Neurocomputing 194 (2016) 192–206

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A statistical comparison of neuroclassifiers and feature selection methods for gearbox fault diagnosis under realistic conditions

Fannia Pacheco [a,*], José Valente de Oliveira [c,e], René-Vinicio Sánchez [e], Mariela Cerrada [b,e], Diego Cabrera [e], Chuan Li [d,e], Grover Zurita [e], Mariano Artés [f]

[a] Independent researcher at GIDTEC-Mechanical Engineering Department, Universidad Politécnica Salesiana, Ecuador
[b] Control Systems Department, Universidad de Los Andes, Venezuela
[c] CEOT, Universidade do Algrave, Faro, Portugal
[d] Chongqing Key Laboratory of Manufacturing Equipment Mechanism Design and Control, Chongqing Technology and Business University, Chongqing, China
[e] Mechanical Engineering Department, Universidad Politécnica Salesiana, Ecuador
[f] Department of Mechanics, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain

ARTICLE INFO

Article history:
Received 22 October 2015
Received in revised form
29 December 2015
Accepted 1 February 2016
Communicated by Shen Yin
Available online 26 February 2016

Keywords:
Neural networks
Statistic tests
Classification
Fault diagnosis
Feature selection
Gearbox

ABSTRACT

Gearboxes are crucial devices in rotating power transmission systems with applications in a variety of industries. Gearbox faults can cause catastrophic physical consequences, long equipment downtimes, and severe production costs. Several artificial neural networks, learning algorithms, and feature selection methods have been used in the diagnosis of the gearbox healthy state. Given a specific gearbox, this study investigates how these approaches compare with each other in terms of the typical fault classification accuracy but also in terms of the area under curve (AUC), where the curve refers to the precision-recall curve otherwise known as receiver operating characteristic (ROC) curve. In particular, the comparison aims at identifying whether there are statistically significant (dis)similarities among six feature selection methods, and seven pairs of neural nets with different learning rules. Genetic algorithm based, entropy based, linear discriminants, principal components, most neighbors first, and non-negative matrix factorization are the studied feature selection methods. Feed forward perceptrons, cascade forward, probabilistic nets, and radial basis function neural nets are evaluated. Six supervised and one unsupervised learning rules are considered. Both parametric and nonparametric statistical tests are employed. A ranking process is defined to elect the best approach, when available. An experimental setup was especially prepared to ensure operating conditions as realistic as possible.

1. Introduction

Gearboxes play a crucial role in power transmission system, as they are designed for speed and torque conversion. In a conventional gearbox this is accomplished by connecting two or more (parallel) shafts through coupled gears with different sizes. Gearboxes can be found in applications as distinct as automotive, industrial, maritime, and power generation. Gearboxes are prone to incur in different types of faults, which in turn cause problems in the coupled device. Early fault detection may prevent further problems related to severe physical damage, breakdowns, production costs, among others. In gear box fault diagnosis, vibration signals are typically employed for assessing the healthy state. Fault diagnosis include tasks such as (i) fault detection, i.e., identify whether or not the equipment is in its healthy state; (ii) fault classification, i.e., given that the equipment has a fault, to identify the type of fault, and (iii) fault prognosis, i.e., to estimate the current remaining useful life of the equipment.

Fault diagnosis methods can be divided into two main groups, i.e., model-based and data-based methods [1–3]. Model-based approaches use mathematical models to describe, control and monitor rotating machines; they require an extensive knowledge of the system and the dynamics associated with it, which in real world applications is hard to achieve. On the other hand, data-based methods, commonly called data-driven approaches, have been widely employed in fault diagnosis to predict and to analyze current and future states of rotating machines. The most popular data-driven techniques for fault diagnosis are presented by [1,2]. The authors categorized the data-driven approaches in: statistical methods that include Hidden Markov model (HMM), Principal Component Analysis (PLC) and Partial Least Squares (PLS) techniques [4,5]; neural networks (NN); Support Vector Machine (SVM); fuzzy logic; and Neuro-fuzzy (NF) system, among others. Particularly, the usage of these techniques depends on their

