Deep ensemble-based classifier for transfer learning in rotating machinery fault diagnosis

FANNIA PACHECO¹, ALIN DRIMUS¹, LARS DUGGEN¹, MARIELA CERRADA² (Senior, IEEE), DIEGO CABRERA² (Member, IEEE), and RÉNE-VINICIO SANCHEZ² (Member, IEEE)

¹ Department of Mechanical and Electrical Engineering, University of Southern Denmark, Sønderborg, 6400, Denmark (e-mail: fpacheco@sdu.dk, drimus@sdu.dk, duggen@sdu.dk)
² GIDTEC, Universidad Politécnica Salesiana, Cuenca, 010105, Ecuador (e-mail: mcerrada@ups.edu.ec, dcabrera@ups.edu.ec, rsanchezl@ups.edu.ec)

Corresponding author: Fannia Pacheco (e-mail: fpacheco@sdu.dk).

ABSTRACT Nowadays, intelligent models can correctly detect faults by analysing signals from rotating machinery. However, most of the studies are run in controlled environments and the performance in industrial real-world environments is not yet fully validated. Hence, a suitable tool to implement fault diagnosers is transfer learning, this topic is under development and challenges persist. This paper proposes a framework for creating accurate 1D-CNN based fault classifiers that can be transferred between different rotating machines and working conditions. Multiple Bayesian processes select architecture parameters and hyperparameters, which minimize a loss function related to their transferability to other machines and to the same machine under different operating conditions (such as load and engine speed). The resulting model is fitted to heterogeneous fault diagnosis data resulting in a 1D-CNN ensemble that improves the performance of the unitary model. Subsequently, the transfer learning capability of the ensemble is analyzed on two source data sets using function and parameter based transfer. The results are compared with classical fault diagnosis classifiers. Finally, additional transfer operations on five target domain datasets validate our framework on limited labeled samples and allow interpretation of the ensemble results. The ultimate goal is to find an ensemble that can generalize fault diagnosis on rotating machinery for easy implementation and update in industrial settings.

INDEX TERMS Fault diagnosis, deep learning, rotating machinery, transfer learning, vibration signal.

I. INTRODUCTION

CONDITION Monitoring (CM) in rotating machinery has been instrumental in preventing failures, scheduling maintenance, and estimating life expectancy. Advances in this field have experienced growth as artificial intelligence has been introduced. CM in rotating machinery has been traditionally assessed by measuring different signals (e.g., vibrations and acoustic signals). Classical approaches extract various time, time-frequency, and frequency domain features on those signals. These features are used to create Machine Learning (ML)-based intelligent models to detect anomalies and faulty conditions. Novel approaches introduce Deep Learning (DL) techniques to take advantage of their generalization capabilities and their property to extract knowledge without explicit feature extraction [1]. Nonetheless, real-world industrial datasets are generally unavailable due to their acquisition cost and commercial sensitivity. The trend is to build controlled environments where failures are induced under multiple working conditions in the rotating machinery. DL models can be designed for CM in such an environment. However, the transition from the controlled setup to the industrial application is not always successful. Moreover, finding the right DL model is a time-consuming task that can be successfully addressed with the appropriate computational resources.

In more mature DL applications, such as image processing and natural language processing, data collection needs have been alleviated with Transfer Learning (TL). TL is to find a way to combine a source domain that contains rich historical data with a new or target domain that lacks data. TL has a pivotal role in addressing data constraints and model implementation in real-world applications. In CM,
TL is commonly approached following three transfer styles [2]: i) transfer between machines aims at conveying knowledge from laboratory-controlled to industrial configurations [3], ii) transfer between working conditions (e.g., speed or load) create models that maintain performance in scenarios different from those in which they were built [4], and iii) finally, transfer between multiple fault types aims to leverage similarities between their feature indicators, for example, from a rich data source of bearing fault problems to a sparse data target of gear fault problems [5]. More on this topic is presented below.

A. RELATED WORKS

The existing literature on transferring between working conditions is extensive and focuses on varying the rotation speed and load on the machine. For example, [4], [6], [7] showed that classical Neural Networks (NN) and Convolutional Neural Networks (CNNs) trained on the source domain result in a set of parameters and hyperparameters that can be reused for another working condition.

Another common and prevalent trend is to transform measured signals into images and tune state-of-the-art models from the field of image processing. Fast Fourier transforms (FFTs), spectrogarams, and wavelet transforms provide visual representations suitable for fault detection with TL [3], [7], [8]. For instance, [9] transformed time-series into image representations by using the frequency-slice wavelet transform (FSWT), and [10] plotted the raw time-series behavior of vibration signals; both tuned well-known image classification architectures for fault diagnosis. This approach is accurate; however, it sustains disadvantages such as loss of performance in the face of noisy measurements, high model tuning cost, lack of interpretability, among others. Finally, most image transformations are on cyclostationary signals or constant operating conditions; thus, faults under varying operating conditions (referred to as non-cyclostationary) might be misclassified.

