

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Contextual knowledge-informed deep domain generalization for bearing fault diagnosis

ADAM LUNDSTRÖM^{1,2}, MATTIAS O'NILS¹ AND FAISAL Z. QURESHI^{1,3}, (Senior Member, IEEE)

¹Department of Computer and Electrical Engineering, Mid Sweden University, 85170 Sundsvall, Sweden

²SCA, 85188, Sundsvall, Sweden (e-mail: adam.lundstrom2@sca.com)

³Faculty of Science, University of Ontario Institute of Technology, Oshawa, ON L1G 0C5, Canada (e-mail: faisal.qureshi@ontariotechu.net)

Corresponding author: Adam Lundström (e-mail: adam.lundstrom@miun.se).

This work was supported in part by The Knowledge Foundation (kks.se) within the Industrial graduate school Smart Industry Sweden.

ABSTRACT Reliable methods for bearing fault diagnosis are of great importance because they provide the possibility of preventing failures in machines. A significant challenge is developing solutions that can handle the variances in the data across different domains i.e., the operational context of the bearing and settings including noise, bearing type, rotational speed, and sampling frequency. To solve this issue, a common approach is to use transfer learning. However, most existing methods either assume that faults are available in the target domain or that only operational conditions on a single machine are changed between the source and target domain. Unfortunately, neither of these assumptions can be made in an industrial environment where there is a need to deploy the method on new machines with no historical faults. Therefore, there is a need to develop methods for bearing fault diagnosis that can provide accurate predictions on a general basis without needing access to historical faults from the target domain. To address this problem, this study develops a novel method including a knowledge-enriched standardization procedure used to lower the domain shift, a feature-targeted metric learning procedure enriched with contextual data that can be deployed to any new domain without having access to historical faults. The method is compared against state-of-the-art solutions in three cases with various tasks where it achieves the highest accuracy and displays high robustness and generalizability. The results also show a low level of false positives, a high level of fault segment recall and the possibility of giving feedback to the user about the model's decision. The conclusion is that the method can be used on a general basis for fault diagnosis of bearings without having access to historical faults from the target domain.

INDEX TERMS deep domain generalization, fault transfer diagnosis, rolling bearing, rotating machinery, XAI

I. INTRODUCTION

Bearings are components that are widely used in rotating equipment and a common cause of machine failure, which is why bearing fault diagnosis is of great importance [1]. Because of this, methods ranging from physics-based [2] to machine learning-based [3] have been suggested and recently, deep learning (DL) has become prominent in developing solutions for fault detection, based on vibration data [4]–[6]. This includes methods based on few-shot learning [7] and transfer learning [6], [8], [9] which can limit the need for labelled examples of faults. A major challenge in the development of general solutions is the heterogeneous nature of the data from different domains i.e., the operational context of the bearing and settings including noise, bearing type, rotational speed,

and sampling frequency [10]. This implies that a solution that works well in one domain does not necessarily work well in another. In combination with the lack of historical examples of faults in the industrial environment, as argued in [11] and shown in [10], the development of methods that perform well independently of the domain is of great value. However, many of the current methods assume that historical faults from the target domain are available during training which means that they might not be applicable in the industrial context [8], [11]. In addition, a large body of research based on more generalized methods, including, [12]–[18] present their result on scenarios where only operational conditions on a single machine are changed between the source and target domain. Because industrial applications will include domain transfer

between machines, these solutions might not work well in practice.

To address these issues, this study aims to develop a new method based on DL called Contextual Knowledge-Informed Deep Domain Generalization (ConKIDDG) for Bearing Fault Diagnosis which achieves domain generalization with high accuracy on faults from unseen domains. The main contribution of the study is that it:

- Proposes a novel method that achieves domain generalization that can be applied to new machines without having access to historical faults from the target domain and performs better than state-of-the-art solutions.

To achieve this, the following novel approaches are suggested:

- A feature extraction technique that can increase domain generalization. This consists of a signal standardization procedure based on prior knowledge used to reduce the domain shift in combination with separate feature extractors designed to target specific features in the data.
- A historical context during training that increases the robustness of the model.

An overview of how the method can be applied in practical scenarios is shown in Figure 1. A model is developed using normal data from the target domain (only if available) and multiple known source domains from different machines and operational conditions. The method comprises two different parts each important to achieve robust domain generalization:

- The first is signal standardization based on known fault characteristics which relate to the vibration frequencies of the bearing components that reduce the domain shift which is shown to not only improve the generalizability of the model but also make it possible to give the user insightful feedback about the model's decision.
- The second is the model development, specifically designed to minimize the discrepancies between domains, capture the most important features related to the health state of the bearing, and make predictions based on a historical context. This approach is shown to increase the performance of the model significantly.

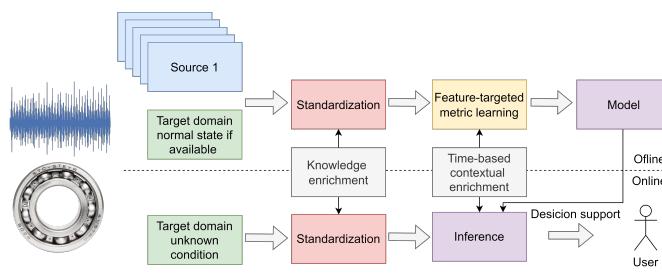


FIGURE 1. Overview of the key components of the method.

II. EXTENDED BACKGROUND

Several different DL methods have been developed to solve the issue with cross-domain bearing fault diagnosis. Firstly,

different levels of preprocessing exist and vary between using raw vibration data to transform the data into the frequency domain [8], [3]. Furthermore, additional preprocessing to ease the learning process exists such as standardizing the data based on known physical characteristics. One example is [9] where the fault frequencies of the bearing were used to standardize the enveloped raw vibration signal.

Furthermore, several different types of transfer learning mechanisms exist applied together with a classifier that enforces the algorithm to generalize its understanding of health states across domains. Two of the main types are adversarial and metric learning [8]. Both techniques can be used for domain adaption by reducing discrepancies in the features generated from a feature extractor between a labelled source and an unlabeled target domain. In addition, it is possible to use them to enforce a network to discard dissimilarities for intra-class samples and enforce differences between inter-class samples to increase the generalizability.

Previous studies can be divided based on the type of problem they intend to solve which is also summarized in Table 1. One area is based on the problem with partial, complete, universal, or open set domain adaption using single or multiple domains [8]. Because this study only considers bearing fault diagnosis where the number of faults is limited, only partial or complete domain adaption is covered. In those cases, unlabeled fault data from the target domain is included in the training data where it is adapted to the health states from the source domain in the training procedure [8]. For example, [19] used adversarial learning to adapt the health states of a bearing from a source to an unlabeled target domain with different operational conditions on a single machine using the features extracted from the bottleneck. Also, [20] used Joint Maximum Mean Discrepancy (JMMD) and adversarial learning for domain adaption of bearing health states using features from the bottleneck from a labelled source domain to an unlabeled target domain with different working conditions on a single machine. Lastly, [21] used adversarial loss and Multi-Kernel Maximum Mean Discrepancy (MK-MMD) for domain adaption of bearings from different machines and operations conditions. A major issue with these approaches is that they assume that faults in the target domain are available which is not necessarily the case in industrial applications [10].

To solve this issue there exist methods built for domain generalization where no faults in the target domain are accessible during training including [9], [11], [22] and [12]–[18]. For example, [11], applied adversarial learning and batch spectral penalization (BSP) to achieve domain generalization using multiple sources from different machines and operational conditions. Furthermore, [9] used instance-based discriminative loss to enforce domain generalization using multiple sources from different conditions and machines. Lastly, [12] used center loss to achieve domain generalization for bearings with different operational conditions. However, many studies, counting [12]–[18] consider the transfer scenario where the only difference between the source and target domain are

the operational conditions on a single machine. This means that the methods are not evaluated considering the variations found in different machines.

There exist methods that consider knowledge transfer between different machines and that historical faults do not exist in the target domain during training. Three of these are [9], [11], and [22] which use several types of sources from different machines and operational conditions to achieve domain generalization which they demonstrate on different transfer tasks. Despite this, as argued in [23], considering the limited research for this scenario, improving the performance of solutions is of great value since they replicate the setting found in industrial applications. Therefore, this study suggests a new method that can outperform state-of-the-art methods. This study differs from previous studies by using a novel standardization procedure based on the enveloped frequency spectrum and separate feature extractors to improve generalizability across domains.

TABLE 1. Summary, issues and examples from the literature of methods for bearing fault diagnosis across domains.

