Deep Transfer Learning for Failure Prediction Across Failure Types

Zhe Li[a], Eivind Kristoffersen[b], Jingyue Li[c,*]

[a] *Central Academe, Shanghai Electric Group Co., Ltd., 200070 Shanghai, China*
[b] *Sopra Steria, 0185, Oslo, Norway*
[c] *Department of Computer Science, Norwegian University of Science and Technology, 7491 Trondheim*

Abstract

With the increasing development of artificial intelligence (AI) technologies, deep learning-driven approaches have been widely applied to predicate different machinery failures. One key challenge of failure prediction is to collect sufficient data, especially data of various failure types, to train the data-driven models. Existing studies focus on using transfer learning to transfer knowledge across machines or domains, but not across failure types. In this study, we hypothesise that knowledge about failure among similar failure types is transferable. Should the hypothesis hold, companies may no longer require a large amount of all types of failure data for predictive maintenance. This will increase the companies' overall implementation feasibility and productivity gains. We tested our hypothesis on knowledge transferability for failure prediction in an experiment performed on rotating machinery with vibration signals. During the experiment, we first calibrated the performance of the trained deep neural network in each impending failure type. Then, we leveraged the architecture and hyper-parameters of the neural network model trained from one type of failure as the pre-trained model for knowledge transfer. The pre-trained model is fine-tuned with data from another type of failure of the same machine. After that, we com-

*Corresponding author

Email addresses: `lizhe7@shanhai-electric.com` (Zhe Li), `eivind.kristoffersen@soprasteria.com` (Eivind Kristoffersen), `jingyue.li@ntnu.no` (Jingyue Li)

pared the performance of the neural network model to predict the second type of failure before and after knowledge transfer. Results showed that transferring knowledge obtained from one type of failure could vastly improve the performance of predicting another type of failure, which may not have sufficient data to train a good prediction model. This result implies that predictive analytics can apply parameter-based deep transfer learning (TL) to address the challenge of insufficient data on all types of machine failures for failure prediction.

Keywords: Deep neural network, Failure prediction, Predictive maintenance, Transfer learning

---

1. Introduction

Advanced manufacturing strategies such as predictive maintenance have the potential to increase equipment lifetime, improve production quality, reduce lead times, prevent accidents and malfunctions, and optimise energy efficiency and resource consumption [1]. One typical task of predictive maintenance is to predict system failures. Machine learning-based failure prediction can handle massive monitoring signals collected from various sensors and identify the working conditions of machines [2] [3]. Traditional failure prediction methods usually require elaborate engineering and considerable domain expertise to design a feature extraction system that can transform raw data into a suitable internal representation or feature vectors [4]. For this reason, as one of the most popular trends in machine learning, deep learning methods have been widely applied for predictive maintenance with various tasks because of their advantages in automatic feature extraction [5]. Many types of deep learning architectures have been studied for predictive maintenance in recent years. Wang et al. [6] applied a fully connected deep neural network (DNN) to identify impending failures for a wind turbine gearbox. By comparing their network with several traditional machine learning approaches, the experiment shows that their method performs better in extracting useful features for indicating failures from vibration signals. Li et al. [7] proposed a deep belief network (DBN) to predict backlash error in machining centres. The proposed deep neural network can discover helpful information about failures from coupled data with good generalisability. Other deep learning-based approaches, such as autoencoder-based deep neural networks [8], deep convolution neural networks [9], and long/short-term memory networks [5], have also been widely applied and researched for predictive maintenance. All the aforementioned studies have achieved specific targets for predictive maintenance and demonstrated successful applications of deep learning-based approaches. However, most methods inevitably require a large amount of data collected under both normal and failure conditions. Therefore, an approach to reduce the necessary amount of data without reducing prediction performance and additional effort for predictive maintenance is imperative. To address the data deficiency challenge, transfer learning (TL) [10][11] arose as a new learning framework to reuse knowledge captured in deep neural networks [12] [13]. A few studies have applied TL for predicting systems' remaining useful life (RUL), e.g., [14][15][16], and failures, e.g., [17][18][19][20][21][22][23]. For studies using TL for failure prediction, they focused either on knowledge transfer across domains [17][22][23], across machines [18][20][21], or across working conditions [19]. Many sensor data of a complex machine may need to be monitored and analysed to predict failures. However, not all types of failures have sufficient data to build AI-based prediction models. Predicting a particular type of failure using the knowledge of another failure type on the same machine could substantially save costs and improve system operation reliability and safety. Feng and Zhao [24] tried to use the attribute transfer method to tackle the zero-sample fault diagnosis challenge. However, the method in [24] requires additional domain knowledge to form attribute descriptions in both training and target failures. Such domain knowledge may not be available for some types of failures. To our knowledge, no study has investigated the possibility of applying TL in failure prediction across different failure types without human intervention.

We hypothesise that deep neural networks for predicting different failures related to the same machine may share specific parameters. We, therefore, piloted using parameter-based TL to predict one type of failure based on knowledge captured in the deep neural network for predicting another type of failure. Our experiment results on predicting two types of failures of a rotation machine showed that TL could improve failure prediction performance across failure types. The contributions of this study are twofold:

- We give empirical evidence that parameter-based TL could transfer knowledge across different types of failures on the same machine to improve the prediction performance and address the insufficient data challenge.

- We proposed a novel deep transfer learning-based predictive maintenance method based on our findings.

The rest of this paper is organised as follows. Section 2 introduces the transfer learning (TL) approaches. Section 3 describes our research design. Section 4 presents our research implementation and results. Section 5 proposes a novel deep transfer learning-based failure prediction approach across failure types. Section 6 discusses our results and compares our approach with related work. The conclusions and future work are summarised in Section 7.

2. Transfer learning

First of all, notations and acronyms used frequently in this paper are summarized in Table 1.

Transferring knowledge from one task or domain to another may vastly improve machine learning efficiency and performance. The insight behind TL is that generalisation may occur within tasks and across tasks [10]. TL aims to unite knowledge from different fields, which enables leveraging validated knowledge from other domains or tasks when the targeted domain's available data is limited. In the definition of TL [25], we consider that a domain D consists of two components: one feature space X and one marginal probability distribution P(X), where X = $\{x_1, ..., x_n\} \in$ X. In general, if two domains are different, they may have different feature spaces or different marginal probability distributions. Given a specific domain, D = {X, P(X)}, a task T consists of two components:

Table 1: Descriptions of notations and acronyms

Notation	**Description**	**Notation**	**Description**
D	Domain	T	Task
X	Feature space	DS	Source domain
Y	Label space	TS	Learning task in source domain
x, y	Single sample	DT	Target domain
n	Sample number	TT	Learning task in target domain
f(·)	Prediction function in source domain	fT(·)	Prediction function in target domain
Acronym	**Description**	**Acronym**	**Description**
TL	Transfer Learning	AI	Artifical Intelligence
Failure Type 1	Friction failure	Failure Type 2	Load imbalance failure
DNN	Deep Neural Networ	SVM	Support vector machine
DBN	Deep belief network	KNNC	k-nearest neighbours classification
BPNN	Backpropagation neural network	ReLU	Rectified Linear
NN	Neural network model	NN-A	NN model for knowledge transfer
NN-B	NN model receiving knowledge transferred	NN-C	Another NN model receiving knowledge transferred

one label space Y and one objective predictive function f(·) (denoted by T = {Y, f(·) }, which cannot be observed but could be learned through training. The training data consist of pairs x_i, y_i, where $x_i \in$ X and $y_i \in$ Y. The function f(·) can be leveraged to predict the corresponding label, f(x), of a new instance

x. Given a source domain DS and a learning task TS and a target domain DT and a learning task TT, TL aims to help improve the learning of the target predictive function fT(·) in DT using the knowledge in DS and TS, where DS ≠ DT or TS ≠ TT. Several approaches are used to achieve TL. For instance, Weiss et al. [11] and Pan and Yang [25] categorised the form of knowledge transfer into four general categories:

- The first category is instance-based TL. The most common approach in this category is for instances from the source domain to be reweighted to adjust for marginal distribution differences. These reweighted instances can be leveraged to train models for the target domain.
- The second category is feature-based TL, which reshapes the features from the source domain by reweighting them to match the target domain. The core idea is to identify potential common feature space with predictive qualities while reducing the marginal distribution between the domains.
- The third category is parameter-based TL, which assumes that the source and the target tasks may share specific parameters and that the knowledge that we want to transfer can be encoded into these shared parameters. Therefore, knowledge can be transferred across tasks by identifying those shared parameters.
- The fourth category is to transfer knowledge based on some defined relationship between the source and target domains. It usually deals with TL in relational domains.