An alternative approach is to transfer features or samples from a source domain to a target domain. This will augment the target domain data and is achieved by distribution matching using methods such as Transfer Component Analysis (TCA), Joint Distribution Adaptation (JDA), Adversarial Domain Adaptation (ADA), and others [11]–[13]. The raw signal from the source domain can also enrich the target domain data by using the Maximum Mean Discrepancy (MMD). The MMD is used to measure the difference between the source and target distributions according to the DL model built [3], [14]. One of the main disadvantages of those approaches is the cost of training such models– the TL process also becomes very greedy.

In this work we are interested on the definition of DL architectures that are transferable between machines and working conditions for CM. In this matter, ensemble of classifiers has shown versatility for fault classification and TL. Traditionally, base models of an ensemble are obtained with bootstrap sample versions of the original training data (as in the Random Forest approach); however, for CM, different approaches propose to train the ensemble on different working conditions or to modify their transfer method to maximize the classification performance on a target domain. For instance, the work in [15] presents an ensemble with base models created on different channels and then transferred by tuning the parameters on different working conditions. In this work, an arbitrary CNN was defined using the signal images as feature maps, and a combination method based on an adjustable weighted majority voting. The work presents higher performance than traditional methods; however, few samples per class are not evaluated, and their tests were performed on the same machine by varying the working conditions. On the other hand, [16], [17] propose to transfer a 1D-CNN deep ensemble by defining different MMD kernels to each base model. Similarly, the work in [18] trains a set of CNNs to obtain feature extraction functions; these are coupled with auto-encoder decision functions. Bootstrap is used to define the training data subsets for each base model. The resulting experiments show improved performance; however, their transferability is not demonstrated. [19] presents a similar approach to the latter. In the aforementioned works, the ensemble results are mostly combined by voting the most frequent class. Moreover, they do not consider how to correctly define deep learning architectures suitable for TL operations in fault diagnosis. These last two points will become the focus of this paper.

B. CONTRIBUTIONS

The main objective of this work is to find a fault diagnosis classifier that can be easily transferable between different types of rotating machines and the same machine under different working conditions. To do so, a deep 1D-CNN ensemble is constructed with various data sources to better generalize the fault patterns. The idea is to find a set of invariant features or decision functions to increase performance in TL operations. Our contributions are threefold

- We propose a framework in which 1D-CNN base models are created independently with different source domains. Then, a Bayesian optimization process is carefully defined for each base model to find its parameters and hyperparameters. However, the Bayesian objective function maximizes not only the traditional model performance on a test set but also its performance after tuning its parameters over other data sources and different working conditions.
- We propose an ensemble model with the results from the previous item. The results of the Bayesian processes are joint by a combination method. Three combination methods were tested and the best one was selected for further analysis.
- Finally, we provide the elements to validate such an ensemble in different transfer tasks (transfer from one working condition to another or from one machine to another). Throughout this paper, the classical approaches
will be compared with the 1D-CNN based result, especially in the presence of few samples per class.

Although the experiments in this paper are developed on datasets from controlled environments, extensive experiments are proposed to demonstrate the transferability of the ensemble model under different conditions such as working conditions, type of machinery, type of sensor and number of samples available. The resulting validated ensemble model, which is publicly available, becomes valuable knowledge for practitioners in the field because it can reduce the effort of collecting large amounts of data to create deep learning solutions. Moreover, the results of the framework composed of multiple Bayesian processed also alleviates the efforts of finding the best Deep Learning architecture to solve the problem.

The remainder of this paper is organized as follows. Section II presents 1D-CNN and TL foundations. Section III presents the framework of our proposal. Section IV describes the two source and the five target domain datasets. Section V reports the experiments and results of the framework. Finally, Section VI outlines the conclusions of the paper.

II. BACKGROUND

A. 1D CONVOLUTIONAL NEURAL NETWORK

1D Convolutional Neural Networks (1D-CNNs) are derived from the ordinary Neural Network with some structural changes. In general, a 1D-CNN is composed of layers with learnable weights or kernels that generates the layer’s activation, i.e., \( g(Z^1) \). After this operation, it is common, but not mandatory, to apply a Pooling layer, which will reduce the dimensions of the activation result.

\[
Z^1 = W^1 \odot X^1 + b^1
\]

where \( W^1 \in \mathbb{R}^{d \times k} \) is a weight vector with \( k \) one-dimensional filters of size \( d \). \( X^1 \in \mathbb{R}^{n \times c} \) is a \( n \)-dimensional input with channel depth \( c \).

Following, Eq. (1) is passed through a nonlinear operator \( g \) which generates the layer’s activation, i.e., \( A^1 = g(Z^1) \). After this operation, it is common, but not mandatory, to apply a Pooling layer, which will reduce the dimensions of the activation result.

The convolution, nonlinear function, and pooling operations form a convolutional layer \( (L) \). The output of a convolutional layer is a new intermediate time-series (See Figure 1). The final intermediate time-series is evaluated by fully connected (FC) layers that predict the class output. The convolution operations commonly represent an automatic feature extraction function \( f_{fe} \). In contrast, the final FC layers are viewed as a mapping function \( f_{m} \), where a Softmax operation will compute the probability distribution of the possible classes \( C_{l_t} \), i.e, \( P = p(y = C_{l_t}|X^1) \). An optimization process should be used to obtain the parameters \{\( W, b \)\} and hyperparameters (number of hidden layers and units, activation function \( g \), learning rate, and others) that minimize a loss function, which in turn measures how well the model fits the data.