General idea	Main issue
Adapting using faults from the target domain [19]–[21]	Faults from the target domain are not necessarily available in industrial applications. There is a need to transfer between different machines.
Domain generalization same machine different operational conditions [12]–[18]	Considering the limited studies for this scenario there is a need to evaluate new methods in order to improve the performance.
Domain generalization across machines [9], [11], [22]	

III. METHOD

A. PROBLEM FORMULATION

We are provided with a set of raw vibration data for training called source data from multiple domains, d , where each domain represents a bearing in an operational condition. This data is defined as $\mathcal{D}_s^j = \{(\mathbf{x}_i^j, \mathbf{y}_i^j)\}_{i=1}^{n_s^j}$ where n_s^j is the number of samples in a domain j and \mathbf{y}_i^j are one-hot-encoded ground truth labels of the source data which contain the possible health states normal (N) which means that no bearing-related fault is present, inner ring fault (IR), outer ring fault (OR) and rolling element fault (RF).

Furthermore, the assumption is that the sampling rate $r = \{r^j\}_{j=1}^d$ in Hz is given, as well as the fault frequencies, $f = \{(f_1^j, f_2^j, f_3^j)\}_{j=1}^d$ in Hz where f_1^j is the inner ring fault frequency, f_2^j the rolling element fault frequency and f_3^j the outer ring fault frequency, for a domain j . The aim is to learn from the source data and predict health states $\hat{\mathbf{y}}$ from a dataset $\mathcal{D}_t = \{(\mathbf{x}_i^t, \mathbf{y}_i^t)\}_{i=1}^{n_t}$ that contains health states where the faults are unknown and come from a new machine. Regarding the label space, to replicate real scenarios the assumption is that the number of health states in the target domain is unknown but that all types of conditions in the target domain exist in some source domain. Since the method should be robust

towards abnormal events unrelated to a bearing fault it should predict N in these cases.

Lastly, because historical vibration data of state N often exist or can be collected from the industrial environment, some samples of state N from the target domain are allowed to be used for training when available. When no samples of type N are given, no data from the target domain are used for training.

B. STANDARDIZATION

The standardization procedure is depicted in Figure 2. It uses known fault characteristics of bearings to transform and standardize the raw vibration measurement to limit the domain shift from differences in bearing type, sampling frequency, amplitude and rotational speed between domains. To achieve this the method needs the raw vibration measurement, the sampling frequency, the rotational speed and the bearing specification (which makes it possible to determine the fault frequencies). The suggested method utilizes the enveloped frequency spectrum's properties, enabling an efficient signal standardization that can assist a deep neural network to perform accurate diagnoses across different domains. This, together with the rest of the method, displays robustness across various tasks and outperforms the method suggested in [9] which uses a different form of signal standardization.

1) Step 1 - Extraction

Initially, the data is collected from each measurement via a sliding window. The size of the window, s^j , for a domain j , is standardized using the inner ring frequency f_1^j and sampling rate r^j so that each window will have a similar measurement time in relation to the fault frequency:

$$s^j = r^j \frac{k}{f_1^j}, \quad (1)$$

where k is the number of rotations. To simplify the description the assumption is that the sampling rate for each measurement in a domain is constant. Choosing a larger k value will give a higher level of detail because of the larger amount of rotations if the shaft rotational speed is approximately constant during the measurement.

2) Step 2 - Transformation

In step 2 each window of \mathbf{x} is transformed into \mathbf{z} by creating enveloped spectrums using Hilbert Transform (HT) and Fast Fourier Transform (FFT) (explained in depth in [24] and [1]). This means that the vibrations from the bearing's components carried by high-frequency resonance are demodulated meaning that peaks in the frequency spectrum occur at the respective fault frequencies. This is advantageous because it makes it possible to limit the input size and, in the standardization, disregard the variation in frequency range due to the difference in sampling rate between domains.

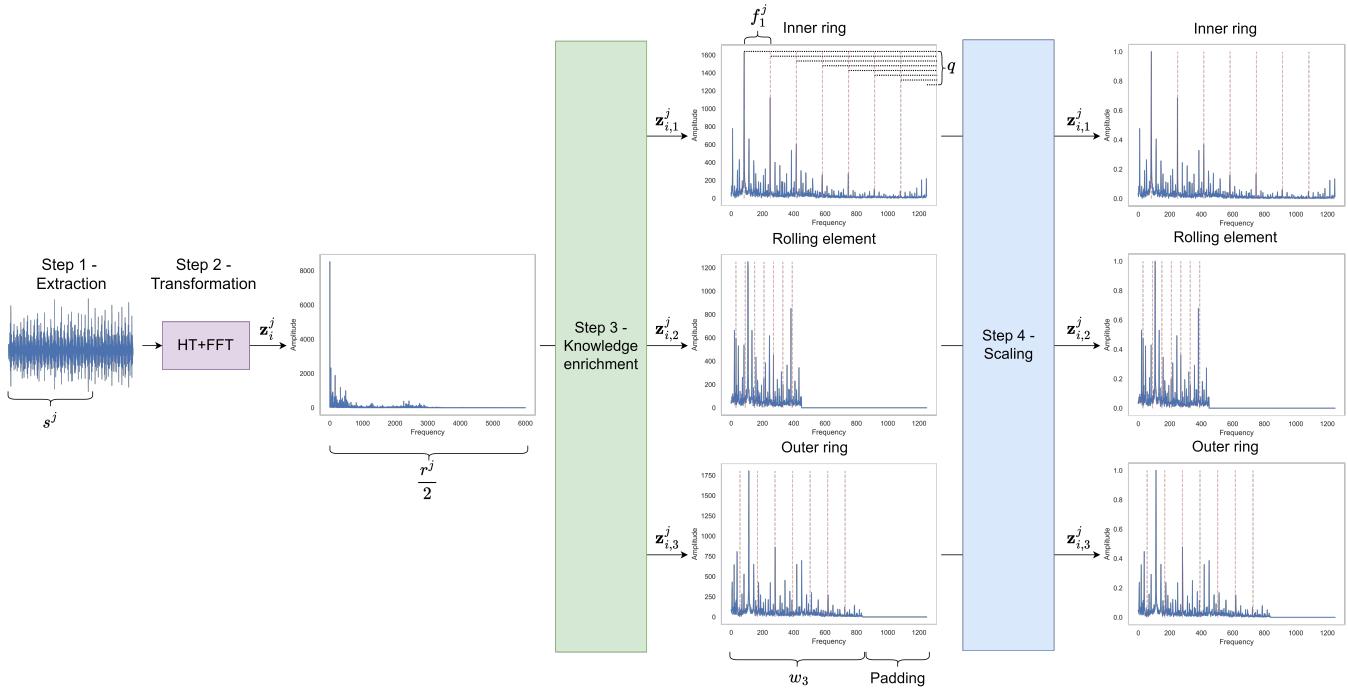


FIGURE 2. Overview of the preprocessing method. It includes transforming the data into the enveloped frequency spectrum and standardizing based on the known fault characteristics of bearings.

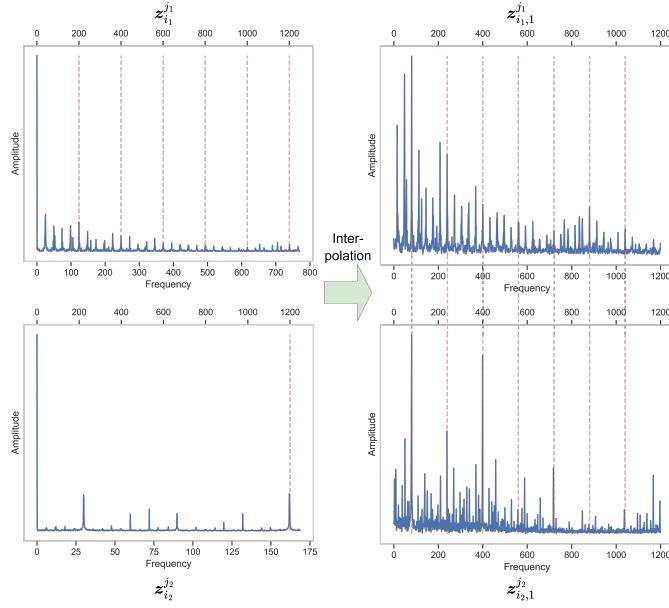


FIGURE 3. Illustration of the knowledge enrichment based on the inner ring frequency. The fault frequency and its harmonics are shown as the dashed red line.

