A fine-tuning deep learning framework to palliate data distribution shift effects in rotary machine fault detection

Nima Rezazadeh[1], Donato Perfetto[1], Mario de Oliveira[2], Alessandro De Luca[1]*, Giuseppe Lamanna[1]

[1] Department of Engineering, University of Campania "L. Vanvitelli", Via Roma 29, 81031 Aversa, Italy.
[2] College of Engineering, Birmingham City University, Birmingham B4 7XG, UK.
*alessandro.deluca@unicampania.it

Abstract. In the condition monitoring of rotating systems, overfitting is a common challenge due to limited data history, which reduces the effectiveness of fault detection frameworks; this limitation often leads to unreliable diagnostics, resulting in unexpected machine failures and increased operational costs in industrial applications. Advances in deep learning suggest using simulated data to address this issue, but operational variabilities still cause significant data distribution shifts, affecting model accuracy. This paper presents a new vibration-based monitoring framework that improves fault detection in rotary machines by effectively managing these shifts. It features a novel fine-tuning approach within sequential domain adaptation, requiring only a limited number of observations from the target domain for accurate model adjustment. The domain adaptation process is elucidated through a novel visualization of internal activation patterns within the sequential network. This method is further enhanced by a hybrid algorithm that combines wavelet transformation, a multi-layer perceptron, and a transformer encoder, followed by domain-specific fine-tuning. The framework's effectiveness is demonstrated through experimental data from two different rotor systems, validated by sensitivity and comparative analyses, highlighting its robustness, generalizability, and practical applicability as a baseline in industrial fault detection scenarios.

Keywords: SHM, Operational Variabilities, Domain Adaptation, Transformer Encoder, Intelligent Systems.

Acronyms List

CNNs	Convolutional Neural Networks
FT	Fourier Transform
KNN	K-Nearest Neighbor
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
MRMR	Minimum Redundancy Maximum Relevance
OVs	Operational Variabilities
RNNs	Recurrent Neural Networks
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TE	Transformer Encoder
TL	Transfer Learning
VCM	Vibration-based Condition Monitoring
WT	Wavelet Transform

1. Introduction

The shift towards data-driven fault detection techniques marks a significant advancement in the field of rotating machinery maintenance. By applying machine learning (ML) and deep learning, industries are now able to predict and diagnose faults with increased precision and at an earlier stage with respect to the available classical inspection methods [1]. This transition to intelligent fault detection methods is consistent with the broader goals of Industry 4.0, aiming to enhance operational reliability, reduce maintenance costs, and ensure the safety of both equipment and personnel [2–4]. Moreover, these

methods address the limitations of traditional vibration monitoring techniques, which may suffer from practical issues such as sensor installation challenges and signal interference [5–8].

Recent research has focused on the development of adaptable ML models for rotor systems, intending to address (operational variabilities) OVs and improve fault detection, even in cases where historical fault data is limited. Cyclostationarity-based machine learning leverages periodic statistical properties of signals to enhance detection, classification, and prediction in time-varying environments. Additionally, physics-informed machine learning integrates physical laws and constraints into machine learning models to enhance accuracy, generalization, and interpretability, particularly in complex systems with limited data[9,10].

Research studies have deepened knowledge in feature-based and model-based methods for this purpose. In their research, Wang et al. [11] proposed a feature-based domain adaptation model that uses wavelet scalograms and bilinear interpolation for bearing fault diagnosis. Although this method improves fault detection and minimizes noise interference, it is constrained to diagnose faults within the same machine and necessitates manual tuning of its parameters. These limitations could hinder the model's effectiveness in diverse industrial environments and increase the operational complexity, potentially impacting its practical deployment. To tackle the challenge of insufficient data in the target domain, Xiang et al. [12] employed an unsupervised learning strategy that includes convolution layers and multiple-kernel maximum mean discrepancy to extract malfunction indicators. However, these approaches could lead to biased diagnostics and reduced model efficacy when the simulated conditions do not precisely mirror actual operational environments. Zhu et al. [13] developed a domain adaptation framework using visual vibration data from high-speed cameras to diagnose machinery faults across OVs, incorporating phase information in image sequences. The study highlighted multiple limitations including the model's potential lack of generalizability to other types of machinery without modifications. Its heavy dependence on high-quality, well-preprocessed data, and the significant computational resources it requires.

Cao et al. [14] introduced an unsupervised adversarial network framework for malfunction diagnosis, incorporating domain-specific batch normalization and clustering to enhance vibration data feature extraction. They demonstrated their methodology's effectiveness with gearbox and bearing datasets, noting potential limitations such as feature scattering and persistent domain-specific statistics that could impact model generalization in new environments. Jiang et al. [15] proposed a fault detection method employing convolutional neural networks (CNNs) and augmented orbit images from denoised vibration signals; the approach was shown to significantly enhance the model's diagnostic accuracy across different machines and fault types. However, the method's effectiveness is contingent on precise noise reduction and substantial data augmentation, has high computational demands, and its general applicability to diverse machinery without risking overfitting remains uncertain. Jiang et al. [16] proposed a gearbox fault diagnosis method combining vibration response-generated simulation signals and enhanced unsupervised domain adaptation to address the lack of high-quality labeled data. This method effectively narrows domain discrepancies and boosts diagnostic accuracy with minimal real samples. Although its success hinges on the robust alignment of simulated and experimental data distributions, the approach may struggle with extremely noisy or non-representative experimental environments.

In model-based fault diagnosis for rotor systems, two main strategies are commonly employed. The first strategy involves fine-tuning a well-known pre-trained network with target domain samples, while the second one consists in training a network with source domain data captured from similar machines and then fine-tuning it with a restricted count of target domain samples. On one hand, the former relies on image processing principles, given the prevalence of trained CNN models. On the other hand, the latter uses simulated (whether experimentally or numerically) data as the source domain. Kumar et al. [17] leveraged VGG19 to classify faults in squirrel cage motors from images

derived from 1D current signals. While the results indicated a potentially generalizable framework, the use of 450 observations for each health state, where 70% dedicated to fine-tuning raises concerns about the model's efficacy when confronted with a smaller target domain, which is more typical in transfer learning (TL) scenarios. Additionally, the omission of factors such as operational noise and other variabilities in the study could compromise the robustness and real-world applicability of the fault detection model, particularly in variable industrial environments. Similarly, Zhang et al. [18] adapted ResNet18 with wavelet and Fourier transform (FT) for diagnosing misalignment, rubbing, and unbalance in machinery. A FEM model was verified with a real rotating system, then the verified model was used to create the target dataset where, for each of the 10 health scenarios, 300 and 100 sample tests were assigned to the fine-tuning and validation, respectively. Asutkar and Tallur [19] introduced a framework to enhance CNNs' generalization in machine fault diagnosis across different domain datasets. They examined one source domain and two target domains, analyzing the effects of the number of retrained layers and observations in the fine-tuning phase for one target domain. The models require significant computational resources for training and inference, potentially limiting their practical application. Additionally, using a minimal number of observations (70) from the target domain resulted in notably low performance, close to 60%.

Zhang et al. [20] enhanced the performance of a neural network for fault detection in rotor systems by fine-tuning it. Data from the same machinery, operating under different conditions, was used as source and target domain data. The network was trained directly using raw vibration signals. During fine-tuning, each health condition was represented by 1208 samples, which does not reflect a scenario of data scarcity in ML. Yang et al. [21] developed a custom CNN for detecting faults in gas turbines, tackling data inconsistencies across manufacturers. They assessed the model using simulated data from different turbines and the impact of frozen layers. The fine-tuned CNN excelled with just 40 faulty observations, while a basic SVM also diagnosed four out of five fault categories effectively. This suggests discrepancies in feature distribution were mostly addressed during feature extraction, allowing for a robust assessment of the model in challenging scenarios. Tang et al. [22] combined a semi-supervised long short-term memory (LSTM) network with entropy gain to detect bearing and gearbox faults, emphasizing multi-domain features evaluation and optimal selection with limited labeled data. However, the study primarily uses the same machinery under varied conditions for both source and target domains, which may limit the model's applicability to entirely new machines or diverse operational scenarios.

Conventional automatic feature extraction layers, such as convolution layers in CNNs and gate mechanisms in recurrent neural networks (RNNs), faced challenges with long-range dependencies and high-dimensional data when analyzing rotating machines, which can potentially affect model accuracy and interpretability. As an alternative, transformers have been shown to provide a powerful solution to these limitations, significantly enhancing the field of feature engineering [23]. Tang et al. [24] employed a signal transformer for fault diagnosis in bearing systems, incorporating signal embedding for high-dimensional data representation, a transformer block for deep feature extraction utilizing multi-head self-attention, and an attention visualization for model interpretability. The implemented feature extraction framework improved the final performance by enhancing interpretability and managing high-dimensional data, however, it did not consider the capturing of short-time dependencies in the signal, which are critical for accurate fault diagnosis in dynamically changing operational conditions.

Jin et al. [25] developed a time-series transformer for fault diagnosis in rotating machinery, using a time-series tokenizer to convert 1D vibration signals into input token sequences. This model implements a transformer architecture with multi-head self-attention and classification layers for fault mode recognition. According to the authors, it outperforms CNNs and LSTMs in capturing long-term dependencies and feature extraction. However, this approach may not effectively address the local variations in the signal, which are essential for detecting subtle anomalies. Weng et al. [26] introduced a

multisensory fusion transformer for fault diagnosis in rotating machinery, which combines adaptive multisensory signal fusion with an enhanced transformer that incorporates a local learning unit for feature extraction. While this model successfully integrates data from multiple sensors, it may not fully optimize the extraction of local features from individual sensors, potentially missing critical fault indicators specific to certain sensor readings.

Li et al. [27] used adversarial domain adaptation with asymmetric mapping and CORAL alignment for intelligent fault diagnosis, verifying through experiments on public and private datasets. The method showed high diagnostic accuracy but lacked verification under variable loads and real operational conditions. Despite achieving good diagnostic effects, applicable only to small-range speed variations and lacking verification for load-induced distribution differences are the limitations of the proposed method. Yang et al. [28] proposed a transformer neural network-based approach for fault diagnosis in rolling bearings, utilizing segmented vibration signal processing and a pure attention mechanism for indicator extraction. Although this approach enhances the extraction of dominant features through segmentation, it struggles with the continuity and temporal dependency of segments, which can be vital for diagnosing evolving faults. Similarly, Hou et al. [29] after employing the FT utilized a multi-feature parallel fusion encoder and a cross-flipped decoder for fault detection in bearings. This approach, while robust for stationary signals, does not adequately adapt to non-stationary signal characteristics that frequently occur in mechanical failures. Xu et al. [30] developed a transfer diagnosis model for unbalanced samples, employing dual attention mechanisms and Siamese feature fusion, supported by feature visualization for interpretability. The designed framework showed high performance for bearing fault data collected under different loads and planetary gear fault data obtained at varying speeds, although it was suggested that a clearer physical interpretation is required.

Recent studies in the field of fault diagnosis for rotor-bearing systems, vital for industrial operations, have identified diverse critical limitations that impede the development of robust and accurate diagnostic tools. Key challenges include the need for enhanced local feature extraction and the handling of non-stationary signals, as well as the management of short-term dependencies and temporal continuity in signal processing. Moreover, generalization across different machine types remains a significant hurdle, coupled with a high dependence on quality data and extensive computational resources. Adapting systems to varied operational conditions typically requires significant manual tuning, and there is a notable shortage of real data available for model training. Furthermore, shifting TL models to interpretable modes is still in its preliminary stages and requires further research. To circumvent these challenges, this work designs a vibration-based condition monitoring (VCM) framework where a combination of (wavelet transform) WT, (multi-layer perceptron) MLP, and (transformer encoder) TE followed by sequential domain adaptation is implemented. Furthermore, a visualization technique, which plots the activation from the dense layers, was employed to graphically illustrate the steps of the used domain adaptation method, aiming to demystify the model's operations. This effort to visualize the domain adaptation process served to make the underlying mechanisms more transparent, thereby reducing the perceived "black-box" nature of the model and enhancing its interpretability.

Despite advancements in fault diagnosis for rotor-bearing systems, challenges persist in effective local feature extraction, handling non-stationary signals, managing temporal dependencies, generalizing across different machines with limited data, and enhancing model interpretability.

In summary, the main contributions of this paper are:

- a hybrid feature extraction framework that effectively addresses both long-term and short-term dependencies in signals. This framework integrates WT and MLP before employing TE, enabling the consideration of both local and global dependencies. The inclusion of WT is particularly advantageous for capturing features in non-stationary signals, which traditional methods like FT may fail to detect.