B. TRANSFER LEARNING

The goal of TL is to use an existing model trained under the best conditions for one domain (source domain) and use it in another similar application (target domain).

Definition 1: Let \( X_s = \{(x^s_1, y^s_1), \cdots, (x^s_m, y^s_m)\} \) be the available source domain dataset, with \( x^s_i \) a sample of \( n \)-dimensional measurements and \( y^s_i \) its label. Equivalently, \( X_t = \{(x^t_1, y^t_1), \cdots, (x^t_k, y^t_k)\} \) represents the target domain.

Definition 2: A classification task consists of creating a decision function \( f(x; W, b) \) that maps the values of each \( x_i \) to \( y_i \). In our case, this function is divided into the feature extraction function \( f_{fe} \) and the mapping function \( f_m \), as in Figure 1.

Statement 1: \( X_s \neq X_t \) and \( f_s \neq f_t \) where \( f_s \) and \( f_t \) are the decision functions of the source and target domain, respectively.

In TL, the premise is that similarities can be found between decision functions and datasets. To find such similarities, domain adaptation approaches are implemented. In the following, we briefly describe the domain adaptation approaches used in this work.

1) Parameter-based transferring

A parametric model in the source domain is defined by \( f_s(x^s, W_s, b_s) \), where \( W_s \) and \( b_s \) are copying into a new model and will undergo small deviations from the original space; they are modified using a new cost function and a fine-tuning regime with \( X_t \) [3], [4], [6]–[8].

2) Feature-based transferring

The feature space of the source and target are matched. There exist two strategies to perform such matching: symmetric and asymmetric feature transfer. In symmetric feature transfer, both domains are mapped to a common latent feature space
where their probability distribution discrepancy is minimized as proposed by the TCA algorithm [11]. Another example is automatic feature extraction with CNNs that can produce a feature space in the source domain suitable for the target domain by fine-tuning. In contrast, an asymmetric method transforms one domain into another (source or target) by minimizing the distance between similar samples as JDA [12].

C. TL CONTEXTS FOR FAULT DIAGNOSIS

Different scenarios can be set up in rotating machinery due to operational requirements. Such scenarios are commonly reproduced in controlled environments for data collection purposes and are referred to as context in this paper. A context consists of different machinery configurations defined by the tuple $C_i = (T_i, F_i, O_i)$, where $T_i$ corresponds to the machinery type, $F_i$ is the fault type, and $O_i$ represents the operating condition. A context $C_i$ leads to the definition of one or more sub-source domain datasets $S_i$.

One can extract a finite number of contexts by finding all the possible combinations between the working conditions, fault types, and machinery. In this work, the possible context compositions are:

- One machine and type of fault operating in different conditions:
  \[ C_i = (T_i, F_i, Comb(O_i, r)) \quad (2) \]
  where $Comb$ is a function that creates all possible combinations without repetition. $r$ is the total operating conditions at $T_i$, i.e., the number of speeds and loads at which the failures were collected.

- Several machines intercepted by their working condition:
  \[ C_i = (T_i \cup T_j, F_i \cup F_j, O_i \cap O_j) \quad i \neq j \quad (3) \]

The contexts are purposely selected to find or highlight important results throughout the evaluation process. For example, in the first element, we would like to ensure that the resulting model works under different operating conditions on the same machinery. In the second case, we combine operating conditions of different machines to improve the generalization capabilities of the same model in the presence of failures in different devices (e.g., gears and bearings).

III. FRAMEWORK

Our framework aims at finding transferable 1D-CNN models between different rotating machines and working conditions (load and speed). To do so, we propose to use various source domains $X_s$ to find such transferable models. Figure 2 outlines the steps to achieve this goal. $k$ source domain signals are divided into contexts $C_i$ and each context $i$ into sub-source domains $S_i$ using the definition in Section II-C. Subsequently, the vibration signals in $S_i$ are sliced into windows of 16,384 samples (equivalent to 0.32 s) in length as proposed by [21]. Furthermore, for each context $C_i$, an optimal 1D-CNN model is obtained through a Bayesian process. During this Bayesian optimization process, a sub-source domain is used to train the 1D-CNN, and the rest acts as sub-target domains. The 1D-CNN is evaluated according to its performance on TL operations over the target domains. This Bayesian process is repeated for all source datasets. The best model ($M_{best}$) is selected by evaluating its performance on those TL operations using a function ($\Gamma_1$ in Figure 2). Moreover, we propose to build an ensemble with base architecture models equal to $M_{best}$ tuned to the available source datasets. The result of the ensemble is given by a combination method $\Gamma_2$. The rationale for using ensembles of models is that they might perform better due to the variance decrement of their results. More details about the framework are presented in the following sections.

![FIGURE 2: Framework to obtain DL models for transfer learning in fault diagnosis.](image)

A. ARCHITECTURE DEFINITION

One of the most well-known tools for finding DL architectures is the Bayesian search. The work in [22] demonstrated that a random selection of the optimization space yields better results than a grid search. In a Bayesian process, the parameters and hyperparameters of the DL architecture ($HP$) are defined as the space of an objective function. The objective function aims to minimize the loss function of the DL architecture by modifying the defined space.