* Corresponding author.
E-mail addresses: fannikaro@gmail.com (F. Pacheco), rsanchezl@ups.edu.ec (R.-V. Sánchez), cerradam@ula.ve (M. Cerrada), dcabrera@ups.edu.ec (D. Cabrera), chuanli@21cn.com (C. Li), gzuritav@ups.edu.ec (G. Zurita).

http://dx.doi.org/10.1016/j.neucom.2016.02.028
0925-2312/

requirements (e.g., computational cost, historical data requirement, etc), and their characteristics that are more favorable for some types of problems than others. This work deals only with fault classification and focuses on the employment of artificial neural networks (ANNs) for such task.

ANNs have proved to be an adequate technique for complex systems with non-linear behaviours, and it is common to classify failures in gears [6]. There are very many works on ANN for gearbox fault classification, the most relevant of which being briefly reviewed below. Actually, ANNs have some characteristics that make them particularly appealing for fault classification. These include (i) universal approximation, i.e., they are able to approximate to an arbitrary degree of precision any function in a compact (closed and bounded) domain, (ii) they exhibit a graceful performance degradation with the fault of their own elements, i.e., artificial neurons, (iii) have the potential to massively distribute and parallel computing, (iv) some ANNs are particularly robust to noisy and incomplete inputs, and (v) more and more learning algorithms are being proposed for learning their parameters (weights). In the fault diagnosis field, ANNs are commonly trained with labeled vibration data and once trained they are supposed to correctly classify a fault from previously unseen data.

ANNs are commonly trained in a supervised learning scheme, where both inputs and outputs are known. Classic ANNs include multilayer perceptrons with different structures such as the feed forward neural network (FFNN), cascade forward network (CFN), among others; the learning method can vary among back propagation (BP), Levenberg–Marquardt (LM) algorithm, etc. Any of these approaches, i.e., any pair (network, learning algorithm), is suitable for fault diagnosis. FFNN with different training methods is one of the most popular approach presented as a diagnoser in rotatory machinery e.g., [7–13].

Radial basic function network (RBFN) is an alternative to multilayer perceptrons. Roughly speaking, an RBFN forms a linear combination of the basis functions (e.g., a multivariate Gaussian) computed by the hidden units of the net. RBFN is also suitable for mechanical fault diagnosis, e.g., [14,15]. On the other hand, some other work reports a fault detection system using a probabilistic neural network (PNN) as classifier [16,17].

Works combining ANN with fuzzy logic for gear fault diagnosis are also available, e.g., [18–20]. The work in [21] presents a classifier using a neural fuzzy scheme for real-time machinery health condition monitoring in gear systems. The classifier is trained by a hybrid method based on recursive Levenberg–Marquardt(LM) and least-squares estimate (LSE) to improve the classifier convergence.

Another approach uses ANNs with an unsupervised learning process to construct and train the model. One of the ANNs belonging to this category is self organizing map (SOM), as it produces in an unsupervised way a low-dimensional discretized representation of the input space. This technique is also suitable for fault diagnosis because it helps preserve the topological relationships of the data [22,23].

Training conventional ANNs requires an a priori definition of some parameters regarding their architecture. Genetic Algorithms (GA) have been integrated to ANN to find an appropriated neural network structure that minimizes the error, number of layers and number of neurons in each layer; in addition GA is widely applied to feature selection in large databases. This particular approach has been applied for fault detection in gearbox recently [24–26].