- a TL model designed to function under the constraints of significant instance shortages during the fine-tuning phase. This model is built on the premise that both source and target domain data are experimentally obtained, with the latter originating from a different machine and subjected to various uncertainties such as operational variations, sensor discrepancies, and background noise.
- a visualization step that delineates the internal operations of the TL model, enhancing its interpretability for subsequent analysis.

The remainder of the paper is organized as follows: Section 2 introduces the methodology that explores WT-MLP-transformer for indicator extraction and sequential domain adaptation for the classification model. Section 3 details the case studies, focusing on the source and target domain data. The experimental validation of the proposed VCM framework is presented in Section 4. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Methodology

This study proposes a novel methodology rooted in the principles of VCM systems. Typically, such systems analyze vibration signals from one or more sources to produce outcomes, identified as class labels, representative of the health status of the machinery. Previous studies have highlighted the challenges faced by VCM systems, including their tendency towards low accuracy and limited generalizability, often originating from the extraction of inferior quality features and OVs. To address these issues, this research aims to enhance the traditional VCM framework by integrating a hybrid feature engineering model (WT, MLP, and TE) with sequential domain adaptation to improve fault diagnosis performance in rotating machinery. The method unfolds through separate phases: (i) the feature extraction from vibration data; (ii) the training of a sequential model on these health indicators; (iii) the fine-tuning of this model to enhance its predictive accuracy, and (iv) the classification of the machinery's health condition based on the model's output. Figure 1 shows the overall process of the designed fault diagnosis based on the domain adaptation. Specifically, the phase (i) is represented by the signal processing box, which includes WT, MLP, and TE; phases (ii) to (iv) can be grouped into the domain adaptation box and described as an integrated framework.

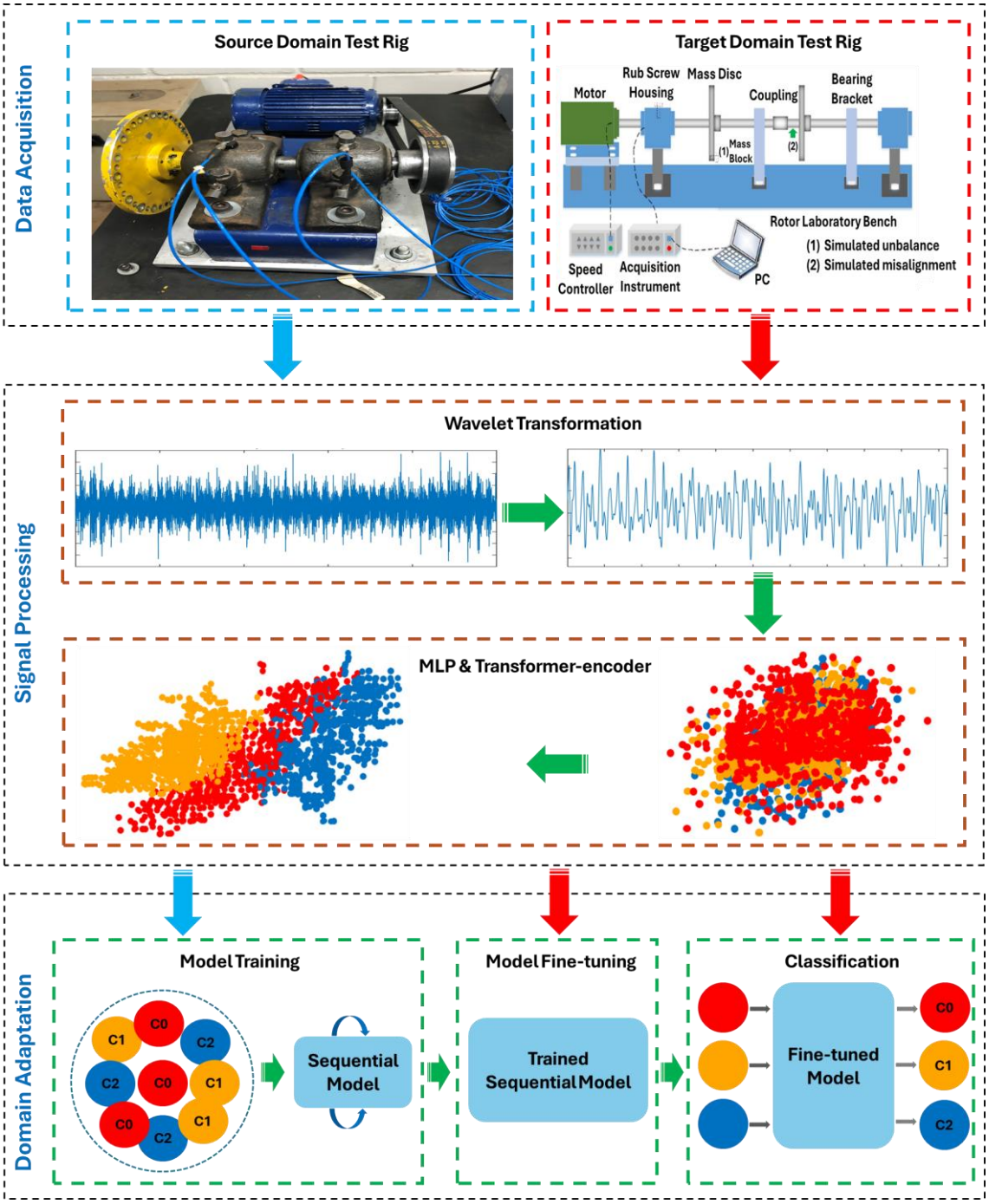


Figure 1. Workflow of the proposed method for fault detection utilizing domain adaptation

2.1. Signal processing

The initial stage of the model's feature extraction process begins with the WT applied to the vibration signals collected from the rotor system. WT effectively isolates frequency components across different scales, capturing transient features from non-stationary signals, providing superior time-frequency localization, focusing on short-term dependencies, and facilitating multi-resolution analysis for enhanced detection and characterization of signal anomalies. Following this phase, the dimensionality of WT coefficients reduced when implementing MLP. The MLP is a stack of fully connected layers that perform a series of linear transformations on the input data. Each layer within the MLP is designed to map the data to a new space via a combination of weight matrices and bias vectors (linear transformation) followed by non-linear activation functions. Non-linear activations introduce the capability to detect and model non-linear relationships within the data, which is a

common characteristic in real-world signals. The sequential application of these layers serves to gradually reduce the dimensionality of the input data, which not only simplifies the data's complexity but also helps in emphasizing the most salient indicators.

After the data has been processed through the MLP, it is fed into a TE. The TE is the cornerstone of the model's ability to oversee sequential data. Unlike the MLP, which treats the input as a collection of independent features, the TE analyzes the data within its sequence, allowing it to understand the temporal dynamics and the contextual relationships between various parts of the data. This is achieved through the multi-head self-attention mechanism, which can be understood as the model's ability to focus on various positions of the input sequence when predicting each element. It gives the model the ability to "look back" at other positions in the input sequence, granting it a form of memory that is capable of capturing long-range dependencies.

Each layer of the TE combines this self-attention with position-wise feed-forward networks (FFNs). The FFNs process the data point-wise, but in a way that allows interactions between the distinctive parameters as transformed by the self-attention mechanism. Moreover, a dropout is applied within these layers as a regularization technique. By randomly zeroing out a subset of features in each layer during training, the model becomes more robust, less sensitive to noise, and less prone to overfitting on the training data.

The ultimate step in the TE is the pooling operation. Once the sequential data has been thoroughly processed by the multi-head self-attention and feed-forward networks, it is imperative to synthesize this information into a coherent form that can be used for further tasks such as classification. The pooling operation condenses the sequence of transformed features into a single vector that encapsulates the essence of the data's attributes. This vector serves as a compact, informative representation of the entire input sequence, summarizing the rich and complex information extracted through the MLP and TE processes. The detailed parameter extraction procedure except for the WT is represented in the following pseudocode. Figure 2 depicts the workflow of the structured feature extraction framework, illustrating the process from feeding wavelet coefficients as input to extract a property vector as the output.

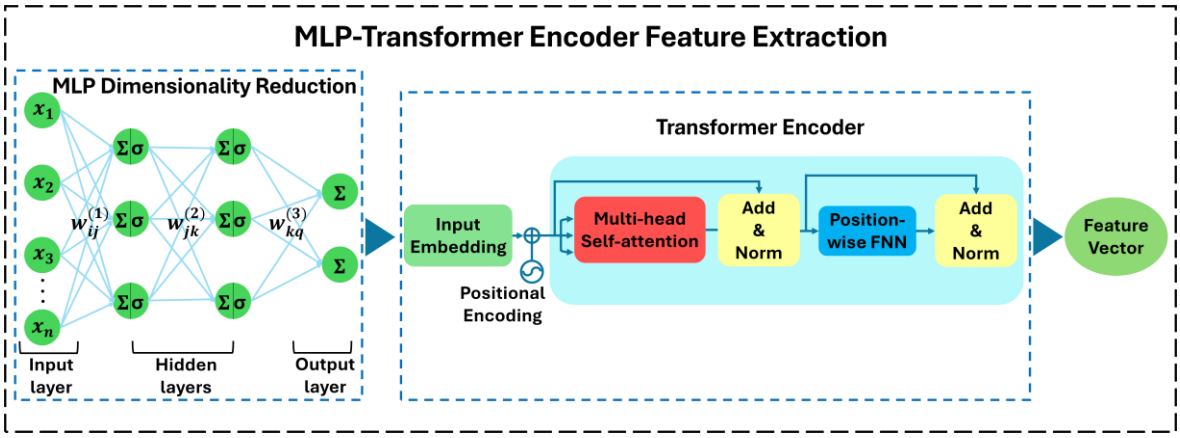


Figure 2. Workflow of the designed MLP-TE feature extraction model

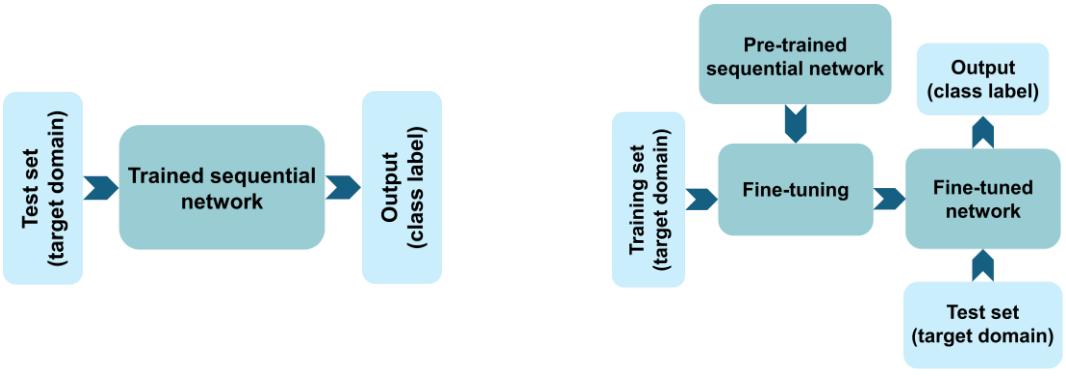
Algorithm 1 presents the hybrid feature extraction implemented in this approach.