In this work, we propose multiple Bayesian processes to find the 1D-CNN model that performs well on the sub-source domains of each context. The training procedure is unfolded as follows.

The first step is to train a standalone model $M_r$ with random initial parameters and hyper-parameter and minimize a loss function $L$ for each source domain $h$ in each context $C_i$, i.e., $\{S^h_i, \forall h \in |S_i| \text{ and } \forall S_i \in C_i \}$.

\[ \min_{W,b} L(y_s, f(x_s, W, b)) \forall x_s \in S^h_i \quad (4) \]
The next step is to evaluate the versatility of the previous models for transfer operations using all target data available, and vary the models’ parameters to minimize the objective function’s loss ($E$) of the Bayesian process.

$$\min_{H, P} E[M_r \rightarrow M_k] \forall k \neq r$$

(5)

where $\rightarrow$ denotes a transfer operation. $M_r$ was trained with one sub-source domain and transferred to the remaining datasets by only tuning the top and last layer in $M_k$. The loss $E$ is defined as a weighted sum by $w$ of the average accuracy ($\overline{ACC}$) and the loss in the training and validation phases as follows.

$$E = w[(1 - \overline{ACC}_{train}) + L_{train}] + (1 - w)[(1 - \overline{ACC}_{val}) + L_{val}]$$

The result of this phase is a set of 1D-CNN models (size equals the number of contexts defined in Section II-C). The best and ensemble model are defined in the following section.

B. MODEL SELECTION

In Figure 2, two functions are proposed to the model selection. Each function is defined as follows.

- $\Gamma_1$ selects the best model denoted as $M_{\text{best}}$ with the minimum Bayesian loss among contexts from eq. (5). Such model serves as the base model and is defined as $M_{\text{base}} = \{f_r, f_m\}$.
- $\Gamma_2$ creates an ensemble of $M_{\text{base}}$. Each base model of the ensemble is trained with each source data available $i$, then the meta model is the union of the feature extraction functions and a combination of the mapping functions as $M_{\text{ensemble}} = \bigcup_{p=1}^{n} f_c[p] \Gamma_2(f_m)\}$. Three combination methods will be tested in $\Gamma_2$. i) Average: the Softmax layer results are averaged. The maximum value gives the class output. ii) Stacked: the Softmax layer results are stacked in a vector, and a new Softmax layer defines the class output. This is similar to a weighted combination. iii) Meta: the Softmax layer is discarded, and the last Dense layer activations are summed. Following, a new Dense and Softmax layer will define the class output. The difference between the above approaches is based on the number of parameters adjusted to make the final decision. The Average and Stacked approaches are based on the decision made by each model using its Softmax layers; they have no parameters or a minimum number of parameters to adjust, respectively. On the other hand, the Meta method constructs a new nonlinear function (with the Dense layer) that combines the results, thus adding more parameters to fit, which are relatively small compared to the parameters of the whole architecture. These approaches will be evaluated to find the one that maximizes the final performance.

The architecture of the final ensemble is illustrated in Figure 3. Since all models have the same convolutional layer dimensions, we can stack the results from each source domain. To evaluate the resulting models, we use datasets from different rotating machines, as detailed below.

IV. DATASETS

The GIDTEC\textsuperscript{3} research group created the selected datasets in controlled environments. Configurations of rotating machinery, i.e., gearbox, compressor, and others, were assembled. Artificial faults were placed (mainly in gears and bearings). The machinery was subjected to different working conditions (loads and speeds). We briefly summarize the data’s Experimental Plan (EP) of the source domain datasets that was strategically selected to have a rich set of vibrations covering two of the main topics of this paper: defective gears and bearings.

1) EP05

Tooth breakage severity failures have been emulated in the driver helical gear of a gearbox. Figure 4 shows the experimental setup. A motor drives the gearbox through a coupling. A frequency inverter controls the motor speed. The output shaft of the gearbox is connected to an electromagnetic torque load via a belt transmission. Four accelerometers measure the vibration signals in a vertical ($V_1$ and $V_2$) and horizontal ($H_1$ and $H_2$) configuration. Three different speeds and three loads were used as working conditions. The severity failures are listed in Table 1. Five repetitions of the experiments were performed with all possible combinations in the ($\text{fault, speed, load}$) tuple.

![FIGURE 4: Real setup of EP05.](image)

| Label | Fault mode |
|-------|------------|
| P1    | Healthy    |
| P2    | Breakage (11.5%) |
| P3    | Breakage (19.5%) |
| P4    | Breakage (28%) |
| P5    | Breakage (37.2%) |
| P6    | Breakage (51.7%) |
| P7    | Breakage (60.5%) |
| P8    | Breakage (70.2%) |
| P9    | Breakage (85.6%) |
| P10   | Breakage (100%) |

![TABLE 1: Severity faults in the tooth.](image)

\textsuperscript{3}https://gidtec.blog.ups.edu.ec/
2) EP02

Figure 5 shows the configuration of this case. Two bearings (B1 and B2) are coupled with a shaft as a supporting device. A motor drives the shaft at different speeds, and flywheels can be on the shaft to induce loads when required. Two accelerometers (V1 and V2) are placed vertically for each bearing. As in the previous case, three speeds and loads were tested. Table 2 shows the faults caused to the bearing components.

