the source domain samples. As the source domain is labelled, the prediction loss can be calculated by comparing the actual source label and the predicted source label. Additionally, the source and target domain features pass through the domain adaptation module. This module aligns the distributions of these two feature representations by minimising the MMD loss, as expressed in equation 5.1. Therefore, the objective function of TCNN consists of two components, namely the prediction loss and MMD loss. Minimsing the prediction loss ensures that the model learns a relation between the features and the bearing RUL and minimising MMD loss ensures that domain invariant features are used for RUL predictions [53].

In their case study, the authors divided bearing vibration data from one operating condition in two failure behaviour groups, the source and target domain [53]. This method achieved accurate RUL prediction results, as can be observed in figure 5.9. The downsampled RUL predictions do follow actual RUL trend. Additionally, the TCNN outperformed methods involving a DANN in terms of mean absolute error and root mean square error [53].

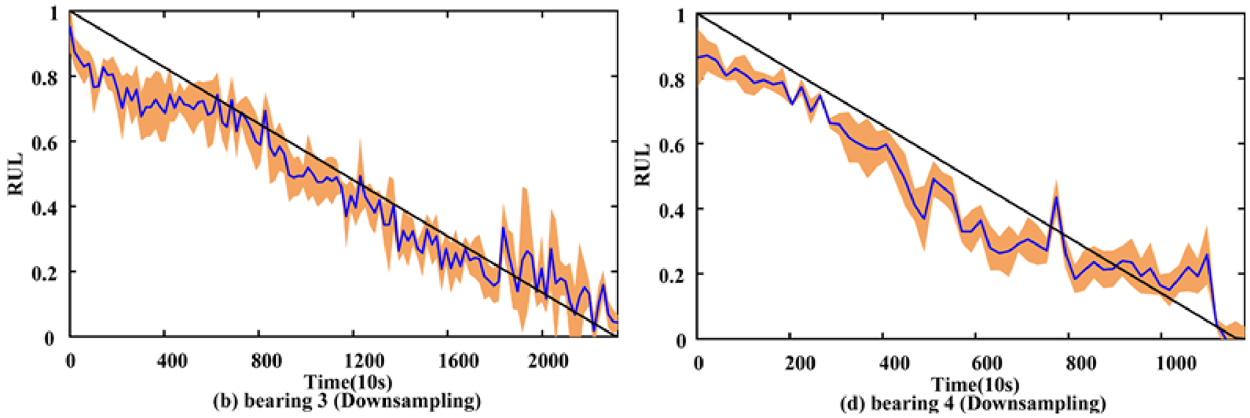


Figure 5.9: The prediction results for bearings 3 and 4 of operating condition 1 of the Pronostia dataset using TCNN [53]

A similar network to TCNN was used for realising cross operating condition RUL predictions, namely the transferable BiLSTM [64]. In this method, different operating conditions were used as source and target domain data. First, features were extracted from the time, frequency, and time-frequency domain. Next, a BiLSTM was used for further feature extraction. The source domain features were passed through a RUL predictor, such that a prediction loss is obtained. The source and target domain features are also passed through the domain adaptation module, which uses the same principle as described in [53]. The authors of [64] performed a case study on transferring between operating conditions in the Pronostia bearing dataset. In this work, some accurate RUL predictions were achieved, though the accuracy varied depending on the operating condition used as source domain data.

5.2.2. Remaining Useful Life Prediction by Multi-Source Domain Adaptation

The aforementioned studies on RUL prediction of bearings by transfer learning have one aspect in common. All these studies use a single domain, or operating condition, as source data. Using more operating conditions as source domain data may enhance model performance, as concluded by Costa et al. [63]. Therefore, a transfer learning approach to RUL prediction by using source domain data from multiple different operating conditions has been proposed by several scholars in order to make use of the bearing degradation information encapsulated in these different source domains [48], [58]. These different source domains cannot be combined into one source domain, due to the distribution discrepancies among the different source domains [58]. Therefore, multi-source domain adaptation models have been developed.

Multi-source domain adaptation involves different mathematical notations compared to single-source domain adaptation. In multi-source domain adaptation, it has been assumed that there are N run-to-failure source domain data sets, denoted by $\{X^{s_j}, Y^{s_j}\}_{j=1}^{N}$. s_j denotes the j^{th} source domain data set. $X^{s_j} = \left\{x_i^{s_j}\right\}_{i=1}^{n_{s_j}}$, where n_{s_j} is the number of samples in source domain data set s_j. The RUL labels are represented by $Y^{s_j} = \left\{y_i^{s_j}\right\}_{i=1}^{n_{s_j}}$. Additionally, there is an unlabelled target domain data set $X^t = \{x_i^t\}_{i=1}^{n_t}$. Here, n_t denotes the number of samples in the target domain [58]. Additionally, the condition that the bearings show a similar degradation trend also holds in this MDAN model [13].

Ding et al. proposed the MDAN model [58], which is shown in figure 5.10. In this network, domain invariant features are learned from multiple sources to provide a generalised model, which enhances accuracy of the RUL predictions [63].

