Applied Intelligence
https://doi.org/10.1007/s10489-024-05373-6

Semi-supervised diagnosis of wind-turbine gearbox misalignment and imbalance faults

Jose Alberto Maestro-Prieto[1] · José Miguel Ramírez-Sanz[1] · Andrés Bustillo[1] · Juan José Rodriguez-Díez[1]

Accepted: 7 March 2024

Abstract
Both wear-induced bearing failure and misalignment of the powertrain between the rotor and the electrical generator are common failure modes in wind-turbine motors. In this study, Semi-Supervised Learning (SSL) is applied to a fault detection and diagnosis solution. Firstly, a dataset is generated containing both normal operating patterns and seven different failure classes of the two aforementioned failure modes that vary in intensity. Several datasets are then generated, maintaining different numbers of labeled instances and unlabeling the others, in order to evaluate the number of labeled instances needed for the desired accuracy level. Subsequently, different types of SSL algorithms and combinations of algorithms are trained and then evaluated with the test data. The results showed that an SSL approach could improve the accuracy of trained classifiers when a small number of labeled instances were used together with many unlabeled instances to train a Co-Training algorithm or combinations of such algorithms. When a few labeled instances (fewer than 10% or 327 instances, in this case) were used together with unlabeled instances, the SSL algorithms outperformed the result obtained with the Supervised Learning (SL) techniques used as a benchmark. When the number of labeled instances was sufficient, the SL algorithm (using only labeled instances) performed better than the SSL algorithms (accuracy levels of 87.04% vs. 86.45%, when labeling 10% of instances). A competitive accuracy of 97.73% was achieved with the SL algorithm processing a subset of 40% of the labeled instances.

Keywords Wind turbine · Powertrain failures · Bearing failures · Semi-supervised learning · Fault detection and diagnosis

1 Introduction

A large number of wind-turbine installations generate a significant proportion of total electricity production. They are a notable source of renewable energy and their continued growth is likely, due to the renewable electricity generation targets that are now established. For example, the aim of the European Union is to generate 32% of its electricity from renewable sources by 2030 [1]. Wind turbines are complex electromechanical systems that transform the wind into electrical energy. For optimal energy production, wind turbines are invariably located in open countryside, at some distance from the point of consumption, especially in urban areas. Following their installation, they also have associated operating and maintenance costs. A good description of the components in a wind turbine drivetrain can be found in [2]. In addition, adverse environmental conditions increase the risks of multiple failures, which can be countered through the use of Failure Detection and Diagnosis (FDD) methods that maximize turbine operating times and minimize operating and maintenance costs [3, 4]. FDD should be focused on those types of failure with higher maintenance costs and downtimes. In windfarms, those failures are related to the power chain or gearbox, due to their mechanical complexity, highly demanding working conditions, and variety of possible failure modes. The most dangerous failures of these components are rotor blade misalignment and imbalance of the power chain caused by bearing fatigue and gear damage [5]. Their

José Miguel Ramírez-Sanz, Andrés Bustillo and Juan José Rodriguez-Díez contributed equally to this work.

✉ Andrés Bustillo
abustillo@ubu.es

Jose Alberto Maestro-Prieto
jamaestro@ubu.es

José Miguel Ramírez-Sanz
jmrsanz@ubu.es

Juan José Rodriguez-Díez
jjrodriguez@ubu.es

[1] Department of Computer Engineering, Universidad de Burgos, Avda. Cantabria, Burgos 09006, Burgos, Spain

Published online: 28 March 2024

repair involves winching a heavy sub-assembly in and out of the nacelle. Lengthy procurement times can also result in prolonged downtimes. Slight deflection of the external axis of the power train, due to the forces transmitted by the rotating blades, is sufficient to cause maladjustment, thereby misaligning the internal power train axis with the external one. Imbalance of the power chain is usually a consequence of damaged axis bearings or gearing mechanisms, usually due to insufficient lubrication, and shock to the mechanical chain (*e.g.*, turning windmills on and off with ice on the blades), *etc*.

As the nature of wind is variable and turbine dynamics are not linear, a wind turbine is an example of a machine that operates under variable loads and speed. For a recent analysis of fixed and floating wind turbine drivetrain loads, see [6]. Although they can have a direct-drive (gearless) design, around 75% of industrial wind turbines have a geared design [7]. Typically, a two or three stage gear set is used, in which planetary and other gearing systems are combined. A planetary gear is used on the low-speed shaft, because it can withstand high torque loads.

A wind turbine consists of fixed and rotatory components that may fail. A component failure can propagate and affect performance and perhaps lead to a general failure. Different types of failures can result in anything from poor performance to increased component failure rates. For instance, torque deviation failure in the generator/converter may be due to an internal fault in the converter electronics or a deviation in the torque estimation of the converter, which in turn may be due to either improper design or manufacturing defects. Torque deviations affect functional control and, therefore, power generation. Fluctuations in turbine dynamics and power generation can cause both material fatigue and power production problems for a wind turbine farm and even for the electricity grid.

Fluctuating weather means that several wind-turbine components are more prone to wear and fatigue than others: the drivetrain, gearbox, and generator are the most affected by maintenance downtime [8]. The rotor and blades, pitch, yaw and tower system and generator and control system are also prone to failure [4]. Wind-turbine gearboxes can often fail early on, due to varying wind loads, and may require replacement parts and maintenance within a few years. The main cause of many common industrial wind-turbine failure modes is related to bearing defects resulting from micropitting, scuffing, and cracking of the white etching area. In addition, bearings can skid during starts and stops, due to short-term dynamic loads [7].

In this paper, both the effects of a few labeled and unlabeled items (a Semi-Supervised Learning (SSL) problem) on the diagnosis of wind-turbine gearbox powertrain failures and the best techniques to predict each failure mode are studied. The fault diagnosis task in this case presented two main limitations under industrial conditions: datasets are usually strongly imbalanced (many instances of functional conditions and very few of fault situations) and working conditions are often not labeled. No expert has time to stop the system, in order to identify small degrees of failure, although the initial stages of damage and degradation provoke further wear that can lead on to catastrophic failure. But both restrictions, imbalance and unlabeling, are almost impossible to test together in the existing datasets, because dataset size under both restrictions is so small that no existing Machine Learning (ML) technique could ever extract useful information. The authors have therefore focused on solving the problem through two steps. The first step was to study the level of imbalance that could be reasonable before the ML techniques loose accuracy [9] In this research, the capabilities of SSL to resolve the limitations of labeled instances are studied. To do so, the methodology described in Fig. 1 is followed. Firstly, an experimental dataset is collected from different testbed working conditions and states. Secondly, the dataset is processed to extract new features using filtering and statistical methods, while datasets are generated with different proportions of labeled instances. Thirdly, both supervised and semi-supervised methods are tested on those datasets to evaluate their performance in terms of different quality indicators. Finally, the best methods are identified and compared with the existing bibliography.

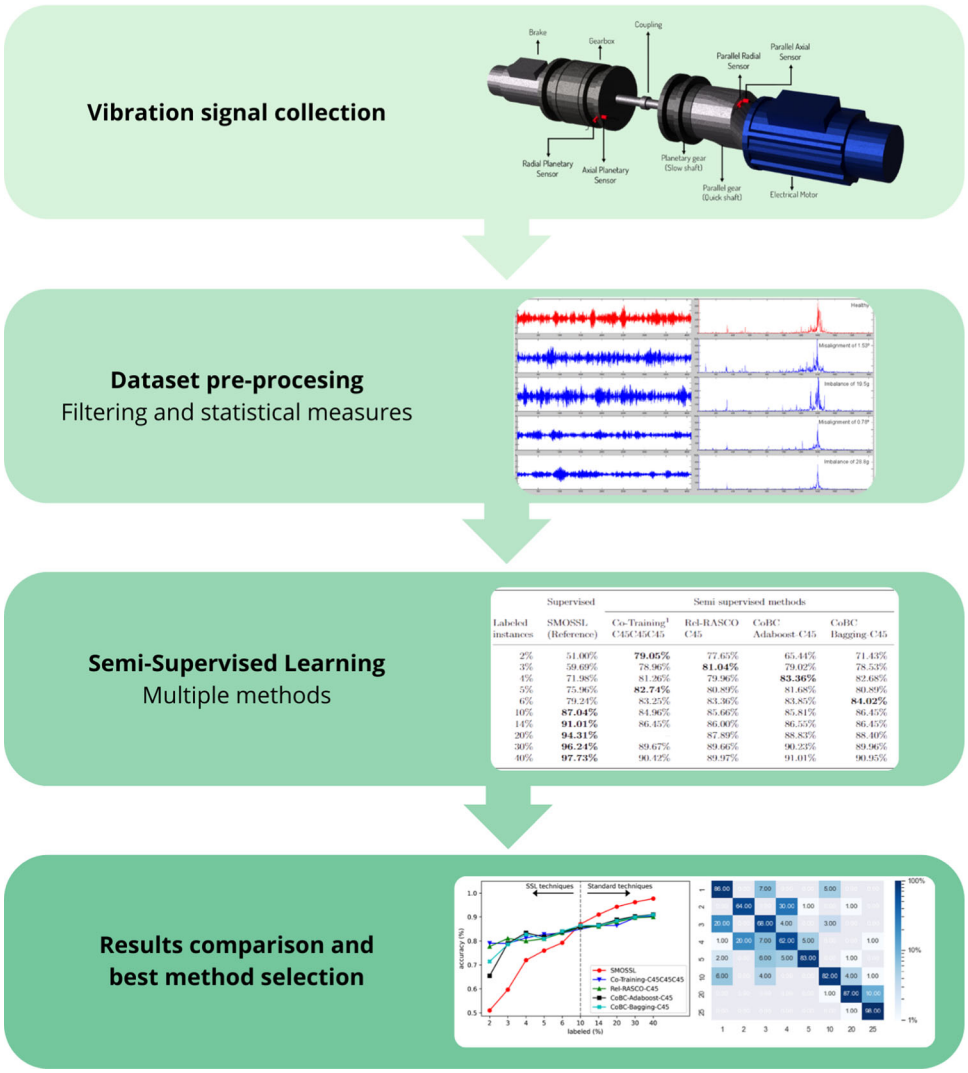
The rest of this paper is organized as follows. In Section 2, the basic background of SSL, the tool used to train and to test the learning algorithms, and SSL approaches related to FDD are briefly presented. In Section 4, the design of the SSL experiment is described. In Section 4.3, the experimental results are commented and compared to other approaches. In Section 5, the conclusions are presented.

2 Background

In this section, a brief description of SSL [10] is provided. Then, the most-recent literature [11–19] that uses SSL techniques and approaches for FDD is reviewed. The section continues with a review of recent literature that examines FDD in typical wind-turbine parts and components. Finally, the open-source machine learning software package, Knowledge Extraction based on Evolutionary Learning (KEEL) [20] is presented for use in this research.

2.1 Semi-Supervised Learning (SSL)

Machine Learning (ML) serves to pinpoint relations within datasets composed of instances that in turn contain features. When these datasets are recorded in an industrial environment, they usually represent the behaviour of industrial processes such as mechanical and chemical processes. Typ-



ical processes include engines, bearings, gearboxes, tanks, flows, temperatures, electrical voltages, *etc*., depending on the type of process. Industrial processes when monitored very often consist of operating patterns under normal conditions, and one or more operating patterns under fault conditions, and they usually share some characteristics, such as the presence of background noise. In the literature, these data are very often preprocessed using typical signal processing to perform time and frequency and time-frequency analyses.