3. Research design and data collection

In this section, we first justify using parameter-based TL to predict different types of failure of the same machine. Then, we explain the machine and its failures we study.

3.1. The rationale for using parameter-based TL

Based on knowledge transferability theory, we hypothesise that phenomena captured by sensors for detecting similar types of failures may share specific common rules. As shown in Figure 1, all phenomena captured from a machine shall follow specific natural rules such as Newton's Law. When a failure type (A) happens, all phenomena under this failure will also follow specific rules. For machine learning-based failure identification, the essence is to learn about those rules by training a data-driven model to recognise the edge of the failure circle. We consider that two similar failure types may have an intersection area that represents their common rules. These common rules cannot be seen with human eyes but could be captured and represented by deep learning models. As the common rules are usually unknown, the attribute transfer method [24], which requires additional domain knowledge to form attribute descriptions in both source and target failure types, is not applicable. The instance-based TL is also difficult to use because it is challenging to know which data or instance of the source failure type is relevant to the target failure type.

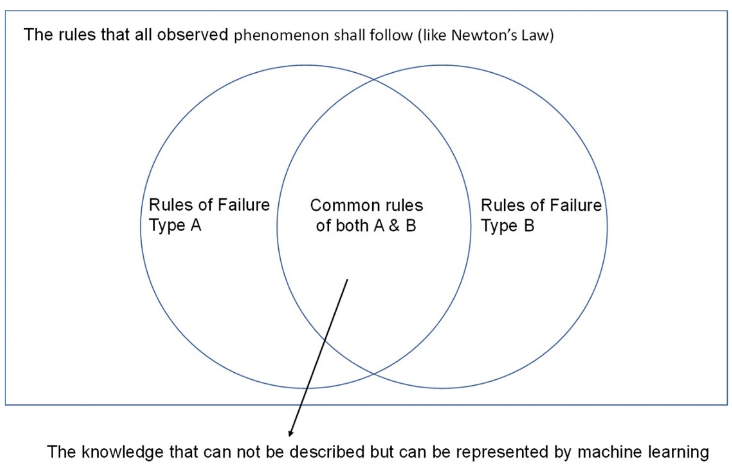

Figure 1: Knowledge representation by machine learning

The parameter-transfer approaches *assume that individual models for related tasks should share some parameters or prior distributions of hyperparameters*

[25]. The parameter-based TL has shown positive results in transferring knowledge across machines [20], and across conditions of the same type of failure [19]. Cho et al. [19] also compared two parameter-based transferring learning strategies, one is to transfer only the neural network architecture, and another is to transfer both the architecture and the neural network parameters. They found that transferring architecture and parameters improves the prediction accuracy of the target domain better. Our research chose to study parameter-based TL since we assume that the source and the target tasks share the same parameter scale and prior distributions of the hyperparameters in the data-driven model. We limit our focus on using sensor data collected from the same equipment to accomplish different tasks. In this study, the tasks are to identify different types of impending machine failures. Furthermore, we consider that the tasks shall have a similar relationship between inputs and outputs. We hypothesise that, by discovering and reusing the shared parameters, knowledge for identifying one type of failure could be transferred to identify the other.

3.2. Machine and failures studied

We tested our hypothesis on a rotating machine during the experiment since it is the most common type of mechanical machinery and usually ran under harsh working conditions with various types of failures [26].

Our laboratory's experiment was set up with a Bently Nevada Rotor Kit to simulate rotating equipment's working conditions. As shown in Figure 2, three accelerometers of Kistler 8702B100 were mounted in the X, Y, and Z directions on the bearing house to collect the vibration signals from the rotating machine. We selected vibration signals as the primary sensor data, given their superior performance in indicating anomalies from the complex environment and broad applicability in mechanical systems [6]. The sampling frequency for vibration signals was 4,096 Hz. The experiment was performed with changing revolving speed to simulate practical working conditions. Vibration monitoring refers to the zero position of the machine. In this position, signals from the accelerometers were recorded and stored as normal samples. The rub generator can be modulated to simulate friction on the main spindle, which we labelled as Failure Type 1. Failure Type 2 (load imbalance) can be injected by adjusting the weight on the adjustable mass load during the experiment. *"Friction and imbalance of components in rotating machines are some of the most recurrent failures that significantly increase vibration levels, thus affecting the reliability of the devices"* [27]. Failures of both types (1 and 2) will largely affect the rotating movement of the main spindle, which will react to the measured vibration signals. Thus, sensor data under impending failure conditions can be obtained through the acceleration metres.

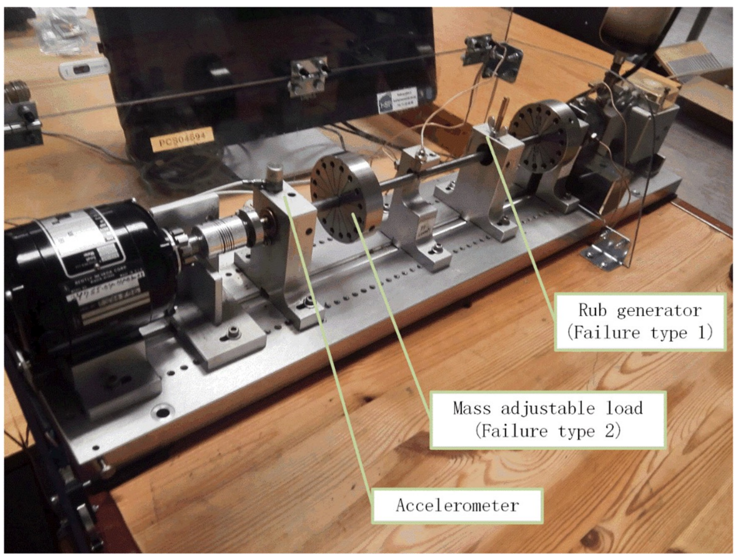

Figure 2: The hardware setup of the experiment

During the experiment, the vibration signals were measured through the accelerometers in three directions at different rotating speeds through proximity sensors and a handheld tachometer for control. We collected 5,035 samples. Among them, 1,608 samples were collected under normal working conditions, which are hereby labelled normal samples. The amounts of samples collected with Failure Types 1 (friction on the main spindle) and 2 (load imbalance) are 1,703 and 1,724, respectively.

In this research, we applied wavelet and Fourier transforms to extract wavelet coefficient-based and energy-based feature sequences from the vibration signals to represent the working conditions of target equipment in both the time and frequency domains [28]. Different types of wavelet functions may cause various time-frequency structures. In this study, the Daubechies 4 wavelet function was selected because of its capacity to derive both conventional and energy-based features from vibration signals [29] as well as its superior performance to estimate the local properties, such as breakdown points [30]. More details about the rationale for choosing the Daubechies 4 wavelet function to extract features from vibration signals and the method to use wavelet function are in [26].

To increase the data-driven model's efficiency, we extracted a total of 33 features as follows from the original vibration signals.

- The first and second peaks of vibration signals in the frequency domain in three directions
- The standard deviation noises of wavelet coefficients in levels 1 to 4 (the decomposition level during the wavelet transform)
- The percentages of energy (acquired from one-dimensional wavelet decomposition) corresponding to the approximation in three directions
- The percentages of energy corresponding to the details in levels 1 to 4

4. Research implementation and results

In this study, we designed a study to validate our hypothesis that knowledge between different failure types of the same machine can be transferred. In particular, we attempted to investigate whether the knowledge collected in the model of predicting Failure Type 2 (imbalance) can facilitate the prediction of Failure Type 1 (friction). The study had three high-level steps as follows:

- Step 1: We used different percentages of Failure Type 1 data to train neural network models to predict its failure. We chose relatively low accuracy models as the baseline for later comparison.

- Step 2: We used different percentages of Failure Type 2 data to train neural network models to predict its failure. We chose a high accuracy model as the model for transferring Failure Type 2 knowledge in later steps.