Algorithm 1. Hybrid feature extraction

Input: Number of input features N_{in}, number of output features N_{out},
number of attention heads H, number of transformer layers L,
dropout rate d, device type *dev*
Output: Extracted feature matrix F, label vector L
Define FeatureExtractor model
Define a sequence of Linear transformations and activation function (af) for the *MLP*:
for layer l in {1,2,3}, compute intermediate representations:
$X^{(l)} = af\ (W^{(l)} * X^{(l-1)} + b^{(l)})$
end
where $W^{(l)}$ and $b^{(l)}$ are the weights and biases for layer l, $X^{(0)}$ is the input data,
and $X^{(l)}$ is the output of layer l;
Define TransformerEncoder *TE* with H attention heads, L layers, and dropout rate d:
for each layer l in {*1*, ..., L} within *TE*, apply multi-head self-attention and
position-wise feedforward networks:
$X^{(l)}$ = TransformerLayer($X^{(l-1)}$, H, d)
end
where TransformerLayer includes multi-head attention and feedforward networks with dropout;
Feature extraction loop
Set FeatureExtractor to evaluation mode;
Initialize empty lists F_{list} and L_{list} to hold features and labels;
for each epoch $i = 1,2, \ldots, E$ do:
for each mini-batch (X_{batch}, Y_{batch}) in *DL* do:
a. Transfer X_{batch} to dev.
b. Compute the MLP feature representation for X_{batch}:
Z_{MLP} = $MLP(X_{batch})$
c. Pass Z_{MLP} through the TE and apply pooling:
Z_{TE} = TE(Z_{MLP}.unsqueeze(1))
Z_{pool} = Mean(Z_{TE}, dim=1) // Apply pooling operation over sequence dimension
d. Append the pooled feature representation Z_{pool} to F_{list} and Y_{batch} to L_{list}.
end
end
Concatenate all feature representations and labels from F_{list} and L_{list} to form F and L.
Result: Extracted features

2.2. Sequential-based domain adaptation

In this work, a straightforward yet effective approach to adapt a model from one area of knowledge (source domain) to another set, referred to as the target domain, was explored. This method falls under the category of parameter-based (model-based) domain adaptation techniques. The process involves the training of the model on the source domain data. Here, emphasis was placed on minimizing errors through an optimization method, specifically utilizing an approach called stochastic gradient descent (SGD). Following the initial training phase, the model's knowledge was refined by adjusting its parameters to focus more on the target domain data; this refinement, often termed as fine-tuning, involved making minor adjustments to the model's parameters, either in all layers or in the layers closer to the output layer. The decision of whether to fine-tune all layers of the network or only select batches of them, with the rest being left unchanged (or "frozen"), is influenced by profuse factors. A schematic of a classification framework without and with a domain adaptation (a model-based approach) is displayed in Figures 3-a and b, respectively.



(a) **(b)**

Figure 3. Schematic of a classification network (a) without a domain adaptation step and (b) with a fine-tuning step

An integral part of the strategy consisted not only of selecting the layers to train but also of tweaking the model's settings(hyperparameter optimization) to ensure optimal performance. This includes the use of densely connected layers, activation functions for facilitating non-linear connections, regularization to prevent overfitting, and batch normalization for more stable and efficient training. This approach was characterized as a supervised domain adaptation technique, as it relied on labeled data from the target domain to guide the fine-tuning process. Through these steps, including the strategic use of model checkpoints to save the best configuration, the framework aimed to achieve an elevated level of adaptability and performance in domain adaptation tasks. After fine-tuning, the test subset of the target domain data was fed into the model to evaluate its performance in terms of classification metrics. Algorithm 2 presents the process of training, fine-tuning, and evaluation of the designed domain adaptation model.

Algorithm 2. Domain adaptation employing sequential network

Input: Source dataset *S*, target dataset *T*, learning rates η_s, η_t, momentum μ, regularization factors λ_1, λ_2, and number of classes *C*

Output: Trained model parameters θ, evaluation metrics

Training on Source Domain

Initialize the neural network with weights θ_0;
Compile the model with *SGD* optimizer with learning rate η_s and momentum μ, loss function $L(\theta) = -\sum y_i \log(p(y_i|\theta))$ and accuracy metric;
for each epoch $i = 1,2, \dots, E$ do:
  for each mini-batch $B_j \subseteq S_{train}$ do:
    Perform forward pass to compute predictions $p(y|\theta)$;
    Apply L_1L_2 regularization: $R(\theta) = \lambda_1 \sum|\theta| + \lambda_2 \sum \theta^2$;
    Compute total loss: $L_{total} = L(\theta) + R(\theta)$;
    Update θ using backpropagation: $\theta \leftarrow \theta - \eta_s \nabla L_{total}(\theta)$;
  end
  Save checkpoint if validation accuracy on S_{test} improves;
end

Fine-tuning on Target Domain

Unfreeze selected layers for fine-tuning;
Compile the model with *SGD* optimizer with learning rate η_t and momentum μ;
for each epoch $i = 1,2, \dots, F$ do:
  for each mini-batch $B_j \subseteq T_{train}$ do:
    Perform forward pass to compute predictions $p(y|\theta)$;
    Compute loss $L(\theta) = -\sum y_i \log(p(y_i|\theta))$;
    Update θ using backpropagation: $\theta \leftarrow \theta - \eta_t \nabla L(\theta)$;
  end
  Save checkpoint if validation accuracy on T_{val} improves;
end

Evaluation on Target Domain

Evaluate the model on T_{test} to compute test accuracy $Acc = \sum \mathbb{1}\left(y_i = argmax\left(p(y|\theta)\right)\right)/T_{test}$;
Generate confusion matrix $CM = [cm_{ij}]$ where cm_{ij} is the number of instances of class *i* predicted as class *j*;

Result: Fine-tuned model parameters θ, test accuracy *Acc*, confusion matrix *CM*

3. Case study

In the present paper, two distinct datasets were chosen to represent the source and target domain data. Each dataset was generated using an experimental setup in the laboratory at contrasting times and by different research groups. The selection of the two datasets for this study is strategic, as they both encompass three health scenarios, i.e., normal, unbalanced, and misaligned conditions. This is crucial because the study aims to develop a TL framework that can effectively transfer knowledge from a larger domain (source) to a smaller one (target). Choosing a limited dataset as the target domain can illustrate data scarcity which is a prevalent issue in practical applications [31]. To ensure the effectiveness of this transfer, the network must be trained on the same types of faults that will be evaluated during the adaptation and testing phases, even if the severity of these faults varies. This approach ensures that the framework is robust and can accurately detect and classify faults under

OVs. While these two datasets are detailed in the following subsections, the primary difference between the source and target domain sets is that the source domain data was released as raw vibration signals, whereas the target domain data was provided as extracted wavelet coefficients.

3.1. Source domain dataset

In a newly published dataset [32], the efficacy of rotary machines was analyzed across various operational health conditions, including unbalance, misalignment, mechanical looseness, and standard operating conditions. This comprehensive dataset is now publicly accessible and provides the source domain data for further examination in this study. Figure 4 presents the test rig at the core of the experiments for producing this dataset. The test bench is equipped with a motor also with a frequency inverter to adjust the motor's speed, bearings to help smooth rotation, pulleys, a belt to transmit power to the rotor, and the rotor itself. To collect data displaying the machine behavior across various health states, a series of five experiments were conducted for every health scenario. Each experiment involved recording four groups of 420 signals, where every signal comprised 25000 data points captured at a sampling rate of 25 kHz. This process resulted in a total of 8400 signals being captured using accelerometers, with each health state providing 2100 signals. The accelerometers were strategically positioned at four critical locations around the machine, on the vertical and horizontal sides of both the coupling and the disk. Thus, each observation yielded four raw vibration signals.

While data generation was carried out in an isolated laboratory environment, before initiating any test, the bench was dismantled and restored to its normal operating condition to introduce faults later. This procedure permits variations, making the tests more reflective of real industrial scenarios. Signals were gathered using accelerometers (PCB 333B30, 2 per bearing), placed in vertical and horizontal orientations. A power supply (PCB 482A16) and an acquisition board (Hi-Speed USB Carrier, NI USB-4432) were utilized. A Python program was developed for data collection. All accelerometer sensitivities were calibrated according to the calibration chart, and the rotational speed was maintained at approximately 1772 rpm.

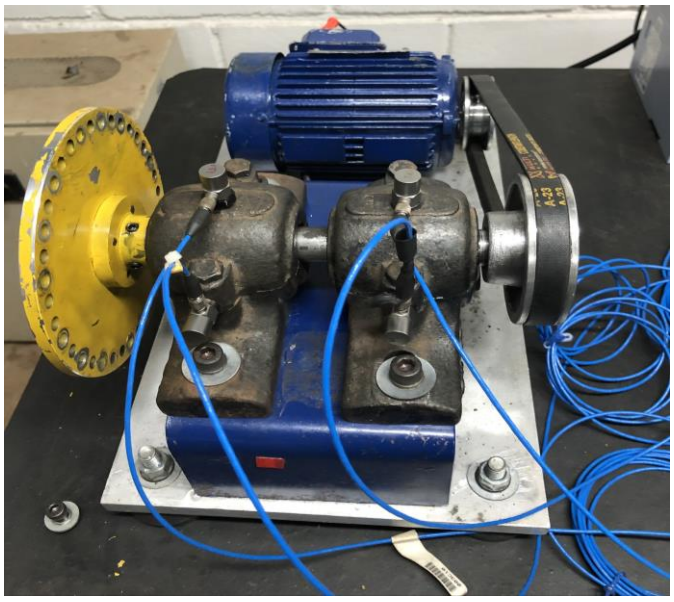


Figure 4. Test rig of source domain dataset [32]

To maintain consistency with the target domain data concerning health conditions, observations belonging to the normal, unbalanced, and misaligned scenarios were preserved. Table 1 provides a comprehensive overview of this dataset, delineating the described conditions, their corresponding labels, and the count of entries for each category. As previously mentioned, each of the 2100 observations encompasses 4 sets of signals gathered from the 4 accelerometers. Specifically, 85% of

the observations (1785) were used for the training of the network, whereas the remaining 15% (315) was used for validation intents.

Table 1. Source domain dataset parameters

Health scenario	No. of observations for training	No. of observations for validation	Class Label
Normal	1785	315	0
Unbalanced	1785	315	1
Misaligned	1785	315	2

3.2. Target domain dataset

As concerns the target domain, the WHU rotor dataset [33] sourced from Wuhan University was considered. This dataset encompasses vibration signals obtained from an experimental rotating machinery system captured in a laboratory scale, displayed in the schematic of Figure 5, including a rotor test platform, a speed controller, a front-end processor, and a computer. Captured through two eddy current sensors, strategically positioned on the sensor bracket directly above the rotor, the vibration data ensures the precise measurement of the rotor's condition. Furthermore, this dataset is designed for the identification of four distinct rotor states: normal, contact-rubbing, unbalance, and misalignment, with each state represented by 45 observations, leading to a total of 180 instances. Moreover, each sample encapsulates 2048 data points, reflecting the vibrations over a duration of 1 second, whilst the rotor operates at a speed of 1200 rpm and a sampling frequency set at 2048 Hz. The vibration signals underwent preprocessing, including denoising utilizing wavelet thresholding-based techniques, and were subsequently resampled into a 2-dimensional matrix format. The dataset was obtained using a rotor laboratory bench driven by a DC motor with a rated current of 1.95A and a maximum output power of 148W, controlled by a DH5600 speed controller. The rotor, with a diameter of 10 mm and a length of 850 mm, included two shafts connected by a coupling and two mass disks with a diameter of 75 mm. A rub screw housing for contact-rubbing experiments and sensor installation was mounted on the bench. The data acquisition device used was the GTS3-TG series test simulator, which transmitted the collected signals to the front-end processor for amplification and filtering before passing them to the computer for analysis and storage. Copious studies have subsequently utilized the data derived from this experiment [34–38].

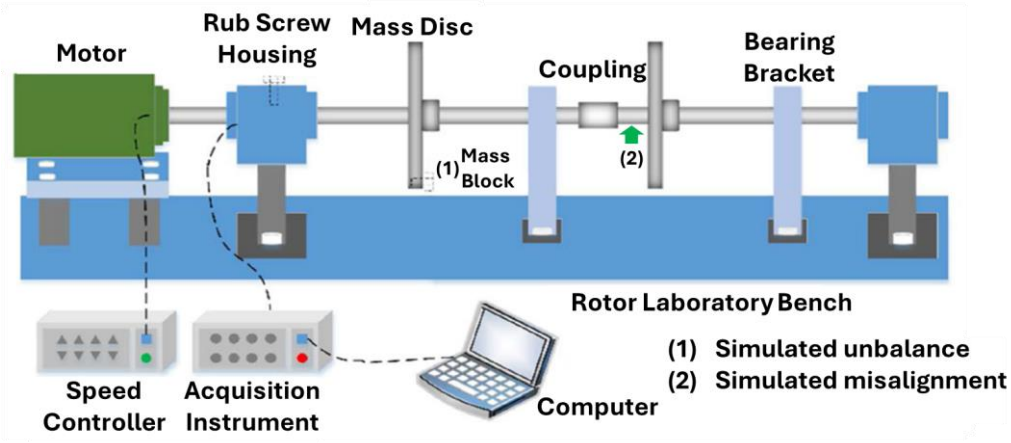


Figure 5. Schematic of test bench of the target domain dataset [33]

It is important to emphasize that for further evaluation, health scenarios matching those in the source domain dataset were chosen as a result the stored target dataset in this study contains 135 observations. Table 2 provides a summary of the selected sample tests from this dataset. Specifically, 80% of the observations (36) were used for the fine-tuning procedure, 4.5% (2) for the validation of the network, and the remaining 15.5% (7) was used for the testing phase.