3) Step 3 - Knowledge enrichment

To incorporate prior knowledge considering known fault characteristics of bearings into the learning procedure, the data is standardized based on the fault frequencies of the bearing. This is achieved by splitting the processed signal \mathbf{z} , and

for each element standardize the signal with an empirically specified multiple, q , of the respective fault frequencies. The procedure, which is illustrated for the inner ring frequency in Figure 3, is carried out for each sample i in domain j using a piecewise linear interpolation function:

$$\forall a \in \{1, 2, 3\} : \mathbf{z}_{i,a}^j = \text{interpolate}(\mathbf{F}', \mathbf{F}, \mathbf{X}_i^j), \quad (2)$$

where $(\mathbf{F}_m)_{m=1}^{j^j} = (0, \dots, \frac{r^j}{2})$ considering the Nyquist Theorem is the original frequency range, $(\mathbf{F}'_m)_{m=1}^{w_a} = (\frac{f_a}{2}, \dots, qf_a)$ is the new frequency range, \mathbf{X} represent \mathbf{z} before the operation and w is the specified length of each sequence for all domains so that it approximately corresponds to the expected size of the preprocessed signal considering the sampling size and q . Notice that the first part of the interpolated signal is removed to simplify it further. Lastly, padding is added so that each sequence has the same size.

4) Step 4 - Scaling

Since the amplitude of the data can differ significantly between different domains, normalization is performed so that for a domain i , domain j and element a , \mathbf{z}_a becomes:

$$\forall \ell \in \{1, 2, \dots, m\} : \mathbf{z}_{a,\ell} = \frac{\mathbf{z}_{a,\ell} - \min(\mathbf{z}_a)}{\max(\mathbf{z}_a) - \min(\mathbf{z}_a)}, \quad (3)$$

where j and i are excluded for simplification.

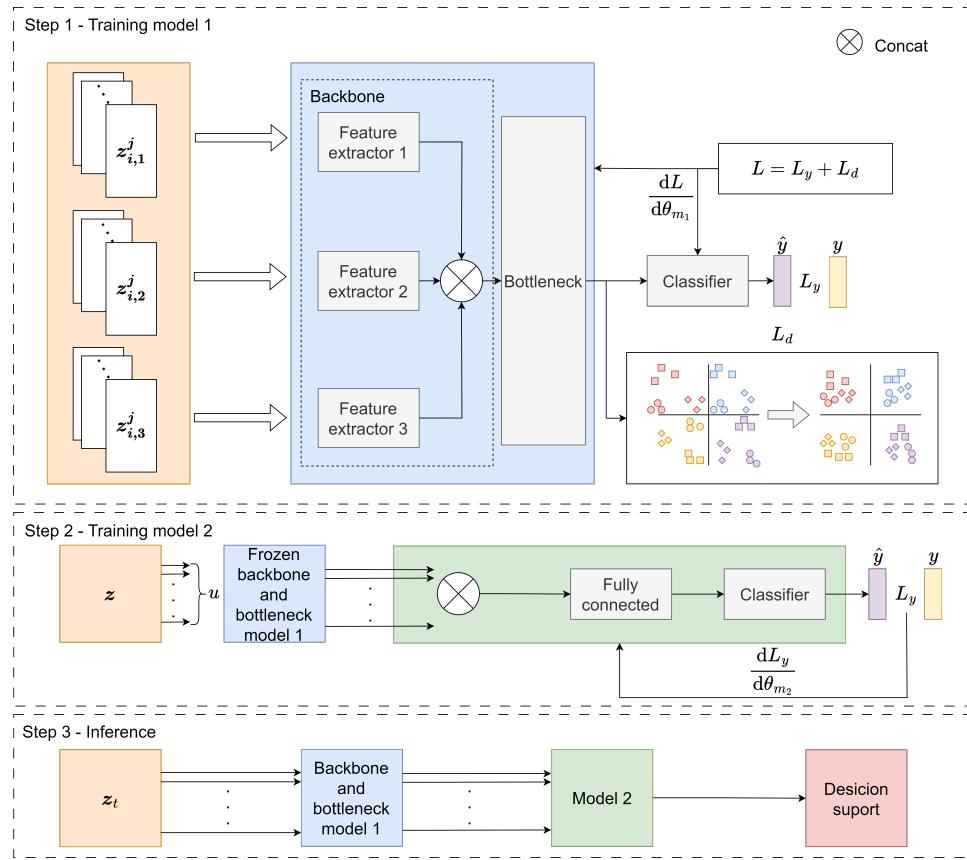


FIGURE 4. Description of the model development. The training is conducted in two steps. In the first step, a model is developed based on the output of the standardization procedure. In the second step, a model is constructed using historical outputs from the bottleneck of model 1 as context. No faults from the target domain are available during training.

C. MODEL DEVELOPMENT

The overview of the model development is illustrated in Figure 4 where z is used to represent the training data constructed from \mathcal{D}_s (and normal samples from \mathcal{D}_t if available) and z_t for the test data based on \mathcal{D}_t . It consists of three different steps, two of the development itself and one stage describing the inference. The training is carried out in two separate steps. In the first step, a metric-based learning approach is deployed with a separated feature extractor to reduce the discrepancies between domains further and simultaneously increase the capability of learning the patterns related to the different fault types. In the second step, a contextual layer is added to increase the robustness.

1) Training stage 1 - Model 1

The layers of training stage 1 are presented in Table 2. Since the three elements of the data input differ, a feature extractor is used for each of them in the backbone and then concatenates the output. This allows the extractor to learn features related to each of the inputs that correspond to the fault types and the weights between the feature extractors are not shared. Then the data is processed through the bottleneck and lastly, the classifier.

The learning procedure has two different objectives. The

first is to learn to classify based on the health state of each input. This loss is defined as L_y and is based on the output of the classifier layer \hat{y} and the ground truth y . The loss is calculated using categorical cross-entropy:

$$L_y = - \sum_{i=1}^b \sum_{j=1}^c y_i^j \log \hat{y}_i^j, \quad (4)$$

where b is the batch size and c is the number of health states. The second objective is to enforce the feature extractor to minimize the distance between the intra-class samples and maximize the distance between the inter-class samples. This means the network is trained to ignore differences between domains within the same class and maximize the difference between classes regardless of domain. The purpose of doing this is to enforce the network to increase its generalizability. Because some samples in the source data are expected to be outliers and not representative of a health state, a method that encourages the mass of features from the same class towards a common objective rather than considering the distance between individual features to reduce the risk of overfitting is requested. Therefore the features from the output of the bottleneck, $\mathbf{h} \in \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_b\}$, are used for one batch and the loss L_d is defined as:

$$L_d = \frac{1}{2cb} \sum_{i=1}^c J(\mathbf{h}, \mathbf{y}, i), \quad (5)$$

where:

$$J(\mathbf{h}, \mathbf{y}, i) = \begin{cases} (||\mathbf{X}||_F)^2 & \text{if } \forall j \in \{1, 2, \dots, l_1\} : (y_j = i) \\ \max(0, (m - ||\mathbf{X}||_F))^2 & \text{if } \forall j \in \{1, 2, \dots, l_2\} : (y_j \neq i). \end{cases} \quad (6)$$

TABLE 2. Description of the layers in training stage 1.

Part	Layers	Parameters	Activation
Feature extractor	Conv 1D + Batch Normalization	Kernel size=15, Channels=16	ReLU
	Max Pooling	Pool size=2	-
	Conv 1D + Batch Normalization	Kernel size=3, Channels=32	ReLU
	Max Pooling	Pool size=2	-
	Conv 1D + Batch Normalization	Kernel size=3, Channels=64	ReLU
	Max Pooling	Pool size=2	-
	Conv 1D + Batch Normalization	Kernel size=3, Channels=128	ReLU
	Max Pooling	Pool size=2	-
Bottleneck	Flatten	-	-
	Fully connected	Units=256	ReLU
Classifier	Dropout	Rate=0.4	-
	Fully connected	Units=4	Softmax

In equation 6, $|| \cdot ||_F$ is the Frobenius norm, $\mathbf{X}_j \in \mathbf{h}$, l_1 and l_2 the number of samples in each of the cases. The intention is to minimize the total distance of the features from the same class and simultaneously maximize the distance of the features from the other classes. Intuitively, the network will encourage the Frobenius norm of all features from the same class to move together towards zero, regardless of the domain and simultaneously make the norm of the features from the other classes move away from zero. The latter is regulated by the margin m , where a high value will increase the encouragement and a low will do the opposite.