![Real setup of EP02](image)

**FIGURE 5: Real setup of EP02.**

The experimental plans for the target domain have similar configurations to those of EP05 and EP02. Table 3 summarizes the source and target domain datasets; the latter consists of a reciprocating compressor, single-stage and two-stage gearboxes, among others. The deployed sensors differ from machine to machine; therefore, we selected the vibration measurement closest to the fault condition.

**TABLE 2: Faults emulated in the bearings.**

| Fault mode | Label | B1 | B2 |
|------------|-------|----|----|
| P1         | Healthy | Healthy | Healthy |
| P2         | Inner race | Healthy | Healthy |
| P3         | Outer race | Healthy | Healthy |
| P4         | Ball | Healthy | Healthy |
| P5         | Inner race | Outer race | Outer race |
| P6         | Inner race | Ball | Ball |
| P7         | Outer race | Ball | Ball |

**TABLE 5: Transfer test from one speed to another in EP05. Average accuracy after 30 iterations.**

| Model(tuning) | F1 → F2 | F1 → F3 | F2 → F3 |
|---------------|---------|---------|---------|
| M_{best}      | 0.99    | 0.99    | 0.99    |
| VGG16         | 0.91 ± 0.016 | 0.90 ± 0.014 | 0.91 ± 0.012 |
| Resnet50      | 0.71 ± 0.023 | 0.70 ± 0.025 | 0.72 ± 0.003 |
| RF            | 0.98    | 0.98    | 0.98    |
| XGBoost       | 0.98 ± 0.011 | 0.98 | 0.98 ± 0.014 |

**TABLE 6: Transfer tests from one machine to another in EP02 to EP05, average accuracy after 30 iterations.**

| Model(tuning) | EP02 → EP05 | EP05 → EP02 |
|---------------|--------------|--------------|
| M_{best}      | 0.70 ± 0.23  | 0.95 ± 0.01  | 0.96 ± 0.006 | 0.98 ± 0.005 |
| VGG16         | 0.93 ± 0.01  | -             | 0.96 ± 0.008 | -             |
| Resnet50      | 0.64 ± 0.042 | -             | 0.73 ± 0.065 | -             |

The above results validate the construction of M_{best}. RF performs fairly well; consequently, it will be retained for the following tests.

To validate M_{best}, we perform transfer operations from one speed to another (F1 → F2 for example) by tuning the top layer (the first convolutional layer) and adding a new Dense and Softmax layer after layer 10. The trainable parameters are reduced from 394,680 to 11,110 with this regime. The results are compared with VGG16, Resnet50, Random Forests (RF), and XGBoost. The first two, which are image-based classifiers, use spectrograms of the vibration signals as input. RF and XGBoost use hand-crafted features in the time, frequency, and time-frequency domain [23]. TL with RF and XGBoost is not applicable; therefore, these models are trained from scratch in the target domain.

Table 5 shows the performance based on the average accuracy of each transfer operation after 30 repetitions. M_{best} outperforms the other approaches on the top layer tuning regime.

**V. EXPERIMENTS AND RESULTS**

In Section V-A, the target domain is evaluated in our framework with state-of-the-art models using classical feature extraction approaches and compared with results from TL. In Section V-B, the ensemble model is built and tested on the target domain. Finally, Section V-C presents an analysis and explanation of the results.

**A. RESULTS ON THE SOURCE DOMAIN**

The raw samples were input to our framework, which resulted in M_{best} in Table 4.

| Layer | Type   | Description                                      |
|-------|--------|--------------------------------------------------|
| 1-7   | Conv2D | kernels: 9 × 11 × 22 × 13 × 28 × 27 × 15       |
|       |        | kernel size: 8 × 3 × 18 × 17 × 6 × 11 × 14     |
|       |        | strides: 2                                      |
| 8     | Max Pooling | 2                     |
| 9     | Dropout  | 0.4                                      |
| 10    | Dense   | 100                                      |
| 11    | Softmax | number of classes                             |

To validate M_{best}, we perform transfer operations from one speed to another (F1 → F2 for example) by tuning the top layer (the first convolutional layer) and adding a new Dense and Softmax layer after layer 10. The trainable parameters are reduced from 394,680 to 11,110 with this regime. The results are compared with VGG16, Resnet50, Random Forests (RF), and XGBoost. The first two, which are image-based classifiers, use spectrograms of the vibration signals as input. RF and XGBoost use hand-crafted features in the time, frequency, and time-frequency domain [23]. TL with RF and XGBoost is not applicable; therefore, these models are trained from scratch in the target domain.

Table 5 shows the performance based on the average accuracy of each transfer operation after 30 repetitions. M_{best} outperforms the other approaches on the top layer tuning regime.