Table 2. Target domain dataset parameters

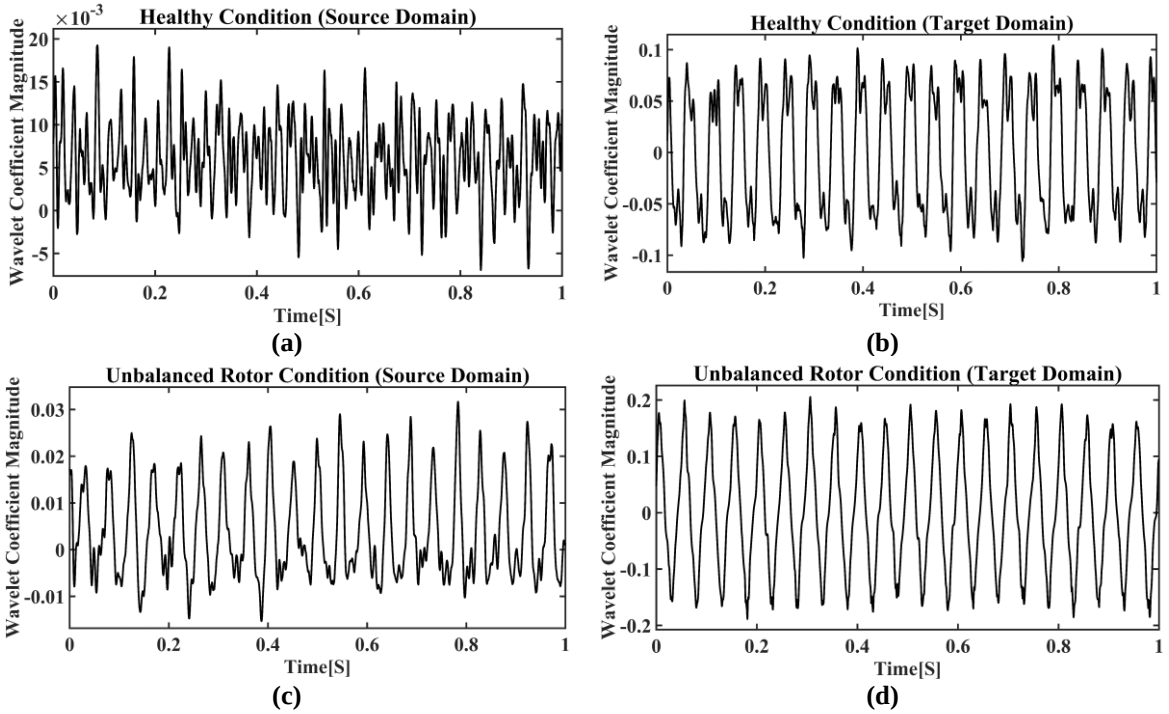
Health scenario	No. of observations for fine-tuning	No. of observations for validation	No. of observations for testing	Class Label
Normal	36	2	7	0
Unbalanced	36	2	7	1
Misaligned	36	2	7	2

4. Results and discussion

The results analysis is preliminarily obtained towards demonstrating the capability of the proposed VCM framework to accurately categorize different health conditions. It operates under the assumption that the target domain lacks adequate observations and is susceptible to uncertainties arising from OVs in rotating machinery. As previously noted, the dataset from the source domain includes readings from four accelerometers, with two oriented horizontally and two vertically. Upon analysis of these signals, it was found that the data collected by sensor number 3, which is positioned vertically near the disk's coupling, appears to be the optimal selection for the training. This preference is because this sensor demonstrated superior efficacy in distinguishing between different class observations.

4.1. Wavelet transformation

As illustrated in the workflow of Figure 1, the signals from the source domain dataset were transformed into the time-frequency domain in the phase (i); this transformation was achieved by applying the discrete WT to level 3, using the "modwt" command within MATLAB®, with Daubechies 8 (db8) serving as the mother wavelet. The "modwt" function provides maximum overlap whilst the digit 8 indicates the number of vanishing moments. The choice of db8 as the mother wavelet was selected by the fact that the target domain data was originally released as wavelet coefficients based on the same function. This choice obviated the need for a preprocessing stage for this data (target data), as any potential noise had already been filtered out using WT preceding the dataset's release. It is imperative to note that the sampling frequencies of the source and target domain data differed, 25000 Hz and 2048 Hz, respectively. Consequently, the number of data points of the source domain data was adjusted to 2048 by employing the "decimate" function in MATLAB®, by applying a decimation factor of 12. Before the application of the WT, the last 35 data points were excluded. Plots in Figure 6 present the wavelet coefficients of an observation in the normal, unbalanced, and misaligned rotor conditions for both the source and target domain datasets.



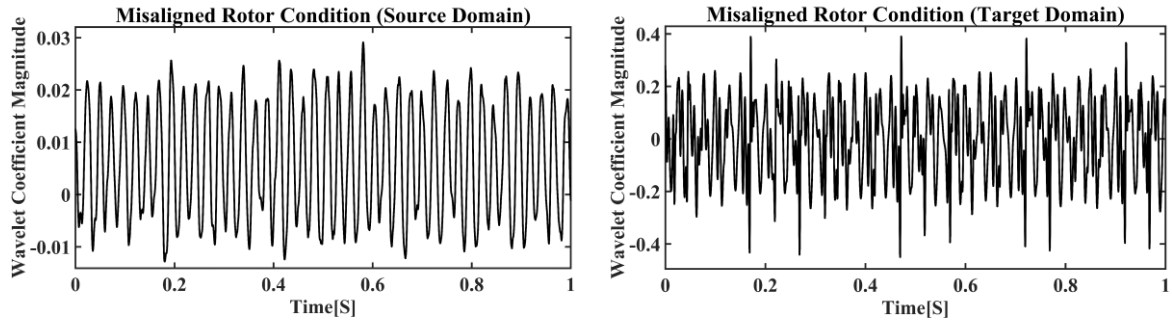


Figure 6. Wavelet coefficients for source and target domain data: (a, b) healthy, (c, d) unbalanced rotor, and (e, f) misaligned rotor

4.2. Feature extraction

This phase begins with dimensionality reduction via the MLP and concludes with processing through a TE. The MLP employs a structure of three hidden layers, each layer utilizing a 'LeakyReLU' activation function featuring a negative slope of 0.01. LeakyReLU was chosen to prevent the issue of dying neurons, which is common when employing ReLU as an activation function, thereby ensuring that all neurons remain active and contribute to the learning process. The design of these layers sequentially reduces the dimensionality by half, culminating in an output from the final layer that aligns with the dimension of the desired feature vector, which is 25. For instance, the outputs of the first, second, and third hidden layers are 1024, 512, and 256, respectively.

To perform the TE operation, a specific set of hyperparameters was established. At the outset, the number of heads in the multi-head attention mechanism was set to 5, providing the model with the ability to simultaneously address distinct parts of the input sequence. Additionally, a dropout rate of 0.1 was chosen to help prevent overfitting by randomly omitting a fraction of the parameter detectors on each iteration. The TE was designed with 10 sub-encoder layers, each contributing to the model's depth and its ability to learn complex patterns. Furthermore, the dimensionality of the feedforward network model within each TE layer was determined to be twice the size of the model's output feature dimension, resulting in a feedforward network size of 50. This configuration is crucial for the transformer's capacity to process and learn from the input data effectively. It is worth noting that the final feature extraction step, including both MLP and TE, was executed using Python.

To demonstrate the efficacy of the three signal processing phases, the outputs from each stage, i.e., WT, MLP, and TE from both the source and target domain data were fed into conventional machine learning classifiers using the Classification Learner App in MATLAB®. A five-fold cross-validation method was employed as the validation strategy, with 10% of the dataset reserved for testing. It is important to note that the feature vector resulting from WT had a dimension of 2048. To ensure a fair comparison of the three phases of feature extraction under similar conditions, the top 25 ranked features were selected using the minimum redundancy maximum relevance (MRMR) algorithm. Tables 3 and 4 present the results of this comparative analysis for the target and source domains, respectively. The 1st, 2nd, and 3rd models shown in these tables represent the classification models (from the Classification Learner App) that achieved the highest performances in the test phase. Additionally, SVM and KNN in the tables stand for support vector machines and k-nearest neighbors, respectively.

Table 3. Performance of feature extraction models for target domain data

Feature extraction	1st model (accuracy)	2nd model (accuracy)	3rd model (accuracy)
WT	Fine Gaussian SVM (0.92)	Fine KNN (0.92)	Subspace KNN (0.91)
WT+MLP	Quadratic Discriminant (0.93)	Quadratic SVM (0.92)	Cubic SVM (0.92)
WT+MLP+TE	Quadratic Discriminant (1.0)	Quadratic SVM (1.0)	Cubic SVM (1.0)

Table 4. Performance of feature extraction models for source domain data

Feature extraction	1st model (accuracy)	2nd model (accuracy)	3rd model (accuracy)
WT	Wide neural network (0.95)	Quadratic SVM (0.93)	Sine KNN (0.93)
WT+MLP	Quadratic SVM (0.97)	Cubic SVM (0.97)	Medium Gaussian SVM (0.95)
WT+MLP+TE	Quadratic SVM (0.99)	Cubic SVM (0.99)	Narrow neural network (0.99)

The data presented in the tables indicate that the accuracy of classification improved in both the source and target domains following the sequential implementation of the proposed property extraction algorithm.

In this work, a violin plot was employed as a visual tool to show the effectiveness of the applied feature engineering framework at its final stage after employing the TE step. A violin plot is a method of plotting numeric data that shows the distribution across distinct categories. It is similar to a box plot but provides a deeper level of understanding by also showing the probability density of the data at different values. Key statistical measures such as the median and interquartile range can be also marked, making it a comprehensive visual summary of the data [39]. Figure 7 exhibits the violin plots for three distinct classes, labeled 0, 1, and 2, and color-coded in green, blue, and red, respectively. The y-axis represents the range of feature values, while the x-axis identifies the individual features. The concentration of the data around the medians is illustrated by the slender middle sections of the violins.

When assessing violin plots, the width along the y-axis conveys the variability or dispersion of the data. A wide violin along the y-axis suggests a high variance within the feature values for that class, which could imply less consistency and potentially complicate the classification process. In contrast, a narrow distribution along the y-axis indicates that the data points are tightly clustered, reflecting greater consistency and potentially more reliable features for classification. Conversely, the width along the x-axis is a matter of scale and design, representing the frequency or density of the data points at various values. It does not inherently suggest anything 'good' or 'bad' but rather visualizes the shape of the data distribution. A broader shape at any point along the x-axis indicates more data points at that value, while a narrower shape indicates fewer data points. The violin plots in Figure 7, characterized by their narrow widths along the y-axis, suggest that all classes exhibit a consistent feature set with limited variability. This homogeneity within classes suggests a solid foundation for distinguishing between them using classification algorithms, especially those that can exploit the subtle differences across the multi-feature landscape.