After defining L_y and L_d , the combined learning objective becomes:

$$L = L_y + \lambda L_d, \quad (7)$$

where λ controls the influence of L_d .

2) Training stage 2 - Model 2

Since model 1 only takes one input at a time it might make inaccurate predictions when readings are unusual. To create a more robust method a second model is built that uses historical context to make the method less sensitive to outliers. In particular, the primary purpose is to reduce the amount of false positives for faults. To achieve this, the output from the bottleneck of model 1, \mathbf{h} , is used and for each input, the previous u outputs are concatenated. Then one fully connected

layer and a classifier are used to achieve classification considering a historical context. The parameters for the different layers are specified in Table 3. In the training to create model 2, the weights in the bottleneck and feature extractor in model 1 are frozen.

TABLE 3. Description of the layers for training stage 2.

Layers	Parameters	Activation
Dropout	Rate=0.4	-
Flatten	-	-
Fully connected	Units=64	ReLU
Fully connected	Units=4	Softmax

3) Inference

When both training procedures have been carried out the output of model 2 is used, based the output of the bottleneck from model 1 as input, to determine the health state of the bearing.

IV. EXPERIMENTS

A. DATASETS

This study uses eight publicly available datasets with different types of faults from various machines and operational conditions. It features artificially damaged bearings (for example the CWRU dataset), naturally damaged bearings (for example the PU dataset) and a dataset from the industrial environment (the SCA dataset).

1) CWRU

The Case Western University bearing dataset [25] contains vibration measurements of bearings with health states N, IF, OF and RF. It includes three different setups: 1) drive side with bearing 6205-2RS JEM SCA and a sampling rate of 64 kHz, 2) drive side with bearing 6205-2RS JEM SCA and sampling rate 12 kHz and 3) bearing 6203-2RS JEM SCA on the fan side with sampling rate 12kHz. Furthermore, each fault was induced by an electro-discharge machining (EDM) with different sizes. In this study, only 0.007, 0.014 and 0.021 inches are used. It also has four different rotational speeds of the shaft which are 1797, 1772, 1750 and 1730 RPM. It should be noted that for the OF, only the centred position is used in this study.

2) PU

The Paderborn (PU) dataset [26] includes vibration measurements of bearing conditions N, IF and OR. All files contain a measurement with a sampling rate of 64 kHz and with the bearing 6203. It has 5 different damage levels with faults artificially created by EDM, drilling and an electric engraver but also natural faults from lifetime tests. In addition, it includes tests with different loads and rotating speeds of the shaft and can be retrieved from [27].

3) JNU

The JNU bearing dataset [28] includes the health states N, OF, IF, RF with three different rotational speeds of the shaft 600,

800 and 1000 RPM. The data was sampled at 50 khz and the bearings used were NU205 and N205.

4) MFPT

The Machine Failure Prevention Technology (MFPT) dataset [29] comprises N, IF and OF vibration measurements of an unknown bearing with known geometry and a sampling rate of 48828 Hz and 97656 Hz. It includes data with different load and shaft speeds.

5) Ottawa

The Time-Varying Bearing dataset from Ottawa University [30] consists of N, IF, OF and RF vibration measurements with 200 kHz using an ER16K bearing. The data includes measurements with varying shaft rotational speed and can be collected from [31].

6) SpectraQuest

The SpectraQuest dataset was retrieved from [32] and includes vibration measurement of bearing ER-16K under conditions OF and IF. The sampling rate used was 51.2 kHz.

7) HUST

The HUST dataset [33] contains vibration measurements for bearing 6204, 6205, 6206, 6207, and 6208 using three different working conditions 0 W, 200 W and 400 W and a sampling rate of 51.2 kHz. It includes measurements for health states N, IR, OR and RF. In this study, only RF is used. The dataset can be retrieved from [34].

8) SCA

The SCA bearing dataset [10] contains data from an industrial environment with N, IF, OF and RF measurements of different bearings, sampling rates and rotational speed of the shaft. There are 11 cases in the dataset and it can be collected from [35].

B. CASES

To show the potential of the solution the performance is compared against state-of-the-art methods. Since these frequently, such as [36], [15] and [37] use data that is not publicly available or does not account for the assumption specified in section III-A, studies whose results are possible to compare are selected. In this search, two previous works are found which are [9] and [11]. In the experiments, the same data is used and the results of the suggested method are compared with the ones presented in their papers. In addition, to verify the method against a dataset from an industrial environment performance based on the SCA bearing dataset is also evaluated on the [10], [35]. The datasets and the files used in the respective cases are listed in Table 4. The comparison against [9] is donated as case 1, [11] case 2, and the SCA bearing dataset, case 3. As can be seen, for case 1, UHIT is not publicly available so it is excluded from the experiment in this study. Regarding case 2, as in the original paper, the

health state N for PU is only used when PU is set as the target domain. Also, in contrast to their developed method, samples of health state N from the target domain are assumed to be accessible during training when available. For case 3, the files with health state N are used in the training, and the rest for evaluation. In addition, both the fan and drive side for CWRU are grouped as domain A but regard the fan side as 1A and the drive side as 2A.

C. METRICS

As the main metric the accuracy score is used which is also applied in [9] and [11]:

$$\text{Accuracy} = 100 \frac{TP + TN}{TP + FP + TN + FN}, \quad (8)$$

where TP is the number of True Positives, TN is the True Negatives, FP is the False Positives and FN is the False Negatives. In addition, because of the importance of not generating false alarms, the metric Fault False Positive Rate, $FFPR$, is defined where the number of Fault False Positives, FFP , is counted and Fault True Negatives, FTN . Then the $FFPR$ is calculated accordingly:

$$FFPR = 100 \frac{FFP}{FFP + FTN}. \quad (9)$$

Lastly, to evaluate to what extent the method captures all fault segments the Fault Segment Recall FSR is defined. This is calculated by first counting all Fault Segment True Positives, $FSTP$ which occurs when a prediction in a segment of a particular fault, considering individual bearing fault separately, is correctly identified and more than 50% of all non-healthy predictions are correct. Then, all Fault Segment False Negatives, $FSFN$ and FSR can be calculated accordingly:

$$FSR = 100 \frac{FSTP}{FSTP + FSFN}. \quad (10)$$

D. SETUP

Each sample is selected using an overlapping window of the raw measurement data. The overlapping size is selected based on the total measurement time to get approximately the same amount of samples from each domain. This means that the datasets with less raw data are given a smaller overlapping size and sets with a larger amount of raw data, a larger one. Furthermore, 100 samples from each domain and health state in each case are randomly selection. Then, based on the initial selection, 200 samples for each health state are randomly chosen for training. For the test data, 100 samples randomly samples are selected for each health state (without overlapping with normal samples from the training set). Overall, as in [11], some minor class imbalances are accepted in the experiment for cases 1 and 2. For case 3, no actions are taken to reduce the imbalances of the original dataset. Furthermore, for cases 1 and 2, as was done in [9], half of the samples of the health state N from the target domain are available as training data and the rest for testing.

In the preprocessing, k is set to 200 (except for dataset Ottawa where s is set to 0.7 times the sampling rate due to

TABLE 4. Description of datasets.

Case	Domain	Dataset	Additional identification/Files used
1 [9]	A	CWRU	Drive side with fault size 0.007 and 0.021 in and sampling rate 12kHz
	B	CWRU	Fan side with fault size 0.007 and 0.021 in and sampling rate 12kHz
	C	PU	Artificially damaged and setting N15_M07_F10, files K001, K002, K003, KA01, KA03, KA05, KI01, KI07, KI08
	D	PU	Natural damaged, files K004, K005, K006, KA04, KA16, KA30, KI04, KI14, KI18
	E	MFPT	All files
	F	SpectraQuest	All files
	G	Ottawa	All files with health, IR and OR fault
	H	UHIT	Not used because not publically available
2 [11]	A	CWRU	All files for drive and sampling rate 12kHz
	B	JNU	All files
	C	Ottawa	All files
	D	MFPT	All files
	E	PU	All files with artificial damages
3	A	CWRU	Drive (1A) and fan side (2A) with sampling rate 12 kHz. From 1797: 97, 105, 169, 209, 234, 278, 274, 294, 313, 315 From 1772: 98, 106, 170, 210, 131, 235, 271, 295, 309, 316 From 1750: 99, 107, 211, 132, 236, 280, 272, 296, 311, 317 From 1730: 100, 108, 212, 133, 237, 281, 297, 312
	B	PU	K001, K002, K003, K004, K005, K006, KA01, KA03, KA06, KA07, KA08, KA09, KI01, KA04, KA16, KI04, KI16, KI17, KI18
	C	JNU	tb1000
	D	MFPT	All files
	E	SpectraQuest	Outer_race_fault_29hz
	F	HUST	B500, B502, B504, B700, B702, B704, B800, B802, B804
	G	SCA	All files

the high variations in shaft speed). In addition, the width w_1 is set to 1200, w_2 to 450, w_3 to 840, the multiple q to 8, and u to 5.