- Step 3: We used the high-accuracy Failure Type 2 model as the pre-trained model (including its architecture and hyperparameters) and fine-tuned the model using the same percentage of data for training the low accuracy Failure Type 1 models. Then, we compared the performance of Failure Type 1 models using and without using the pre-trained model. We expected that the pre-trained model's performance should be better because we believed that the knowledge to predict Failure Type 2 was captured in the pre-trained model's parameters, and the knowledge could be transferred to predict Failure Type 1.

4.1. Step 1 and 2: Training models for predicting Failure Types 1 and 2

The applied deep learning model is established through fully connected deep neural networks (DNN) with six layers. Lu et al. [22] used DNN in their studies for fault diagnosis across domains and argued that *"DNN is able to disentangle fundamental factors of variations underlying the samples, and then group features hierarchically in accordance with their relatedness to shared factors, which makes representations robust to transfer."* In addition, DNNs have also shown a superior ability in studies using domain adaption benchmark datasets [31][32][33]. Deep learning can *"capture abstract features and recognize patterns in ways many once thought impossible for computers"* [34], and DNN has *"the ability to learn multiple nonlinear transformations with high complexity through multiple hidden layers, which helps to capture the main variations and discover the discriminative information from the industrial data"* [26]. Li et al. [4] argued that deep learning has attracted not only researchers' but also engineers' attention due to the strong capacity to capture abstract features and recognize patterns in ways many once thought impossible for computers [14]. Li et al. [26] compared DNN with support vector machine (SVM) [35], deep belief network

(DBN) [36], k-nearest neighbours classification (KNNC) [37], and backpropagation neural network (BPNN) [38] to classify bearing looseness, main spindle friction, and load imbalance failures in an experimental environment similar to this study. The results demonstrate DNN's superiority in failure classification.

In our DNN, we used leaky Rectified Linear Unit (ReLU) functions as the activation functions of the hidden layers and softmax functions as the activation functions of the output layer. Unlike the DNN used in [26] for fault classification, we chose to use ReLU as the activation function of the hidden layers instead of using Tanh. The calculations of ReLU and Tanh are shown in Equations (1) and (2).

$$\mathrm{Tanh} f(a) = \frac{\sinh(a)}{\cosh(a)} = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}} \qquad f(a) \in [-1, 1], \tag{1}$$

$$\text{Rectified Linear} \qquad f(a) = \max(0, a) \qquad f(a) \in \mathbb{R}_{+}, \tag{2}$$

where a represents the weighted combination, which is $a = \sum_i x_i w_i + b$, and x_i and w_i are the input values of the firing neurons and their weights, respectively.

A general problem with Tanh functions is that it saturates, which leads to the vanishing gradient problem and prevents deep (multi-layered) networks from learning effectively [39]. In modern neural networks, the default recommendation is to use the rectified linear unit or ReLU [39].

Since the dimension of inputs is 34 (33 features extracted from the vibration signals and the rotating speed), the numbers of nodes selected in the hidden layers of the constructed deep neural network are 32, 32, 16, 16, 8, and 2 (i.e., 32 nodes in the first and second layers, 16 nodes in the third and fourth layers, 8 nodes in the fifth layer, and 2 nodes in the last layer) to learn and represent the input data smoothly. We selected Adam as the optimiser. Adam is *"an algorithm for first-order gradient-based optimization of stochastic objective functions."* [40]. Adam is a popular deep learning algorithm because it achieves good results fast [41]. We used TensorFlow Keras to train the model and applied its default Adam configuration parameters, i.e., learning_rate=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0. We used categorical cross-entropy as the loss function because of its broad applicability. Studies show that using the cross-entropy error function instead of the sum-of-squares for a classification problem leads to faster training as well as improved generalisation [42].

We first calibrated the neural network's performance in Failure Types 1 and 2 with a different number of training samples. We first used 20% of the randomly selected Failure Type 1 samples and 20% of the randomly selected normal samples as the training dataset to train the neural networks, and the remaining Failure Type 1 samples and normal samples as the test dataset. The results of the prediction of Failure Type 1 using the test dataset are shown in Tables 2 and Figure 3. Table 2 shows the recorded test loss and accuracy for all the steps (each step has been run and recorded five times). The test loss is obtained by computing the cross-entropy loss between the labels and the predictions. In Figure 3, condition 0 represents the failure condition, while condition 1 means the normal condition. The blue line in Figure 3 represents the prediction result from the softmax layer of the neural network. The prediction results of using 40% of the Failure Type 1 samples and 40% of the normal samples are shown in Table 2 and Figure 4. The prediction results using 60% of the training samples are shown in Table 2 and Figure 5. From 3 to Figure 5, we found Failure Type 1 can already be predicted with acceptable test accuracy (on average more than 99%) by using only 40% of the training data.

Similar to the calculation we had done for Failure Type 1, for Failure Type 2, we calculated the test accuracy and loss using 10%, 20%, and 40% of the training data. The test accuracy and loss results are shown in Table 2 and Figure 6, Figure 7, and Figure 8. For Type 2 failures, we observed that using only 20% of the training data can already achieve acceptable test accuracy (on average 89.9%).

Table 2: Prediction Loss and Accuracy Using Different Numbers of Training Samples for Failure Type 1

	Percentage and number of training samples of Failure Type 1		
	20% (662 sample)	40% (1,324 sample)	60% (1,986 samples)
Test loss 1	0.429	0.052	0.001
Test accuracy 1	78.5%	100%	100%
Test loss 2	0.499	0.029	0.004
Test accuracy 2	76.9%	100%	100%
Test loss 3	0.401	0.090	0.000
Test accuracy 3	77.7%	100%	100%
Test loss 4	0.226	0.169	0.003
Test accuracy 4	79.3%	97.2%	100%
Test loss 5	0.512	0.043	0.002
Test accuracy 5	64.9%	100%	100%
Average test loss	0.413	0.077	0.002
Average test accuracy	75.5%	99.4%	100%

4.2. Step 3: Transferring knowledge captured from Failure Type 2 data to predicting Failure Type 1

As shown in Table 1, using 40% of Failure Type 2 and 40% of the normal samples to train the NN can achieve 100% prediction accuracy, which means that the NN has learned the knowledge to predict Failure Type 2. We call this model NN-A. Results in Table 1 also show that the model trained using 20% of Failure Type 1 and 20% of its normal samples has room to be improved. The same goes for the NN model trained using 40% of Failure Type 1 and 40% of its normal samples. This step aims to show that using NN-A as the pre-trained model and fine-tuning it using a certain amount of Failure Type 1 data can predict Failure Type 1 better than using the same amount of Failure Type 1

Table 3: Prediction Loss and Accuracy Using Different Numbers of Training Samples for Failure Type 2

	Percentage and number of training samples of Failure Type 2		
	10% (333 samples)	20% (666 samples)	40% (1,332 samples)
Test loss 1	0.392	0.183	0.007
Test accuracy 1	54.4%	99.9%	100%
Test loss 2	0.451	0.161	0.018
Test accuracy 2	65.9%	99.4%	100%
Test loss 3	0.377	0.161	0.000
Test accuracy 3	69.2%	100%	100%
Test loss 4	0.434	0.192	0.062
Test accuracy 4	51.8%	84.5%	100%
Test loss 5	0.508	0.319	0.000
Test accuracy 5	66.0%	65.9%	100%
Average test loss	0.432	0.203	0.017
Average test accuracy	61.4%	89.9%	100%

data without using NN-A. In this step, we first retrained NN-A using 20% of Failure Type 1 data and compared the performance improvement. Then, we did the same by using 40% of Failure Type 2 data.

4.2.1. Experiment 1: Fine-tune NN-A using 20% of Failure Type 1 data

1. Build a neural network called the NN-B with the same structure as the NN-A and use the hyperparameters in the NN-A as the initial weights of the NN-B.
2. Randomly select 20% of the samples from Failure Type 1 and 20% of the normal samples, as Training Dataset 2.
3. Fine-tune NN-B using Training Dataset 2.
4. Use the remaining (80%) Failure Type 1 data as the test dataset and

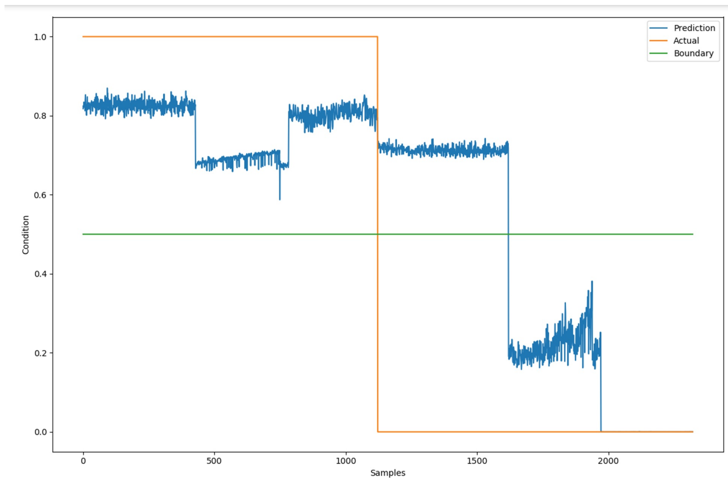

Figure 3: Prediction result of Failure Type 1 using 20% of the Failure Type 1 samples and 20% of the normal samples as the training dataset.