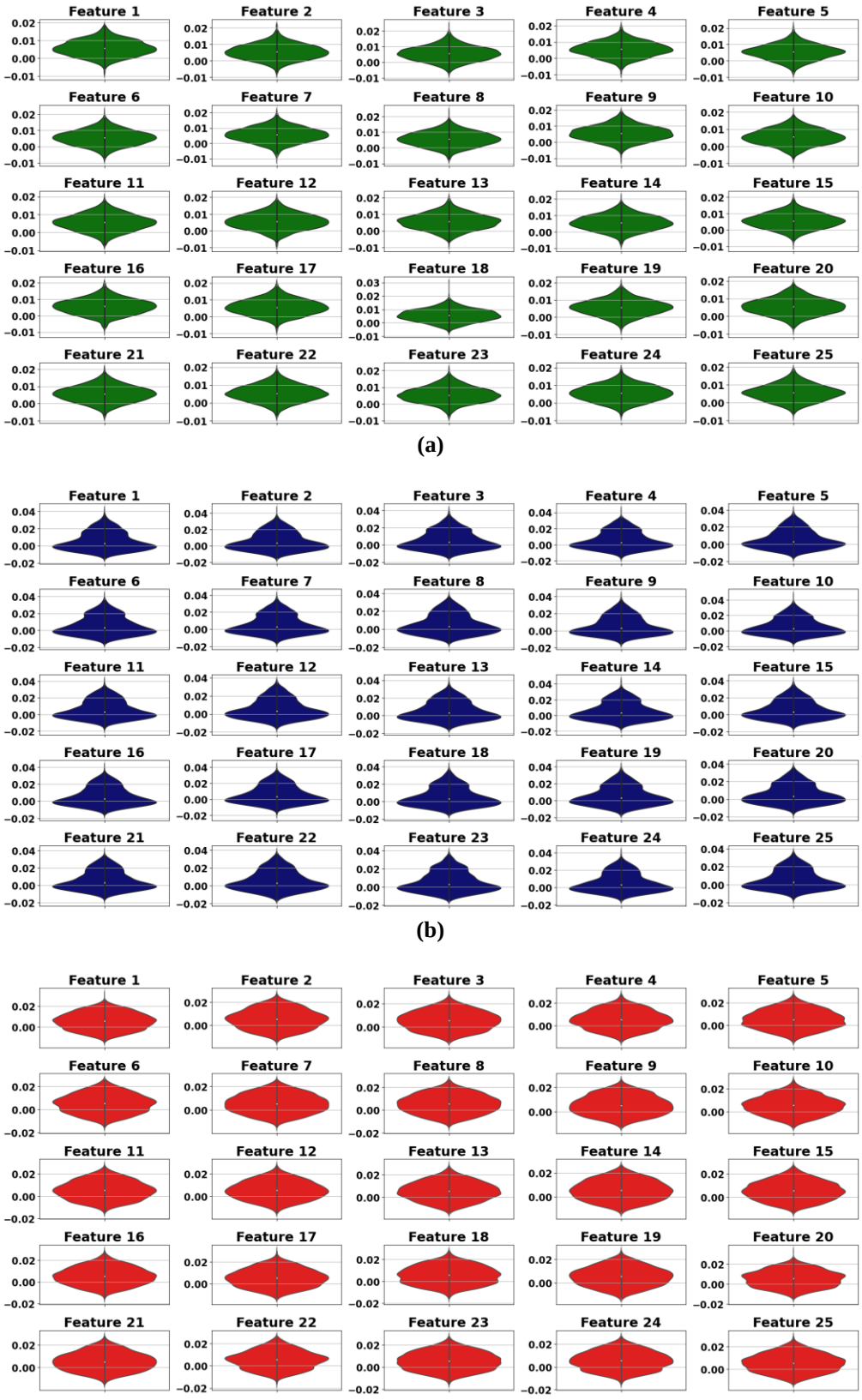


Figure 7. Violin plots for feature distribution: (a) Class 0, (b) Class 1, and (c) Class 2

4.3. Classification without domain adaptation

To elucidate the impact of the designed VCM framework on fault diagnosis in rotary machines, the attributes extracted from the preceding section were initially utilized, assuming the absence of the

domain adaptation phase. For a fair comparison between the outcomes presented in this section and those in the subsequent section, the target domain data was randomly divided into training, validation, and testing subsets, whilst the source domain data was divided into training and validation subsets. This division was achieved using the 'train_test_split' function from the scikit-learn library in Python, adhering to a split ratio of 80% for training, 5% for validation, and 15% for testing, based on the assumption of the target domain data. For the source domain data, the dataset was divided into training and validation subsets with a ratio of 0.85:0.15. These subsets were preserved for implementation in the following section as well. It should be noted that the 'stratify' option in the mentioned function was activated to ensure the split datasets are balanced. The results of these divisions can be found in Tables 1 and 2.

A classification task was executed using a sequential neural network crafted with layers configured for optimal data processing. It begins with a 512-unit dense layer featuring ReLU activation, coupled with L1 and L2 regularization (both set at 0.001), followed by batch normalization and a dropout rate of 0.4 to combat overfitting. A 256-unit layer with LeakyReLU activation (alpha=0.01), batch normalization, and the same dropout rate precedes further dense layers of 128 and 64 units, each with ReLU activation, identical regularization, batch normalization, and dropout rates of 0.4 and 0.3, respectively. The network culminates in a dense layer with the same number of units as the number of classes, here 3, followed by a Softmax-activated output layer. This architecture was optimized using SGD with a learning rate of 5×10^{-5} and a momentum of 0.9, focusing on minimizing categorical cross-entropy. The training spans 1000 epochs with batches of 10, incorporating a checkpoint mechanism that preserves the iteration exhibiting the highest performance. Figure 8 depicts the constructed classification model, where each dense layer and its subsequent layers are delineated as distinct blocks. Following the completion of the training phase, the trained network was fed with the test subset of the target domain.

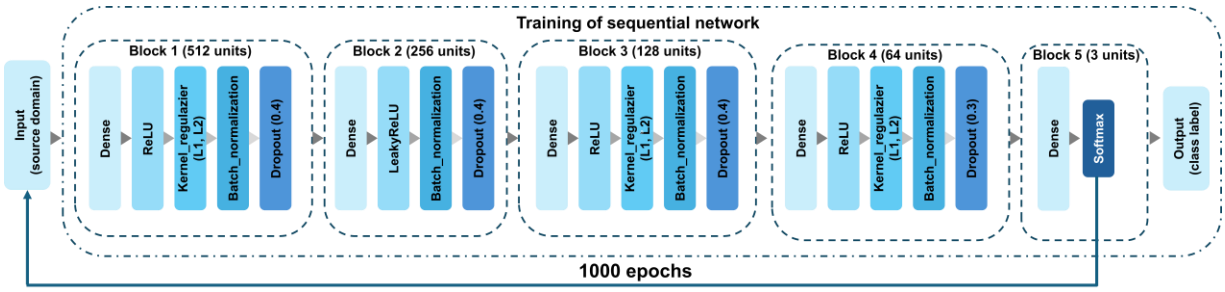


Figure 8. Sequential neural network architecture with integrated layers for classification

Figure 9 illustrates the trajectory of training and validation losses throughout 1000 epochs. It initially presents high losses (5.2 for training and 4.4 for validation), but demonstrating a consistent decrease, notably reaching a validation loss of approximately 0.24 by the 600th epoch. This pattern, also mirrored in the accuracy improvements of both training and validation phases, underscores the network's effective training with data from the source domain. Beyond the 600th epoch, the losses underwent minor fluctuations, indicating a stable condition in the learning process for both training and validation, with the validation loss reaching its minimum magnitude of 0.21 at epoch 685. At this point, the network is deemed optimally trained, as captured by the defined 'Callback' class, based on its performance with source domain data.

Further analysis of the network's generalizability is conducted post-validation for each epoch through assessments using the test subset from the target domain. This evaluation led to the identification of an alternative optimized model, specifically at the 23rd epoch. For these two networks, the optimum model based on its performance in the test phase and the best model referring to the validation phase were saved as Model 1 and Model 2, ordinary for further analysis.

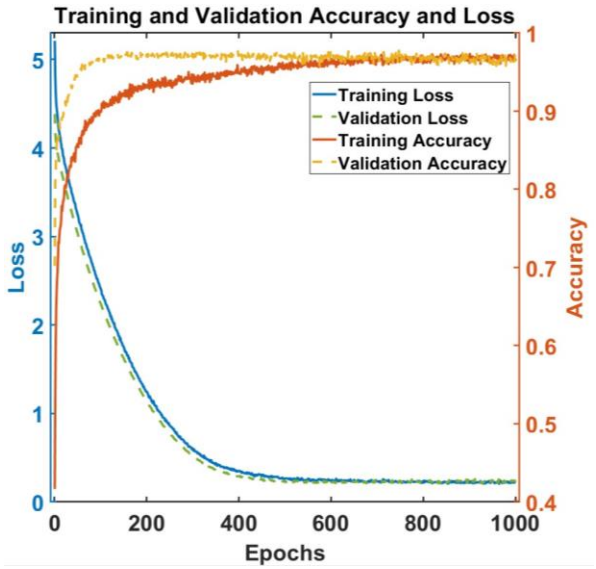
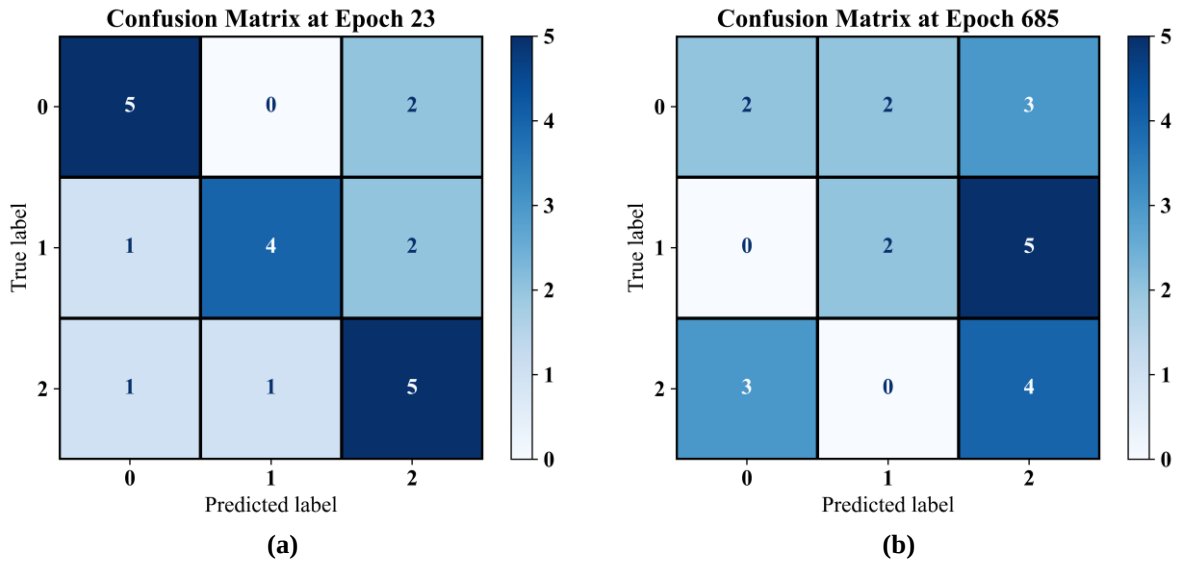


Figure 9. Training and validation loss progress on source domain data

The performances of Model 1 and Model 2 in the test phase are highlighted in Table 5, which elaborates on classification performance metrics, accuracy, precision, recall, and F1-score. Moreover, confusion matrices illustrated in Figures 10-a and b display the classification results achieved with these two models.

Table 5. Classification metrics without domain adaptation at the optimum checkpoints

Model name	Number of epochs	Accuracy	Class label	Precision	Recall	F1-score
Model 1	23	0.67	0	0.71	0.71	0.71
			1	0.8	0.57	0.67
			2	0.56	0.71	0.63
		Macro average		0.69	0.67	0.67
Model 2	685	0.38	0	0.4	0.29	0.33
			1	0.5	0.29	0.36
			2	0.33	0.57	0.42
		Macro average		0.41	0.38	0.37



(a) **(b)**

Figure 10. Confusion matrices of test phase classification on target data, without domain adaptation (a) finishing 23rd epoch (Model 1), (b) finishing 685th epoch (Model 2)

A thorough examination of the figures and table reveals a significant challenge to the classification models' ability to generalize within the target domain. Despite performing well in the validation and testing phases, the networks were unable to accurately classify fault scenarios in the target domain. However, based on Model 1 (Figure 10-a), the classification performance reached close to 67%, demonstrating the effectiveness of the designed hybrid feature extraction framework. This inefficiency suggests that domain adaptation could be a valuable follow-up strategy to improve fault diagnosis accuracy. It is important to note that the real-time nature of condition monitoring systems and the lack of fault history records make it impractical to rely on target domain data to determine the most effective network.

4.4. Classification with domain adaptation

To address the issue highlighted in the preceding section, the suboptimal accuracy of fault detection in the target domain, the trained networks detailed in section 4.3 underwent a domain adaptation step. This domain adaptation algorithm can be considered a model-based technique, as it fine-tunes the weights of various layers using a limited number of sample tests from the target domain. Throughout the fine-tuning stage, it is critical to adjust four key hyperparameters: the learning rate and momentum of the SGD optimizer, the maximum epochs, and the batch size. These were configured to 5×10^{-5}, 0.9, 600, and 5, respectively. The determination of these hyperparameters resulted from a trial-and-error approach.

4.4.1. Impact of layer freezing on domain adaptation

Preference of the trainable layer(s), whilst freezing the others, is a major challenge that arose during the fine-tuning process. A range of factors can impact this election, including the similarity of the data from the source and target domains, the quality of the extracted features, and the number of labeled observations available in the target domain. This choice can have a significant impact on the fault detection framework's performance. To identify the optimal number of trainable layers and their optimal positions within the network, a sensitivity analysis was conducted. The study examined four different configurations: (1) fine-tuning the initial dense layer with 512 units along with its subsequent batch normalization layer (referred to as block 1 in Figure 8), (2) modifying the middle three blocks, which include dense layers with 256, 128, and 64 units respectively, and their accompanying batch normalization layers (blocks 2 to 4), (3) adjusting the final dense layer that contains 3 units (block 5), and (4) enabling fine-tuning across all 5 blocks. It is important to note that both Model 1 and Model 2 were evaluated in this analysis. The confusion matrices for the test phase using fine-tuned versions of Model 1 are displayed in Figure 11-a-d; those for Model 2, across the four scenarios mentioned, are shown in Figure 12-a-d. Furthermore, Figure 13-a and b illustrate the effects of freezing different blocks within the fine-tuned network on the F1-score for Model 1 and Model 2, respectively.