Four main approaches to ML are considered [10, 21, 22]: Supervised Learning (SL), Unsupervised Learning (UL), SSL, and Reinforcement Learning (RL). RL is used for further improvement of a previously trained model while being used for its intended purpose. The main difference between supervised and unsupervised approaches is the presence of one or more special dataset features that contain one or more expected solution values or one or more labels that classify each instance. Generally, obtaining the expected output(s) or labeling the instances with their corresponding error type can be costly, and time-consuming, and will usually require expert assistance. SSL is an intermediate approach between SL and UL. For a recent review of SSL methods for FDD in industry, see [23]. And in [24] there is a review of recent ML proposals for wind turbine fault diagnosis, including some semi-supervised methods.

There are several approaches towards generating models of higher accuracy that usually employ a few labeled instances together with many unlabeled ones. For instance, Active Learning [25] processes the unlabeled instances to select those that contribute more than any others to the model that is being learned and its improvement, before an oracle (invariably an expert) is asked to label them. In that way, active learning attempts to minimize the number of true labeled instances, thereby reducing both labeling time and cost. SSL algorithms are programmed to improve the accuracy of models that are learned from datasets that consist of a limited number of labeled instances and a certain number of unlabeled instances. SSL processes reduce the number

of labeled instances to the minimum needed for high accuracy, *i.e.*, equal or close to the accuracy obtained using a fully labeled dataset. A good up-to-date review of semi-supervised methods, arranged in a taxonomy, can be found in [10].

SSL is usually considered to have two central approaches [10]: (*i*) transductive learning; and (*ii*) inductive learning. Both SSL approaches use unlabeled instances to improve the model that is being learned, but their main difference is the way in which the unlabeled instances are considered. Transductive learning aims to label only the unlabeled instances, so it does not usually create a proper model, as there are no new instances to be classified. The aim of inductive learning is to improve the model that is being learned, by using information from both the unlabeled and the labeled instances, for generalization purposes. While transductive learning mainly relies on graph-based methods, various inductive learning methods have been proposed, based on different assumptions. Inductive learning can be categorized into various SSL approaches, including unsupervised pre-processing, wrapper methods, and intrinsically semi-supervised methods. These categories can be further sub-divided into more specific approaches. Further details on the underlying assumptions of SSL, taxonomy, and the diverse methods used in each category, can be found in [10, 23].

2.2 Semi-Supervised Learning (SSL) for wind-turbine Failure Detection and Diagnosis (FDD)

Applying semi-supervised techniques and methods to wind-turbine FDD is a new research field reported in only a few very recent papers.

In [11], semi-supervised condition monitoring was proposed for bearing fault diagnosis in offshore wind turbines. A coupled residual CNN was proposed for an information fusion approach. Both vibration sensor data and acoustic signals were used as inputs. For testing purposes, an experimental platform was used to simulate different bearing failures affecting offshore wind turbines. Data were recorded under normal conditions and four failure modes, including typical inner ring, ball, and outer ring failures, and a compound failure. Two hundred instances consisting of signal segments were recorded for each condition and randomly divided into training and test sets. Ten percent of the training set was labeled. White Gaussian noise was added to all recorded segments to simulate the real operating environment. The proposed method achieved an Accuracy of 98.18%.

Accuracy is one of several metrics that can be used to measure the goodness of a classifier. Typically, those instances that are correctly classified as positive are called True Positives (TP). Those correctly classified as negative are called True Negatives (TN), and those incorrectly classified would be False Positives (FP) or False Negatives (FN). Accuracy is defined in (1).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Qian et al. approached blade cracking detection in a number of ways. In [14], they studied the detection and diagnosis of wind-turbine-blade faults using SSL with class-imbalanced datasets. Industrial datasets are often class-imbalanced: more instances are available for normal condition or some failure classes than for others. It all poses a problem, as the learning algorithm may be focused on increasing the detection or diagnostic accuracy in the over-represented failure classes and can ignore the under-represented classes, a scenario which corresponds to an overfitting problem [26]. As usual, when using class-imbalanced datasets, other measures were calculated instead of accuracy. The F1 score ranged from 0.785 to 0.964, depending on which of the five wind-turbine datasets obtained from real wind turbines were used.

The F1 score, an accuracy metric that attempts to account for differences in the number of instances in a class-imbalanced dataset, is defined below in Equation 2.

$$F1\ score = \frac{2TP}{2TP + FP + FN} \tag{2}$$

In [13], a hybrid network called PUHN, which combines a Deep Neural Network (DNN) and Positive Unlabeled (PU) learning, was proposed as a semi-supervised fault detection solution for blade cracking in wind turbines. PU learning [27] learns a binary classifier and requires only some positively labeled instances along with other unlabeled positive or negative instances. A non-negative risk PU network trained a binary classifier, a deep stacked AutoEncoder (AE) performed feature extraction, and a clustering layer was incorporated to improve class separability and class prior estimation of PU learning. Accuracy, Recall, and F1 score were used as metrics. The authors reported an Accuracy of 0.822 (Recall 0.907 and F1 score 0.832) using a dataset of instances from 24 wind turbines.

Recall, or sensitivity, is an accuracy metric that measures the capability of the model to detect positive instances. It is calculated as the number of positive correctly classified instances (TP) out of all instances classified as positive in the dataset. Recall is defined in (3).

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

And in [12], it was proposed to apply a PU learning method called Probability Ratio Least-Square Importance Fitting (PRL-SIF) under Labeling Bias (LB) to the problem of wind turbine blade early cracking fault detection. Feature

extraction and dimensionality reduction based on functional analysis was performed first of all and then the PRL-SIF method was applied. Accuracy was used together with the F1 score and the Area Under the Curve of the Receiver Operating Characteristic (AUC-ROC) metrics. It was reported that by using 20% of normal labeled instances, a 90% classification accuracy was achieved on a dataset consisting of instances from 23 wind turbines. The AUC-ROC curve is a graphical representation and a way of measuring the performance of an ML model. It measures the capability of a binary classifier to distinguish between classes.

There are several proposals on the detection of wind turbine blade icing. A recent review [28] of icing detection for wind turbine blades included a section on semi-supervised methods. In [18], Unified Imbalanced Semi-Supervised Contrastive Learning (UISSCL) was proposed to address the usual class imbalanced data problem and the semi-supervised approach simultaneously. The proposed method included a data augmentation step where Gaussian noise (RandomAddGaussian) was randomly added to generate new data sequences and the data sequence was multiplied by a random factor (RandomScale) to scale it. Semi-supervised contrastive learning was then applied, using both labeled and unlabeled data instances, and including a regularization term in the contrastive loss function, to compensate for the class imbalance problem. Evaluation metrics included Accuracy, Precision, Recall, G-mean, and F1 score. Two different datasets were used for testing. Both datasets were class-imbalanced: 88.92% and 88.68% were, respectively, normal condition instances, 6.09% and 5.58% were, respectively, faulty condition instances, and the remainder (4.99% and 5.74%, respectively) were unlabeled instances. The accuracy and the G-mean metrics, and the F1 scores reported for both datasets were between 0.9839 and 0.9990.

Precision is an accuracy metric that measures the capability of the model to detect negative instances. It is calculated as the number of negative correctly classified instances (TN) over all instances classified as negative in the dataset. Precision is defined in (4).

$$Precision = \frac{TN}{TN + FP} \tag{4}$$

G-mean is an attempt to combine the Recall and Precision metrics into one measure. The G-mean measure is defined in (5).

$$G\text{-}mean = \sqrt{Recall \times Precision} \tag{5}$$

In [19], it was proposed to use the XGBoost algorithm [29] as a base algorithm for semi-supervised Tri-Training [30], to solve the early detection of blade icing in wind turbines. In addition, instead of using over-sampling or under-sampling techniques to deal with the class imbalance problem, a cost-sensitive approach was chosen and a focal loss function replaced the usual loss function in the XGBoost classifiers to solve the common class imbalance problem and inaccurate labels in the datasets. The focal loss function used different weights to compute the loss depending on the difficulty of classifying the instances. Three new features were constructed using existing features, those features that were correlated with others using Pearson correlation coefficients were removed, and then the data were normalized. Data from 3 wind turbines were used. The training set consisted of 70% of the instances and the remaining 30% comprised the test set. Different percentages of labeled instances from 10% to 90% were used for the experiments and the metrics Accuracy, Precision, Recall, F1 score and Matthews Correlation Coefficient (MCC) were computed. Using 60% of labeled instances, an Accuracy of 0.974 was reported (0.92 of MCC, 0.93 of F1 score, 0.94 of Precision, and 0.933 of Recall).

MCC is defined to produce a high score only if all the four basic measures (TP, TN, FP, FN) are close to their best value. MCC is defined in (6).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{6}$$

In [15], Chen et al. proposed an enhanced version of Random Forest (RF) using Graph-based Semi-Supervised Learning (GSSL) and a Decision Tree (DT) whenever there were insufficient labeled instances for fault diagnosis in a wind-turbine gearbox. GSSL and DT methods were used for increasing the labeled instances when training the RF model. If both methods predicted an unlabeled instance, then it was added to the labeled dataset together with the predicted label (pseudo-labeling). The SpectraQuest's Wind-Turbine Drivetrain Diagnostic Simulator (WTDS) [31] was used for testing. Six different operating conditions were used, combining motor frequencies of 6, 10, and 14 Hz, and load voltages of 5, and 8. Normal and four abnormal (worn surface, missing tooth, chipped tooth and cracked tooth) gear working operations were used. In all, 16 signal segment instances were collected (each totaling 96 instances) for the different types of operations and condition, each with 112 features, grouped using the 5 gear working conditions. Experiments were performed using 180 labeled instances and 300 unlabeled instances. Sixty of the 300 unlabeled instances were randomly chosen for pseudo-labeling, but only 50 of the 60 unlabeled instances were pseudo-labeled, so the final labeled set for training the RF consisted of 230 instances. This approach yielded an RF accuracy of 99.38%.

In [16], Wang et al. approached the diagnosis of wind-turbine bearing faults by using Multiscale Permutation Entropy (MPE) to extract the feature information of bearing

vibration signals and to construct a high-dimensional feature representation; Mahalanobis distance along with SSL and manifold learning were used to reduce the dimensionality of the representation; and an SVM classifier was trained with the help of a Beetle Antenna Search (BAS) algorithm to search for the best SVM parameters. The experiment was conducted using the SpectraQuest's WTDS platform, with the motor speed set to 0.8 Hz, a constant load of 10 volts was applied, and ER-12K bearings were used. There were four working conditions: the normal working condition, and inner raceway, outer raceway, and bearing failure modes. Eighty sets of vibration acceleration signals were collected, each containing 3000 sampling points. In total, 320 instances were collected. For each working condition, 20 instances were randomly chosen for label removal. The proposed method achieved 100% recognition accuracy.