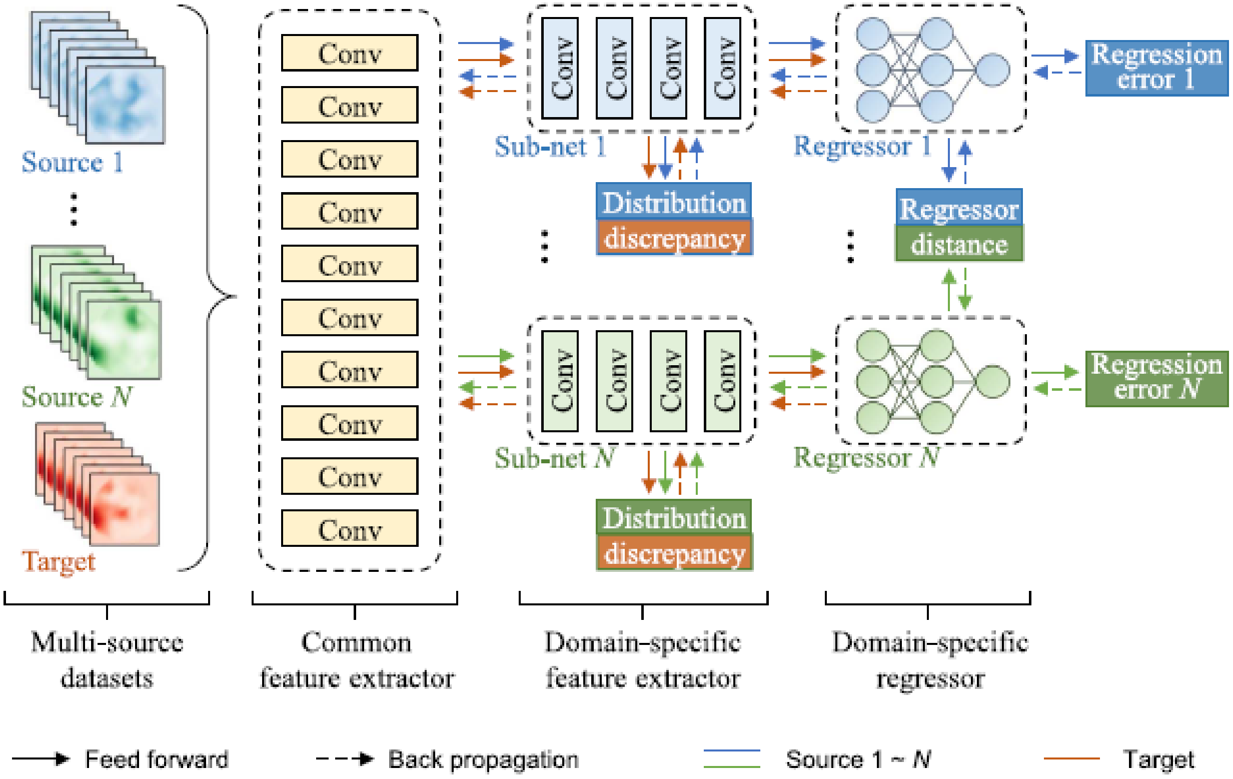


Figure 5.10: A schematic overview of the MDAN proposed by Ding et al. [58]

The architecture of the MDAN model proposed by Ding et al. can be observed in figure 5.10 [58]. The MDAN model consists of a common feature extractor, domain-specific feature extractors, and domain-specific regressors. Any input data x_i, whether source or target domain data, passes through the common feature extractor first. This results in a common feature representation $F(x_i)$. If the sample is from a source domain j, $x_i^{s_j}$, then the data subsequently passes through the domain-specific feature extractor. This way, a domain-specific feature representation $H_j(F(x_i^{s_j}))$ is obtained. Afterwards, the domain specific features pass through the domain-specific regressor. This results in an estimated RUL label denoted by $\hat{y}_i^{s_j} = C_j(H_j(F(x_i^{s_j})))$. Hence, based on the i^{th} sample of input data in source domain s_j, a corresponding RUL label ($\hat{y}_i^{s_j}$) is predicted. The objective of MDAN is to minimise the MSE between the true RUL and the predicted RUL. Thus, the parameters of the common feature extractor, domain-specific feature extractor, and domain-specific regressor (θ_F, θ_{H_j}, and θ_{C_j} respectively) are optimised according to the following loss function:

$$L_{err}\left(\theta_F, \theta_{H_j}, \theta_{C_j}\right) = \frac{1}{\sum_{j=1}^{N} n_{s_j}} \sum_{j=1}^{N} \sum_{i=1}^{n_{s_j}} \left(y_i^{s_j} - \hat{y}_i^{s_j}\right)^2 \tag{5.5}$$

The domain adaptation is done in two stages. First, the distribution discrepancy between each pair of source and target domains is minimised. Minimisation of this discrepancy is done by making use of two metrics, namely the MMD and the CORAL loss between the source and target domains. The MMD is calculated in equation 5.1. The CORAL loss is defined as the difference between the covariances of the source and target domain samples [65]:

$$CORAL = \frac{1}{4d^2} \|C^s - C^t\|_F^2 \tag{5.6}$$

where d is the dimension of the source and target domain data. $\|\cdot\|_F^2$ represents the Frobenius norm of the squared matrix. C^s and C^t are calculated by:

$$C^s = \frac{1}{n_S - 1}\left(X^{s\top} X^s - \frac{1}{n_s}\left(\mathbf{1}^\top X^s\right)^\top \left(\mathbf{1}^\top X^s\right)\right) \tag{5.7}$$

$$C^t = \frac{1}{n_t - 1}\left(X^{t\top} X^t - \frac{1}{n_t}\left(\mathbf{1}^\top X^t\right)^\top \left(\mathbf{1}^\top X^t\right)\right) \tag{5.8}$$

Here, $\mathbf{1}$ is a column vector where all values equal to one.