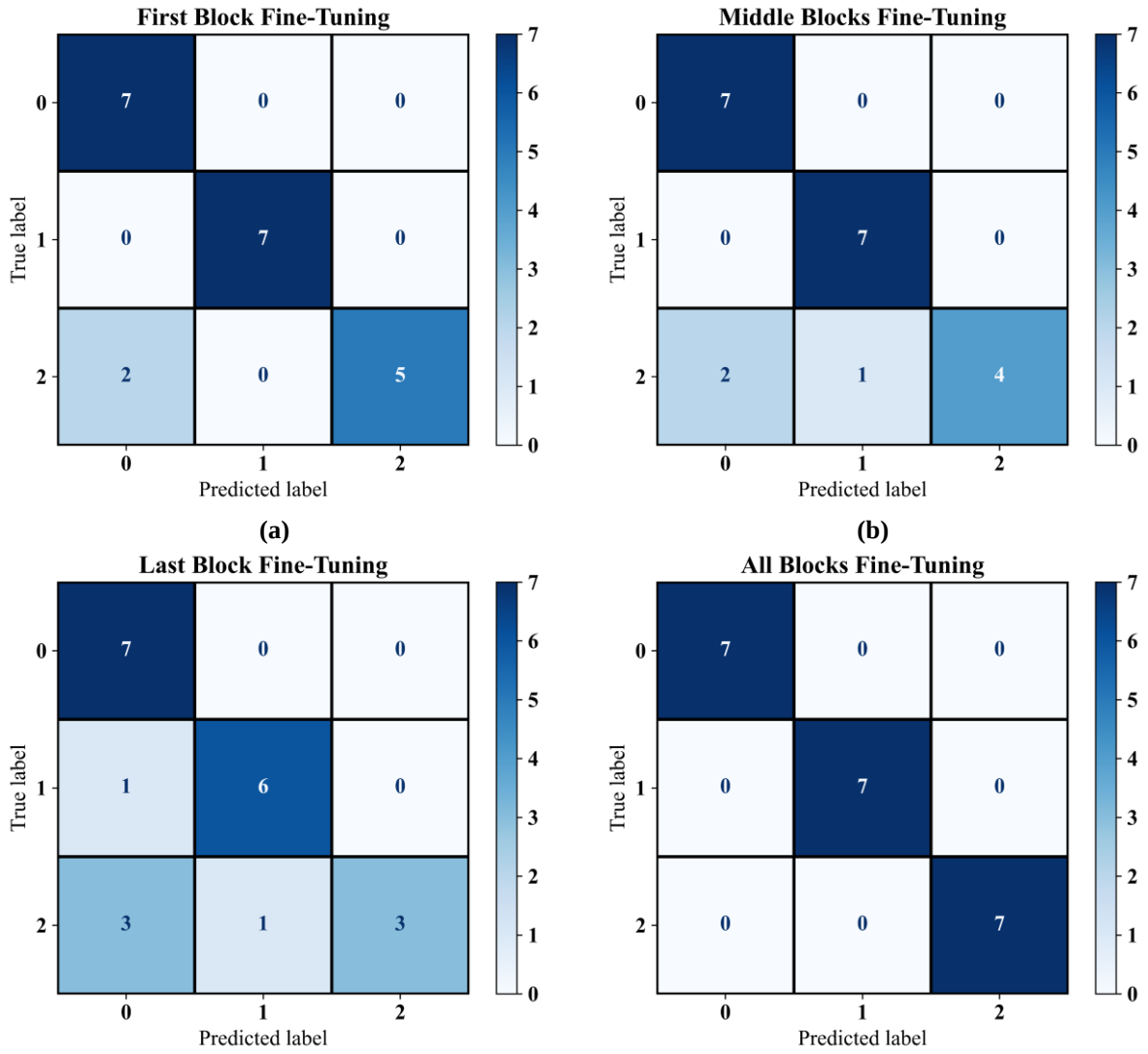
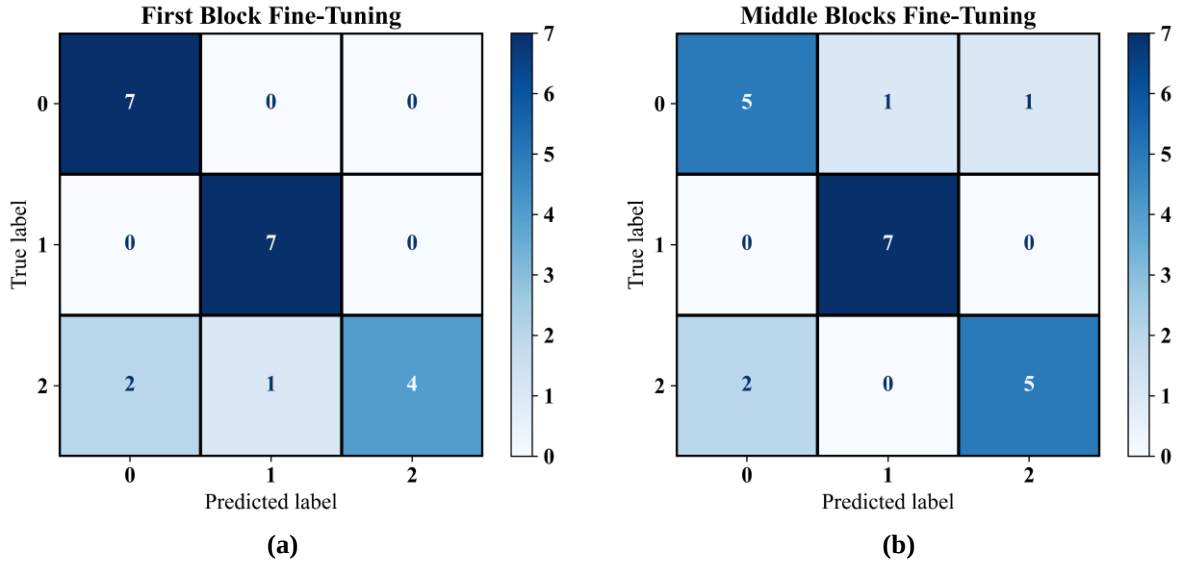
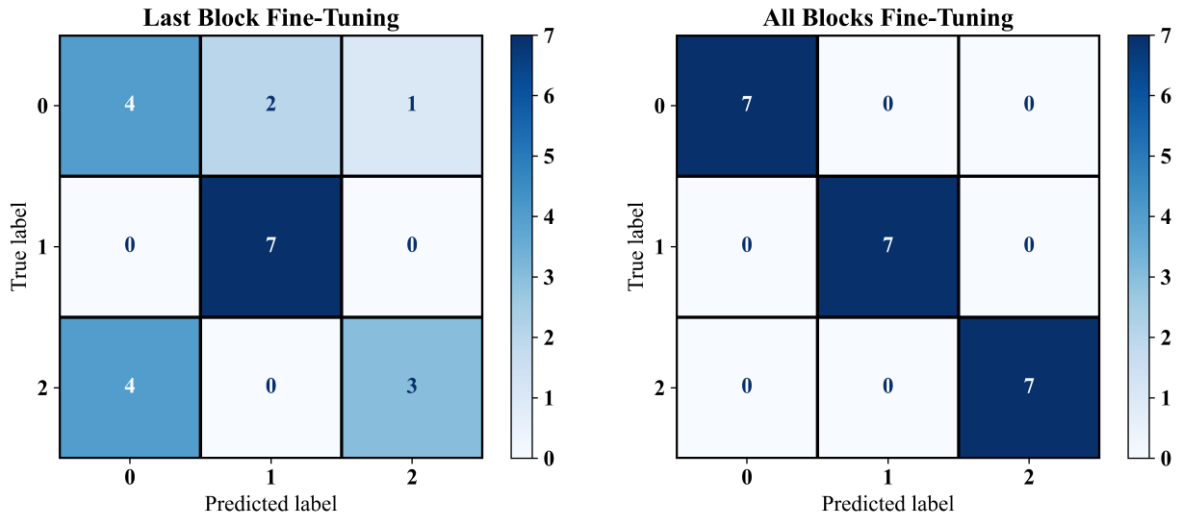


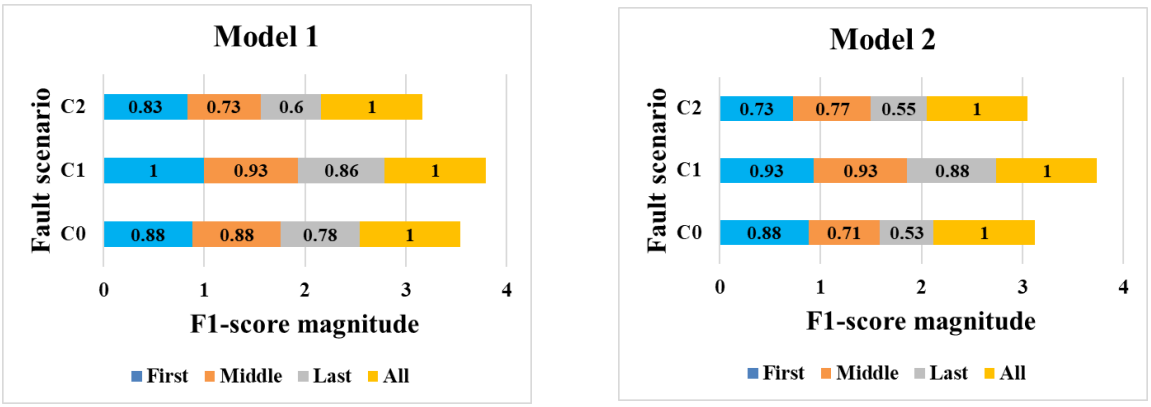
Figure 11. Confusion matrices for Model 1 with tuning of (a) first block, (b) middle blocks, (c) last block, and (d) all blocks





(c) **(d)**

Figure 12. Confusion matrices for Model 2 with tuning of (a) first block, (b) middle blocks, (c) last block, and (d) all blocks



(a) **(b)**

Figure 13. F1-score variations due to freezing blocks in fine-tuning for (a) Model 1 and (b) Model 2

The comparison of Model 1 and Model 2 through sensitivity analysis clearly shows differences in their performance under various fault conditions. Adjusting the first block in both models led to fairly reasonable results in recognizing the three classes, with Model 1 performing slightly more effective in terms of the F1-score. When the middle blocks were fine-tuned, a similar pattern emerged, but with both models performing slightly worse. Adjusting the last block proved to be the least effective, as both models exhibited inferior performance. The data suggests that for a classification model to perform well overall, it is necessary to adjust all blocks, regardless of whether the pre-trained network was selected during the validation or test stages. This is supported by the confusion matrices in Figures 11 and 12, which show that the highest level of accuracy, 100%, was achieved when all blocks were adjusted. In the real-world applications, selecting the best version of a pre-trained network based on its performance on the target domain data is not possible, as a result in this work Model 1 was not considered for further analysis and it has designed only as a comparative for Model 2.

4.4.2. Effect of data scarcity on domain adaptation

Another aspect of domain adaptation challenges is determining the number of observations required for effective model fine-tuning. Consequently, the impact of adjusting the amount of target domain data used in the fine-tuning process is explored in Model 2, wherein all its layers are configured to be

trainable. To achieve this, 8 different scenarios based on the size of the training subset of the target domain were evaluated as listed in Table 6. It is worth noting that the subset "SZ8" is the one employed in the preceding section where 80% of the target domain data was assigned as the training, and the remaining subsets were extracted randomly from this subset. All these subsets are balanced with respect to the three fault conditions, and the subsets for the validation and testing phases are the same as in the previous section. The other hyperparameters are tantamount to the ones mentioned in the previous section.

Table 6. Details of training, validation, and testing subset sizes in different scenarios

Model name	Subset name	Class-wise size in fine-tuning phase	Class-wise size in validation phase	Class-wise size in testing phase
Model 2	SZ1	4	2	7
	SZ2	9	2	7
	SZ3	14	2	7
	SZ4	18	2	7
	SZ5	22	2	7
	SZ6	27	2	7
	SZ7	31	2	7
	SZ8	36	2	7

Following the fine-tuning of Model 2 under the various training size scenarios discussed, Figure 14 presents the classification results for each class, illustrated through the obtained F1-scores.