**TABLE 5: Transfer test from one speed to another in EP05. Average accuracy after 30 iterations.**

| Model(tuning) | F1 → F2 | F1 → F3 | F2 → F3 |
|---------------|---------|---------|---------|
| M_{best}      | 0.99    | 0.99    | 0.99    |
| VGG16         | 0.91 ± 0.016 | 0.90 ± 0.014 | 0.91 ± 0.012 |
| Resnet50      | 0.71 ± 0.023 | 0.70 ± 0.025 | 0.72 ± 0.003 |
| RF            | 0.98    | 0.98    | 0.98    |
| XGBoost       | 0.98 ± 0.011 | 0.98 | 0.98 ± 0.014 |

**TABLE 6: Transfer tests from one machine to another in EP02 to EP05, average accuracy after 30 iterations.**

| Model(tuning) | EP02 → EP05 | EP05 → EP02 |
|---------------|--------------|--------------|
| M_{best}      | 0.70 ± 0.23  | 0.95 ± 0.01  | 0.96 ± 0.006 | 0.98 ± 0.005 |
| VGG16         | 0.93 ± 0.01  | -             | 0.96 ± 0.008 | -             |
| Resnet50      | 0.64 ± 0.042 | -             | 0.73 ± 0.065 | -             |

- the accuracy is lower than 0.5 for a maximum of 60 epochs

The above results validate the construction of M_{best}. It is noteworthy that RF performs fairly well; consequently, it will be retained for the following tests.

**B. RESULTS ON THE TARGET DOMAIN**

In this case, the vibration signals were also sliced into windows (win) of 16,384 samples. However, for target domains containing less than 40 samples per class, a data augmentation process was proposed that consists of slicing the original raw signals with overlapping to produce more samples. For instance, for 3 samples per class, we randomly select 1 vibration signal and take 3 successive samples of size win. In Figure 6, a 50% overlap increases the number of samples to 5. For our experiments, the best trade-off between performance and number of samples was found with a 25% overlap.
TABLE 3: Dataset summary.

| Data   | Machinery Type       | Fault type       | Gear/ bearing | Number of faults in gears | Number of faults in bearings | Fault type | Gear/ bearing | Number of faults in gears | Number of faults in bearings | Fault type | Gear/ bearing | Number of faults in gears | Number of faults in bearings | Fault type | Gear/ bearing | Number of faults in gears | Number of faults in bearings | Fault type | Gear/ bearing | Number of faults in gears | Number of faults in bearings |
|--------|----------------------|------------------|---------------|---------------------------|------------------------------|------------|------------------|---------------------------|------------------------------|------------|------------------|---------------------------|------------------------------|------------|------------------|---------------------------|------------------------------|------------|------------------|---------------------------|------------------------------|
| EP02   | Single-stage gearbox | Severity Helical gear | EP03_2       | 6                          | 4                            | F1         | 12               | 15                        | 3                            | 12         | 15               | 3                          | 12               | 15               | 3                          | 15               | 12               | 15                         |
| EP05   | Single-stage gearbox | Severity Helical gear | EP03_2       | 6                          | 4                            | F2         | 12               | 15                        | 3                            | 12         | 15               | 3                          | 12               | 15               | 3                          | 15               | 12               | 15                         |
| EP02+  | Two-stage gearbox    | Multiple Ball bearing | EP03_2       | 6                          | 4                            | F3         | 12               | 15                        | 3                            | 12         | 15               | 3                          | 12               | 15               | 3                          | 15               | 12               | 15                         |

FIGURE 6: Data augmentation on target domains with less than 40 labeled samples per class.

The ensemble is built with $M_{best}$ using EP02, EP05, and EP02+EP05 as source domains. The combination methods are Average (Avg), Stack, and Meta (refer to Section III-B). For realistic TL scenarios, the number of labeled samples is limited to 10, 20, and 40 per class. The rest of the samples are marked as unlabeled (test set). Figure 7 shows the accuracy on EP03 test set, while Figure 8 shows the results on EP05. The Meta model (in orange) outperforms RF in most cases. It is also notable that the standalone 1D-CNN ($M_{best}$ in dark blue) provides the worst accuracy. This gives a clear indication that the ensemble approach defined by the Meta model is more beneficial for TL in our application. Therefore, the rest of the experiments will be focused on the Meta model performance against the second-best model (RF).

FIGURE 7: Transferring to EP03 at the working conditions: F1 and F2. Accuracy on the test set after 30 iterations.

FIGURE 8: Transferring to EP04 at the working conditions: F2 and F3. Accuracy on the test set after 30 iterations.

In addition, the average accuracy and Mean Square Error (MSE) are computed for each speed using the Meta and RF models. The results of each speed are averaged to present the summary in Figure 9. For EP03 and EP04, our approach performs as well as RF. EP06_1 was tested on two ensembles: Meta − 3 formed by EP02, EP05 and EP02+EP05 and Meta − 5 formed by Meta − 3, EP03 and EP04. The figure reveals that RF accuracy is low as the labeled samples are limited for EP06_1; a contrary case occurs with the proposal. Moreover, the results indicate that increasing the number of ensembles might slightly boost the performance when few labeled samples are available.

FIGURE 9: Average accuracy among speeds after transferring from the source domains to EP03, EP04 and EP06_1.

For EP06_2 and EP11, we will vary the size of the ensembles and the sensor. In Table 7, the transfer operation is performed on a multi-label classification task due to the
multi-fault signals in EP06_2. Meta − 5 created from different source domains yields better results. It suffices to use only one accelerometer placed vertically at any bearing (V₁ or V₂). However, the accuracy is considerably low using 10 and 20 labeled samples. This is because one of the working conditions provides very low throughput, while in another speed, the accuracy was upper than 0.8 on average with 10 samples. Therefore, it is more beneficial to use the working condition that maximizes the performance and then transfer to another working condition as in Table 5.