In order to align the distributions, the MMD and CORAL loss are calculated between any source domain s_j and target domain t. The source domain data passes through the common feature extractor (F) and the domain-specific feature extractor (H_j). The data from the target domain also passes through the common and domain-specific feature extractors. The MMD loss and CORAL loss are defined as follows [58]:

$$L_{\text{MMD}} = \frac{1}{N}\sum_{j=1}^{N} \text{MMD}\left(H_j\left(F\left(X^{s_j}\right)\right), H_j\left(F\left(X^{t}\right)\right)\right) \quad (5.9) \qquad L_{\text{CORAL}} = \frac{1}{N}\sum_{j=1}^{N} \text{CORAL}\left(H_j\left(F\left(X^{s_j}\right)\right), H_j\left(F\left(X^{t}\right)\right)\right) \quad (5.10)$$

These two losses are combined into a total discrepancy loss as follows:

$$L_{dis} = L_{MMD} + \lambda' \cdot L_{CORAL} \tag{5.11}$$

Then, the next stage is the domain-specific regressor adaptation. If a target domain sample passes through all the different domain-specific feature extractors and regressors, then the output of the network varies significantly. This undesirable effect is mitigated by aligning these various outputs. Therefore, another loss term is introduced by the authors. This loss term is the distance between the outputs of two regressors. Therefore, for calculating the distance, every possible combination of two regressors is used. This distance is calculated as follows:

$$L_{reg} = \frac{1}{C(N,2)\cdot n_t}\sum_{j=1}^{N-1}\sum_{k=j+1}^{N}\sum_{i=1}^{n_t}(C_j(H_j(F(x_i^t))) - C_k(H_k(F(x_i^t))))^2 \tag{5.12}$$

In this equation, $C(N,2)$ represents the number of possible combinations, which equals $N(N-1)/2$.

For training the MDAN, an objective function is constructed by combining the three aforementioned loss terms. The parameters θ_F, θ_{H_j}, and θ_{C_j} are optimised such that the objective function is minimised. The objective function of MDAN is as follows:

$$J(\theta_F, \theta_{H_j}, \theta_{C_j}) = L_{err} + \gamma L_{dis} + \mu L_{reg} \tag{5.13}$$

Here, γ and μ are trade-off coefficients. After obtaining the parameters θ_F, θ_{H_j}, and θ_{C_j}, the MDAN model can estimate the RUL labels for an unseen test data set from the target domain, $X^{test} = \{x_i^{test}\}_{i=1}^{n_{test}}$. This is done by letting the sample pass through the common feature extractor and all the different domain-specific feature extractors and regressors. Hence, the RUL is the average of N predictions:

$$\hat{y}_i^{test} = \frac{1}{N}\sum_{j=1}^{N} C_j(H_j(F(x_i^{test}))) \tag{5.14}$$

Before training the MDAN model, the raw vibration signals are converted into a time-frequency representation by a wavelet transform. Additionally, the source domain data has to be labelled by the RUL. After this data pre-processing, the model can be constructed and trained. After training the model, the previously unseen test data can be fed through the model for RUL predictions. In this work, bearing data from Xi'an Jiaotong University is used. In this data set, the horizontal and vertical acceleration are measured. The authors decided to use the data from both sensors as input to the model. Therefore, for the domain-specific feature extractor, a two dimensional CNN is used to extract features from a time-frequency representation of the vibration data, consisting of three convolutional layers. The third convolutional layer is connected to a pooling layer, where average pooling is used.

An example of the RUL prediction results can be observed in figure 5.11 [58]. Here, Task E and Task F denote the transfer tasks. Task E means that operating conditions one and three are used as source domain data. Operating condition two is used as target domain data. Task F means that operating conditions two and three are used as source domain data and operating condition one is used as target domain data. It can be observed that the RUL prediction results follow the linear degradation trend of the bearing. Also in the severe degradation stages, the RUL prediction results are also close the true RUL. The authors also compared their results to other transfer learning approaches, such as the LSTM-DANN model approach described by Costa et al. [63]. In the LSTM-DANN model, the source domain providing the best prediction results was used. It was demonstrated that the MDAN model provided superior results to other models such as LSTM-DANN.

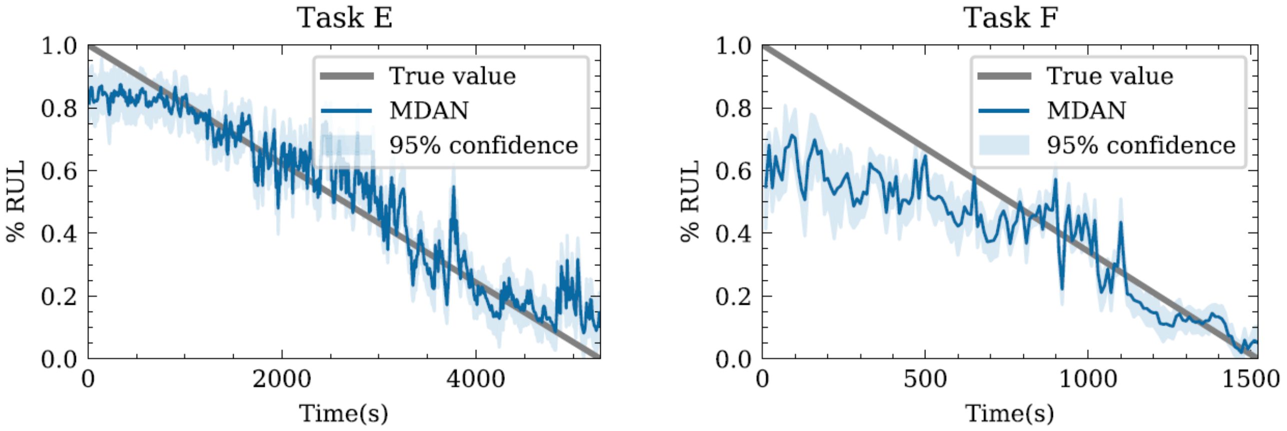
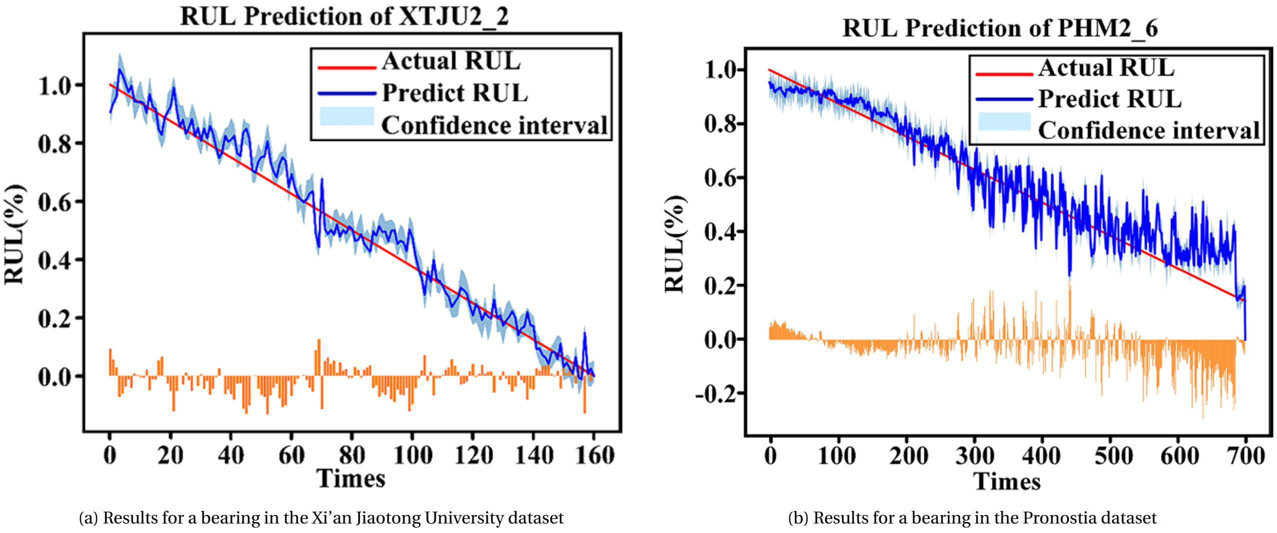