In [17], Tang et al. introduced a fault-detection method for the wind-turbine pitch system. They proposed the use of a semi-supervised Optimal margin Distribution Machine (ssODM), optimized using a Dynamic State Transition Algorithm (DSTA) that selects the best hyperparameters for improving the fault detection model. Data were acquired from a domestic wind farm of 1.5 MW double-fed wind turbines. The dataset was sampled at intervals of 1 second and the samples were isolated 30 minutes before the onset of the faults up until 30 minutes after the faults. Three kinds of wind-turbine pitch faults were considered: (*1*) emergency stop fault of the pitch system; (*2*) a CANBUS communication fault between pitch PLC and pitchmaster (the servo driver) of blade 1; and (*3*) a low temperature invoked blade-2 axle-box fault affecting the pitch. The raw data were pre-processed and feature selection was performed by applying an RF to rank the importance of the features. After eliminating features that showed strong correlations, the features were reduced from the initial 58 to 24. The proposal was tested using instances labeled at 5% and 10%, however, each type of failure was tested independently as a binary problem, detecting whether the instance was faulty or normal. The proposal obtained the lowest false positive and false negative rates compared to the other 3 possible alternatives.

2.3 Overview of recent SSL proposals for wind turbine-related technologies

Various SSL approaches from ML, Deep Learning (DL), and RL have been successfully applied to typical wind-turbine tasks, such as: fault detection, fault identification, condition-based monitoring, and related tasks for bearings, drivetrains, gearboxes, rotating machinery, and others. A brief overview of some recent proposals is presented below.

Recent surveys on Deep Semi-Supervised Learning (DSSL) can be found in [32, 33]. In [32], DSSL proposals were classified into five main groups, namely: generative, consistency regularization, graph-based, pseudo-labeling, and hybrid methods. In [33], proposals focusing on consistency regularization methods using DSSL approaches with image datasets were reviewed. It should however be noted that learning from images requires and permits some techniques that are neither common nor even possible when the datasets have other characteristics, such as individual instances consisting of a few instantaneous measurements of an industrial plant or a physical system, rather than graphical information. Therefore, Data Augmentation (DA) techniques occupy a large part of [33], as it is recognized that they often produce great improvement in the capabilities of the model that is learnt. It is worth mentioning that the image datasets usually used for benchmarking contain a considerable number of images. For example, the CIFAR-10 and CIFAR-100 datasets [34] contain 60K images, MNIST [35] contains 70K images, the SVHN dataset [36] contains more than 99K images, the STL-10 dataset [37] contains 113K images (100K of which are unlabeled), NORB [38] contains nearly 350K images, and the ImageNet dataset [39] contains over 14 million images.

In some recent references, there are proposals to perform semi-supervised fault detection, diagnosis and condition monitoring in bearings in different ways. For a recent review on SSL methods for anomaly detection, see [40]. A DSSL approach [41] and a Safe Semi-Supervised Support Vector Machine (S4VM) [42] were used for incipient fault detection in bearings. A recent review of condition-based maintenance and recent references using SSL approaches and proposals for fault detection in bearings, gearboxes, induction motors, generators, and other typical industrial machinery can also be consulted in [43].

For fault diagnosis in bearings, a cross-domain approach and Transfer Learning (TL) were proposed in [44, 45]; Generative Adversarial Network (GAN) approaches in [46, 47]; Convolutional Neural Networks (CNNs) in [48, 49]; a Deep Adversarial Semi-Supervised (DASS) method was proposed in [50]; a Deep Reinforcement Learning (DRL) approach in [51]; Graph-based learning methods can be found in [52, 53]; Laplacian Regularization (LapR) in [54]; a consistency regularization-based approach in [55]; and Local Fisher Discriminant Analysis (LDA) in [56].

In [16], it was proposed to use a swarm intelligence approach for bearing fault diagnosis in wind turbines. This reference is described in more detail below. In [57], it was proposed to use metric learning techniques for bearing condition monitoring.

Rotating machinery has also received some attention and some semi-supervised proposals can be found in the recent literature. In [58], it was proposed to use a consistency-based approach for fault diagnosis in rotating machinery. More specifically, fault diagnosis in gearboxes was proposed by modifying an AE in [59] and using a graph-based approach in [60]. Fault diagnosis in planetary gearboxes was pro-

posed using Semi-Supervised Multiple Association Layers Networks (SSMALN) in [61] and using TL in [62]. Fault diagnosis in drivetrains was proposed using a GAN in [63].

A very recent review of condition monitoring approaches using ML techniques can be found in [64]. Unfortunately, only one of the reviewed references is classified as SSL.

It may be of interest to note that there is no consensus over the reference number or percentage for labeled instances in the SSL datasets. Some authors consider 30 as the maximum number of labeled instances per class for a sample to be considered *small* [65]. 10 has also been proposed as the maximum number to consider for an *extremely limited* sample [66].

2.4 KEEL

Knowledge Extraction based on Evolutionary Learning (KEEL) is an open-source software tool programmed in Java, which includes evolutionary algorithms and soft computing techniques for standard Data-Mining problems such as regression, classification, and association rules, as well as data pre-processing techniques [20].

KEEL consists of three main modules: a module for SL, a module for SSL, and a module for learning with imbalanced datasets. The SSL module includes several methods such as Self-training [67], Co-Training [68], RASCO [69], Rel-RASCO [70], CoForest [71], ADE-CoForest [72], Democratic Co-learning [73], CLCC [74], CoBC [75], APSSC [76], SETRED [77], SNNRCE [78], various regression types (LDA, logistic, and others), several types of neural networks and versions of basic supervised methods (C45, Nearest Neighbor (NN), Naive Bayes (NB), and Support Vector Machine (SVM)) for use with semi-supervised datasets.

Experimental work can be performed using cross validation in KEEL. The 10-fold cross validation consists of dividing the dataset into 10 equal randomly generated folds, 9 of the 10 are used for training and the other for testing. In that case, 10 different experiments are run, each time using a different fold for testing and the results are averaged. Using 5×2-fold cross validation, the dataset can be divided into two equal parts. In that case, half of the instances are used for training and the other half for testing. Five different (random) splits are generated for the experiment and the average of the results is calculated.

3 Methodology

A series of experiments were conducted using various datasets that varied in the number of labeled and unlabeled instances, to evaluate the effectiveness of SSL algorithms within KEEL for diagnosing real-world problems with multiple classes. The output of a reference supervised algorithm was computed using only the labeled instances in each dataset, to evaluate whether the SSL approach improved upon the results of the SL approach.

3.1 Semi-supervised methods

KEEL implements several SSL algorithms. Some of these algorithms require a base classifier to be specified, usually a simple one. KEEL uses the well-known base classifiers C45, NB, NN and SMO[1]. For example, Self-training, Co-Training, and some variants such as RASCO, Rel-RASCO, and CoBC, among others, require one or more base classifiers to be specified.

A total of 209 algorithms and algorithm combinations were trained and tested for each different dataset. Most of the cases corresponded to the Co-Training algorithm, as 3 different or equal base classifiers were selected for the Co-Training implementation in KEEL, so all possible combinations could be tested. The best results were obtained using a C45 decision tree as the base classifier in combination with Co-Training, Rel-RASCO and CoBC, which cover all the different kinds of Co-Training implemented in KEEL. These algorithms are described in greater detail below.

Co-Training [68] is a sort of bootstrapping, with which a large number of unlabeled instances are used in an attempt to improve the performance of a learning algorithm when a small set of labeled instances is available. One assumption of Co-Training is that the dataset has to be split into two views (instance features are split into two subsets) and both views must be sufficient for learning. Two learning algorithms are separately trained on each view and, the predictions of each algorithm on unlabeled instances are used to extend the training set of the other algorithm.

Formally, an instance space is divided into two different sets of features (views) $X = X_1 \times X_2$ and each view is supposed to be sufficient for correct classification. Let D be a distribution over X and C_1, and let C_2 be classes defined over X_1 and X_2, respectively. A target function $f = (f_1, f_2) \in C_1 \times C_2$ is termed compatible with D, if D assigns 0 probability to the set of instances (x_1, x_2) that $f_1(x_1) \neq f_2(x_2)$. The distribution, D, can be represented as a weighted bipartite graph, $G_D(X_1, X_2)$, where an edge, (x_1, x_2), exists, if and only if the instance, (x_1, x_2), has non zero probability under D. That same probability is attached to the edge weight. The authors assume a fully compatibility scenario where the two views of an instance are equally labeled by the two functions, f_1 and f_2, (7) and where Ω is

[1] Instead of using the usual acronym SVM to refer to Support Vector Machines, it is replaced in KEEL with the one corresponding to the SVM training algorithm. As KEEL uses the Sequential Minimum Optimization algorithm, SMO acronym is used instead of the standard SVM acronym.

the set of defined labels.

$$\forall x_1 \in X_1, \forall x_2 \in X_2, f_1(x_1) = f_2(x_2) = l, (l \in \Omega), \quad (7)$$

In the same way, a graph, G_S, is defined for the unlabeled set of instances, S, as a bipartite graph with an edge (x_1, x_2) for each $(x_1, x_2) \in S$. Basically, two instances connected to the same component (the same values in x_1) in S must be equally labeled.

The hypothesis of Blum et al. in [68] is as follows: given an assumption of conditional independence in the distribution, D, if the target class can be learned from random classification noise in the PAC [79] learning model, then Co-Training can improve any initial weakly learned model, to achieve any arbitrarily high accuracy using unlabeled instances. However, minimizing the empirical error on the instances labeled by the weak predictor may not minimize the true error.

Also, the assumption that instances (x_1, x_2) showing $f_1(x_1) \neq f_2(x_2)$ will never appear can be relaxed. It will be sufficient if (8) is fulfilled.

$$p[f_1(x_1) = 1, f_2(x_2) = 1] \times [f_1(x_1) = 0, f_2(x_2) = 0] > p[f_1(x_1) = 1, f_2(x_2) = 0] \times [f_1(x_1) = 0, f_2(x_2) = 1] + \delta. \quad (8)$$

The Co-Training example described in [68] used the same classifier (an NB classifier) for both views. Certain parameters had to be set: namely, the number, p, of positive labeled instances selected and the number, n, of negative instances selected in each iteration (the example was a binary classification), the number $k = 30$ of iterations, and the number $u = 75$ of unlabeled instances selected from the unlabeled set U. The authors proposed the use of a subset $U' \subset U$ for pseudo-labeling, as it had shown better performance in empirical tests. Each classifier selects the most confident n and p instances classified from U', which together with their predicted label are added to the labeled set of the other algorithm. At each iteration, the subset of U' is completed after randomly extracting $2n + 2p$ instances from the set U.

The algorithm implemented in KEEL was a variation of the above algorithm. A parameter, p, that establishes the number of instances to be selected can be activated in KEEL, as well as the parameters k and u. Perhaps the greatest variation is that 3 classifiers can be selected. The third classifier is used for computing the results of each iteration.

The idea underlying Co-training is that, by using two views of the same dataset, if both unlabeled and labeled instances are also used together, then the number of labeled instances needed to obtain an accurate classifier can be reduced.

However, maintaining the conditional independence between both views, as is required when just one dataset is available, can be difficult in practice. Some modifications to the original Co-Training algorithm have been proposed, in order to overcome that problem. RASCO (Random Subspace Method for Co-training) [69] is a multiview Co-Training method that obtains different feature splits with the random subspace method. If there are n features in the instances of the dataset, random subspaces of dimension m ($m < n$) are selected. Then, the set of labeled instances, L, is projected into the subspace of m dimensions (L_{sub}). This process is repeated K times, so K different views of the feature space are created (L_{sub_k} with $1 \leq k \leq K$) and K different classifiers are trained, each with a different view of the dataset.