In the training of model 1, multiple parameters for the setups are tested (see section VI-B) and based on the best performance m is set to 100, λ to 0.25, and training is carried out for 40 epochs using a batch size of 128. For model 2, 50 epochs are used for training with a batch size of 64 and in both training procedures, Adam optimizer with a fixed learning rate of 0.001 is used.

To get a fair evaluation, considering randomness, each case is run five times with different seeds for the data selection and weight initialization. The implementation is based on TensorFlow's GPU-based platform with an Intel Core i7-9750H processor, 32 GB RAM, and an NVIDIA Quadro T2000 graphics card.

E. COMPARISON

This study compares the results against the experiments carried out in [9] and [11]. The largest differences between the suggested method and their methods are the training procedure, as explained in section II, and the data input. In comparison to the suggested method that adopts standardization based on the enveloped frequency spectrum, [9] uses a standardization procedure based on the enveloped raw signal and [11] uses the raw vibration data as input.

Instead of redoing their experiment, the results they presented in their studies are reused. They applied different variants of support vector machine (SVM), CNN, multi-layer perception (MLP), stacked sparse autoencoder (sSAE) and other methods suggested by other studies which are also

included in the results.

In case 1 from [9], they used their developed method DDGFD (described in section II), the baselines SVM, MLP, sSAE, CNN, DTL [38] and WDCNN [39] from other studies. They used four different preprocessing steps with extensions -E, -F, -S, -V where E stands for a preprocessed enveloped raw signal, F for handcrafted features, S, for ordered spectra and V for a preprocessed raw signal.

Case 2 based on [11] used, apart from their suggestion method DT-DDG (described in section II), the baselines MLP, CNN, DAN and DL-ADAN [40], DCTLN [41], WDAN [42] and CNN-C [12] from other studies. They used the extensions -b for the best result on a single-source-to-target and -c for combining several domains as the training data.

Lastly, in addition to these, the performance considering variants of ConKIDDG is tested. These are: only using model 1, not using metric learning and not using standardization of the data based on fault frequencies where only one feature extractor is used.

V. RESULTS

The results of the experiment for all cases and tasks are shown in Tables 5-7 where a task ABCD→E means that sources A, B, C and D are used as source and E as target. As can be seen, the suggested method ConKIDDG outperforms DDGFD and all other methods presented in case 1 by more than 10 percentage points (not counting the variants of ConKIDDG). In addition, ConKIDDG performs best of all methods on case 2 including a 0.62 percentage points higher score than DT-DDG. Interestingly it can be seen that there are high variances in the performance of the respective models in the different tasks. For example, in the case ABD→E, ConKIDDG dis-

TABLE 5. Accuracy for the tasks in case 1

Method	ABCD →E	ABCD →F	ABCD →G	ABEG →C	ABEG →D	ABEG →F	CDEG →A	CDEG →B	CDEG →F	Average
SVM-S [9]	40.33	0.00	33.33	33.33	33.33	40.85	66.67	60.23	0.00	34.23
SVM-F [9]	39.50	6.55	66.99	62.03	55.49	56.00	66.67	57.60	49.80	51.18
MLP-S [9]	33.33	0.00	33.33	33.33	33.33	38.85	65.98	63.40	15.48	35.23
MLP-F [9]	33.99	7.23	31.58	43.12	43.46	56.05	64.98	66.22	62.98	46.18
sSAE-V [9]	45.71	16.69	33.88	36.48	40.23	26.11	61.85	57.21	15.36	37.06
sSAE-S [9]	50.63	12.58	33.78	34.10	35.01	39.59	56.13	64.15	17.67	38.18
sSAE-E [9]	67.94	69.94	56.30	66.06	63.36	95.15	52.00	71.45	64.60	67.42
CNN-V [9]	63.56	36.01	34.48	48.04	55.31	80.19	81.74	66.73	67.25	59.26
CNN-S [9]	38.45	4.60	33.62	41.80	44.20	50.00	68.16	67.66	51.55	44.41
CNN-E [9]	79.09	82.87	53.38	60.27	68.13	86.23	78.90	68.92	63.56	71.26
WDCNN [39]	41.71	34.40	30.35	26.06	20.37	18.18	25.43	29.30	50.95	30.42
DTL [38]	36.30	-	32.68	33.94	52.25	-	37.78	36.67	-	38.27
DDGFD-V [9]	67.11	27.88	36.92	52.85	60.35	93.27	78.40	66.68	82.39	62.87
DDGFD [9]	86.52	73.33	60.14	63.59	64.41	94.78	84.28	63.39	89.97	76.27
ConKIDDG (our)	98.79	69.33	61.07	73.60	100.0¹	88.90	96.52	94.78	100.0¹	87.00
- w/o model 2	96.81	65.67	60.40	71.87	100.0¹	89.10	95.65	92.32	100.0¹	85.76
- w/o metric learning	87.14	65.93	62.87	72.53	100.0¹	91.50	97.68	92.32	100.0¹	85.55
- w/o standardization	73.08	34.13	47.93	63.80	86.33	78.30	96.67	94.35	100.0¹	74.95

TABLE 6. Accuracy for the tasks in case 2

Method	ABCD→E	ABD→E	ABCE→D	ABC→D	ABDE→C	ACDE→B	BCDE→A	Average
MPL-b [11]	53.56	53.56	53.93	53.93	66.09	35.25	42.42	51.25
MPL-c [11]	43.33	44.22	58.25	61.31	42.44	34.75	28.92	44.75
CNN-b [11]	86.56	86.56	70.89	70.89	58.22	42.33	47.92	66.20
CNN-c [11]	75.22	72.63	62.59	74.97	45.15	44.25	60.92	62.25
DAN [11]	70.22	92.77	73.89	88.67	39.67	75.00	53.50	70.53
DL-ADAN [40]	79.73	93.45	78.21	86.37	60.33	75.11	62.46	76.52
DCTLN [41]	81.32	93.11	76.71	89.02	62.49	74.67	63.37	77.24
WDAN [42]	74.21	89.91	80.45	86.33	59.42	73.50	60.72	74.93
CNN-C [12]	71.06	72.33	65.79	82.26	48.89	46.17	59.54	63.72
DT-DDG [11]	84.89	94.76	90.56	88.56	63.38	76.72	64.67	80.51
ConKIDDG (our)	65.80	55.93	95.05	97.58	91.33	91.50	70.70	81.13
- w/o model 2	64.40	54.60	98.02	98.24	84.67	87.15	68.65	79.39
- w/o metric learning	63.07	52.93	97.80	99.23	81.80	70.20	66.15	75.88
- w/o standardization	43.87	47.33	76.59	56.59	33.33	43.00	66.20	52.42

TABLE 7. Accuracy for the tasks in case 3

Method	A-F →G1	A-F →G2	A-F →G3	A-F →G4	A-F →G5	A-F →G6	A-F →G7	A-F →G8	A-F →G9	A-F →G10	A-F →G11	Average
ConKIDDG (our)	37.00	99.79	59.18	92.50	84.72	90.54	93.16	94.14	47.60	53.60	100.0	77.47
- w/o model 2	37.20	97.87	61.22	93.46	86.02	88.99	90.79	88.45	54.80	50.60	100.0	77.22
- w/o metric learning	21.00	99.57	58.16	90.58	81.30	85.74	87.11	36.72	24.60	52.80	74.6	64.74
- w/o standardization	25.80	98.94	58.16	87.88	81.30	51.01	85.79	93.97	90.40	24.80	62.02	69.11

¹ In some instances the model achieves 100% accuracy largely because the samples in the target domain, due to the standardization, resemble them in the source domains.

plays poor performance while DT-DDG shows high accuracy and in the case ABDE→C the opposite is observable. Considering that one of the biggest differences in these methods is the input, namely raw data for DT-DDG and standardized enveloped frequency for ConKIDDG, it can indicate that the most appropriate input type depends on the task. Lastly, ConKIDDG shows a strong result on case 3 based on the industrial dataset with an average score of 77.47%. In summary, considering the variety of cases, the result indicates that ConKIDDG has a strong generalisability and can be applied on a general basis. To give a general understanding of the training time for ConKIDDG considering the setup, the time

for task ABCD→E case 1 is measured to 34.5 seconds for stage 1 and 4.8 seconds for stage 2.