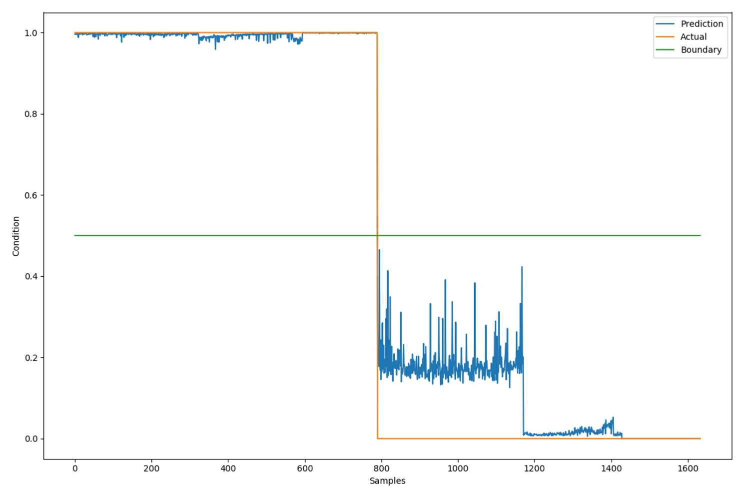

Figure 4: Prediction result of Failure Type 1 using 40% of the Failure Type 1 samples and 40% of the normal samples as the training dataset.

test how well the NN-B can predict Failure Type 1. The testing results represent the performance of predicting Failure Type 1 after TL from the NN model, i.e., the NN-A, which is built from the data of Failure Type 2.

5. Compare the prediction performance of Step 4 with the baseline prediction

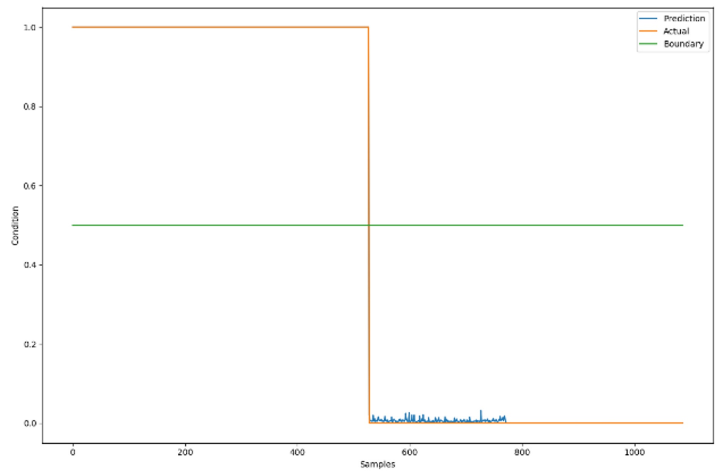

Figure 5: Prediction result of Failure Type 1 using 60% of the Failure Type 1 samples and 60% of the normal samples as the training dataset.

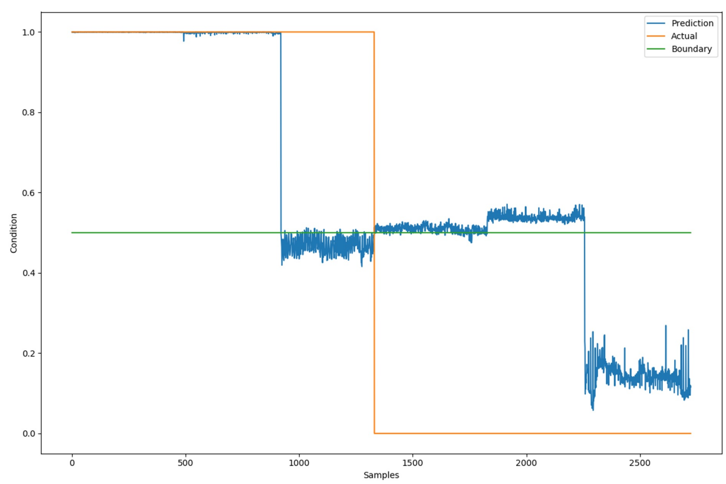

Figure 6: Prediction result of Failure Type 2 using 10% of the Failure Type 2 samples and 10% of the normal samples as the training dataset.

performance (i.e., the performance shown in Table 1 and Figure 3).

The results of Experiment 1 are shown in Table 4. The data in the second column of Table 4 show the prediction results without using the knowledge of

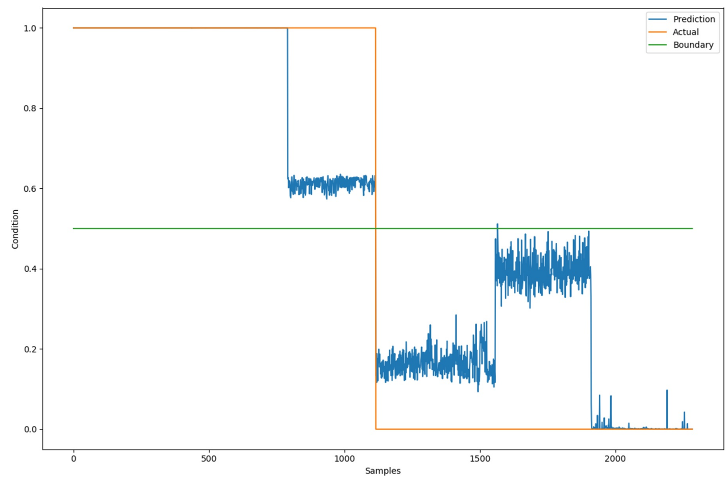

Figure 7: Prediction result of Failure Type 2 using 20% of the Failure Type 2 samples and 20% of the normal samples as the training dataset.

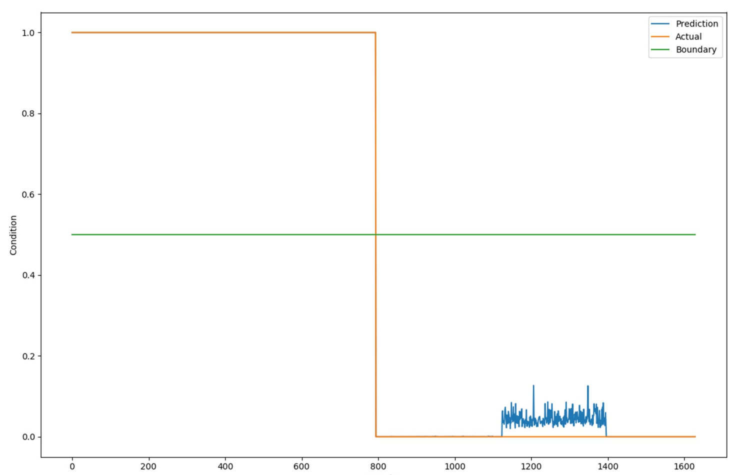

Figure 8: Prediction result of Failure Type 2 using 40% of the Failure Type 2 samples and 40% of the normal samples as the training dataset.

Failure Type 2, i.e., without using TL. The data in the third column of Table 4 show the prediction results using the knowledge of Failure Type 2. This experiment shows that the test accuracy of using the knowledge of Failure Type 2 is, on average, 99.4%, which is higher than the test accuracy (i.e., 75.5%)

without using the knowledge of Failure Type 2. In addition, the average test loss of prediction without using TL is around ten times the average loss of prediction using TL. The results in Figure 9 show the prediction results using TL. By comparing the results shown in Figure 9 and Figure 3, we can see that using TL can provide higher prediction accuracy and lower the loss of Type 1 failures.