RASCO can improve the results on datasets with many features and achieve lower errors than the traditional Co-Training algorithm [69]. However, when there are many irrelevant features, RASCO may not choose the best features to produce a good classifier. Rel-RASCO (Relevant Random Subspace Method for Co-training) [70] scores features to overcome this problem, using mutual information between features and classes (labels). Feature selection is performed based on probabilities that depend on relevance scores, to maintain randomness.

CoBC [75] is a special kind of Co-Training. A two-view approach is used in CoBC to improve results by combining the tree-structured (ensemble) approach and Co-Training. It can be especially useful for improving classification when a large number of classes and low volumes of labeled data are involved. CoBC was designed for classification problems with four characteristics: (*i*) sufficient redundant views may be defined; (*ii*) there is a large number of classes (Ω); (*iii*) there are a few labeled instances; and (*iv*) there are large number of unlabeled instances. CoBC entails combining a tree structure and Co-Training in two ways. On one hand, a *co-train-of-trees* is defined as an ensemble of binary Radial Base Function (RBF) networks trained on each view. Then unlabeled instances are labeled and the most confident one(s) is(are) added to the training dataset of the other decision tree classifier(s). On the other hand, a *co-training tree* is defined as a K-class problem decomposed into a (K-1)-class binary class problem using a tree structure. Then, a binary RBF network is trained on each view to solve the binary problems. Instead of just traversing the decision tree and pseudo-labeling with the predicted class, it uses a method based on Dempster-Shafer evidence theory [80, 81] for obtaining a combination based on probabilities of the intermediate results of the internal nodes within the decision trees. In this method, not only do classifiers on the path from the root to the leaf node of the decision tree contribute to the estimation of the class probability, but all classifiers that are not on the path can also contribute.

KEEL CoBC uses an ensemble approach to SSL and one of two different types of ensembles may be selected: a boosting method (Adaboost) and a bootstrap method (Bagging). Both ensembles need a base learning algorithm, and the pre-

viously mentioned base classifiers C45, NN, and SMO can be selected.

3.2 Supervised method

Supervised SVMs obtained the best results for the wind-turbine dataset in [5], which was therefore the supervised algorithm used for comparison. KEEL includes an implementation of the SMO (Sequential Minimum Optimization) algorithm [82] for training an SVM that can be used with semi-supervised datasets. The KEEL SMOSSL algorithm basically filters out unlabeled instances and only uses the labeled ones to train an SVM with the SMO algorithm, thereby permitting the same semi-supervised dataset to be used as an input for the supervised algorithm. The balance between classes is a major requirement in the datasets, to assure proper comparison with supervised techniques [9].

4 Experiment description and results

In this section, the test-bed platform and the different semi-supervised datasets are briefly described, as well as the algorithms with the best results.

4.1 Platform and data description

The experimental dataset was obtained using a test-bed to simulate the behaviour of wind turbines under faulty operational conditions. The test-bed (Fig. 2) consisted of two parts: the first was an electrical drive, a parallel gearbox (fast shaft), and a planetary gearbox (slow shaft), which simulated the powertrain of a real wind turbine. These components were connected to the second part, composed of a two-stage planetary gearbox and a brake with which wind conditions were simulated. Data were collected and recorded using seven sensors. Four of them were ICP accelerometers that measured axial and radial vibration signals from the two gearboxes. Another three sensors measured the current, the torque of the electrical drive, and the rotation speed. The test-bed design simulated misalignments on both parts of the test-bed for generating and measuring in degrees one of the two failure modes: misalignment of the powertrain. The other failure mode of interest was linked to imbalance failures of the fast shaft, due to damaged bearings and raceways. The damage was measured in grams. Data for four different damages were generated using the test-bed. Simulations of both failure modes reflected progressive degradation of the wind-turbine powertrain using two levels for misalignment and four levels for imbalance. Although the presence of both failures at the same time was considered uncommon, it was also simulated and included in the dataset. Wind conditions were simulated by means of random profiles of speed (between 1000 and 1800 rpm) and load (from 0 to 100%) which covered the range of real working conditions. Each working condition was run 100 times. Each run lasted for 72 seconds and the sampling frequency was 25 600 Hz. Each run of 72 seconds generated an instance. The test-bed, data collection process, and signal treatment are explained in more detail in [8] and [83], respectively.

Failure types and number of instances of each failure mode in the initial dataset are shown in Table 1. The dataset was composed of 6551 instances obtained under different working conditions. Each instance was composed of 544 features or variables, containing information on the operational state (torque, speed, electric input and output currents), information on vibrations from the accelerometers, such as energy distribution statistics (average, root mean square, skewness, kurtosis, and interquartile range), energy in standard frequency bands, and in the harmonics of the rotating speed.

4.2 Experiment design

In all, 10 different semi-supervised datasets were generated from the initial dataset. The original dataset was randomly divided into two halves: one for training and the other for testing. Training instances were randomly selected for label removal. A different percentages of labeled instances, ranging between 2% and 40%, were retained in each dataset. The unlabeling process was performed in a stratified man-

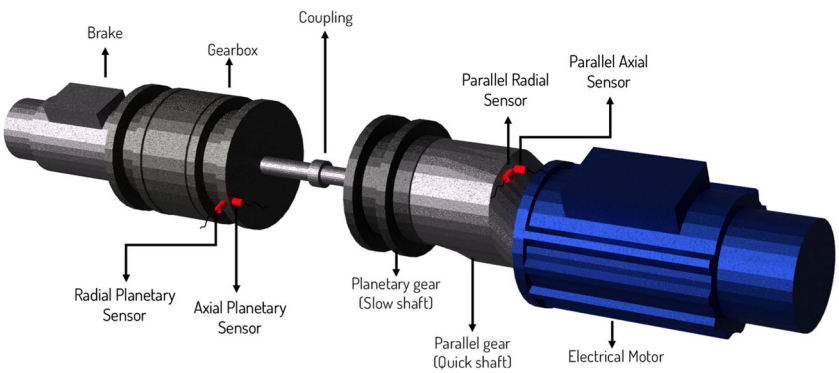


Fig. 2 Scheme of the wind-turbine test-bed

Table 1 Dataset description

Id.	Misalignment	Bearings imbalance	Instances	Percentage
1	No misalignment	No imbalance	887	13.54%
2	No misalignment	Imbalance of 5.79 g	847	12.94%
3	No misalignment	Imbalance of 9.13 g	856	13.07%
4	No misalignment	Imbalance of 19.5 g	838	12.79%
5	No misalignment	Imbalance of 28.8 g	864	13.19%
10	Misalignment of 0.78º	No imbalance	872	13.31%
20	Misalignment of 1.53º	No imbalance	835	12.75%
25	Misalignment of 1.53º	Imbalance of 28.8 g	552	8.43%

The first column contains a numerical identifier for the faults, which are used to label the instances. The second and third columns are textual descriptions of the failure types, which include the misalignment angle (in degrees) and the imbalance weight (in grams). The fourth column and fifth columns, respectively, show the number of instances of each failure type and the percentage over all instances in the dataset. The combination no misalignment-no imbalance in the first row represents the no failure condition

ner, taking into account the failure modes and the number of instances of each failure mode in the dataset to maintain representativeness. Labeled instances for each dataset were chosen independently, so that different semi-supervised datasets could not share common labeled instances. 5×2-fold cross validation experiments were performed. Thus, for each percentage of labeled instances, five different training and test files were created, and the results were averaged.

In Table 2, the number of labeled and unlabeled instances are summarized that constitute the 10 different semi-supervised datasets generated by randomly selecting the corresponding percentage of labeled instances and unlabeling the rest.

4.3 Results and discussion

Table 3 shows the best results of the semi-supervised algorithms for each dataset and the results obtained with the supervised benchmark algorithm. The first column contains the percentage of labeled dataset instances, ranging from 2% to 40%. The second column contains the result obtained with the SMOSSL algorithm, used as the supervised benchmark. The next four columns contain the results obtained with the SSL algorithms: Co-Training, Rel-RASCO, CoBC using Adaboost, and Bagging ensembles. All SSL algorithms yielded the best result using the C45 algorithm as the base classifier. The bold numbers are the best result for each dataset. As can be seen, for datasets with fewer labeled instances, no more than 10%, the highest accuracy was obtained using some SSL algorithm. For datasets labeled 10% or more, the supervised SMO (SVM) algorithm, which was used as a benchmark for comparison, yielded the highest accuracy.

Table 2 Description of the different datasets generated for SSL

Percentage of labeled instances	Number of labeled instances	Number of unlabeled instances
2%	65	3210
3%	98	3177
4%	131	3141
5%	164	3111
6%	196	3079
10%	327	2948
14%	458	2817
20%	655	2620
30%	982	2293
40%	1310	1965

The first column represents the percentage of labeled instances; the second column, the number of labeled instances; and the third shows the number of unlabeled instances for each training dataset. Ten different datasets from 2% to 40% of labeled instances were generated

The data in Table 3 are plotted in Fig. 3. As can be seen in the figure, the SMO algorithm performed poorly on datasets with fewer labeled instances, although it outperformed all SSL algorithms, at 10% and above of labeled instances. The SMO algorithm using the 40% labeled dataset produced comparable results to those shown in [9] using the fully labeled dataset. It can also be seen from the figure that no SSL algorithm was systematically better than the others, although the differences were not very important and sometimes even negligible.

It is worth noting the differences between the results of the SSL algorithms, which were greater when using the 2% labeled dataset. The Co-Training algorithm and the CoBC-Bagging algorithm, respectively, yielded the best and the worst results for that dataset. However, despite still using a small number of labeled instances, the disparity of the results tended to diminish as from the above-mentioned percentage, and similar results were obtained for all the SSL algorithms. Focusing on the SSL algorithms, despite the combination of base classifiers that were tested, the best results were obtained with the Co-Training algorithm or some vari-

Table 3 Accuracy of the different algorithms in percentages

Labeled instances	Supervised SMOSSL (Reference)	Semi-supervised methods Co-Training[1] C45C45C45	Rel-RASCO C45	CoBC Adaboost-C45	CoBC Bagging-C45
2%	51.00%	**79.05%**	77.65%	65.44%	71.43%
3%	59.69%	78.96%	**81.04%**	79.02%	78.53%
4%	71.98%	81.26%	79.96%	**83.36%**	82.68%
5%	75.96%	**82.74%**	80.89%	81.68%	80.89%
6%	79.24%	83.25%	83.36%	83.85%	**84.02%**
10%	**87.04%**	84.96%	85.66%	85.81%	86.45%
14%	**91.01%**	86.45%	86.00%	86.55%	86.45%
20%	**94.31%**	–	87.89%	88.83%	88.40%
30%	**96.24%**	89.67%	89.66%	90.23%	89.96%
40%	**97.73%**	90.42%	89.97%	91.01%	90.95%

Each row represents the results, using a different percentage of labeled instances, ranging from 2% to 40%
[1] Co-Training produced no results with files containing 20% of labeled instances.The algorithm had been running for a week when it was decided to stop the process, even though the first of the five folds had yet to be completed

ant (Rel-RASCO, CoBC) combined with the C45 decision tree. It was also interesting that the best results for the SSL algorithms included approaches that used (boosting and bagging) ensembles.