TABLE 7: Average accuracy after 30 iterations of transfer operations to EP06_2. Two accelerometers V₁ and V₂ are utilized.

| Sensor | Model | 10 | 20 | 40 | 70 |
|--------|-------|----|----|----|----|
| V₁     | RF    | 0.50 ± 0.019 | 0.59 ± 0.0049 | 0.66 ± 0.016 | 0.97 ± 0.0027 |
| V₁     | Meta − 3 | 0.56 ± 0.024 | 0.66 ± 0.0387 | 0.74 ± 0.03 | 0.93 ± 0.023 |
| V₁     | Meta − 5 | 0.66 ± 0.046 | 0.81 ± 0.042 | 0.87 ± 0.089 | 0.97 ± 0.0057 |
| V₁     | Meta − 2 | 0.66 ± 0.027 | 0.77 ± 0.13 | 0.90 ± 0.048 | 0.97 ± 0.0066 |
| V₁, V₂ | Meta − 5 | 0.61 ± 0.051 | 0.76 ± 0.071 | 0.88 ± 0.044 | 0.97 ± 0.0076 |

1. Meta − 3: EP02, EP05, EP02+EP05.
2. Meta − 5: EP02, EP05, EP02+EP05, EP03, EP04.

On the other hand, in Table 8, the vibration and acoustic measurements of EP11 were used (denoted as V₁ and A₁, respectively). In general, the acoustic signals yield better results using Meta − 5 for 10 and 20 labeled samples. Our approach correctly works for acoustic and probably for any cyclo-stationary signal.

TABLE 8: Average accuracy after 30 iterations of transfer operations to EP11

| Sensor | Model | 10 | 20 | 40 | 70 |
|--------|-------|----|----|----|----|
| A₁     | Meta − 3 | 0.88 ± 0.023 | 0.91 ± 0.011 | 0.96 ± 0.011 | 0.94 ± 0.014 |
| A₁     | Meta − 5 | 0.94 ± 0.012 | 0.97 ± 0.013 | 0.98 ± 0.0032 | 0.99 ± 0.010 |
| V₁     | Meta − 5 | 0.91 ± 0.043 | 0.94 ± 0.038 | 0.98 ± 0.011 | 0.99 ± 0.0034 |
| A₁, V₁ | Meta − 5 | 0.90 ± 0.044 | 0.95 ± 0.017 | 0.99 ± 0.0048 | 0.99 ± 0.0049 |

1. Meta − 3: EP02, EP05, EP02+EP05
2. Meta − 5: EP02, EP05, EP02+EP05, EP03, EP04.

C. ANALYSIS

Regarding the results in the source domain, Table 6 indicates that accurate transfer between EP02 and EP05 is only possible in one direction (EP05 → EP02) by tuning only the top layers. To understand the failed transfer (EP02 → EP05), we analyze the automatic features from the CNN layers in the frequency domain. To do so, we compute a modified FFT spectrum, which consists of convolving the intermediate time-series with the FFT of the filters of each layer. The result of each layer is summed to obtain a modified FFT spectrum. More details on this operation can be found in [21].

In Figure 10, the modified spectrum of a sample in the transfer operation EP02 → EP05 is compared to the original spectrum. The figures show that the automatic feature extraction filters out the higher frequencies and highlights the frequencies of interest. The average Dynamic Time Warping (DTW) distance between the original FFT and the modified for 1000 samples is 65.92 and 81.38 for tuning top and tuning all layers, respectively. Whereas the average DTW between the last two is 14.76. The automatic feature extraction of both tuning top and all is very similar. This analysis led us to conclude that the difference between the models lies in the mapping function.

To study the mapping function of the transferred models, we visualize the output distribution of the last Dense layer of two ensembles and their base models. The ensembles were tuned with 10 labeled samples from the bearing severity dataset EP06_1. The output space of the layer is reduced to a 2D space with t-distributed Stochastic Neighbor Embedding (t-SNE). Figure 11 shows that the contribution of each ensemble helps to separate the faults better, and in consequence, to build an accurate fault classifier.

One important result of this analysis is that it is possible to implement only one automatic feature extraction and create an ensemble of mapping functions. The new configuration would have given the same results as our adopted approach with lower computational cost. Finally, the models and framework are publicly available in the GitHub repository.

FIGURE 10: Original and modified spectrum of a vibration signal in EP05 with fault P6 and working condition F3.

FIGURE 11: Results on EP06_1: t-SNE plots of unlabeled samples predictions in the last Dense output layer.

https://github.com/fanikaro/fpacheco21
VI. CONCLUSION

In this work, 1D-CNNs were defined with Bayesian optimization to find the architecture best suited for TL operations. Initially, different feature extraction and mapping decision functions were obtained from which the best ones were selected for the base models of the ensemble. Next, the base models were trained on different data sources. We tested different combination methods and compared our approach with classical TL methods and stand-alone 1D-CNN. The results show improved performance, especially when few labels are available per class.