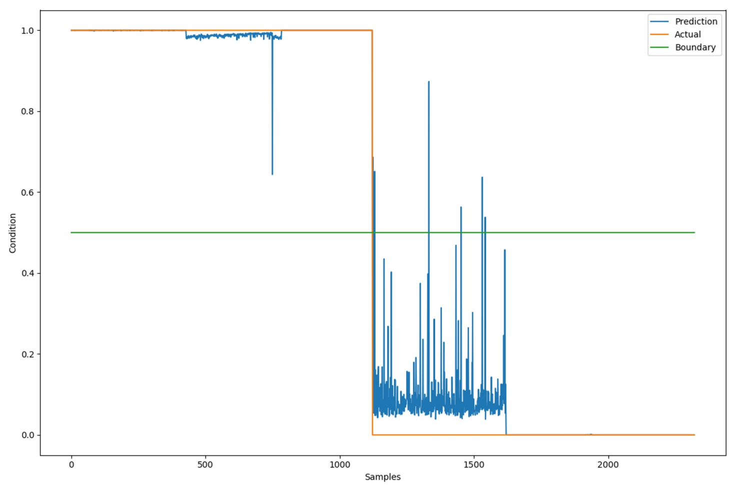

Figure 9: Prediction result of Failure Type 1 using 20% of the Failure Type 1 samples and 20% of the normal samples as the training dataset and TL.

4.2.2. Experiment 2: Fin-tune NN-A using 40% of Failure Type 1 data

1. Build a neural network called the NN-C with the same structure as the NN-A and use the hyperparameters in the NN-A as the initial weights of the NN-C.
2. Randomly select 40% of the samples from Failure Type 1 and 40% of the normal samples, as Training Dataset 3.
3. Fine-tune the NN-C using Training Dataset 3.
4. Use the remaining (60%) Failure Type 1 data as the test dataset and test how well the NN-C can predict Failure Type 1. The testing result represents the performance of predicting Failure Type 1 after TL from the NN model, i.e., the NN-A, which is built from the data of Failure Type 2.

5. Compare the prediction performance of Step 4 with the baseline prediction performance (i.e., the performance shown in Table 1 and Figure 4).

The data in the fourth and fifth columns of Table 4 show the prediction results without using and using the knowledge of Failure Type 2, respectively. Although this experiment shows no significant differences in test accuracy are found between using TL and not using TL, the test loss of not using TL is still ten times that of using TL. Figure 10 visualises the prediction results of using TL and 40% of the sample data to predict Type 1 failures. Comparing such results to those shown in Figure 4, we can see that the test loss shown in Figure 10 is much less than that shown in Figure 4.

Table 4: Result of Using and Not Using TL to Predict Type 1 Failures

	20% (662 training samples)		**40% (1,324 training samples)**	
	Without TL	With TL	Without TL	With TL
Test loss 1	0.429	0.024	0.052	0.002
Test accuracy 1	78.5%	99.7%	100%	100%
Test loss 2	0.499	0.112	0.029	0.001
Test accuracy 2	76.9%	99.0%	100%	100%
Test loss 3	0.401	0.018	0.090	0.000
Test accuracy 3	77.7%	100%	100%	100%
Test loss 4	0.226	0.069	0.169	0.000
Test accuracy 4	79.3%	98.7%	97.2%	100%
Test loss 5	0.512	0.007	0.043	0.027
Test accuracy 5	64.9%	99.9%	100%	100%
Average loss	0.413	0.046	0.077	0.006
Average accuracy	75.5%	99.4%	99.4%	100%

In addition, we found that TL can accelerate training convergence by comparing results in training the neural network using and not using TL. Figure

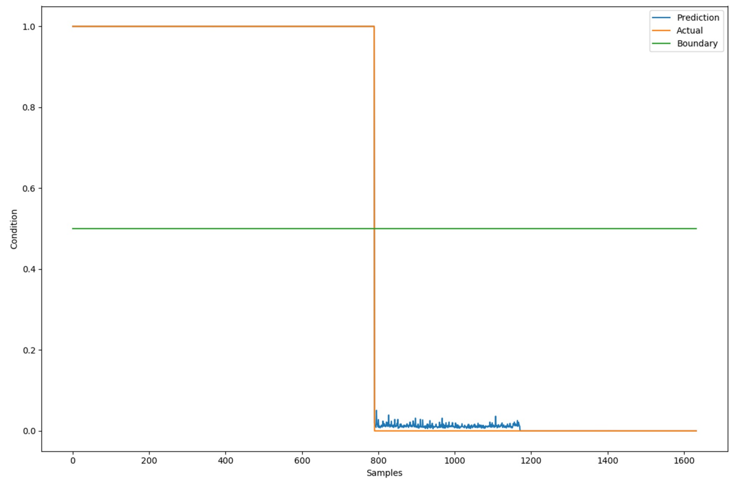

Figure 10: Prediction result of Failure Type 1 using 40% of the Failure Type 1 samples and 40% of the normal samples as the training dataset and TL.

11 and Figure 12 illustrate the training error with epochs and the prediction result without using and using TL, respectively, and show that TL can help accelerate the training convergence in the training process in Experiment 1. A similar trend, as shown in Figure 13 and Figure 14, is observed in Experiment 2.

5. Deep transfer learning-based predictive maintenance method

As mentioned above, to achieve predictive maintenance, most currently available methods inevitably need large amounts of data collected from both normal and failure conditions. After carrying out the aforementioned experiment and analyses, we validate our hypothesis that knowledge between different failure types of the same machine can be transferred. For this reason, it is imperative to develop a method further to identify or predict potential failures based on the hypothesis, especially when the available data is not sufficient for traditional approaches. Therefore, under this background, we propose a deep transfer learning-based method to leverage data across different failure types.

The essential idea is to use one type of failure data to train a neural network.

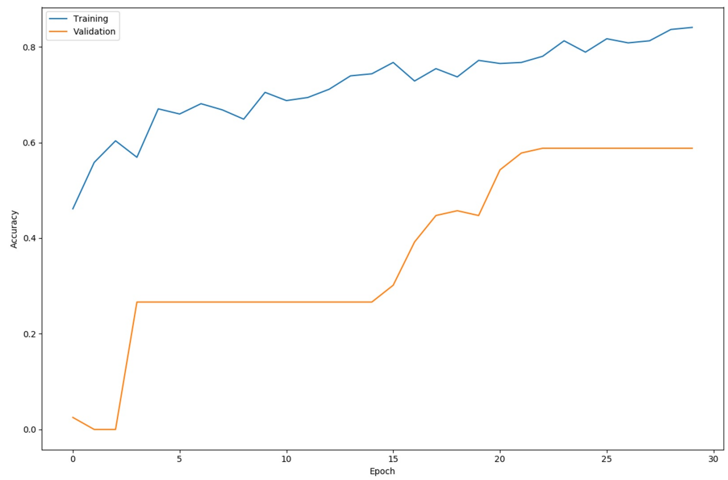

Figure 11: Training accuracy with epochs using 20% of the Failure Type 1 samples and 20% of the normal samples as the training dataset and without TL.

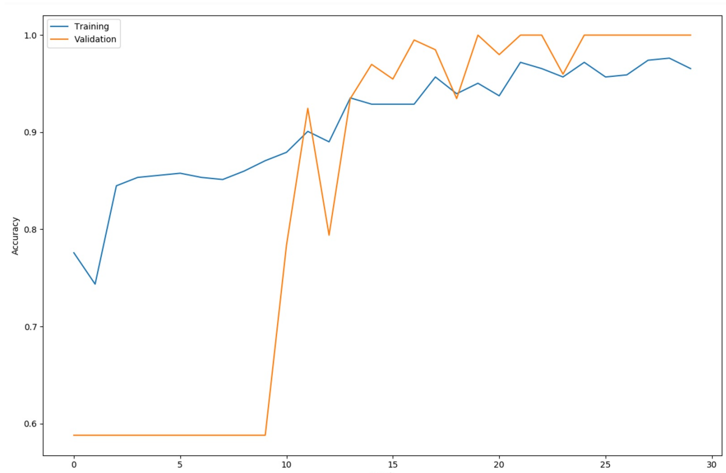

Figure 12: Training accuracy with epochs using 20% of the Failure Type 1 samples and 20% of the normal samples as the training dataset and TL.