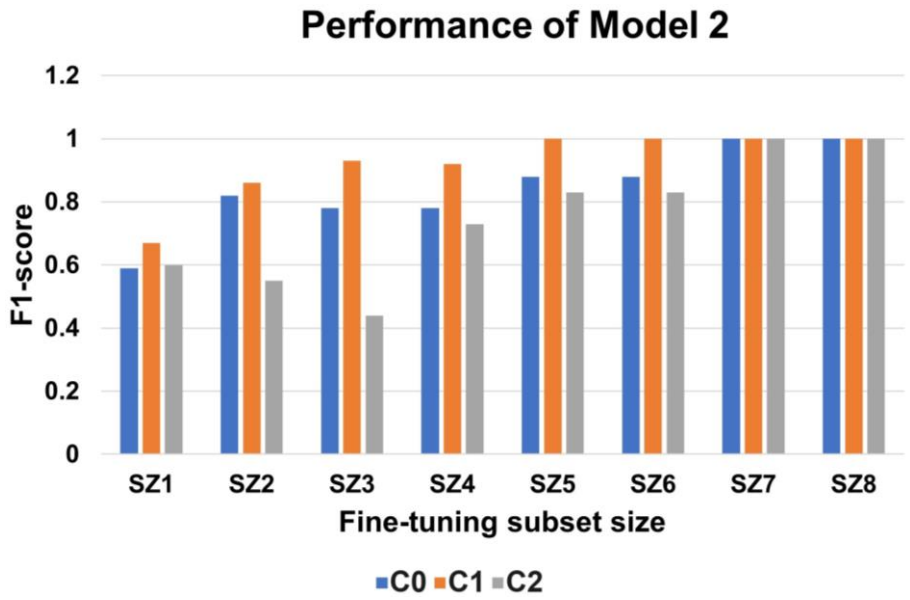


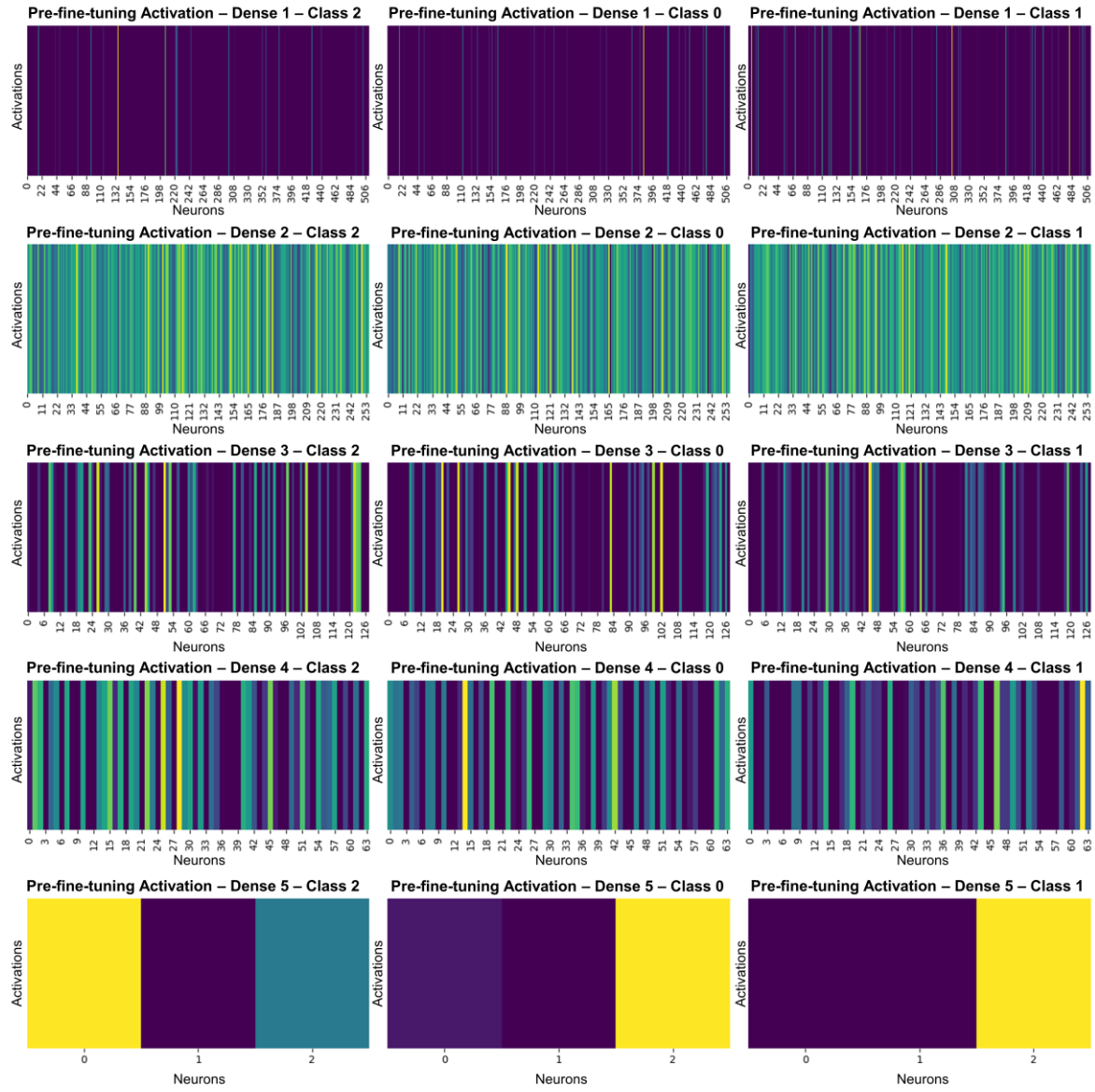
Figure 14. Impact of training size on the performance of Model 2 throughout fine-tuning

The bar chart illustrates that as the sample size for fine-tuning Model 2 increases, its ability to classify various defects with greater accuracy improves. While the F1-score for classes 0 and 1 gradually enhanced with larger training subsets, the performance of Cass 2 experienced fluctuations. Notably, under the conditions labeled 'SZ7' and 'SZ8', peak accuracy was achieved with correct classifications across all three classes. These findings indicate the network's effectiveness even with fewer data in the target domain than initially assumed, which suggests that 31 samples for each fault condition might be sufficient (SZ8 scenario). Moreover, integrating an optimization process might potentially create a more generalized framework in situations of significant data scarcity.

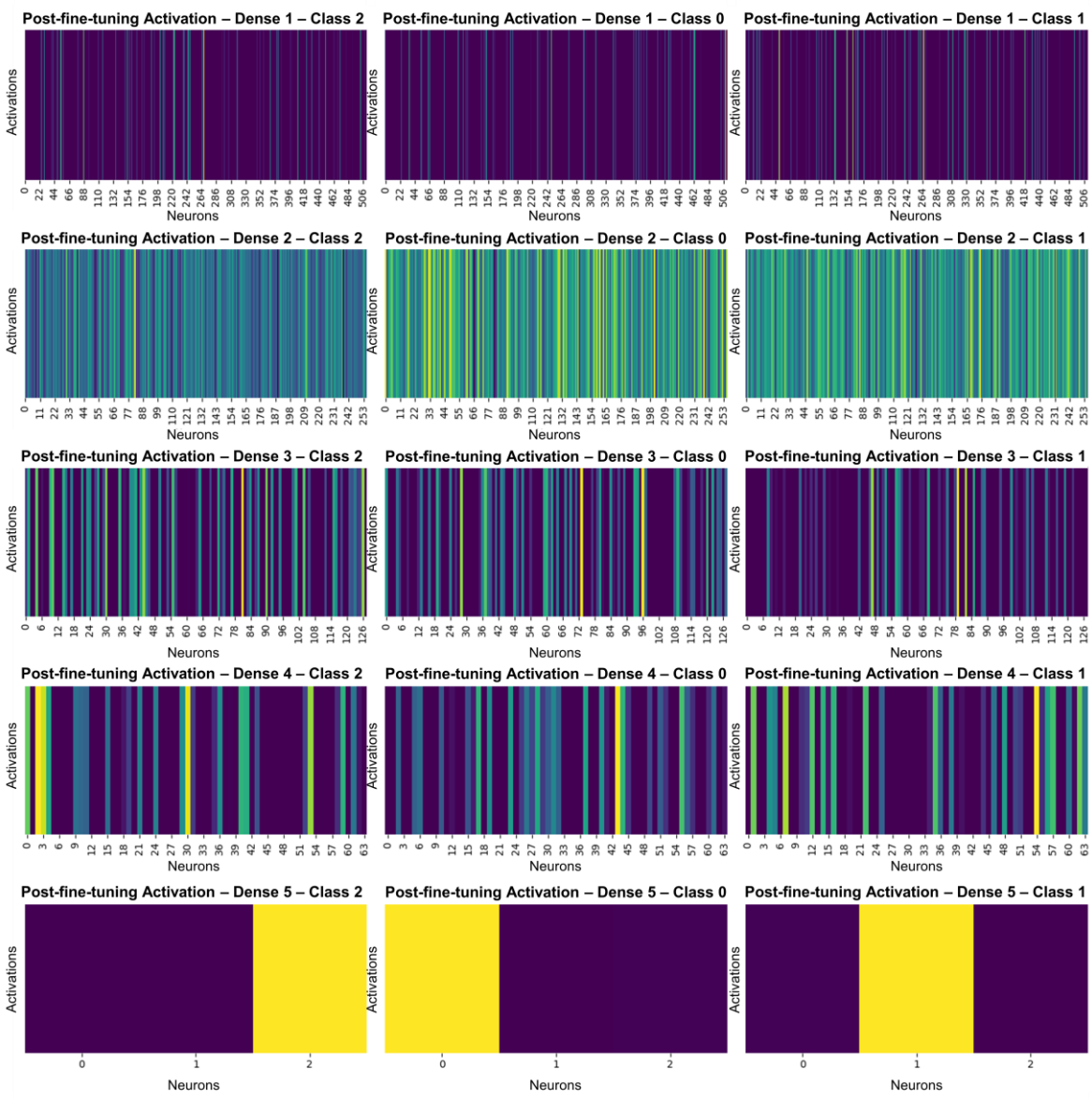
4.5. Visualization of the domain adaptation process

The domain adaptation process is elucidated through the comparative analysis of internal activation patterns within the designed sequential network, as observed in the dense layers of Model 2, both

without and with the fine-tuning phase. A visual evaluation was conducted using heat maps (Figure 15-a and b) to clarify the model's adaptability to the target domain. For this purpose, three observations corresponding to discrepant classes, Class 2, Class 0, and Class 1 were selected from the test subset of the target domain.



(a)



(b)

Figure 15. Comparative activation patterns in dense layers (a) pre-fine-tuning and (b) post-fine-tuning

Initially, uniform activation patterns were observed across all neurons within layers dense 1, 3, and 4, as depicted in the pre-fine-tuning visualizations (Figure 15-a). This uniformity, characterized by consistent activation levels, was indicative of the model's early stage, lacking specific adjustments for the target domain. Concurrently, the output layer (dense 5) exhibited dispersed activation, signaling the model's predictive uncertainty.

Following the fine-tuning, as shown in Figure 15-b, a meaningful change in the activation patterns is observed. An increase in the diversity of the activations signifies a refined adjustment to the target data. Notably, after fine-tuning, dense layer 1 maintains its high activation but now exhibits variation among neurons, indicating improved feature recognition. Dense layer 2, after tuning, displays distinct activation differences, suggesting a deeper analysis of the input. Dense layers 3 and 4 show considerable changes, with a range of activations that reflect a reorganization in how attributes are represented and combined. These modifications suggest that the network has been adjusted to extract features more effectively for the target domain. The output layer, dense 5, undergoes the most noticeable change. The differentiation in neuron activations becomes clear post-tuning, with the brightest color indicating the neuron with the highest confidence in its classification. This distinction

demonstrates an improvement in the model's decision-making ability, marking a significant step in domain adaptation. The network is now better prepared to provide reliable predictions for each input.