VI. DISCUSSION AND ANALYSIS

A. EFFECTIVENESS OF THE SUGGESTED METHOD

The accuracy for different setups of the suggested method is displayed in Figure 5 and shows that ConKIDDG and ConKIDDG without model 2 achieve the best result. It should be noted that ConKIDDG without model 2 sometimes performs slightly better than ConKIDDG, such as ABCE→D in case 2. This is expected because it will be more inclined to predict a fault in a segment of uncertain measurements

which sometimes can be beneficial considering accuracy but likely makes it less robust. The biggest decline in performance comes from not employing standardization. In addition, not using metric learning gives reasonable performance in cases 1 and 2 but significantly worse in case 3.

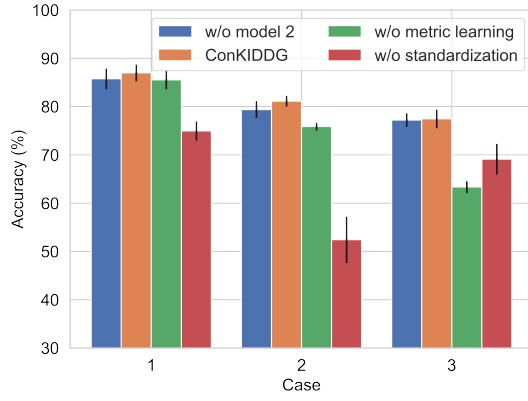


FIGURE 5. Comparison of average accuracy between the different setups of the method.

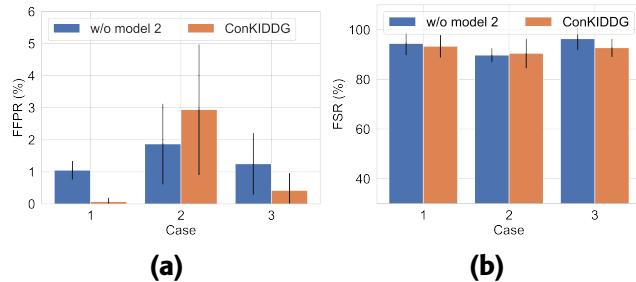


FIGURE 6. FFPR and FSR on average for all cases and methods ConKIDDG and ConKIDDG without model 2. (a) Comparison of FFPR. (b) Comparison of FSR.

To make additional comparisons between ConKIDDG and ConKIDDG without model 2 the metrics *FFPR* and *FSR* are applied which can be seen in Figure 6. The result shows that the historical context gives lower *FFPR* overall considering all cases (1.14% for ConKIDDG and ConKIDDG without model 2 1.39%) and close to zero for cases 1 and 3. In case 2 *FFPR* is higher but the standard deviation is so substantial (outcome varies from 0% to 4%) that more data is likely needed to give an accurate estimate. Regarding the *FSR*, ConKIDDG without model 2 performs slightly better than ConKIDDG overall considering all cases (92.18% for ConKIDDG and ConKIDDG without model 2 93.52%) however both methods consistently show a high detection rate of fault segments.

To understand the differences between using and not using the metric learning procedure, t-SNE is used to describe the features extracted from the output of the bottleneck. Figure 7 illustrates this on the testing data for case 1 task ABCD-E, which includes the health states N, IF and OF, using the prediction \hat{y} . It can be observed that when not using metric

learning (a) the model makes incorrect predictions, which does not occur when using metric learning (b) where features from the different classes are more separated from each other.

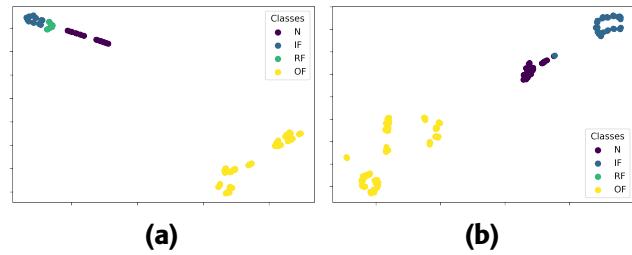


FIGURE 7. Features from the bottleneck described using t-SNE and the predictions \hat{y} on case 1 task ABCD-E (test data). (a) Without metric learning. (b) With metric learning.

B. SENSITIVITY ANALYSIS

Lastly, to evaluate the sensitivity of the parameters for the metric learning method, different parameters are tested on case 2 using a single seed. The result for average accuracy and *FFPR* is shown in Figure 8 with different values of the margin m and the impact of the metric learning method λ . It shows that the model is robust given the high performance for all the different settings. Nevertheless, the highest accuracy in the experiment is achieved by setting m to 125 closely followed by 100 and λ to 0.25. In addition, increasing λ and m generally seem to increase *FFPR*. Considering these aspects, choosing a value for m to 100 and λ to 0.25 is arguably a reasonable trade-off.

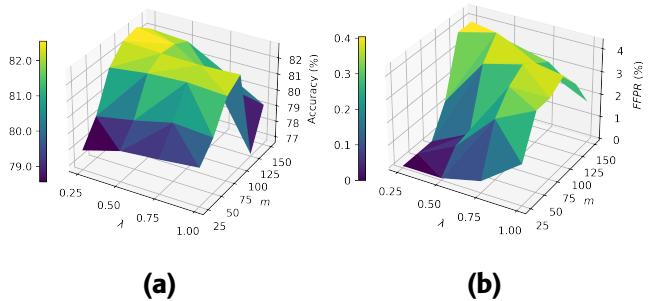


FIGURE 8. Performance for a single seed on case 2 considering different parameters for the metric learning method. (a) Accuracy. (b) FFPR.

C. USER FEEDBACK

A significant challenge with deep learning models is that they are difficult to interpret which can be an issue concerning the trustworthiness and robustness of industrial applications. In this regard, Explainable AI (XAI) has the potential to resolve these issues by creating interpretive solutions valuable to the end-user by providing confidence in the model's predictions and to developers by assisting them in improving the model [43].

A secondary advantage of the standardization method as input to the model is that XAI-based methods can give feedback to the user about the model predictions by utilising the known characteristics of bearing faults. Specifically, when a fault is present and the model correctly identifies the fault, the features of the highest importance should correspond to peaks related to the fault frequency of the specific bearing fault. In addition, the model should, regardless of ground truth, disregard abnormal segments in the spectra that do not relate to the fault frequencies of the bearing. Because models can make accurate predictions without having the correct understanding of what constitutes a fault, the value is that the user can examine if the model made its prediction on features that correlate with the known characteristics of bearing faults. In this way, it is possible to provide new training examples when the model is shown to have an inaccurate understanding of this and gradually increase the end user's confidence in the model when it bases its decision on features that are known to be related to bearing faults.

To demonstrate this, Temporal Saliency Rescaling (TSR) developed by [44] using the implementation of TSR by TSInterpret [45] is used on model 1. This is a post hoc method that, based on a specified saliency method, gives an importance score for each feature related to the output of the time series classification model. In the experiment, gradient (GRAD) was used as the saliency method and the details of the implementation can be found in the following repository [46].

The output from the TSR method is extracted as heat map graphs to show which features in the input data had the most importance for the prediction. The red dotted lines show the fault frequency harmonics of the corresponding fault and are, as previously described, extracted from the bearing specification. Figure 9 shows the explanation for (a) case 1 with task ABCD-F and OF, (b) case 2 with task ABCE-D and IF, (c) case 3 with task ABCDEH-I5 and RF, and (d) case 1 with task ABEG-F and N based on the output IF. As can be seen, the model highlights peaks related to the fault frequency of the bearing which corresponds to the ground truth for each task. In addition, when no fault is apparent the model does not highlight any segments which is desirable.