The neural network, including architecture and hyperparameters, is used as the pre-trained model for another failure type but is fine-tuned using data that type of failure. The fine-tuned model can then be used for predicting the that type of failure.

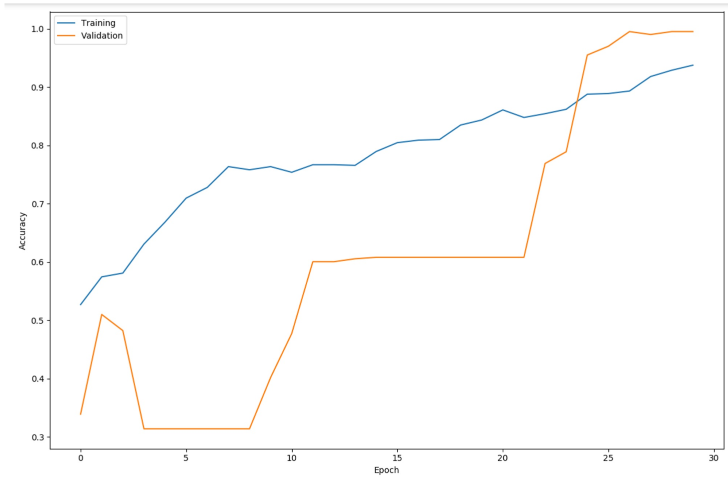

Figure 13: Training accuracy with epochs using 40% of the Failure Type 1 samples and 40% of the normal samples as the training dataset and without TL.

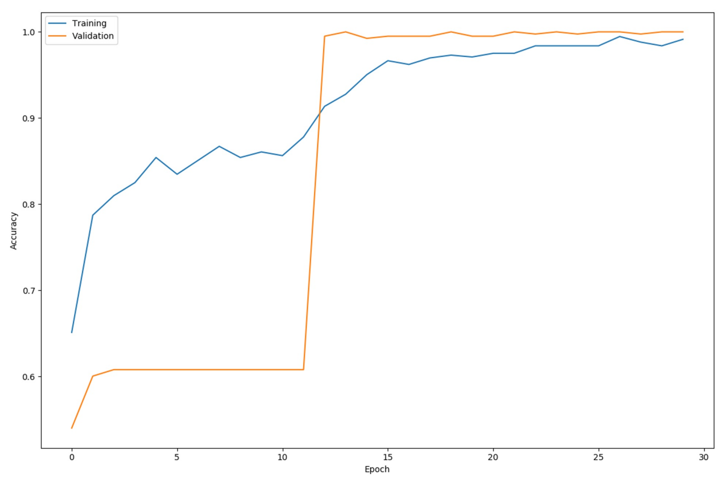

Figure 14: Training accuracy with epochs using 40% of the Failure Type 1 samples and 40% of the normal samples as the training dataset and TL.

The architecture of our proposed method is shown in Figure 15. Our proposed method includes three phases as follows:

- Pre-training phase. We first introduce sensor data and the corresponding ground truth (labels that can indicate the working condition of equipment). After feature engineering, the extracted features are applied to train a deep neural network for the source task (e.g., failure prediction with sufficient history data). The architecture and hyperparameters of the neural network model will be shared by the target task (e.g., failure prediction with insufficient data).

- Adaption phase. The training data may be insufficient to train the machine learning model for the target task alone. The model generated in the pre-training phase will be fine-tuned using the data of the target task.

- Prediction phase. Once the transfer learning-based deep neural network is adapted to the target task, we can use the network for predicting even though the available data for the target task is rare and insufficient for traditional methods. In this stage, the knowledge about source task, e.g., failure learned from similar failure types, can be inherited in the deep neural network and reused for prediction for the target task.

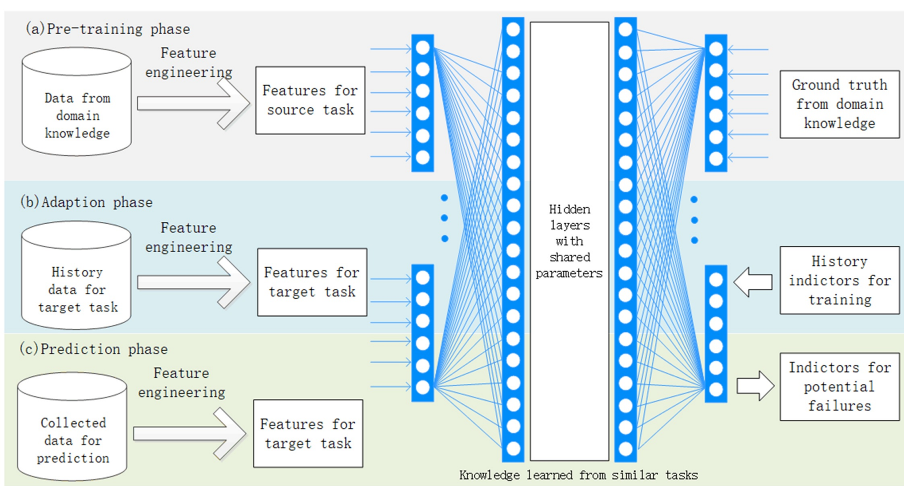

Figure 15: The architecture of deep transfer learning-based predictive maintenance method

6. Discussion

Quantitative or qualitative lack of data or labeled data is a common dilemma in practical applications of predictive maintenance [43][44]. To fill the gap and mitigate the impact of insufficient data, reusing of analytics insights, models, and data, also known as reusing analytics profiles [45], from related data or models have been proposed. Transfer learning is one possible approach for reusing analytics profiles.

Our experiments show that we can improve Type 1 failures' prediction accuracy using knowledge learned for predicting Type 2 failures. This improvement is more apparent when we use 20% of the data in the training dataset than when we use 40% of the data. For industrial practitioners, our results can provide valuable insights to improve the performance of machine learning when data related to some failure types are insufficient. Suppose a company wants to predict a particular type of failure but does not have sufficient data to train the neural network. In that case, the company can investigate the possibility of transferring knowledge learned for predicting other failure types for predicting this type of failure.

6.1. Comparison with related work

According to [11] [25], there are four categories of TL. Three categories of the TL approach, namely instance-based, feature-based, and parameter-based, have been used for predictive maintenance. The TL approach to transfer knowledge based on some defined relationship between the source and target domain has not been used, probably because it usually focuses on TL in the relational domain, which may be irrelevant to predictive maintenance data. Compared to our work, existing studies have either different focuses or use different approaches.

Instance-based TL was applied by Zhang et al. [18] to predict failures of minority disks using data acquired from majority disks. The purpose of [18] is to predict the same type of failure across systems, which is different from the purpose of our study.

Feature-based TL was used for predicting RUL and faults. In [14], feature-based TL was applied to predict the remaining useful life (RUL) across machines by leveraging the weights of models from the same kinds of machines. The RUL in their research is represented by testing the machine's tool wear during the experiment. Ragab et al. [15] also used feature-based TL and DNN for predicting the RUL of the same machine across operation conditions. Lu et al. [22] used DNN with domain adaption for fault diagnosis. Their approach can learn transferable features and strengthen the representative information obtained from the source domain to predict the faults of the target domain. However, their approach requires *"the prior known set of faults remains the same in source and target domains"*. Our approach focuses on using knowledge learned from one failure type to predict failures of another type. We do not focus on a set of faults in source and target domains. Wen et al. [23] use a three-layer sparse auto-encoder to extract the features of raw data and apply the maximum mean discrepancy term to minimise the discrepancy penalty between the features from training data and testing data. The purpose of the approach in [23] is to predict the same type of failure across operation conditions. The approach needs to select a proper third dataset closer to the target dataset than the source dataset and requires the same sample ratios in the source domain and the target domain. Different from [23], our approach targets failure prediction across failure types and does not need the third dataset when transferring knowledge from one failure type to another. Feng and Zhao [24] propose to use the attribute transfer method (similar to feature-based TL) to tackle the zero-sample fault diagnosis challenge, i.e., fault diagnosis when no samples of the target faults are available for model training. The advantage of their approach is that faults can be diagnosed based on the defined fault descriptions without any additional data-based training. However, it requires additional domain knowledge to form attribute descriptions in both training and target faults. Such domain knowledge may not be available for some types of failures. Our study shows that parameter-based deep transfer learning can transfer the knowledge learned from one type of failure to another type of failure without human intervention.