Considering sample 1 as a case study, there is a clear difference in the activation levels across neurons without and with fine-tuning. Initially, dense 1 shows widespread and intense activation, which then evolves to reveal distinguishable differences in activation levels after tuning. In an analogous manner, dense 2 moves from a more uniform activation pattern before tuning to a varied spectrum after tuning, indicating improved capability in feature distinction. Dense layers 3 and 4 transition from uniformity to diversity, highlighting the network's upgraded ability to differentiate subtle features relevant to the target task. The uniform activation seen in the output layer prior to tuning sharply contrasts with the prevalent neuron activation post-tuning, showing a heightened level of confidence in predictions. Figures 15-a and b demonstrate that the pre-fine-tuned network misclassified all the three fault scenarios, whereas the fine-tuned version correctly classified them. The changes seen in the activation patterns before and after tuning illustrate that the network adapted to align with the specific characteristics of the target domain, thereby enhancing its predictive performance.

5. Conclusion

This study proposed a fine-tuning framework to enhance the effectiveness and adaptability of fault diagnosis systems in rotating machinery when faced with both effects: data distribution shifts caused by OVs and limited data for the training procedure. This approach also tackled usual challenges in domain adaptation, such as low-quality features, the necessity of transforming vibration data into visual formats, the requirement of adjusting models with extensive data from the new domain, and the difficulties in interpreting how deep learning models make decisions. By integrating wavelet transformation, a multi-layer perceptron, and a TE for feature extraction, followed by sequential domain adaptation, the framework was designed to improve the classification accuracy of common rotor system conditions: normal, unbalanced, and misaligned. The approach was evaluated on two distinct datasets, one serving as the source domain for initial training, and the other as the target domain for model fine-tuning, demonstrating the adaptability of the model to data from different test setups.

The most notorious finding in this approach was that making all network layers trainable resulted in the best performance for damage identification. This was achieved through the conducted sensitivity analysis that underscored the model's capability to achieve 100% accuracy in fault diagnosis, emphasizing the critical role of comprehensive fine-tuning in overcoming the challenges posed by OVs. Furthermore, the visualization of the adaptation process through heat maps elucidated how the model adjusted to the target domain. This enhanced transparency and provided insights into the model's internal mechanisms, contributing to a deeper understanding of its decision-making process. Despite the limitations related to pre-processing requirements and the need for similarity in fault scenarios between datasets, the proposed framework demonstrated promise in addressing the critical issue of data scarcity. In conclusion, this study presents a significant advancement in VCM for rotating machinery. Future work will explore additional optimization strategies to enhance the framework's efficiency with fewer labeled data from the target domain, aiming to extend its applicability and effectiveness in industrial settings. Further research could also focus on more complex health scenarios, such as wear and tear or malfunctioning bearings, to broaden the framework's scope.

References

1. Espinoza-Sepulveda N, Sinha J. Two-step vibration-based machine learning model for the fault detection and diagnosis in rotating machine and its blind application. *Struct Health Monit*. Epub ahead of print 14 May 2024. DOI: 10.1177/14759217241249055.

2. Ozcan IH, Devecioglu OC, Ince T, et al. Enhanced bearing fault detection using multichannel, multilevel 1D CNN classifier. *Electrical Engineering*; 104. Epub ahead of print 2022. DOI: 10.1007/s00202-021-01309-2.

3. Ibrahim A, Anayi F, Packianather M, et al. New Hybrid Invasive Weed Optimization and Machine Learning Approach for Fault Detection. *Energies (Basel)*; 15. Epub ahead of print 2022. DOI: 10.3390/en15041488.

4. Rezazadeh N, de Oliveira M, Perfetto D, et al. Classification of Unbalanced and Bowed Rotors under Uncertainty Using Wavelet Time Scattering, LSTM, and SVM. *Applied Sciences (Switzerland)*; 13. Epub ahead of print 2023. DOI: 10.3390/app13126861.

5. Pacheco-Chérrez J, Fortoul-Díaz JA, Cortés-Santacruz F, et al. Bearing fault detection with vibration and acoustic signals: Comparison among different machine leaning classification methods. *Eng Fail Anal*; 139. Epub ahead of print 2022. DOI: 10.1016/j.engfailanal.2022.106515.

6. Guo J, He Q, Zhen D, et al. Multi-sensor data fusion for rotating machinery fault detection using improved cyclic spectral covariance matrix and motor current signal analysis. *Reliab Eng Syst Saf*; 230. Epub ahead of print 2023. DOI: 10.1016/j.ress.2022.108969.

7. Rezazadeh N, Felaco A, Fallahy S, et al. Application of Supervised and Unsupervised Machine Learning to the Classification of Damaged Rotor-Bearing Systems. *Macromol Symp*; 411. Epub ahead of print 2023. DOI: 10.1002/masy.202200219.

8. Almutairi KM, Sinha JK. A Comprehensive 3-Steps Methodology for Vibration-based Fault Detection, Diagnosis and Localization in Rotating Machines. *Journal of Dynamics, Monitoring and Diagnostics*. Epub ahead of print 25 March 2024. DOI: 10.37965/jdmd.2024.484.

9. Ni Q, Ji JC, Halkon B, et al. Physics-Informed Residual Network (PIResNet) for rolling element bearing fault diagnostics. *Mech Syst Signal Process* 2023; 200: 110544.

10. Feng K, Ni Q, Chen Y, et al. A cyclostationarity-based wear monitoring framework of spur gears in intelligent manufacturing systems. *Struct Health Monit* 2023; 22: 3092–3108.

11. Wang M, Li J, Xue Y. A New Weight-Based Dual Domain Adaptation Transfer Model for Bearing Fault Diagnosis Under Noisy and Cross-Domain Conditions. *IEEE Access* 2023; 11: 123766–123783.

12. Xiang L, Zhang X, Zhang Y, et al. A novel method for rotor fault diagnosis based on deep transfer learning with simulated samples. *Measurement (Lond)*; 207. Epub ahead of print 2023. DOI: 10.1016/j.measurement.2022.112350.

13. Zhu M, Liu X, Peng C, et al. Intelligent fault diagnosis for variable working conditions of rotor-bearing system based on vibration image and domain adaptation. *Meas Sci Technol*; 34. Epub ahead of print 2023. DOI: 10.1088/1361-6501/aceb83.

14. Cao H, Shao H, Liu B, et al. Clustering-Guided Novel Unsupervised Domain Adversarial Network for Partial Transfer Fault Diagnosis of Rotating Machinery. *IEEE Sens J*; 22. Epub ahead of print 2022. DOI: 10.1109/JSEN.2022.3182727.

15. Jiang X, Li Y, Wang Z, et al. OrbitDANN: A Mechanism-Informed Transfer Learning Method for Automatic Fault Diagnosis of Turbomachinery. *IEEE Sens J* 2024; 24: 2228–2241.

16. Jiang F, Lin W, Wu Z, et al. Fault diagnosis of gearbox driven by vibration response mechanism and enhanced unsupervised domain adaptation. *Advanced Engineering Informatics*; 61. Epub ahead of print 2024. DOI: 10.1016/j.aei.2024.102460.

17. Kumar P, Hati AS, Kumar P. A transfer learning-based deep convolutional neural network approach for induction machine multiple faults detection. *Int J Adapt Control Signal Process*; 37. Epub ahead of print 2023. DOI: 10.1002/acs.3643.

18. Zhang Q, He Q, Qin J, et al. Application of Fault Diagnosis Method Combining Finite Element Method and Transfer Learning for Insufficient Turbine Rotor Fault Samples. *Entropy*; 25. Epub ahead of print 2023. DOI: 10.3390/e25030414.

19. Asutkar S, Tallur S. Deep transfer learning strategy for efficient domain generalisation in machine fault diagnosis. *Sci Rep*; 13. Epub ahead of print 2023. DOI: 10.1038/s41598-023-33887-5.

20. Zhang R, Tao H, Wu L, et al. Transfer Learning with Neural Networks for Bearing Fault Diagnosis in Changing Working Conditions. *IEEE Access*; 5. Epub ahead of print 2017. DOI: 10.1109/ACCESS.2017.2720965.

21. Yang X, Bai M, Liu J, et al. Gas path fault diagnosis for gas turbine group based on deep transfer learning. *Measurement (Lond)*; 181. Epub ahead of print 2021. DOI: 10.1016/j.measurement.2021.109631.

22. Tang Z, Bo L, Liu X, et al. A semi-supervised transferable LSTM with feature evaluation for fault diagnosis of rotating machinery. *Applied Intelligence*; 52. Epub ahead of print 2022. DOI: 10.1007/s10489-021-02504-1.

23. Liu D, Cui L, Wang G, et al. Interpretable domain adaptation transformer: a transfer learning method for fault diagnosis of rotating machinery. *Struct Health Monit*. Epub ahead of print 23 May 2024. DOI: 10.1177/14759217241249656.

24. Tang J, Zheng G, Wei C, et al. Signal-Transformer: A Robust and Interpretable Method for Rotating Machinery Intelligent Fault Diagnosis under Variable Operating Conditions. *IEEE Trans Instrum Meas*; 71. Epub ahead of print 2022. DOI: 10.1109/TIM.2022.3169528.

25. Jin Y, Hou L, Chen Y. A Time Series Transformer based method for the rotating machinery fault diagnosis. *Neurocomputing*; 494. Epub ahead of print 2022. DOI: 10.1016/j.neucom.2022.04.111.

26. Weng C, Lu B, Gu Q, et al. A Novel Multisensor Fusion Transformer and Its Application Into Rotating Machinery Fault Diagnosis. *IEEE Trans Instrum Meas*; 72. Epub ahead of print 2023. DOI: 10.1109/TIM.2023.3244822.

27. Li R, Li S, Xu K, et al. Adversarial domain adaptation of asymmetric mapping with CORAL alignment for intelligent fault diagnosis. *Meas Sci Technol*; 33. Epub ahead of print 2022. DOI: 10.1088/1361-6501/ac3d47.

28. Yang Z, Cen J, Liu X, et al. Research on bearing fault diagnosis method based on transformer neural network. *Meas Sci Technol*; 33. Epub ahead of print 2022. DOI: 10.1088/1361-6501/ac66c4.

29. Hou Y, Wang J, Chen Z, et al. Diagnosisformer: An efficient rolling bearing fault diagnosis method based on improved Transformer. *Eng Appl Artif Intell*; 124. Epub ahead of print 2023. DOI: 10.1016/j.engappai.2023.106507.

30. Xu K, Li S, Miao X, et al. Interpretable Siamese dual attention enhancement transfer compound diagnostic model for unbalanced samples. *Struct Health Monit*. Epub ahead of print 7 March 2024. DOI: 10.1177/14759217241232985.

31. Rezazadeh N, Perfetto D, Polverino A, et al. Guided wave-driven machine learning for damage classification with limited dataset in aluminum panel. *Struct Health Monit*. Epub ahead of print 12 August 2024. DOI: 10.1177/14759217241268394.

32. Brito LC, Susto GA, Brito JN, et al. Mechanical faults in rotating machinery dataset (normal, unbalance, misalignment, looseness). Epub ahead of print 2022. DOI: 10.17632/zx8pfhdtnb.2.

33. Liu D, Xiao Z, Hu X, et al. Feature extraction of rotor fault based on EEMD and curve code. *Measurement (Lond)*; 135. Epub ahead of print 2019. DOI: 10.1016/j.measurement.2018.12.009.

34. Peng B, Wan S, Bi Y, et al. Automatic Feature Extraction and Construction Using Genetic Programming for Rotating Machinery Fault Diagnosis. *IEEE Trans Cybern*; 51. Epub ahead of print 2021. DOI: 10.1109/TCYB.2020.3032945.

35. Zhang W, Zhang T, Cui G, et al. Intelligent Machine Fault Diagnosis Using Convolutional Neural Networks and Transfer Learning. *IEEE Access*; 10. Epub ahead of print 2022. DOI: 10.1109/ACCESS.2022.3173444.

36. Wu Q, Zhang X, Zhao B. A novel adaptive kernel-guided multi-condition abnormal data detection method. *Measurement (Lond)*; 206. Epub ahead of print 2023. DOI: 10.1016/j.measurement.2022.112257.

37. Li X, Hu H, Zhang S, et al. A Fault Diagnosis Method for Rotating Machinery With Semi-Supervised Graph Convolutional Network and Images Converted From Vibration Signals. *IEEE Sens J*; 23. Epub ahead of print 2023. DOI: 10.1109/JSEN.2023.3267427.

38. Wang J, Ran R, Fang B. GNPENet: A Novel Convolutional Neural Network With Local Structure for Fault Diagnosis. *IEEE Trans Instrum Meas* 2024; 73: 1–16.

39. Bagheri S, Taghvaei M, Familiar A, et al. Statistical plots in oncologic imaging, a primer for neuroradiologists. *Neuroradiology Journal*. Epub ahead of print 2023. DOI: 10.1177/19714009231193158.