In Figure 10 (a) the model incorrectly predicts an RF when the ground truth is an OF. As shown, segments not related to the outer ring fault frequency are highlighted strongly suggesting that the model bases the prediction on segments not related to OF. In addition, in (b) the ground truth is N but the model predicts IF and highlights random segments in the data. These scenarios can be used as an indicator that more

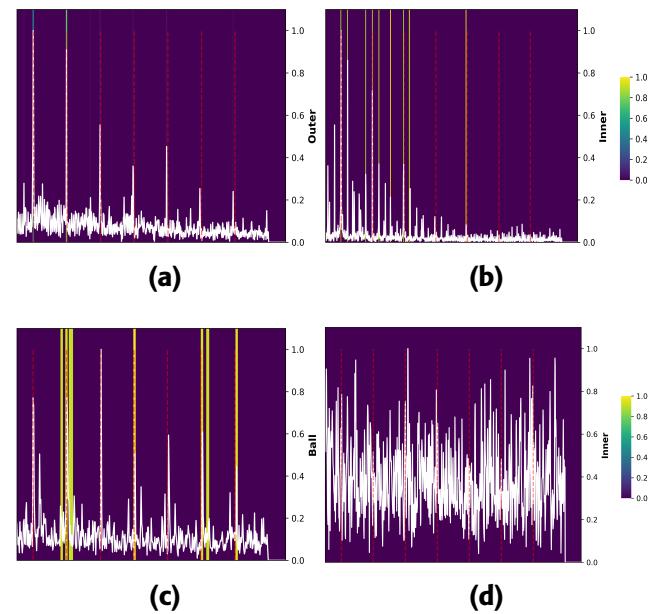


FIGURE 9. Explanations of correct classifications. The heat map shows which parts of the data are important for the model's decision. The red dotted lines show the fault frequency of the bearing for the ground truth fault. (a) Case 1, task ABCD-F, OF. (b) Case 2, task ABCE-D, IF. (c) Case 3, task ABCDEH-I5, RF. (d) Case 1, task ABEG-F, N.

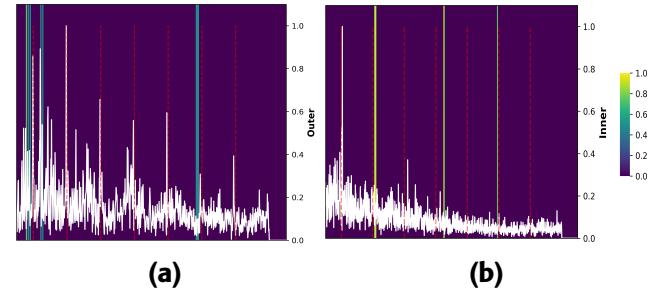


FIGURE 10. Explanations of incorrect classifications. The heat map shows which parts of the data are important for the model's decision. (a) Case 1, task ABEG-F, ground truth OF, predicted RF. The red dotted line shows the outer ring fault frequency. (b) Case 3, task ABCDEH-I8, ground truth N, predicted IR. The red dotted line shows the inner ring fault frequency.

examples in training are needed to improve the robustness of the model.

D. LIMITATIONS AND FUTURE WORK

The results show the effectiveness of the standardization method. However, there are some limitations and possible improvements to be explored to improve it. Firstly, the suggested method will not be as effective when the rotation of the shaft varies during the measurement. To overcome this, it is encouraged that future research explore standardization techniques based on time-frequency methods such as Hilbert-Huang Transform or Wavelet Transform [47]. In addition, possible improvements to the suggested method exist such as using adaptive filters, suggested in for example [48], which can further ease the learning task. Lastly, regarding the stan-

dardization of the size of each input, techniques other than piecewise linear interpolation are worth exploring.

VII. CONCLUSION

This study aims to develop a new bearing fault diagnosis method that gives the user reliable decision support without having access to historical faults from the target domain. To achieve this, a standardization method is applied to reduce the domain shift using the enveloped spectrum and knowledge enrichment based on the fault frequencies of the bearing. In addition, a DL method that includes separate feature extractors designed to capture the key features in the data, metric-based learning and a contextual layer to achieve domain generalization is suggested. The performance of the method is evaluated by comparing it to state-of-the-art and tested on three cases including several transfer tasks. The result shows that the method consistently gives the highest accuracy in the different tasks with a high fault segment recall and a low number of false positives for bearing faults. In addition, a user feedback method shows the possibility of explaining the model's decisions and potentially increasing trustworthiness and robustness in industrial applications. The conclusion based on the result is that the suggested method can be applied to diagnose the health state of bearings on a general basis without having access to any historical examples of faults from the target domain. Despite this, limitations of the method are scenarios with high variances in the rotational speed during the vibration measurement and noise in the preprocessed signal. Therefore, future work is recommended to examine ways of improving the standardization method and to provide novel suggestions for methods that increase the accuracy of bearing fault diagnosis across machines.

References

- [1] R. B. Randall and J. Antoni, "Rolling element bearing diagnostics—A tutorial," *Mechanical Systems and Signal Processing*, vol. 25, no. 2, pp. 485–520, Feb. 2011.
- [2] I. El-Thalji and E. Jantunen, "A summary of fault modelling and predictive health monitoring of rolling element bearings," en, *Mechanical Systems and Signal Processing*, vol. 60-61, pp. 252–272, Aug. 2015.
- [3] M. H. Mohd Ghazali and W. Rahiman, "Vibration Analysis for Machine Monitoring and Diagnosis: A Systematic Review," en, *Shock and Vibration*, vol. 2021, e9469318, Sep. 2021.
- [4] D. Neupane and J. Seok, "Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset With Deep Learning Approaches: A Review," *IEEE Access*, vol. 8, pp. 93 155–93 178, 2020.
- [5] M. Hakim, A. A. B. Omran, A. N. Ahmed, M. Al-Waily, and A. Abdellatif, "A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning: Taxonomy, overview, application, open challenges, weaknesses and recommenda-
- [6] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review," *IEEE Access*, vol. 8, pp. 29 857–29 881, 2020.
- [7] A. Zhang, S. Li, Y. Cui, W. Yang, R. Dong, and J. Hu, "Limited Data Rolling Bearing Fault Diagnosis With Few-Shot Learning," *IEEE Access*, vol. 7, pp. 110 895–110 904, 2019.
- [8] Z. Zhao, Q. Zhang, X. Yu, *et al.*, "Applications of Unsupervised Deep Transfer Learning to Intelligent Fault Diagnosis: A Survey and Comparative Study," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–28, 2021.
- [9] H. Zheng, Y. Yang, J. Yin, Y. Li, R. Wang, and M. Xu, "Deep Domain Generalization Combining A Priori Diagnosis Knowledge Toward Cross-Domain Fault Diagnosis of Rolling Bearing," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.
- [10] A. Lundström and M. O'Nils, "Factory-Based Vibration Data for Bearing-Fault Detection," en, *Data*, vol. 8, no. 7, p. 115, Jul. 2023.
- [11] Y. Shi, A. Deng, M. Deng, *et al.*, "Domain Transferability-Based Deep Domain Generalization Method Towards Actual Fault Diagnosis Scenarios," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 6, pp. 7355–7366, Jun. 2023.
- [12] Y. Yang, J. Yin, H. Zheng, Y. Li, M. Xu, and Y. Chen, "Learn Generalization Feature via Convolutional Neural Network: A Fault Diagnosis Scheme Toward Unseen Operating Conditions," *IEEE Access*, vol. 8, pp. 91 103–91 115, 2020.
- [13] X. Li, W. Zhang, H. Ma, Z. Luo, and X. Li, "Domain generalization in rotating machinery fault diagnostics using deep neural networks," en, *Neurocomputing*, vol. 403, pp. 409–420, Aug. 2020.
- [14] H. Zheng, R. Wang, Y. Yang, Y. Li, and M. Xu, "Intelligent Fault Identification Based on Multisource Domain Generalization Towards Actual Diagnosis Scenario," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1293–1304, Feb. 2020.
- [15] Y. Liao, R. Huang, J. Li, Z. Chen, and W. Li, "Deep Semisupervised Domain Generalization Network for Rotary Machinery Fault Diagnosis Under Variable Speed," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 8064–8075, Oct. 2020.
- [16] Q. Zhang, Z. Zhao, X. Zhang, *et al.*, "Conditional Adversarial Domain Generalization With a Single Discriminator for Bearing Fault Diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.
- [17] T. Han, Y.-F. Li, and M. Qian, "A Hybrid Generalization Network for Intelligent Fault Diagnosis of Rotat-

tions," en, *Ain Shams Engineering Journal*, p. 101 945, Sep. 2022.