Figure 5.11: RUL prediction results obtained by the MDAN model on the Xi'an Jiaotong bearing data set [58]

Ding's MDAN model provided a basis for more advanced multi-source domain adaptation approaches. For example, the MDAN model was expanded with a domain classifier in [48]. This approach manually extracts features from the time and frequency domains. These features pass through the common feature extractor, domain-specific feature extractors, and domain-specific regressors in a manner similar to Ding's MDAN model. This expanded model also uses the prediction loss, domain adaptation loss, and regression output distance loss terms. However, this approach does not use the same domain adaptation method as Ding's model. In the expanded model [48], domain adaptation is achieved by minimising MMD loss between source and target domain features, as extracted by the domain-specific extractors. Additionally, these source and target domain features are passed to a domain classifier. As a result, this model incorporates an adversarial component into the training process, similar to DANN models [48].

The expanded model also demonstrated accurate RUL prediction results based on features from the time and frequency domains. Additionally, the transfer tasks in this model not only included cross operating condition scenarios, but also cross bearing type scenarios. Using multiple bearing types allows for generating RUL predictions for a bearing even if no failures have been recorded for that specific type. Additionally, the information on bearing degradation can be expanded by using multiple bearing types in the source domain. An example of the results for such a scenario can be observed in figure 5.12 [48]. These experiments indicate that a multi-source domain adaptation model structure as proposed by Ding et al. may be used for generating RUL predictions across operating conditions and bearing types.



(a) Results for a bearing in the Xi'an Jiaotong University dataset

(b) Results for a bearing in the Pronostia dataset

Figure 5.12: An example of the cross bearing type RUL prediction results [48]

6 Conclusion of the Literature Review

The objective of this literature review was to acquire knowledge about machine learning methods for data-driven RUL prediction for bearings, as well as the concept of transfer learning. Bearing vibration data is often used as the starting point for RUL predictions. Therefore, first, the vibration measurement system (HUMS) in helicopters was discussed, as such systems allow for the implementation of CBM [12]. In several studies, the bearing vibration data is processed by calculating various condition indicators, such as RMS and kurtosis. Next, the condition indicators are fed into a data-driven model for RUL prediction [25], [28], [46]. These data-driven models often include variants of the LSTM or CNN for further feature extraction and RUL prediction. Furthermore, it has been observed that the CNN is capable of automatic feature extraction from raw vibration data, a process that is less time consuming than manual feature extraction [54], [55].

However, the aforementioned models were trained and tested on the same working condition. In reality, bearing vibration data can be collected in different conditions, which causes a mismatch in the distributions of the data collected in different operating conditions. This leads to a deterioration in RUL prediction performance if a model is trained on data from a certain operating condition and is used to predict the RUL of a bearing in a different operating condition [13]. Additionally, run-to-failure data in a specific operating condition may be scarce, as letting a bearing fail on purpose may involve cost and safety concerns [13].

A solution to these challenges is transfer learning. This techniques aims to transfer bearing degradation knowledge from a source domain to a target domain. For example, the source domain may be a certain operating condition and the target domain is a different operating condition. For transfer learning, several techniques can be used such as fine-tuning an existing model, using distribution discrepancy metrics, and DANNs [13]. Additionally, transfer learning can be used for RUL estimation, even if there is no run-to-failure data in the target domain. In this case, bearing run-to-failure data is only needed in the source domain. Then, the bearing degradation knowledge included in the source domain can be transferred to an unlabelled target domain. Several studies have been performed on this RUL estimation scenario. Examples of such studies are Costa's DANN model [63] or the transferable CNN, a method that makes use of the MMD metric [53].

Costa et al. observed that if the source domain containing multiple operating conditions, then the RUL predictions tend to be more accurate [63]. Therefore, multi-source domain adaptation models have been developed. These models allow for the use of more labelled bearing degradation information for generating RUL predictions. Ding et al. proposed the MDAN model, which consists of a common feature extractor and domain-specific feature extractors and regressors [58]. In a recent study, this model has been expanded such that cross bearing type predictions are possible [48]. Such a model allows for generating RUL predictions for a bearing even if no failures have been recorded for that specific bearing type. Additionally, the information on bearing degradation can be expanded by using multiple bearing types in the source domain.