An analysis of the ensemble model revealed that the feature extraction remains unchanged from one base model to the other, thus demonstrating that the resulting functions extract invariant features. It is the mapping function that plays a key factor in the data discrimination of the ensemble model.

Finally, the resulting ensemble model represents a useful tool for practitioners in the field. Challenges still exist to improve accuracy when labeled samples are limited. In future works, the target domains can be enriched with the source domain, unlabeled data sources, or artificial data from finite element-based simulations or physical models of rotating machinery. Finally, further efforts are planned to cope with non-cyclo stationary signals.

REFERENCES

[1] M. Cerrada, R.-V. Sánchez, C. Li, F. Pacheco, D. Cabrera, J. Valente de Oliveira, and R. E. Vásquez, “A review on data-driven fault severity assessment in rolling bearings,” Mechanical Systems and Signal Processing, vol. 99, pp. 169 – 196, 2018.

[2] R. Yan, F. Shen, C. Sun, and X. Chen, “Knowledge transfer for rotary machine fault diagnosis,” IEEE Sensors Journal, vol. 20, no. 15, pp. 8374–8389, 2020.

[3] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, “Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data,” IEEE Transactions on Industrial Electronics, vol. 66, no. 9, pp. 7316–7325, 2019.

[4] H. MJ and K. J-M, “Bearing fault diagnosis under variable rotational speeds using stockwell transform-based vibration imaging and transfer learning,” Applied Sciences, vol. 8, no. 12, p. 2357, 2018.

[5] T. Han, C. Lin, W. Yang, and D. Jiang, “Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application,” ISA Transactions, vol. 97, pp. 269 – 281, 2020.

[6] P. Cao, S. Zhang, and J. Tang, “Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning,” IEEE Access, vol. 6, pp. 26 241–26 253, 2018.

[7] S. Shao, S. McAleer, R. Yan, and P. Baldi, “Highly accurate machine fault diagnosis using deep transfer learning,” IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2446–2455, 2019.

[8] M. J. Hasan and J.-M. Kim, “Bearing fault diagnosis under variable rotational speeds using stockwell transform-based vibration imaging and transfer learning,” Applied Sciences, vol. 8, no. 12, 2018.

[9] P. Cao, S. Zhang, W. Fan, C. Wang, G. Wen, and X. Zhang, “A novel bearing fault diagnosis method based on 2d image representation and transfer learning-convolutional neural network,” Measurement Science and Technology, vol. 30, no. 5, p. 055402, 2019.

[10] P. Cao, S. Zhang, and J. Tang, “Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning,” IEEE Access, vol. 6, pp. 26 241–26 253, 2018.

[11] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” IEEE Transactions on Neural Networks, vol. 22, no. 2, pp. 199–210, 2011.

[12] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, “Transfer feature learning with joint distribution adaptation,” in 2013 IEEE International Conference on Computer Vision, 2013, pp. 2200–2207.
ALIN DRIMUS is an Associate Prof. at the Department of Mechanical and Electrical Engineering at the University of Southern Denmark since 2016. He obtained a PhD degree in 2012 from University of Southern Denmark, where he worked on development of novel tactile sensors and the use of machine learning for tactile data interpretation. His research is in the area of biologically inspired tactile sensors, use of machine learning for decision making and development of intelligent embedded systems.

RENÉ-VINICIO SANCHEZ is the director of GIDTEC. Received the B.S. in Mechanical Engineering in 2004, from the Universidad Politécnica Salesiana (UPS), Ecuador. He got PhD in industrial technologies research in 2017 at the UNED, Spain. Currently, he is Professor of the Department of Mechanical Engineering in the UPS. His research interests are machinery health maintenance, pneumatic and hydraulic systems, artificial intelligence and engineering education.

LARS DUGGEN is an associate professor within mathematical modelling at the University of Southern Denmark and currently heading the Centre for Industrial Mechanics and SDU Mechatronics. His research focus is concentrated on modelling and optimization of mechatronic systems as well as acoustic sensors, including macro- and microscale piezoelectric effects.

MARIELA CERRADA, IEEE senior member, received her B.Sc. in system engineering in 1993, and the M.Sc. in control engineering in 1998, from the University of Los Andes (ULA), Venezuela. She received her Ph.D. from the National Institute of Applied Sciences, Toulouse, France, in 2003. She was with the Control Systems Department at ULA from 1994 to 2016; she was Prometeo researcher at the Salesian Politechnique University (UPS) in Cuenca, Ecuador, from 2014-2015 and 2016-2017. She is with the Mechatronics Department at UPS since 2017. Her main research interests are industrial intelligent supervision and control, fault diagnosis and prognostics, and machine learning.

DIEGO CABRERA received his B.Sc. in Electronic Engineering from the Universidad Politécnica Salesiana, Ecuador, in 2012. He received his M.Sc. degree in Logic, Computation and Artificial Intelligence, and his Ph.D. degree in Computer Science from the Seville University, Spain, in 2014 and 2018 respectively. Currently, He is a Postdoctoral Fellow at the Dongguan University of Technology, China, and a Professor at the Universidad Politécnica Salesiana, Ecuador, from 2014. His research interests include intelligent systems, data-driven modelling and fault diagnosis.