Parameter-based TL has been used in a few studies for predictive maintenance. Zhang et al. [16] used parameter-based TL to predict RUL across operation conditions. Our study focuses on predicting machinery failures, not its RUL. So, we do not use the sensor sequence information, such as in [16]. Zhang et al. [17] used parameter-based TL to transfer knowledge from the source domain to the target domain for fault diagnosis. Their neural network structure needs to be modified to dimensions of new data and labels. Our study focuses on transferring knowledge between different failure types on the same machine. Thus, the source domain neural network structure can be used as-is. Liu et al. [20] and Fan et al. [21] used convolutional neural network and model-based (equivalent to parameter-based) transfer learning models for fault diagnosis in building chillers. In their approach, the pre-trained neural network model is fine-tuned using some labeled target domain data. However, their studies and experiments focus on transfer learning across machines of the same type, meaning that they also expect the prior known set of faults remains the same in source and target domains. Cho et al. [19] studied how to use parameter-based TL to transfer knowledge on the same failure type, i.e., flaking failure, but under different working conditions, namely low speed and high speed. Although our study also applies parameter-based TL, different from [20] [21] [19], our study focuses on predicting different types of failures from the same machine, not the same failure type across machines or working conditions.

6.2. Implications to academia

Our study showed that parameter-based TL could help transfer knowledge and provide evidence to support our hypothesis that knowledge about failure among similar failure types is transferable. Furthermore, we noticed that TL could also help to accelerate training convergence in the second round of training. One possible reason is that the two data-driven models share specific parameters or similar distributions of the hyperparameters in the first several layers. For instance, they may share a way to extract certain key features from the raw data so neurons linked with the path will have similar values or distributions. Therefore, knowledge can be transferred by inheriting the architecture and hyperparameters in the trained network. Our experiments demonstrated the process that knowledge for identifying one failure type could be stored in the shared hyperparameters of deep neural networks and later be transferred to identify other failure types. We consider our findings could offer insights about using deep transfer learning to reuse knowledge across failure types in the research fields of predictive maintenance, which can supplement the existing knowledge of transferring failure prediction knowledge across machines and domains.

6.3. Known limitations

Our experiment successfully leveraged the data from one failure type of a rotating machine to predict another failure type for the same machine. Although the two failure types are among the most recurrent ones [27], generalizing the results of this study to other failure types of rotating machine [46], such as misalignment and mechanical looseness, needs to be evaluated further. In addition, we need to validate knowledge transfer across failure types of different equipment types.

Like many studies [17][18][19][20][21][22][23], our current work is limited to understanding the applicability of using TL to predict failures without considering the dynamic aspects of a system, e.g., degradation. Previous studies, e.g., [26] showed that DNN could also be used for degradation assessment. Understanding how to combine DNN and TL for degradation assessment can bring more insights to data-driven predictive maintenance.

7. Conclusion and future work

This study builds on previous deep learning and knowledge transfer research to develop a generic approach for predictive maintenance. We propose to leverage knowledge transfer between failure types to increase overall predictive maintenance feasibility and productivity gains for firms. The proposed approach could resolve the data insufficiency challenge, which is one of the main challenges for predictive maintenance. The theoretical basis of the proposed approach is the hypothesis that knowledge about failure among similar failure types is transferable. We designed a two-step training procedure to validate the hypothesis to transfer the knowledge learned from one failure type to another. The study results provided substantial evidence to support our hypothesis. Our experiment considers that the two tasks are similar because the data are collected from the same equipment through the same collection system. To the best of our knowledge, our study is the first to automatically transfer knowledge about impending failure identification across different failure types.

Our study is limited to knowledge transfer from one failure type to another in the same equipment. One of our planned future studies is to study knowledge transfer among different failure types across multiple types of equipment. Another one is to understand how to use DNN and TL to facilitate degradation assessment in predictive maintenance.

References

[1] M. Garetti, M. Taisch, Sustainable manufacturing: trends and research challenges, Production Planning & Control 23 (2-3) (2012) 83–104. arXiv:https://doi.org/10.1080/09537287.2011.591619, doi:10.1080/09537287.2011.591619.
URL https://doi.org/10.1080/09537287.2011.591619

[2] Y. Bao, Z. Chen, S. Wei, Y. Xu, Z. Tang, H. Li, The state of the art of data science and engineering in structural health monitoring, Engineering 5 (2) (2019) 234–242. doi:https://doi.org/10.1016/j.eng.2018.11.027.
URL https://www.sciencedirect.com/science/article/pii/S2095809918308026

[3] L. Guo, Y. Lei, S. Xing, T. Yan, N. Li, Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with

unlabeled data, IEEE Transactions on Industrial Electronics 66 (9) (2019) 7316–7325. doi:10.1109/TIE.2018.2877090.

[4] Z. Li, J. Li, Y. Wang, K. Wang, A deep learning approach for anomaly detection based on sae and lstm in mechanical equipment, The International Journal of Advanced Manufacturing Technology 103 (1) (2019) 499–510. doi:10.1007/s00170-019-03557-w.
URL https://doi.org/10.1007/s00170-019-03557-w

[5] K. T. Nguyen, K. Medjaher, A new dynamic predictive maintenance framework using deep learning for failure prognostics, Reliability Engineering System Safety 188 (2019) 251–262. doi:https://doi.org/10.1016/j.ress.2019.03.018.
URL https://www.sciencedirect.com/science/article/pii/S0951832018311050

[6] T. Wang, Q. Han, F. Chu, Z. Feng, Vibration based condition monitoring and fault diagnosis of wind turbine planetary gearbox: A review, Mechanical Systems and Signal Processing 126 (2019) 662–685. doi:https://doi.org/10.1016/j.ymssp.2019.02.051.
URL https://www.sciencedirect.com/science/article/pii/S0888327019301426

[7] Z. Li, Y. Wang, K. Wang, A data-driven method based on deep belief networks for backlash error prediction in machining centers, J. Intell. Manuf. 31 (7) (2020) 1693–1705. doi:10.1007/s10845-017-1380-9.
URL https://doi.org/10.1007/s10845-017-1380-9

[8] M. Xia, T. Li, T. Shu, J. Wan, C. W. de Silva, Z. Wang, A two-stage approach for the remaining useful life prediction of bearings using deep neural networks, IEEE Transactions on Industrial Informatics 15 (6) (2019) 3703–3711. doi:10.1109/TII.2018.2868687.

[9] L. Ren, Y. Sun, H. Wang, L. Zhang, Prediction of bearing remaining useful

life with deep convolution neural network, IEEE Access 6 (2018) 13041–13049. doi:10.1109/ACCESS.2018.2804930.

[10] M. E. Taylor, P. Stone, Transfer learning for reinforcement learning domains: A survey, J. Mach. Learn. Res. 10 (2009) 1633–1685.

[11] K. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, Journal of Big Data 3 (1) (2016) 9. doi:10.1186/s40537-016-0043-6. URL https://doi.org/10.1186/s40537-016-0043-6

[12] C. Battistella, A. F. De Toni, R. Pillon, Inter-organisational technology/knowledge transfer: a framework from critical literature review, The Journal of Technology Transfer 41 (5) (2016) 1195–1234. doi:10.1007/s10961-015-9418-7. URL https://doi.org/10.1007/s10961-015-9418-7

[13] J.-T. Huang, J. Li, D. Yu, L. Deng, Y. Gong, Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 7304–7308. doi:10.1109/ICASSP.2013.6639081.

[14] C. Sun, M. Ma, Z. Zhao, S. Tian, R. Yan, X. Chen, Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing, IEEE Transactions on Industrial Informatics 15 (4) (2019) 2416–2425. doi:10.1109/TII.2018.2881543.

[15] M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, Adversarial transfer learning for machine remaining useful life prediction, in: 2020 IEEE International Conference on Prognostics and Health Management (ICPHM), 2020, pp. 1–7. doi:10.1109/ICPHM49022.2020.9187053.

[16] A. Zhang, H. Wang, S. Li, Y. Cui, Z. Liu, G. Yang, J. Hu, Transfer learning with deep recurrent neural networks for remaining useful life estimation, Applied Sciences 8 (12). doi:10.3390/app8122416. URL https://www.mdpi.com/2076-3417/8/12/2416

[17] R. Zhang, H. Tao, L. Wu, Y. Guan, Transfer learning with neural networks for bearing fault diagnosis in changing working conditions, IEEE Access 5 (2017) 14347–14357. doi:10.1109/ACCESS.2017.2720965.