However, the multi-source domain adaptation models make use of vibration data in the time domain [48] or the time-frequency domain [58]. Some systems, such as helicopter drive trains, only record vibration data in the frequency domain. In this case, the model must provide RUL predictions based exclusively on frequency domain data. As a result, the model cannot use important degradation information provided in the time domain or time-frequency domain, complicating bearing RUL prediction [29], [48].

According to the literature review, no studies have yet been published on bearing RUL prediction across different bearing types using multi-source domain adaptation based solely on frequency domain data, highlighting a research opportunity for this study. If such a generalised RUL prediction model can be constructed, this model can be used a first step towards transfer learning approaches from run-to-failure data collected in a laboratory to a helicopter drive train. The run-to-failure data can provide extensive information on bearing degradation, as this data may be collected under various operating conditions and even across different bearing types. Such an approach would also avoid the need for large amounts of run-to-failure data from a real helicopter drive train bearing, which is difficult to collect [13].

Bibliography

[1] H. Liu, Z. Mo, H. Zhang, X. Zeng, J. Wang, and Q. Miao, "Investigation on Rolling Bearing Remaining Useful Life Prediction: A Review," in *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, 2018, pp. 979–984. DOI: 10.1109/PHM-Chongqing.2018.00175.

[2] AIBN, "Accident at Turoy, near Bergen, Norway on 29 April 2016, involving Airbus helicopters H225, LN-OJF," Accident Investigation Board Norway, Tech. Rep., 2016.

[3] F. Elasha and D. Mba, "Improving condition indicators for helicopter health and usage monitoring systems," *International Journal of Structural Integrity*, vol. 7, no. 4, pp. 584–595, Aug. 2016, ISSN: 17579872. DOI: 10.1108/IJSI-09-2015-0032.

[4] A. Mauricio, J. Qi, L. Zhou, W. Wang, D. Mba, and K. Gryllias, "Perspectives on Health and Usage Monitoring Systems (HUMS) of helicopters," *MATEC Web of Conferences*, vol. 314, p. 02 008, 2020. DOI: 10.1051/matecconf/202031402008.

[5] T. Zaman and A. Bayoumi, "Analysis of health and usage monitoring system (hums) users' perspective towards mission benefits using regression analysis," vol. 2, May 2014, pp. 1621–1626.

[6] P. J. Dempsey, J. A. Keller, and D. R. Wade, "Signal Detection Theory Applied to Helicopter Transmission Diagnostic Thresholds," Tech. Rep., 2008.

[7] R. Wang, H. Chen, and C. Guan, "A Bayesian inference-based approach for performance prognostics towards uncertainty quantification and its applications on the marine diesel engine," *ISA Transactions*, vol. 118, pp. 159–173, 2021. DOI: https://doi.org/10.1016/j.isatra.2021.02.024.

[8] D. Wu, C. Jennings, J. Terpenny, R. X. Gao, and S. Kumara, "A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 7, p. 071 018, Apr. 2017. DOI: 10.1115/1.4036350.

[9] M. S. Kan, A. C. Tan, and J. Mathew, "A review on prognostic techniques for non-stationary and non-linear rotating systems," *Mechanical Systems and Signal Processing*, vol. 62-63, pp. 1–20, 2015. DOI: https://doi.org/10.1016/j.ymssp.2015.02.016.

[10] I. El-Thalji and E. Jantunen, "A summary of fault modelling and predictive health monitoring of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 60-61, pp. 252–272, 2015. DOI: https://doi.org/10.1016/j.ymssp.2015.02.008.

[11] Y. Yoo and J.-G. Baek, "A Novel Image Feature for the Remaining Useful Lifetime Prediction of Bearings Based on Continuous Wavelet Transform and Convolutional Neural Network," *Applied Sciences*, vol. 8, no. 7, 2018, ISSN: 2076-3417. DOI: 10.3390/app8071102.

[12] C. Ferreira and G. Gonçalves, "Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods," *Journal of Manufacturing Systems*, vol. 63, pp. 550–562, 2022. DOI: https://doi.org/10.1016/j.jmsy.2022.05.010.

[13] J. Chen, R. Huang, Z. Chen, W. Mao, and W. Li, "Transfer learning algorithms for bearing remaining useful life prediction: A comprehensive review from an industrial application perspective," *Mechanical Systems and Signal Processing*, vol. 193, Jun. 2023, ISSN: 10961216. DOI: 10.1016/j.ymssp.2023.110239.

[14] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li, "Transfer learning using deep representation regularization in remaining useful life prediction across operating conditions," *Reliability Engineering System Safety*, vol. 211, p. 107 556, 2021. DOI: https://doi.org/10.1016/j.ress.2021.107556.

[15] J. Zhu, N. Chen, and C. Shen, "A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions," *Mechanical Systems and Signal Processing*, vol. 139, p. 106 602, 2020. DOI: https://doi.org/10.1016/j.ymssp.2019.106602.

[16] A. Zhang, H. Wang, S. Li, *et al.*, "Transfer learning with deep recurrent neural networks for remaining useful life estimation," *Applied Sciences*, vol. 8, no. 12, Nov. 2018, ISSN: 20763417. DOI: 10.3390/app8122416.

[17] M. S. Azari, F. Flammini, S. Santini, and M. Caporuscio, "A Systematic Literature Review on Transfer Learning for Predictive Maintenance in Industry 4.0," *IEEE Access*, vol. 11, pp. 12 887–12 910, 2023, ISSN: 21693536. DOI: 10.1109/ACCESS.2023.3239784.