The SSL methods achieved 91% accuracy with 40% labeled instances in the training set, and the SL method achieved 97.7% accuracy using only the labeled instances in the semi-supervised training set.

The supervised SMO algorithm obtained better results than the SSL algorithms above 10% of labeled instances (327) in the dataset. In this specific problem, if they represent the different normal and abnormal working conditions, having more than 327 labeled instances, the best results can be obtained using an SMO (SVM) algorithm, regardless of the number of available unlabeled instances.

Figure 4 shows the different confusion matrices for the five algorithms and the 10% labeled dataset. A logarithmic scale was used to color the confusion matrices for better visibility.

As can be seen, in general, the misaligned cases (identified as 10 and 20) and the mixed case (identified as 25) are generally correctly identified with high accuracy. The imbalanced bearing cases are more complex and the test instances are less accurately identified. This problem is important, because bearings with less imbalance (identified as 1, 2, or 3) may not require preventive maintenance. However, bearings with more imbalance (identified as 5 and 6) may require repair work quickly, so accurate diagnosis is important. Figures 4b, 4c and 4d appear to show a more accurate diagnosis of the instances identified as 4 and 5 than Fig. 4a, particularly for case 5.

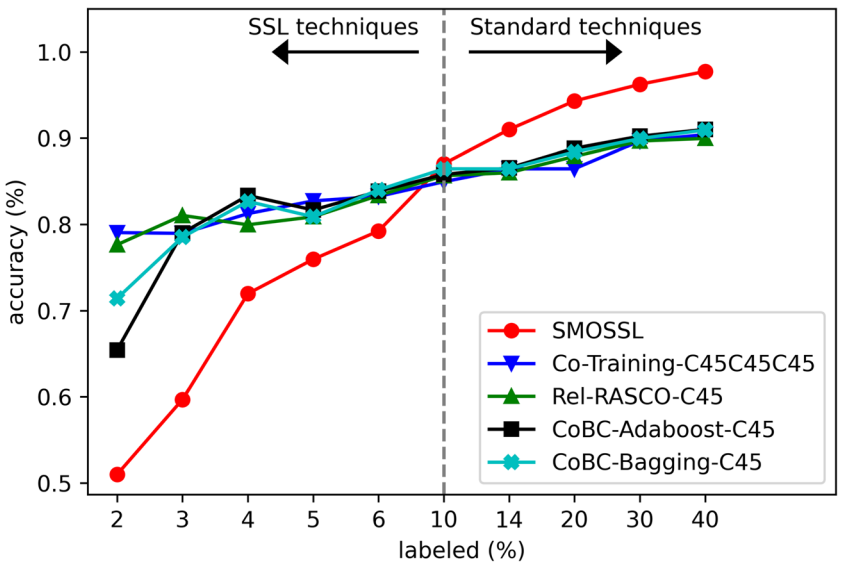
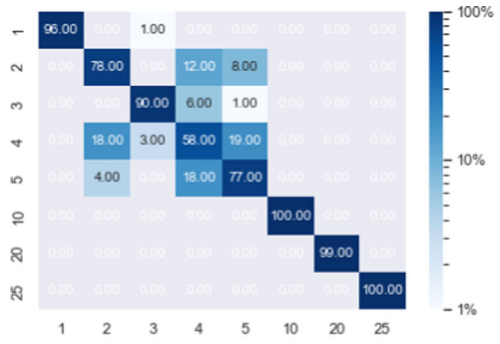
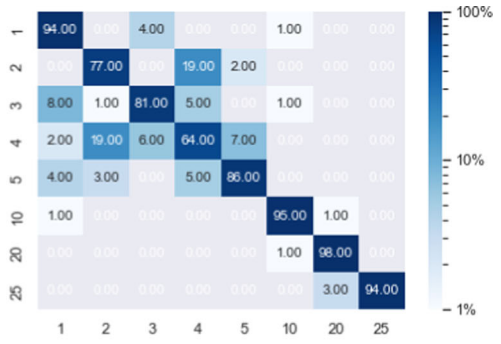


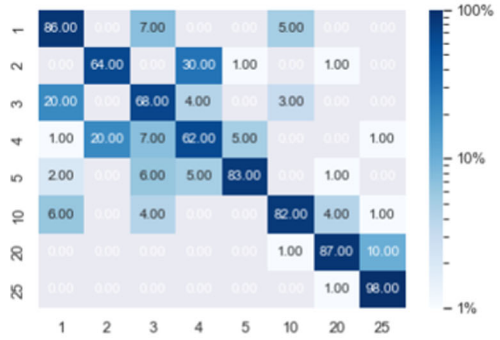
Fig. 3 Accuracy in percentages of each algorithm in Table 3, for each dataset with a different percentage of labeled instances



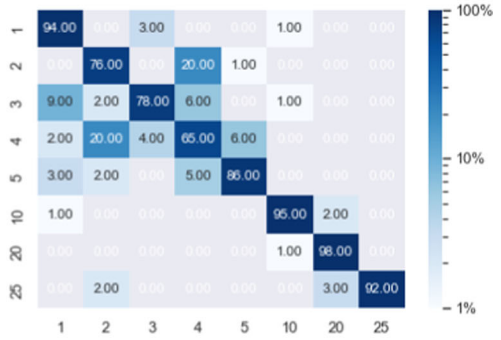
(a) SMOSSL algorithm.



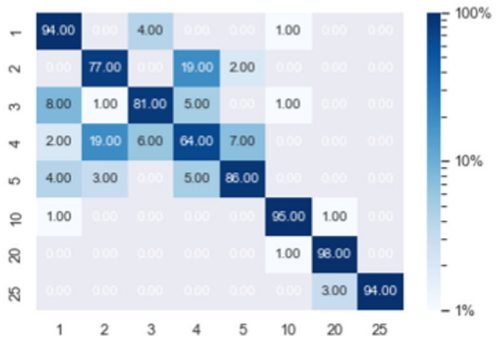
(b) Co-Training-C45C45C45 algorithm.



(c) Rel-RASCO-C45 algorithm.



(d) CoBC-Adaboost-C45 algorithm.



(e) CoBC-Bagging-C45 algorithm.

Fig. 4 Confusion matrices for the results of the different algorithms trained using the 10% labeled dataset. The numbers on each figure axis refer to the different types of faults that are described in Table 1. Logarithmic color scaling is used to aid visualization

Table 4 Computation time (in seconds) of the training and testing process for the different algorithms in Table 3

Labeled instances	Supervised SMOSSL (Reference)	Semi-supervised methods Co-Training C45C45C45	Rel-RASCO C45	CoBC Adaboost-C45	CoBC Bagging-C45
2%	10.8 s	143.4 s	189.9 s	67.0 s	253.8 s
40%	132.3 s	300.2 s	3060.6 s	259.1 s	6702.1 s

The first row shows the different computation times for the dataset containing 2% labeled instances and the second row shows the different computation times for the dataset containing 40% labeled instances

Table 4 shows the computation time, in seconds, for training and testing the algorithms in Table 3. The table contains the computation times for the dataset with 2% labeled instances and the dataset with 40% labeled instances. Substantially different computation times were observed, depending on both the different algorithms and the number of labeled instances in the datasets. As expected, training the algorithms using the less labeled dataset was faster than using the most labeled dataset. All algorithms required more computational time for model training as more labeled instances were included in the training dataset. A great difference was also noticeable among the algorithms, even when the same dataset was used. The fastest algorithm was 25 times faster than the slowest algorithm when using the 2% labeled dataset and about 50 times when using the 40% labeled dataset. However, accuracy (the metric used to compare the results of different models using the same subset of tests) remained comparable, despite the apparent difference in the computation times that were needed to train the model.

The lower percentages of labeled instances in the dataset may fall within what are considered extremely limited and small samples in [65, 66]. Two percent of labeled instances are less than 10 instances per class type in the dataset, which can also be considered an extremely limited number of labeled samples. Moreover, 3%-6% of labeled instances are fewer than 30 instances per class in the dataset, which can be considered a small number of labeled samples.

Although each SSL proposal in [14–16], and [17] was related to a different failure mode, some brief comparisons will help us to assess the potential of the SSL approach when used for FDD in wind turbines.

Macro and micro F1 scores were calculated in relation to the dataset containing 10% labeled instances, for a fair comparison with the proposal in [14]. The scores are shown in Table 5. When facing a multi-class classification problem, there are at least two ways to calculate the metric score: calculate the metric for each class separately and average the results across classes (macro- average), or calculate only a global metric without taking into account whether each instance belongs to one class or another (micro-average). Both provide a slightly different measure with its own interpretation. In a dataset with class imbalance, it is recommendable to use micro-average metrics. The macro and micro F1 scores were calculated by averaging the results obtained from the KEEL outputs for the five folds using the Python scikit-learn library. The macro and micro F1 scores for each algorithm were very similar. The models trained using SMOSSL, Co-Training, Rel-RASCO, and CoBC-Adaboost obtained values for the macro and micro F1 scores that fell within the range of values reported in [14] for diagnosing wind-turbine blade faults. The macro and micro F1 scores of the model trained with the CoBC-Bagging algorithm fell outside that range.

Both [15] and [16] used a SpectraQuest WTDS platform to obtain the dataset for training and testing their respective proposals. In [15], the result of a five class semi-supervised gear problem was reported, which was solved using its own pseudo-labeling process. A 99.38% accuracy level was reported using 230 labeled and pseudo-labeled instances. In [16], the results of a solution to a four class, semi-supervised bearing fault using an SVM algorithm were reported. Twenty out of 320 instances were unlabeled and 100% accuracy was obtained. In our case, the results using the 10% labeled dataset were not even close to those results, however, the 10% labeled dataset had fewer labeled instances per class, as the problem to be solved was an eight-class problem where the last class was a mix of the two failure types that had been diagnosed. It is interesting that the best results were also obtained for SVM in [16]. Finally, it should be outlined that the proposed scenario of unlabeled levels and dataset imbalance influenced the results. Most authors have sought to avoid both problems at the same time or to maintain soft conditions (low imbalance or low labeling rates), some way off real industrial conditions, so the door remains open to new research to find ML solutions that can be applied to both unlabeled levels and dataset imbalance problems at the same time.

In [17], the high complexity of the proposal to detect faults within the pitch system of wind turbines made comparisons difficult. The False Positive Rate (FPR) and False Negative Rate (FNR) were the metrics chosen for comparing the different alternatives of its four-class problem. Furthermore, each comparison was separately performed for each fault class with respect to the non-fault class.