[18] J. Zhang, K. Zhou, P. Huang, X. He, Z. Xiao, B. Cheng, Y. Ji, Y. Wang, Transfer learning based failure prediction for minority disks in large data centers of heterogeneous disk systems, in: Proceedings of the 48th International Conference on Parallel Processing, ICPP 2019, Association for Computing Machinery, New York, NY, USA, 2019. doi:10.1145/3337821.3337881.
URL https://doi.org/10.1145/3337821.3337881

[19] S. H. Cho, S. Kim, J.-H. Choi, Transfer learning-based fault diagnosis under data deficiency, Applied Sciences 10 (21). doi:10.3390/app10217768.
URL https://www.mdpi.com/2076-3417/10/21/7768

[20] J. Liu, Q. Zhang, X. Li, G. Li, Z. Liu, Y. Xie, K. Li, B. Liu, Transfer learning-based strategies for fault diagnosis in building energy systems, Energy and Buildings 250 (2021) 111256. doi:https://doi.org/10.1016/j.enbuild.2021.111256.
URL https://www.sciencedirect.com/science/article/pii/S0378778821005405

[21] C. Fan, X. Liu, P. Xue, J. Wang, Statistical characterization of semi-supervised neural networks for fault detection and diagnosis of air handling units, Energy and Buildings 234 (2021) 110733. doi:https://doi.org/10.1016/j.enbuild.2021.110733.
URL https://www.sciencedirect.com/science/article/pii/S0378778821000177

[22] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, T. Zhang, Deep model based domain adaptation for fault diagnosis, IEEE Transactions on Industrial Electronics 64 (3) (2017) 2296–2305. doi:10.1109/TIE.2016.2627020.

[23] L. Wen, L. Gao, X. Li, A new deep transfer learning based on sparse auto-encoder for fault diagnosis, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (1) (2019) 136–144. doi:10.1109/TSMC.2017.2754287.

[24] L. Feng, C. Zhao, Fault description based attribute transfer for zero-sample industrial fault diagnosis, IEEE Transactions on Industrial Informatics 17 (3) (2021) 1852–1862. doi:10.1109/TII.2020.2988208.

[25] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1345–1359. doi:10.1109/TKDE.2009.191.

[26] Z. Li, Y. Wang, K. Wang, A deep learning driven method for fault classification and degradation assessment in mechanical equipment, Computers in Industry 104 (2019) 1–10. doi:https://doi.org/10.1016/j.compind.2018.07.002.
URL https://www.sciencedirect.com/science/article/pii/S0166361517307832

[27] I. Torres-Contreras, J. C. Jáuregui-Correa, S. Echeverría-Villagómez, J. P. Benítez-Rangel, S. Camacho-Martínez, Diagnosis of friction on an unbalanced rotor by phase-shift empirical mode decomposition integration and recurrence plot, Applied Sciences 11 (17). doi:10.3390/app11177973.
URL https://www.mdpi.com/2076-3417/11/17/7973

[28] Q. Hu, Z. He, Z. Zhang, Y. Zi, Fault diagnosis of rotating machinery based on improved wavelet package transform and svms ensemble, Mechanical Systems and Signal Processing 21 (2) (2007) 688–705. doi:https://doi.org/10.1016/j.ymssp.2006.01.007.
URL https://www.sciencedirect.com/science/article/pii/S0888327006000306

[29] M. Murugappan, N. Ramachandran, Y. Sazali, Classification of human

emotion from eeg using discrete wavelet transform, Journal of Biomedical Science and Engineering 03 (2010) 390–396.

[30] C. B. R. Ferreira, D. L. Borges, Analysis of mammogram classification using a wavelet transform decomposition, Pattern Recognition Letters 24 (7) (2003) 973–982. doi:https://doi.org/10.1016/S0167-8655(02)00221-0.
URL https://www.sciencedirect.com/science/article/pii/S0167865502002210

[31] M. Long, Y. Cao, J. Wang, M. I. Jordan, Learning transferable features with deep adaptation networks, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, JMLR.org, 2015, p. 97–105.

[32] J. Hu, J. Lu, Y.-P. Tan, Deep transfer metric learning, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 325–333. doi:10.1109/CVPR.2015.7298629.

[33] E. Tzeng, J. Hoffman, T. Darrell, K. Saenko, Simultaneous deep transfer across domains and tasks, CoRR abs/1510.02192. arXiv:1510.02192.
URL http://arxiv.org/abs/1510.02192

[34] R. Miyajima, Deep learning triggers a new era in industrial robotics, IEEE MultiMedia 24 (4) (2017) 91–96. doi:10.1109/MMUL.2017.4031311.

[35] N. Ramesh Babu, B. Jagan Mohan, Fault classification in power systems using emd and svm, Ain Shams Engineering Journal 8 (2) (2017) 103–111. doi:https://doi.org/10.1016/j.asej.2015.08.005.
URL https://www.sciencedirect.com/science/article/pii/S2090447915001306

[36] Z. Zhang, J. Zhao, A deep belief network based fault diagnosis model for complex chemical processes, Computers Chemical Engineering 107 (2017) 395–407, in honor of Professor Rafiqul Gani.

doi:https://doi.org/10.1016/j.compchemeng.2017.02.041.
URL https://www.sciencedirect.com/science/article/pii/S0098135417301059

[37] D. Ha, U. Ahmed, H. Pyun, C.-J. Lee, K. H. Baek, C. Han, Multi-mode operation of principal component analysis with k-nearest neighbor algorithm to monitor compressors for liquefied natural gas mixed refrigerant processes, Computers Chemical Engineering 106 (2017) 96–105, eSCAPE-26. doi:https://doi.org/10.1016/j.compchemeng.2017.05.029.
URL https://www.sciencedirect.com/science/article/pii/S0098135417302466

[38] A. Asuhaimi Mohd Zin, M. Saini, M. W. Mustafa, A. R. Sultan, Rahimuddin, New algorithm for detection and fault classification on parallel transmission line using dwt and bpnn based on clarke's transformation, Neurocomputing 168 (2015) 983–993. doi:https://doi.org/10.1016/j.neucom.2015.05.026.
URL https://www.sciencedirect.com/science/article/pii/S0925231215006669

[39] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, book in preparation for MIT Press.
URL http://www.deeplearningbook.org

[40] D. Kingma, J. Ba, Adam: A method for stochastic optimization, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015 (2014).
URL http://arxiv.org/abs/1412.6980

[41] S. Ruder, An overview of gradient descent optimization algorithms, CoRR abs/1609.04747. arXiv:1609.04747.
URL http://arxiv.org/abs/1609.04747

[42] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), 1st Edition, Springer, 2007.

[43] J. Zenisek, F. Holzinger, M. Affenzeller, Machine learning based concept drift detection for predictive maintenance, Computers Industrial Engineering 137 (2019) 106031. doi:https://doi.org/10.1016/j.cie.2019.106031.
URL https://www.sciencedirect.com/science/article/pii/S0360835219304905

[44] Z. Li, J. Li, Y. Wang, K. Wang, A deep learning approach for anomaly detection based on sae and lstm in mechanical equipment, The International Journal of Advanced Manufacturing Technology 103 (1) (2019) 499–510. doi:10.1007/s00170-019-03557-w.
URL https://doi.org/10.1007/s00170-019-03557-w

[45] E. Kristoffersen, O. O. Aremu, F. Blomsma, P. Mikalef, J. Li, Exploring the relationship between data science and circular economy: An enhanced crisp-dm process model, in: I. O. Pappas, P. Mikalef, Y. K. Dwivedi, L. Jaccheri, J. Krogstie, M. Mäntymäki (Eds.), Digital Transformation for a Sustainable Society in the 21st Century, Springer International Publishing, Cham, 2019, pp. 177–189.

[46] C. Mongia, D. Goyal, S. Sehgal, Vibration response-based condition monitoring and fault diagnosis of rotary machinery, Materials Today: Proceedings 50 (2022) 679–683, 2nd International Conference on Functional Material, Manufacturing and Performances (ICFMMP-2021). doi:https://doi.org/10.1016/j.matpr.2021.04.395.
URL https://www.sciencedirect.com/science/article/pii/S2214785321033150