[18] B.-J. van Bruchem, "Helicopter drive train modelling for manoeuvre load alleviation," M.S. thesis, Faculty of Aerospace Engineering, Delft University of Technology, Delft, Aug. 2017.

[19] B. Dykas, "Detection of Naturally Occurring Gear and Bearing Faults in a Helicopter Drivetrain," U.S. Army Research Laboratory, Tech. Rep., Jan. 2014.

[20] J. Wiig, “Optimisation du système de surveillance des hélicoptères pour l’amélioration du diagnostic et de la maintenance,” Theses, Arts et Métiers ParisTech, Dec. 2006.

[21] P. r, M. Kreidl, and R. Smid, “Condition indicators for gearbox condition monitoring systems,” *Acta Polytechnica*, vol. 45, pp. 35–, Jun. 2005.

[22] L. Wang, M. Hu, B. Ma, and Z. Jiang, “Time Synchronous Averaging Based on Cross-power Spectrum,” *Chinese Journal of Mechanical Engineering*, vol. 36, no. 1, p. 51, Apr. 2023, ISSN: 2192-8258. DOI: `10.1186/s10033-023-00867-9`.

[23] L. Hu, L. Zhang, F. Gu, N. Hu, and A. Ball, “Extraction of the largest amplitude impact transients for diagnosing rolling element defects in bearings,” *Mechanical Systems and Signal Processing*, vol. 116, pp. 796–815, 2019, ISSN: 0888-3270. DOI: `https://doi.org/10.1016/j.ymssp.2018.07.022`.

[24] K. Fraser, “An overview of health and usage monitoring sytems (HUMS) for military helicopters,” Tech. Rep., 1994.

[25] F. Elasha, S. Shanbr, X. Li, and D. Mba, “Prognosis of a wind turbine gearbox bearing using supervised machine learning,” *Sensors*, vol. 19, no. 14, Jul. 2019, ISSN: 14248220. DOI: `10.3390/s19143092`.

[26] J. Zhu, T. Nostrand, C. Spiegel, and B. Morton, “Survey of Condition Indicators for Condition Monitoring Systems,” *Annual Conference of the PHM Society*, vol. 6, no. 1, Sep. 2014. DOI: `https://doi.org/10.36001/phmconf.2014.v6i1.2514`.

[27] P. D. Samuel and D. J. Pines, “A review of vibration-based techniques for helicopter transmission diagnostics,” *Journal of Sound and Vibration*, vol. 282, no. 1-2, pp. 475–508, Apr. 2005, ISSN: 0022460X. DOI: `10.1016/j.jsv.2004.02.058`.

[28] H. Ding, L. Yang, Z. Cheng, and Z. Yang, “A remaining useful life prediction method for bearing based on deep neural networks,” *Measurement: Journal of the International Measurement Confederation*, vol. 172, Feb. 2021, ISSN: 02632241. DOI: `10.1016/j.measurement.2020.108878`.

[29] L. Nie, L. Zhang, S. Xu, W. Cai, and H. Yang, “Remaining useful life prediction for rolling bearings based on similarity feature fusion and convolutional neural network,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 44, no. 8, Aug. 2022, ISSN: 18063691. DOI: `10.1007/s40430-022-03638-0`.

[30] G. Pagiatakis, L. Dritsas, G. Chatzarakis, G. Todorov, and B. Stoev, “Introducing concepts and methodologies of fault detection into electrical engineering education: The induction machine example,” in *2017 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, Apr. 2017, pp. 381–388, ISBN: 978-1-5090-5467-1. DOI: `10.1109/EDUCON.2017.7942876`.

[31] X. Yang, R. Huang, X. Liu, J. Cai, and Z. Xie, “Fault mechanism and system structure of in-transit vehicles safety condition monitoring in rail transit,” in *Proceedings of the 6th International Conference on Electrical Engineering and Information Technologies for Rail Transportation (EITRT) 2023*, M. Gong, L. Jia, Y. Qin, J. Yang, Z. Liu, and M. An, Eds., Singapore: Springer Nature Singapore, 2024, pp. 656–663, ISBN: 978-981-99-9319-2.

[32] M. Chandravanshi and S. Poddar, “Ball bearing fault detection using vibration parameters,” *International Journal of Engineering Research Technology*, vol. 2, Dec. 2013, ISSN: 2278-0181.

[33] M. Alonso-González, V. G. Díaz, B. L. Pérez, B. C. P. G-Bustelo, and J. P. Anzola, “Bearing fault diagnosis with envelope analysis and machine learning approaches using cwru dataset,” *IEEE Access*, vol. 11, pp. 57 796–57 805, 2023. DOI: `10.1109/ACCESS.2023.3283466`.

[34] K. R. Al-Balushi, A. Addali, B. Charnley, and D. Mba, “Energy Index technique for detection of Acoustic Emissions associated with incipient bearing failures,” *Applied Acoustics*, vol. 71, no. 9, pp. 812–821, Sep. 2010, ISSN: 0003682X. DOI: `10.1016/j.apacoust.2010.04.006`.

[35] K. Medjaher, N. Zerhouni, and J. Baklouti, “Data-driven prognostics based on health indicator construction: Application to pronostia’s data.,” in *2013 European Control Conference, ECC 2013*, Jul. 2013, pp. 1451–1456. DOI: `10.23919/ECC.2013.6669223`.

[36] J. Guo, Z. Li, and M. Li, *A Review on Prognostics Methods for Engineering Systems*, Sep. 2020. DOI: `10.1109/TR.2019.2957965`.

[37] W. Tiddens, J. Braaksma, and T. Tinga, “Decision Framework for Predictive Maintenance Method Selection,” *Applied Sciences*, vol. 13, no. 3, p. 2021, Feb. 2023, ISSN: 2076-3417. DOI: `10.3390/app13032021`.