FPR is an accuracy metric that calculates the rate of false positives out of all negatives. FPR is defined in (9).

$$FPR = \frac{FP}{FP + TN} \tag{9}$$

Table 5 Micro and macro F1 scores computed for 10% labeled instances dataset for the different algorithms shown in Table 3

Metric	Supervised SMOSSL (Reference)	Semi-supervised methods Co-Training C45C45C45	Rel-RASCO C45	CoBC Adaboost-C45	CoBC Bagging-C45
Micro-F1	0.87	0.84	0.85	0.85	0.71
Macro-F1	0.87	0.85	0.86	0.86	0.71

Table 6 Macro and micro FPR and FNR average scores computed for 10% labeled instances dataset for the different algorithms in Table 3

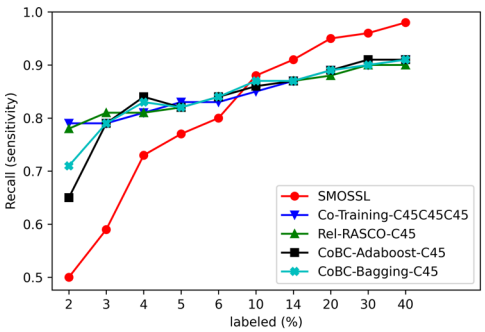
	Supervised		Semi-supervised methods							
	SMOSSL (Reference)		Co-Training C45C45C45		Rel-RASCO C45		CoBC Adaboost-C45		CoBC Bagging-C45	
Metric	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
Micro	0.02	0.13	0.02	0.15	0.03	0.22	0.02	0.14	0.02	0.14
Macro	0.02	0.12	0.02	0.15	0.03	0.21	0.02	0.14	0.02	0.13

FNR is an accuracy metric that calculates the rate of false negatives out of all negatives. FPR is defined in (10).

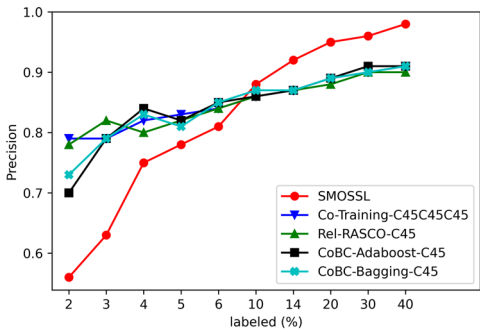
$$FNR = \frac{FN}{FN + TN} \tag{10}$$

Instead of calculating each of the eight FPR and FNR values for each of the five algorithms for the 10% labeled dataset, micro and macro FPR and FNR scores were calculated for each of the algorithms, as it would in any case be difficult to compare a four-class problem and an eight-class problem. Furthermore, rather than providing exact numerical values, separate boxplots were provided in [17], for each failure mode and for 5% and 10% of instances with labeled failures. Each of the three failure modes was represented with a set containing between 1158 and 2144 instances and 24 features.

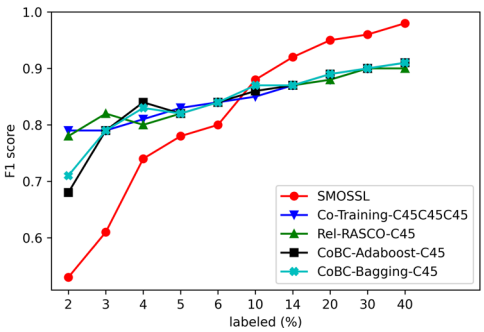
The macro and micro FPR and FNR scores of the 10% labeled dataset are shown in Table 6 for each algorithm in Table 3. In general, it can be said that FPR values in Table 6 were lower than those shown in [17], as all the algorithms shown in Table 6 yielded very low values ranging between 0.02 and 0.03, whereas those shown in [17] ranged between 0.03 and 0.05 for the 10% labeled datasets when using the ssODM-DSTA approach. On the other hand, FNR values for the ssODM-DSTA approach were lower (between 0.05 and 0.07) than the values shown in Table 6, which ranged between 0.12 and 0.22.



(a) Recall metric for the different algorithms.



(b) Precision metric for the different algorithms.



(c) F1 score metric for the different algorithms.

Fig. 5 Recall, precision and F1 score metrics for the different algorithms and percentages of labeled instances in the datasests

As shown in Fig. 5, several metrics other than accuracy, such as recall, precision, and F1 score, all micro-versions, presented similar patterns to the accuracy metrics (Fig. 3). The behavior of the semi-supervised methods was clearly better with lower percentages of labeled instances in the datasets. Consistent with the results obtained using the accuracy metric, for the dataset containing 10 percent labeled instances, the results obtained for the supervised SMOSSL algorithm outperformed the semi-supervised algorithms for high-labeled datasets (>10%), but not for low-labeled datasets (<10%), so that 10% of labeled instances were a trend at a crossroads that was likely to change.

Training time is also a parameter to be taken into account in ML techniques, due to energy consumption reduction requirements, and it is becoming increasingly relevant in computing [84]. Figure 6 shows the learning times of the different algorithms tested for each percentage of labeled instances in the datasets. As can be seen, the supervised SMOSSL and the semi-supervised CoBC-Adaboost-C45 and Co-Training-C45C45C45 algorithms took a very low linear learning time, and generated a very gentle slope on the results curve. Although learning algorithms can show very different behavior in learning time depending on the set of instances, in this particular case, an increase in the number of labeled instances appeared to yield a small increase in learning time. However, the same behavior was not observed when using the semi-supervised Rel-RASCO-C45 and CoBC-Bagging-C45 algorithms. The increment in learning time was greater, as the number of labeled instances increased. And clearly, the learning time of the CoBC-Bagging-C45 algorithm was longer than that of the Rel-RASCO-C45 algorithm. If the time needed to train a model with an algorithm were more important than the accuracy it achieved, it might be better not to choose one of the slowest algorithms, as accuracy (and the other metrics) are close for all semi-supervised algorithms for almost all percentages of labeled instances.

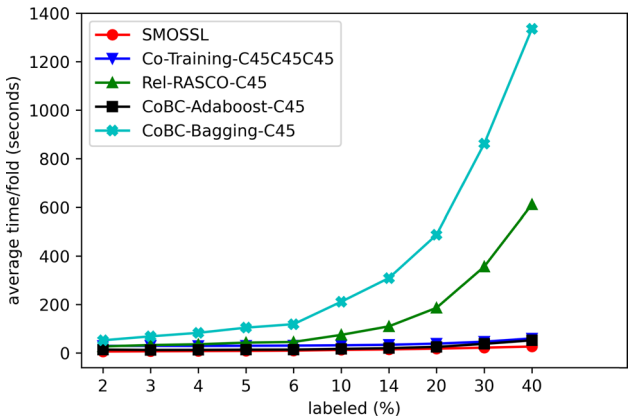


Fig. 6 Computing time for the different algorithms and percentages of labeled datasets

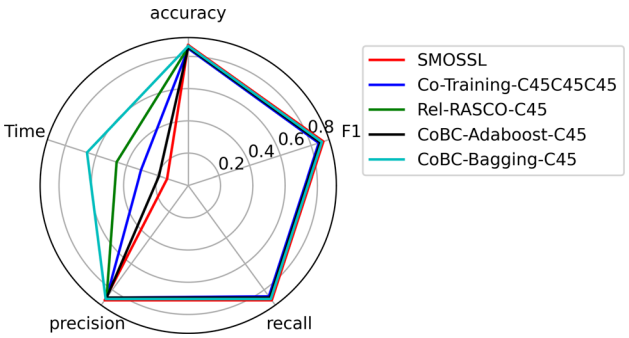


Fig. 7 Polar plots for the results of using the 10% of labeled instances dataset

Finally, as a summary of algorithm performance, Figure 7 shows the polar plot containing the accuracy, F1 score, recall and precision metrics and the average fold learning time for the 10% labeled instances dataset. The natural logarithm was taken for the learning times and then the values were scaled down to values between 0 and 1, to keep the same proportions as the other axes. As can be seen, the results of the scores for each axis are very close for all algorithms for each dataset, and the main differences occur on the time axis. The SMOSSL algorithm had the shortest learning time, and there was a clear difference in learning time for the different algorithms, with the CoBC-Bagging-C45 algorithm having the largest learning time.

5 Conclusions

Although competitive results have been achieved with SSL approaches that are very close to those obtained with more conventional supervised approaches, not many SSL approaches have been found in recent reviews specifically for wind turbine FDD. Therefore, recent approaches are reviewed and described along with more SSL approaches for wind turbine component FDD for related problems. A greater focus on semi-supervised methods would minimize the number of instances required in the datasets, the time-consuming collection of instances, tiresome human labeling processes, and the time needed to run sufficient simulations. The training time of ML proposals could therefore be reduced, while still achieving sufficient accuracy and competitive solutions.

A concise overview of recent approaches towards FDD in wind turbines, as well as in their associated parts and components, has been provided. As wind turbines have been gaining increasing attention over recent years, it is worth mentioning that some wind-turbine problems have received more attention than others. Literature and semi-supervised methods proposed for FDD in bearings are abundant, while they are scarce for gearboxes and transmissions. In addition, there are few semi-supervised FDD proposals that include

more than one type of failure. First, regarding the problem of FDD imbalanced bearings and gearbox misalignment.

Regarding the problem of FDD imbalanced bearings and gearbox misalignment, labeling between about 2% of the training instances (65 instances) and 10% of the training instances (327 instances) can be reasonable for a real-world problem, and can produce a model whose accuracy varies between 79.05% and 86.45%, in an eight-class classification problem. It makes the SSL approach viable for real-world industrial problems when a very limited number of labeled instances and additional unlabeled instances are available. Using up to 40% labeled instances in the dataset, the accuracy levels were as high as 91% using the SSL approach and up to 97.7% using the SL approach. If there were 10% or more labeled instances in the training set, then the supervised SMO method outperformed the tested SSL methods.

Similar and consistent results were obtained using different metrics. However, the learning times showed the greatest differences between the different learning algorithms.

In this problem, the SSL approach obtained better results when there were fewer labeled instances in the dataset (below 10% of labeled instances with the rest unlabeled).

Therefore, the use of unlabeled instances may help to improve the results obtained with SL methods, using only the corresponding subset of labeled instances.

Furthermore, no SSL algorithm is consistently better than the others, when using these semi-supervised datasets. As shown in Table 3, even though different SSL algorithms achieved slightly different accuracies on different datasets, the behaviour was generally similar and homogeneous.

It should be noted that there can be a clear and noticeable difference in the computational time required to train the various learning algorithms (Table 4). As expected, a clear difference in the training time required as a function of the number of labeled instances in the dataset was found: more labeled instances implied more training time. Furthermore, differences of up to 50 times the time taken by the slowest algorithm with respect to the fastest algorithm have been observed using the same dataset. However, this latter difference in computation times produced no large difference in accuracy when testing the corresponding models.

Finally, a 40% labeled subset of the training set was able to generate a supervised SMO model (SVM) that achieved an accuracy comparable to that of the model proposed in [9] (also an SVM). That model was generated with an SL approach, using 100% of the labeled training set instances.

Thus, it may be worth trying to use a smaller subset of the training set and to evaluate the results whenever the learning algorithm either takes too long to train the model or requires too much memory. In any case, SSL algorithms have shown their capability to process a complex industrial failure detection problem in a wind-turbine power train under 7 failure modes of 2 different types where labeled instances are rare, but unlabeled conditions are extensively available. Therefore, they can be useful to extend the accuracy of standard supervised ML models, although the effect of imbalance in the training dataset (few instances of failure conditions versus many instances of normal conditions) should still be simultaneously evaluated with high levels of unlabeled instances. Unfortunately, the dataset used in this research was not sufficiently extensive to test both industrial requirements at the same time.