From the above, it is clear that ANNs are common tools for fault diagnosis in rotatory machinery conditions. A question that naturally arises is this: Given a specific gearbox, how all these approaches compare to each other, e.g., in terms of fault classification accuracy? What is the best approach, if any? Are there significant dissimilarity between the ANN results when it comes to diagnose a fault? Curiously enough answers to these questions are not easily found in the literature, with some worth nothing exception though. In [27] FFNN and RBFN are compared offering thus at least a partial answer. In an attempt to fill this literature gap, the paper proposes the following contributions: (i) A statistical evaluation of seven neurocomputing approaches, i.e., different combinations of neural networks and learning algorithms for gearbox fault classification. Six are supervised while the remaining one is unsupervised. (ii) In addition, six feature selection techniques are also statistically evaluated. These include GA based, entropy based, linear discriminant analysis (LDA), principal component analysis (PCA), most neighbours first (MNF) and non-negative matrix factorization (NMF). Feature selection is a critical step for optimizing efficiency, accuracy and for mitigating overtraining; (iii) a systematic statistical approach to compare algorithms for a real world application, and (iv) a comprehensive and update literature review on the above methods in the context of fault diagnosis.

The analysis includes the usual accuracy and, following the suggestion of [28], the area under curve (AUC) metric. Non-parametric statistical tests such as Friedman test and post-hoc procedures, are included to support our conclusions.

An experimental setup was especially prepared for this study that ensures operating conditions as realistic as possible.

This paper is organized as follows. Section 2 reviews the theory used by our proposal. Section 3 introduces the experimental framework proposed. Section 4 presents the data extraction process to get the fault detection database. Section 5 presents the setting established to test the ANNs. Section 6 summarizes the statistical methods for multicomparison used by our proposal. Section 7 shows the results obtained once the experimental framework is evaluated. Section 8 outlines a discussion about the results given by the previous section, and finally Section 9 concludes the paper.

2. Background

The section briefly reviews the required background for the study. First of all, ANN concepts are described and the ANNs considered in this paper are shown. Finally, the feature selection process is described, and some of the algorithms chosen to perform this task in our proposal.

2.1. Artificial neural network

The main learning problem in classification using an ANN is related to the minimization of an objective function, also known as cost function. One of the most popular learning algorithms is backpropagation. It aims to detect the minimum of the cost function in the weight space using an optimization method such as gradient descent. Given the data set $\{(x^1, y^1), \ldots, (x^m, y^m)\}$, where x^i is the input vector and y^i its corresponding output; when a multi-classification problem is placed $y^i \epsilon \mathcal{R}^K$, in this sense x^i belongs to the class C_k where $y^i_k = 1$. One possible cost function, resulting from a Maximum Likelihood Estimation (MLE) can be given by:

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y^i_k \log\left(h_\theta(x^i)\right)_k + (1-y^i_k)\log\left(1-h_\theta(x^i)\right)_k\right] \quad (1)$$

Particularly, gradient descent (GD) method uses the Eq. (2) to update the weights, but there are other approaches; e.g., scaled gradient descent (SGD), gradient descent with adaptive learning rate α, gradient descent with momentum and adaptive learning rate; where this equation slightly varies in order to find improvements in the optimization problem. On the other hand, there are some other techniques used to solve the optimization problem, e.g., Levenberg–Marquardt (LM) algorithm, Newton's method, among others. LM has demonstrated to be one of the best options getting fast and stable convergence, but this algorithm is

suitable only for small and medium sized training sets.

$$\theta_i^j := \theta_i^j - \alpha \frac{\partial}{\partial \theta_{jk}^l} J(\Theta) \quad (2)$$

2.1.1. Some classic ANNs

Also known as multi-layer (feed forward) perceptrons, the feed forward network (FFNN) is one of the most popular ANNs characterized by the absence of feedback loops in the network. The layers are arranged such that the information flows from the input to the output, i.e., the output of layer l is the input of layer $l+1$ ($l=0,\ldots,n-1$) where n is the number of layers. Backpropagation is the most common learning algorithm for this type of networks. Another particular structure is the cascade forward network (CFN). The CFN includes a weighted connection from the input layer to each layer in the network. Any learning method used for FFNN can also be applied to CFN.