[38] B. Wang, Y. Lei, N. Li, and N. Li, “A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings,” *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 401–412, Mar. 2020, ISSN: 15581721. DOI: `10.1109/TR.2018.2882682`.

[39] Z. Yang, P. Baraldi, and E. Zio, “A comparison between extreme learning machine and artificial neural network for remaining useful life prediction,” in *2016 Prognostics and System Health Management Conference (PHM-Chengdu)*, 2016, pp. 1–7. DOI: `10.1109/PHM.2016.7819794`.

[40] P. Wang, Z. Long, and G. Wang, "A hybrid prognostics approach for estimating remaining useful life of wind turbine bearings," *Energy Reports*, vol. 6, pp. 173–182, Dec. 2020, ISSN: 23524847. DOI: 10.1016/j.egyr.2020.11.265.

[41] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, May 2018, ISSN: 10961216. DOI: 10.1016/j.ymssp.2017.11.016.

[42] C. Cheng, G. Ma, Y. Zhang, *et al.*, "A Deep Learning-Based Remaining Useful Life Prediction Approach for Bearings," *ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1243–1254, Jun. 2020, ISSN: 1083-4435. DOI: 10.1109/TMECH.2020.2971503.

[43] J. Li, F. Huang, H. Qin, and J. Pan, "Research on Remaining Useful Life Prediction of Bearings Based on MBCNN-BiLSTM," *Applied Sciences*, vol. 13, no. 13, Jul. 2023, ISSN: 20763417. DOI: 10.3390/app13137706.

[44] B. Zhang, L. Zhang, and J. Xu, "Degradation feature selection for remaining useful life prediction of rolling element bearings," *Quality and Reliability Engineering International*, vol. 32, no. 2, pp. 547–554, 2016. DOI: https://doi.org/10.1002/qre.1771.

[45] J. Li, Z. Wang, X. Liu, and Z. Feng, "Remaining Useful Life Prediction of Rolling Bearings Using GRU-DeepAR with Adaptive Failure Threshold," *Sensors*, vol. 23, no. 3, Feb. 2023, ISSN: 14248220. DOI: 10.3390/s23031144.

[46] J. Yang, Y. Peng, J. Xie, and P. Wang, "Remaining Useful Life Prediction Method for Bearings Based on LSTM with Uncertainty Quantification," *Sensors*, vol. 22, no. 12, Jun. 2022, ISSN: 14248220. DOI: 10.3390/s22124549.

[47] T. Han, J. Pang, and A. C. Tan, "Remaining useful life prediction of bearing based on stacked autoencoder and recurrent neural network," *Journal of Manufacturing Systems*, vol. 61, pp. 576–591, Oct. 2021, ISSN: 02786125. DOI: 10.1016/j.jmsy.2021.10.011.

[48] L. Shuang, X. Shen, J. Zhou, H. Miao, Y. Qiao, and G. Lei, "Bearings remaining useful life prediction across equipment-operating conditions based on multisource-multitarget domain adaptation," *Measurement*, vol. 236, p. 115 026, 2024. DOI: https://doi.org/10.1016/j.measurement.2024.115026.

[49] G. Li, L. Yang, C.-G. Lee, X. Wang, and M. Rong, "A Bayesian Deep Learning RUL Framework Integrating Epistemic and Aleatoric Uncertainties," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8829–8841, Sep. 2021, ISSN: 0278-0046. DOI: 10.1109/TIE.2020.3009593.

[50] S. Schwendemann and A. Sikora, "Transfer-Learning-Based Estimation of the Remaining Useful Life of Heterogeneous Bearing Types Using Low-Frequency Accelerometers," *Journal of Imaging*, vol. 9, no. 2, Feb. 2023, ISSN: 2313433X. DOI: 10.3390/jimaging9020034.

[51] B. Zhao, X. Zhang, H. Li, and Z. Yang, "Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions," *Knowledge-Based Systems*, vol. 199, p. 105 971, 2020. DOI: https://doi.org/10.1016/j.knosys.2020.105971.

[52] S. Sonoda and N. Murata, "Neural network with unbounded activation functions is universal approximator," *Applied and Computational Harmonic Analysis*, vol. 43, no. 2, pp. 233–268, 2017. DOI: https://doi.org/10.1016/j.acha.2015.12.005.

[53] H. Cheng, X. Kong, G. Chen, Q. Wang, and R. Wang, "Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors," *Measurement*, vol. 168, p. 108 286, 2021. DOI: https://doi.org/10.1016/j.measurement.2020.108286.

[54] Y. Shang, X. Tang, G. Zhao, P. Jiang, and T. Ran Lin, "A remaining life prediction of rolling element bearings based on a bidirectional gate recurrent unit and convolution neural network," *Measurement*, vol. 202, p. 111 893, 2022. DOI: https://doi.org/10.1016/j.measurement.2022.111893.

[55] D. Yao, B. Li, H. Liu, J. Yang, and L. Jia, "Remaining useful life prediction of roller bearings based on improved 1D-CNN and simple recurrent unit," *Measurement*, vol. 175, p. 109 166, 2021, ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2021.109166.

[56] X. Dong, C. Zhang, H. Liu, D. Wang, and T. Wang, "A multi-constrained domain adaptation network for remaining useful life prediction of bearings," *Mechanical Systems and Signal Processing*, vol. 206, p. 110 900, 2024, ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2023.110900.

[57] 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, ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.191.

[58] Y. Ding, P. Ding, X. Zhao, Y. Cao, and M. Jia, "Transfer Learning for Remaining Useful Life Prediction Across Operating Conditions Based on Multisource Domain Adaptation," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 4143–4152, Oct. 2022, ISSN: 1941014X. DOI: 10.1109/TMECH.2022.3147534.

[59] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, *Deep Domain Confusion: Maximizing for Domain Invariance*, 2014. arXiv: 1412.3474 [cs.CV].