Further experiments could be carried out with these datasets. First, it would be interesting to explore the importance of having imbalanced datasets and the impact on the calculated metrics. Secondly, it should be tested whether the number of classes can affect the calculated metrics. It was also found that the most difficult problem is dealing with imbalanced bearings, a subject that may deserve more attention and testing of alternative and specific solutions. For instance, the use of deep learning techniques might be a suitable solution, offering the chance to avoid the pre-processing stage, due to the capabilities of these methods to extract complex information from extensive raw datasets.

Acknowledgements This work was supported by the Junta de Castilla y León under project BU055P20 (JCyL/FEDER, UE), the Spanish Ministry of Science and Innovation under projects PID2020-119894GB-I00/AEI/10.13039/501100011033 and TED2021-129485B-C43, co-financed through European Union FEDER funds. It also was supported through the Consejeria de Educacion of the Junta de Castilla y Leon and the European Social Fund through a pre-doctoral grant (EDU/875/2021). Special thanks to the CARTIF FOUNDATION for providing the original dataset and for performing all the experimental tests on the test-bed. Our thanks also go to Ines Miguel Alonso for the artwork in Fig. 2.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data Availability The datasets generated during and/or analysed during the current study are not publicly available due to they were generated by a third party (CARTIF FOUNDATION) but are available from the corresponding author on reasonable request.

Declarations

Conflicts of interest All authors declare that they have no conflicts of interest.

References

1. The European Commission (2018) Renewable Energy Directive (EU) 2018/2001. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2018.328.01.0082.01.ENG&toc=OJ:L:2018:328:TOC
2. Nejad AR, Keller J, Guo Y, Sheng S, Polinder H, Watson S, Dong J, Qin Z, Ebrahimi A, Schelenz R, Gutiérrez Guzmán F, Cornel D, Golafshan R, Jacobs G, Blockmans B, Bosmans J, Pluymers B, Carroll J, Koukoura S, McDonald Hart E, A, Natarajan A, Torsvik J, Moghadam FK, Daems P-J, Verstraeten T, Peeters C, Helsen J (2022) Wind turbine drivetrains: state-of-the-art technologies and future development trends. Wind Energy Science 7(1):387–411. https://doi.org/10.5194/wes-7-387-2022
3. Yu D, Chen ZM, Xiahou KS, Li MS, Ji TY, Wu QH (2018) A radically data-driven method for fault detection and diagnosis in wind turbines. International Journal of Electrical Power & Energy Systems 99:577–584. https://doi.org/10.1016/j.ijepes.2018.01.009
4. Kabir MJ, Oo AMT, Rabbani M (2015) A brief review on offshore wind turbine fault detection and recent development in condition monitoring based maintenance system. In: 2015 Australasian Universities Power Engineering Conference (AUPEC), pp 1–7. https://doi.org/10.1109/AUPEC.2015.7324871
5. Santos P, Villa LF, Reñones A, Bustillo A, Maudes J (2015) An SVM-Based Solution for Fault Detection in Wind Turbines. Sensors 15(3):5627–5648. https://doi.org/10.3390/s150305627
6. Remigius WD, Natarajan A (2022) A review of wind turbine drivetrain loads and load effects for fixed and floating wind turbines. WIREs Energy and Environment 11(1):417. https://doi.org/10.1002/wene.417
7. Jantara VL, Papaelias M (2020) Chapter 5 - Wind turbine gearboxes: Failures surface treatments and condition monitoring. In: Papaelias M. Márquez F.P.G. Karyotakis A. (eds.) Non-Destructive Testing and Condition Monitoring Techniques for Renewable Energy Industrial Assets, pp 69–90. Butterworth-Heinemann. https://doi.org/10.1016/B978-0-08-101094-5.00005-8
8. Villa L, F, Reñones A, Perán JR, de Miguel LJ(2012) Statistical fault diagnosis based on vibration analysis for gear test-bench under non-stationary conditions of speed and load. Mech Syst Signal Process 29:436–446. https://doi.org/10.1016/j.ymssp.2011.12.013
9. Santos P, Maudes J, Bustillo A (2018) Identifying maximum imbalance in datasets for fault diagnosis of gearboxes. J Intell Manuf 29(2):333–351. https://doi.org/10.1007/s10845-015-1110-0
10. van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109(2):373–440. https://doi.org/10.1007/s10994-019-05855-6
11. Zhang Y, Yu K, Lei Z, Ge J, Xu Y, Li Z, Ren Z, Feng K (2023) Integrated intelligent fault diagnosis approach of offshore wind turbine bearing based on information stream fusion and semi-supervised learning. Expert Syst Appl 232:120854. https://doi.org/10.1016/j.eswa.2023.120854
12. Qian M, Wu H, Li Y-F (2023) Wind turbine blade early fault detection with faulty label unknown and labeling bias. IEEE Trans Industr Inform 19(7):8116–8126. https://doi.org/10.1109/TII.2022.3216816
13. Qian M, Li Y-F, Han T (2022) Positive-unlabeled learning-based hybrid deep network for intelligent fault detection. IEEE Trans Industr Inform 18(7):4510–4519. https://doi.org/10.1109/TII.2021.3121777
14. Qian M, Li Y-F (2022) A weakly supervised learning-based oversampling framework for class-imbalanced fault diagnosis. IEEE Trans Reliab 71(1):429–442. https://doi.org/10.1109/TR.2021.3138448
15. Chen S, Yang R, Zhong M (2021) Graph-based semi-supervised random forest for rotating machinery gearbox fault diagnosis. Control Eng Pract 117:104952. https://doi.org/10.1016/j.conengprac.2021.104952
16. Wang Z, Yao L, Cai Y, Zhang J (2020) Mahalanobis semi-supervised mapping and beetle antennae search based support vector machine for wind turbine rolling bearings fault diagnosis. Renewable Energy 155:1312–1327. https://doi.org/10.1016/j.renene.2020.04.041
17. Tang M, Hu J, Wu H, Wang Z (2021) Wind Turbine Pitch System Fault Detection Using ssODM-DSTA. Frontiers in Energy Research 502. https://doi.org/10.3389/fenrg.2021.750983
18. Wang Z, Qin B, Sun H, Zhang J, Butala MD, Demartino C, Peng P, Wang H (2023) An imbalanced semi-supervised wind turbine blade icing detection method based on contrastive learning. Renewable Energy 212:251–262. https://doi.org/10.1016/j.renene.2023.05.026
19. Man J, Wang F, Li Q, Wang D, Qiu Y (2023) Semi-supervised blade icing detection method based on Tri-XGBoost. Actuators 12(2). https://doi.org/10.3390/act12020058
20. Triguero I, González S, Moyano JM, García S, Alcalá-Fdez J, Luengo J, Fernández A, del Jesús MJ, Sánchez L, Herrera F (2017) KEEL 3.0: An open source software for multi-stage analysis in data mining. Int J Comput Intell Syst 10:1238–1249. https://doi.org/10.2991/ijcis.10.1.82
21. Alloghani M, Al-Jumeily D, Mustafina J, Hussain A, Aljaaf AJ (2020) In: Berry M.W. Mohamed A. Yap B.W. (eds.) A systematic review on supervised and unsupervised machine learning algorithms for data science, pp 3–21. Springer . https://doi.org/10.1007/978-3-030-22475-2_1
22. Iqbal M, Yan Z (2015) SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. Int J Soft Comput 5:946–952. https://doi.org/10.21917/ijsc.2015.0133
23. Ramírez-Sanz JM, Maestro-Prieto JA, Álvar Arnaiz-González Bustillo A (2023) Semi-supervised learning for industrial fault detection and diagnosis: A systemic review. ISA Transactions. https://doi.org/10.1016/j.isatra.2023.09.027
24. Tang M, Zhao Q, Wu H, Wang Z, Meng C, Wang Y (2021) Review and perspectives of machine learning methods for wind turbine fault diagnosis. Frontiers in Energy Research 9. https://doi.org/10.3389/fenrg.2021.751066
25. Kumar S, Ghosh J (2000) Crawford M. A hierarchical multiclassifier system for hyperspectral data analysis 1857:270–279. https://doi.org/10.1007/3-540-45014-9_26
26. Gui Q, Zhou H, Guo N, Niu B (2023) A survey of class-imbalanced semi-supervised learning. Machine Learning, pp 1–30. https://doi.org/10.1007/s10994-023-06344-7
27. Bekker J, Davis J (2020) Learning from positive and unlabeled data: a survey. Machine Learning 109:719–760. https://doi.org/10.1007/s10994-020-05877-5
28. Cai C, Guo J, Song X, Zhang Y, Wu J, Tang S, Jia Y, Xing Z, Li Q (2023) Review of Data-Driven Approaches for Wind Turbine Blade Icing Detection. Sustainability 15(2). https://doi.org/10.3390/su15021617
29. Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. KDD '16, pp 785–794. Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785
30. Zhou Z-H, Li M (2005) Tri-training: exploiting unlabeled data using three classifiers. IEEE Trans Knowl Data Eng 17(11):1529–1541. https://doi.org/10.1109/TKDE.2005.186