- ing Machinery Under Unseen Working Conditions,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.
- [18] Y. Xie, J. Shi, C. Gao, *et al.*, “Rolling Bearing Fault Diagnosis Method Based On Dual Invariant Feature Domain Generalization,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–11, 2024.
- [19] Z. Chen, G. He, J. Li, Y. Liao, K. Gryllias, and W. Li, “Domain Adversarial Transfer Network for Cross-Domain Fault Diagnosis of Rotary Machinery,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 11, pp. 8702–8712, Nov. 2020.
- [20] J. Jiao, M. Zhao, J. Lin, and K. Liang, “Residual joint adaptation adversarial network for intelligent transfer fault diagnosis,” *Mechanical Systems and Signal Processing*, vol. 145, p. 106962, Nov. 2020.
- [21] L. Wan, Y. Li, K. Chen, K. Gong, and C. Li, “A novel deep convolution multi-adversarial domain adaptation model for rolling bearing fault diagnosis,” *Measurement*, vol. 191, p. 110752, Mar. 2022. (visited on 11/15/2024).
- [22] L. Jia, T. W. S. Chow, Y. Wang, and J. Ma, “Dynamic Balanced Dual Prototypical Domain Generalization for Cross-Machine Fault Diagnosis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–10, 2024.
- [23] X. Chen, R. Yang, Y. Xue, M. Huang, R. Ferrero, and Z. Wang, “Deep Transfer Learning for Bearing Fault Diagnosis: A Systematic Review Since 2016,” *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–21, 2023. (visited on 11/14/2024).
- [24] S. Kim, D. An, and J.-H. Choi, “Diagnostics 101: A Tutorial for Fault Diagnostics of Rolling Element Bearing Using Envelope Analysis in MATLAB,” en, *Applied Sciences*, vol. 10, no. 20, p. 7302, Jan. 2020.
- [25] Case Western reserve University (CWRU) Bearing Data Center, Aug. 2021. [Online]. Available: <https://engineering.case.edu/bearingdatacenter/download-data-file> (visited on 03/14/2023).
- [26] C. Lessmeier, J. K. Kimotho, D. Zimmer, and W. Sextro, “Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification,” en, *PHM Society European Conference*, vol. 3, no. 1, 2016.
- [27] Konstruktions- und Antriebstechnik (KAt) - Bearing DataCenter (Universität Paderborn), de. [Online]. Available: <https://mb.uni-paderborn.de/kat/forschung/kat-datacenter/bearing-datacenter/data-sets-and-download> (visited on 03/14/2023).
- [28] K. Li, X. Ping, H. Wang, P. Chen, and Y. Cao, “Sequential fuzzy diagnosis method for motor roller bearing in variable operating conditions based on vibration analysis,” eng, *Sensors (Basel, Switzerland)*, vol. 13, no. 6, pp. 8013–8041, Jun. 2013.
- [29] Society for Machinery Failure Prevention Technology (MFPT), *Fault Data Set*. [Online]. Available: <https://www.mfpt.org/fault-data-sets/> (visited on 03/24/2023).
- [30] H. Huang and N. Baddour, “Bearing vibration data collected under time-varying rotational speed conditions,” en, *Data in Brief*, vol. 21, pp. 1745–1749, Dec. 2018.
- [31] H. Huang and N. Baddour, *Bearing Vibration Data under Time-varying Rotational Speed Conditions*, en, Feb. 2019. [Online]. Available: <https://data.mendeley.com/datasets/v43hmbwxpm/2> (visited on 05/30/2023).
- [32] G. Forbes, *Inner and Outer Race Bearing Fault Vibration Measurements*, Jan. 2023. [Online]. Available: <https://github.com/vibraneur-capstone/dataset> (visited on 11/17/2023).
- [33] N. D. Thuan and H. S. Hong, “HUST bearing: A practical dataset for ball bearing fault diagnosis,” *BMC Research Notes*, vol. 16, no. 1, p. 138, Jul. 2023.
- [34] Hoang Si Hong and Nguyen Thuan, *HUST bearing*, 2023. [Online]. Available: <https://data.mendeley.com/datasets/cbv7jyx4p9> (visited on 02/15/2023).
- [35] A. Lundström and M. O’Nils, *SCA bearing dataset*, Jun. 2023. [Online]. Available: <https://data.mendeley.com/datasets/tdn96mkkpt/1> (visited on 09/01/2023).
- [36] B. Yang, Y. Lei, F. Jia, N. Li, and Z. Du, “A Polynomial Kernel Induced Distance Metric to Improve Deep Transfer Learning for Fault Diagnosis of Machines,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, pp. 9747–9757, Nov. 2020.
- [37] J. Li, C. Shen, L. Kong, D. Wang, M. Xia, and Z. Zhu, “A New Adversarial Domain Generalization Network Based on Class Boundary Feature Detection for Bearing Fault Diagnosis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–9, 2022.
- [38] L. Wen, L. Gao, and X. Li, “A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 136–144, Jan. 2019.
- [39] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, “A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals,” en, *Sensors*, vol. 17, no. 2, p. 425, Feb. 2017.
- [40] J. Jiao, J. Lin, M. Zhao, and K. Liang, “Double-level adversarial domain adaptation network for intelligent fault diagnosis,” *Knowledge-Based Systems*, vol. 205, p. 106236, Oct. 2020.
- [41] J. Zhu, N. Chen, and C. Shen, “A New Multiple Source Domain Adaptation Fault Diagnosis Method Between Different Rotating Machines,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4788–4797, Jul. 2021.

- [42] D. Wei, T. Han, F. Chu, and M. J. Zuo, "Weighted domain adaptation networks for machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 158, p. 107744, Sep. 2021.
- [43] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. Díaz-Rodríguez, *Explainable Artificial Intelligence (XAI) on TimeSeries Data: A Survey*, Apr. 2021.
- [44] A. A. Ismail, M. Gunady, H. C. Bravo, and S. Feizi, "Benchmarking deep learning interpretability in time series predictions," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20, Red Hook, NY, USA: Curran Associates Inc., Dec. 2020, pp. 6441–6452.
- [45] J. Höllig, C. Kulbach, and S. Thoma, "TSInterpret: A Python Package for the Interpretability of Time Series Classification," en, *Journal of Open Source Software*, vol. 8, no. 85, p. 5220, May 2023.
- [46] F. F. Informatik, *Tsinterpret*, Nov. 2024. [Online]. Available: <https://github.com/fzi-forschungszentrum-informatik/TSInterpret> (visited on 11/17/2024).
- [47] Z. K. Peng, P. W. Tse, and F. L. Chu, "A comparison study of improved Hilbert–Huang transform and wavelet transform: Application to fault diagnosis for rolling bearing," *Mechanical Systems and Signal Processing*, vol. 19, no. 5, pp. 974–988, Sep. 2005.
- [48] W. Dai, Z. Mo, C. Luo, J. Jiang, H. Zhang, and Q. Miao, "Fault Diagnosis of Rotating Machinery Based on Deep Reinforcement Learning and Reciprocal of Smoothness Index," *IEEE Sensors Journal*, vol. 20, no. 15, pp. 8307–8315, Aug. 2020.



FAISAL Z. QURESHI (Senior Member, IEEE; Member ACM; Secretary and Member CIPPR) received the B.Sc. degree in Mathematics and Physics from Punjab University, Lahore, Pakistan, in 1993, the M.Sc. degree in Electronics from Quaid-e-Azam University, Islamabad, Pakistan, in 1995, and the M.Sc. and Ph.D. degrees in Computer Science from the University of Toronto, Toronto, Canada, in 2000 and 2007, respectively.

He is a Professor of Computer Science in the Faculty of Science, Ontario Tech University, where he leads the Visual Computing Lab. His research focuses on computer vision, and his scientific and engineering interests center on the study of computational models of visual perception to support autonomous, purposeful behavior in the context of ad hoc networks of smart cameras. Dr. Qureshi is also active in journal special issues and conference organizations. He served as the general co-chair for the Workshop on Camera Networks and Wide-Area Scene Analysis (co-located with CVPR) in 2011-13. He also served as the co-chair of Computer and Robot Vision (CRV) conference 2015/16 meetings.

...



ADAM LUNDSTRÖM received the M.S. degree in Industrial Engineering and Management from Mid Sweden University and the M.S. degree in Computer and Systems Sciences from Stockholm University in 2020. He is currently a Ph.D. student within the Industrial graduate school Smart Industry Sweden and studying the potential for machine learning solutions in predictive maintenance. He is employed by SCA and a part of the Department of Computer and Electrical Engineering at Mid Sweden University.



MATTIAS ONILS received the B.S. degree in electrical engineering from Mid Sweden University, Sundsvall, Sweden, in 1993, and the Licentiate and Ph.D. degrees in electronic systems design from the Royal Institute of Technology, Stockholm, Sweden, in 1996 and 1999, respectively. He is currently a professor with the Department of Computer and Electrical Engineering and leads a research group in Embedded IoT Systems at Mid Sweden University. His current research interests include design methods and implementation of embedded DNN-based systems, especially in the implementation of real-time video processing and time series processing systems.