[60] Y. Ganin, E. Ustinova, H. Ajakan, *et al.*, “Domain-adversarial training of neural networks,” in *Domain Adaptation in Computer Vision Applications*, G. Csurka, Ed. Cham: Springer International Publishing, 2017, pp. 189–209, ISBN: 978-3-319-58347-1. DOI: 10.1007/978-3-319-58347-1_10.

[61] M. HassanPour Zonoozi and V. Seydi, “A Survey on Adversarial Domain Adaptation,” *Neural Processing Letters*, vol. 55, no. 3, pp. 2429–2469, Jun. 2023, ISSN: 1370-4621. DOI: 10.1007/s11063-022-10977-5.

[62] S. Siahpour, X. Li, and J. Lee, “A Novel Transfer Learning Approach in Remaining Useful Life Prediction for Incomplete Dataset,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022, ISSN: 15579662. DOI: 10.1109/TIM.2022.3162283.

[63] P. R. d. O. da Costa, A. Akçay, Y. Zhang, and U. Kaymak, “Remaining useful lifetime prediction via deep domain adaptation,” *Reliability Engineering and System Safety*, vol. 195, Mar. 2020, ISSN: 09518320. DOI: 10.1016/j.ress.2019.106682.

[64] M. Singh Rathore and S. Harsha, “Rolling bearing prognostic analysis for domain adaptation under different operating conditions,” *Engineering Failure Analysis*, vol. 139, p. 106 414, 2022. DOI: https://doi.org/10.1016/j.engfailanal.2022.106414.

[65] B. Sun and K. Saenko, “Deep coral: Correlation alignment for deep domain adaptation,” in Nov. 2016, pp. 443–450, ISBN: 978-3-319-49408-1. DOI: 10.1007/978-3-319-49409-8_35.

A

Research Overview

From the literature review, a research gap was identified. Following this research gap, this chapter will introduce the research gap and research questions in Section A.1. Additionally, the set-up for this research will be described in Section A.2.

A.1. Research Gap

In the literature review, it has been observed that much research has been done on RUL predictions for rolling bearings. These prediction methods involved CNNs or LSTMs for RUL prediction, based on either automatically or manually extracted features. These models are typically trained and tested on the same operating condition, which means that they require large amounts of run-to-failure data for bearings under specific operating condition. In practice, bearing run-to-failure data is only available in certain operating conditions. As a result, the aforementioned models fail to provide accurate prediction results when applied to operating conditions where run-to-failure data is unavailable.

To address this problem, some researchers transferred degradation knowledge from certain operating conditions to a different operating condition using transfer learning approaches. Such approaches reduce the need for run-to-failure data, as this data is only needed for source domain operating conditions and not for target domain conditions. Previous transfer learning methods have shown promising prediction results. In particular, models that learned degradation information from multiple operating conditions and transferred it to a different target condition demonstrated good performance. However, these models make use of vibration data in the time and time-frequency domain. Some systems only record vibration data in the frequency domain. Consequently, a prediction model for such a system will not have access to degradation characteristics provided in the time or time-frequency domain. According to the literature review, no studies have yet been published on a transfer learning approach for RUL prediction based solely on frequency domain data. Therefore, research objective will be to **construct a prediction model that is able to estimate bearing Remaining Useful Life across operating conditions and bearing types based solely on frequency domain data**.

If such a model can be constructed, then it can be used as a first step towards transfer learning approaches applied to a real helicopter drive train bearing. In this case, bearing degradation information can be learned from run-to-failure data collected in a laboratory environment and transferred to a real bearing.

The main research question of this study is as follows:

How can a predictive model be constructed to estimate the Remaining Useful Life of bearings across different operating conditions and bearing types using only frequency domain data?

This research question is divided into several sub-questions:

- What backbone data-driven prediction model can be used to estimate bearing Remaining Useful Life using only frequency domain data?
- How can the feature distributions from multiple operating conditions and bearing types, including several source conditions and one target condition, be aligned?

A.2. Method

This research aims to construct a RUL prediction model using only frequency domain data based on transfer learning. A first step of this research is to replicate a system that only provides frequency domain data. This can be done by applying a Fourier Transform to a time domain bearing vibration signal, such that a one-sided frequency spectrum is obtained. Next, a backbone RUL prediction model will be constructed. As CNNs posses the ability to automatically extract features from the vibration data, a CNN structure could be a promising backbone. In order to align the feature distributions from various operating conditions and bearing types, a domain adaptation module will be needed. This module will be inspired by Ding's multi-source domain adaption model.

Once the multi-source domain adaptation RUL prediction model is constructed, the performance of the model should be evaluated by metrics such as Mean Absolute Error (MAE), RMSE, or Score and compared to models proposed by other researchers. Additionally, in order to increase the reliability of the predictions, the model uncertainty should be estimated. This can be done by a Monte Carlo Dropout technique, which uses the dropout layers of the model. It will also be useful to visualise the effect of transfer learning by comparing the feature distributions of models trained with and without the domain adaptation module.

When training the model, it requires access to run-to-failure data from multiple operating conditions, as well as data from conditions without run-to-failure data. Additionally, if the model is tasked with estimating the RUL across different bearing types, it requires access to data from different bearing types. An example of such a task is to estimate the RUL of a bearing type under a certain operating condition where the source domain contains data of a different bearing type. This data can be obtained from publicly available bearing data sets, namely the IEEE PHM 2012 prognostic challenge data set provided by the FEMTO-ST Institute in France and the XJTU-SY data set provided by Xi'an Jiaotong University in China. Both data sets can be retrieved from Github. Additionally, this study requires access to a computer with a Pytorch environment for constructing and training the prediction model. It will be very beneficial if the computer is equipped with a powerful graphics processing unit, in order to accelerate the model training procedure using Pytorch training with Cuda.