31. SpectraQuest's Wind Turbine Drivetrain Diagnostics Simulator. https://spectraquest.com/drivetrains/details/del_wtds/. Accessed: 2023-01-10
32. Yang X, Song Z, King I, Xu Z (2023) A Survey on Deep Semi-Supervised Learning. IEEE Trans Knowl Data Eng 35(09):8934–8954. https://doi.org/10.1109/TKDE.2022.3220219
33. Vanyan A, Khachatrian H (2021) Deep semi-supervised image classification algorithms: a Survey. JUCS - Journal of Universal Computer Science 27(12):1390–1407. https://doi.org/10.3897/jucs.77029
34. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
35. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791
36. Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning 2011. http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf
37. Coates A, Ng A, Lee H (2011) An analysis of single-layer networks in unsupervised feature learning. In: Gordon, G., Dunson, D., Dudík, M. (eds.) Proceedings of the fourteenth international conference on artificial intelligence and statistics. Proceedings of Machine Learning Research, vol 15, pp 215–223. PMLR. http://proceedings.mlr.press/v15/coates11a/coates11a.pdf
38. LeCun Y, Huang FJ, Bottou L (2004) Learning methods for generic object recognition with invariance to pose and lighting. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004., vol 2, pp 104–2. https://doi.org/10.1109/CVPR.2004.1315150
39. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255. https://doi.org/10.1109/CVPR.2009.5206848
40. Villa-Pérez ME, Álvarez-Carmona MA, Loyola-González O, Medina-Pérez MA, Velazco-Rossell JC, Choo K-KR (2021) Semi-supervised anomaly detection algorithms: A comparative summary and future research directions. Knowl-Based Syst 218:106878. https://doi.org/10.1016/j.knosys.2021.106878
41. Ding Y, Zhuang J, Ding P, Jia M (2022) Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings. Reliab Eng Syst Saf 218:108126. https://doi.org/10.1016/j.ress.2021.108126
42. Mao W, Tian S, Fan J, Liang X, Safian A (2020) Online detection of bearing incipient fault with semi-supervised architecture and deep feature representation. J Manuf Syst 55:179–198. https://doi.org/10.1016/j.jmsy.2020.03.005
43. Sharma J, Mittal ML, Soni G (2022) Condition-based maintenance using machine learning and role of interpretability: a review. International Journal of System Assurance Engineering and Management. https://doi.org/10.1007/s13198-022-01843-7
44. Luo S, Huang X, Wang Y, Luo R, Zhou Q (2022) Transfer learning based on improved stacked autoencoder for bearing fault diagnosis. Knowl-Based Syst 256:109846. https://doi.org/10.1016/j.knosys.2022.109846
45. Lin J, Shao H, Min Z, Luo J, Xiao Y, Yan S, Zhou J (2022) Cross-domain fault diagnosis of bearing using improved semi-supervised meta-learning towards interference of out-of-distribution samples. Knowl-Based Syst 252:109493. https://doi.org/10.1016/j.knosys.2022.109493
46. Cui L, Tian X, Shi X, Wang X, Cui Y (2021) A Semi-Supervised Fault Diagnosis Method Based on Improved Bidirectional Generative Adversarial Network. Appl Sci 11(20). https://doi.org/10.3390/app11209401
47. Fu W, Jiang X, Tan C, Li B, Chen B (2022) Rolling bearing fault diagnosis in limited data scenarios using feature enhanced generative adversarial networks. IEEE Sensors Journal 22(9):8749–8759. https://doi.org/10.1109/JSEN.2022.3160762
48. Wu Y, Zhao R, Jin W, He T, Ma S, Shi M (2021) Intelligent fault diagnosis of rolling bearings using a semi-supervised convolutional neural network. Appl Intell 51(4):2144–2160. https://doi.org/10.1007/s10489-020-02006-6
49. Zhang J, Kong X, Cheng L, Qi H, Yu M (2023) Intelligent fault diagnosis of rolling bearings based on continuous wavelet transform-multiscale feature fusion and improved channel attention mechanism. Eksploatacja i Niezawodność – Maintenance and Reliability 25(1). https://doi.org/10.17531/ein.2023.1.16
50. Zhang Y, Ren Z, Zhou S (2020) An intelligent fault diagnosis for rolling bearing based on adversarial semi-supervised method. IEEE Access 8:149868–149877. https://doi.org/10.1109/ACCESS.2020.3016314
51. Wang Z, Xuan J, Shi T (2022) A novel semi-supervised generative adversarial network based on the actor-critic algorithm for compound fault recognition. Neural Comput & Applic 34(13):10787–10805. https://doi.org/10.1007/s00521-022-07011-z
52. Gao Y, Yu D (2021) Intelligent fault diagnosis for rolling bearings based on graph shift regularization with directed graphs. Adv Eng Inform 47:101253. https://doi.org/10.1016/j.aei.2021.101253
53. Chen X, Wang Z, Zhang Z, Jia L, Qin Y (2018) A semi-supervised approach to bearing fault diagnosis under variable conditions towards imbalanced unlabeled data. Sensors 18(7):2097. https://doi.org/10.3390/s18072097
54. Gao Y, Yu D (2021) Fault diagnosis of rolling bearing based on laplacian regularization. Appl Soft Comput 111:107651
55. Yu K, Ma H, Lin TR, Li X (2020) A consistency regularization based semi-supervised learning approach for intelligent fault diagnosis of rolling bearing. Measurement 165:107987. https://doi.org/10.1016/j.measurement.2020.107987
56. Tao X, Ren C, Li Q, Guo W, Liu R, He Q, Zou J (2021) Bearing defect diagnosis based on semi-supervised kernel local fisher discriminant analysis using pseudo labels. ISA Transactions 110:394–412. https://doi.org/10.1016/j.isatra.202010.033
57. Yu K, Lin TR, Ma H, Li X, Li X (2021) A multi-stage semi-supervised learning approach for intelligent fault diagnosis of rolling bearing using data augmentation and metric learning. Mech Syst Signal Process 146:107043. https://doi.org/10.1016/j.ymssp.2020.107043
58. Zhao B, Cheng C, Zhao S, Peng Z (2023) Hybrid semi-supervised learning for rotating machinery fault diagnosis based on grouped pseudo labeling and consistency regularization. IEEE Trans Instrum Meas 72:1–12. https://doi.org/10.1109/TIM.2023.3269112
59. Cheng C, Shan D, Teng Y, Zhao B, Peng Z, He Q (2023) Semisupervised fault diagnosis for gearboxes: a novel method based on a hybrid classification network and weighted pseudo-labeling. IEEE Sensors Journal 23(14):16373–16383. https://doi.org/10.1109/JSEN.2023.3281428
60. Shan D, Cheng C, Li L, Peng Z, He Q (2023) Semisupervised fault diagnosis of gearbox using weighted graph-based label propagation and virtual adversarial training. IEEE Trans Instrum Meas 72:1–11. https://doi.org/10.1109/TIM.2022.3225013
61. Zhang K, Tang B, Qin Y, Deng L (2019) Fault diagnosis of planetary gearbox using a novel semi-supervised method of multiple association layers networks. Mech Syst Signal Process 131:243–260. https://doi.org/10.1016/j.ymssp.2019.05.049
62. Qin Y, Yao Q, Wang Y, Mao Y (2021) Parameter sharing adversarial domain adaptation networks for fault transfer diagnosis of plane-

tary gearboxes. Mech Syst Signal Process 160:107936. https://doi.org/10.1016/j.ymssp.2021.107936
63. Li J, Wang Y, Zi Y, Sun X, Yang Y (2021) A Current Signal-Based Adaptive Semisupervised Framework for Bearing Faults Diagnosis in Drivetrains. IEEE Trans Instrum Meas 70:1–12. https://doi.org/10.1109/TIM.2020.3046051
64. Surucu O, Gadsden SA, Yawney J (2023) Condition Monitoring using Machine Learning: A Review of Theory, Applications, and Recent Advances. Expert Syst Appl 221:119738. https://doi.org/10.1016/j.eswa.2023.119738
65. Zhang T, Chen J, Li F, Pan T, He S (2021) A small sample focused intelligent fault diagnosis scheme of machines via multimodules learning with gradient penalized generative adversarial networks. IEEE Trans Ind Electron 68(10):10130–10141. https://doi.org/10.1109/TIE.2020.3028821
66. Xie Z, Chen J, Feng Y, He S (2022) Semi-supervised multi-scale attention-aware graph convolution network for intelligent fault diagnosis of machine under extremely-limited labeled samples. J Manuf Syst 64:561–577. https://doi.org/10.1016/j.jmsy.2022.08.007
67. Yarowsky D (1995) Unsupervised word sense disambiguation rivaling supervised methods. In: 33rd Annual meeting of the association for computational liguistics, pp 189–196
68. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the annual ACM conference on computational learning theory, pp 92–100
69. Wang J, Luo S-w, Zeng X-h (2008) A random subspace method for co-training. In: 2008 IEEE International joint conference on neural networks (IEEE World Congress on Computational Intelligence), pp 195–200. https://doi.org/10.1109/IJCNN.2008.4633789
70. Yaslan Y, Cataltepe Z (2010) Co-training with relevant random subspaces. Neurocomputing 73(10):1652–1661. https://doi.org/10.1016/j.neucom.2010.01.018
71. Li M, Zhou ZH (2007) Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. Systems and Man and Cybernetics and Part A: Systems and Humans and IEEE Transactions on 37(6):1088–1098
72. Deng C, Guo MZ (2011) A new co-training-style random forest for computer aided diagnosis. J Intell Inf Syst 36(3):253–281
73. Zhou Y Goldman S (2004) Democratic co-learning. In: IEEE International Conference on Tools with Artificial Intelligence, pp 594–602
74. Huang T, Yu Y, Guo G, Li K (2010) A classification algorithm based on local cluster centers with a few labeled training examples. Knowl-Based Syst 26(6):563–571
75. Hady M, Schwenker F, Palm G (2010) Semi-supervised learning for tree-structured ensembles of RBF networks with Co-Training. Neural Networks 23(4):497–509. https://doi.org/10.1016/j.neunet.2009.09.001
76. Halder A, Ghosh S, Ghosh A (2013) Aggregation pheromone metaphor for semi-supervised classification. Pattern Recogn 46:2239–2248
77. Li M, Zhou Z-H (2005) SETRED: Self-training with Editing. In: Ho TB, Cheung D, Liu H (eds) Advances in knowledge discovery and data mining pp 611–621. Springer
78. Wang Y, Xu X, Zhao H, Hua Z (2010) Semi-supervised learning based on nearest neighbor rule and cut edges. Knowl-Based Syst 23(6):547–554
79. Valiant L (2013) Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World. Basic Books, Inc.,
80. Dempster AP (1968) A Generalization of Bayesian Inference. Journal of the Royal Statistical Society: Series B (Methodological) 30(2):205–232. https://doi.org/10.1111/j.2517-6161.1968.tb00722.x
81. Shafer G (1976) A Mathematical Theory of Evidence. Princeton University Press
82. Platt J (1998) Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14, Microsoft
83. Villa LF, Reñones A, Perán JR, de Miguel LJ (2011) Angular resampling for vibration analysis in wind turbines under non-linear speed fluctuation. Mech Syst Signal Process 25(6):2157–2168. https://doi.org/10.1016/j.ymssp.2011.01.022
84. Gutiérrez M, Moraga MA, García F, Calero C (2023) Green-in machine learning at a glance. Computer 56(6):35–43. https://doi.org/10.1109/MC.2023.3254646

Jose Alberto Maestro-Prieto received his B.S. (1997) and his M.S. (2000) in Computer Science from the Universidad de Valladolid. In 2000, he joined the Computer Science Department of the Universidad de Valladolid, where he was a member of the Intelligent Systems Group and obtained his Ph.D. (2011). He is currently at the University of Burgos, where he is a member of the ADMIRABLE research group (Advanced Data Mining Research And Business Intelligence/Big Data/Bioinformatics Learning), and his research interests are mainly related to fault detection and diagnosis.

José Miguel Ramírez-Sanz received his B.S. (2019) in Computer Science from the Universidad de Burgos and his M.S. (2020) in Business Intelligence and Big Data in Cyber-Secure Environments from the Universidad de Burgos. In 2020, he joined the Computing Department at the Universidad de Burgos, where he is pursuing his Ph.D. on Semi-supervised Learning applied to Industry problems. He is a member of the ADMIRABLE research group (Advanced Data Mining Research And Business Intelligence/Big Data/Bioinformatics Learning).

Andrés Bustillo is Full Professor at Burgos University (Spain). He holds a Ph.D. in Physics from the University of Valladolid (Spain). His Ph.D. examined laser development for Plasma Diagnostics using an experimental set-up developed at the Physikalish-Technische Bundesanstalt (PTB) Berlin. Subsequently, he worked for 7 years as an R&D Project Manager at Nicolas Correa S.A., a leading industrial group in the design of large-scale milling machines. Over this period, he gained experience in the simulation and optimization of high speed milling, laser cladding, and other industrial technologies related to the manufacturing industry. Nowadays, his research interests are focused on the industrial applications of various machine learnibng and data mining techniques.

Juan José Rodriguez-Díez received the BS, MS and Ph.D. degrees in Computer Science from the University of Valladolid, Spain, in 1994, 1998 and 2004, respectively. He worked with the Department of Computer Science, University of Valladolid from 1995 to 2000. Since then he has worked at the Universidad de Burgos, Spain, where he is a Full Professor in the Department of Computer Science. His interests include data science, machine learning and pattern recognition. He has worked on methods for classifier and regression ensembles, time series, feature selection, instance selection, multi-output, semi-supervised and big data; with industrial, health, bioinformatics, ecological and educational applications.