The radial basic function network (RBFN) is a feed forward network with the peculiarity that only has three layers: input, hidden and output layer. The Euclidean distance is the common metric to compute the distance between two points. Generally speaking, the main objective is to find a function $f(x)$ that can map the input vector to its corresponding target. Typically, the RBFN training process includes two main steps: (1) define the center vectors c_n of the hidden layers; e.g., one of the simplest ways is to randomly choose them from the input data x, but they can also be selected by an unsupervised method such as k-means clustering; (2) the second step simply fits a linear model with coefficients θ_n to the hidden layer's outputs with respect to some objective function, such as the least squares objective function.

Probabilistic neural network (PNN) has a similar structure and training to RBFN, but it is composed of four layers: the input layer, hidden layer, summation layer and output layer. The input layer has one neuron for each case, and the distance between the input and the samples in the training data is computed; getting as a result a vector where its elements define how close an input is to the training data. The summation layer sums the results gained from the hidden layer and gets the probabilistic density function value in each class for the input. Finally, the previous result is given as an input to a transfer function in the output layer, this last one selects the maximum probability value.

The ANN structures analyzed are based on a supervised scheme to be trained, where the desirable output of each input in the training set are known. In the literature there can be found several approaches that aim to find a neural network structure using an unsupervised learning method to train the net. Self-organizing Maps (SOM) is a popular ANN based on unsupervised learning. Neurons dealing with closely related pieces of information are kept close together in order to facilitate the interaction via short synaptic connections, so that the neighborhood relations are preserved. The most popular SOM structure called Kohonen Network has a feed-forward structure with a single computational layer arranged in rows and columns.

2.1.2. Genetic algorithm with ANN

A genetic algorithm (GA) operates on a population of individuals each one of them being a candidate solution to the considered optimization problem. In the classical form, each individual is composed of one or more chromosomes which in turn are viewed as a set of genes, coded as a binary string. Associated with each individual there is a figure of merit (fitness function) which indicates how good this candidate solution is. The algorithm itself is an iterative stochastic procedure where in each iteration, individuals are stochastically selected based on their fitness. Selected individuals (progenitors) will have a chance to crossover their genes among each other, thus propagating part of their genes to new individuals (offspring). During this process sometimes mutation occurs, i.e., one gene flips its value with a (very) low probability. Offspring can replace total or partially the old population. The whole process repeats until a given stop criterion is met.

To use ANNs, the hidden layers (among other parameters) need to be defined a priori. Different hidden layers result in different ANNs with different approximation capabilities. GA can be used to find an adequate ANN structure. GA is usually used with ANN to reduce the number of features and find a suitable number of layers. The GA process starts by creating a random population of individuals, each one of them codifying a candidate neural network structure for a certain subset of features. Afterwards, actual neural networks are instantiated from this population. These are trained and evaluated. If the stopping criteria is met, the process finishes and the best chromosomes are obtained; if not the process continues selecting a new population to create a new ANN and so on. Once the GA process finishes, a classification phase is executed with the parameters obtained in the previous step.

2.2. Feature selection

Throughout the past years, the data that can be collected from real world applications has grown dramatically. These applications are notable for their measurable properties called features (attributes), which in turn are used by machine learning techniques to knowledge extraction. In particular, for classification tasks a large space of attributes may carry several problems such as: decreasing accuracy, increasing computational burden, and introducing bias. Feature selection is to find a low set of attributes that better describe a process.

In general, the feature selection techniques are divided into two groups supervised and unsupervised learning. In supervised learning the feature selection procedure is performed searching relations between the features and an objective function. The objective function in classification is the one that maximizes the classifier performance that assures the selection of features that truly influence the objective function results. On the other hand, unsupervised based feature selection methods try to gather together those features that present high similarity between themselves. We briefly present as follows the theoretical background of the techniques used by our proposal.

2.2.1. Feature selection by supervised learning approaches

- Random forest Decision tree splits the feature space into disjoint regions that are associated to classes. This partition is represented by a tree, the nodes are the features (attributes) and the leaves are classes. Each level of the tree is built using the features that distinguish between one class and another. In order to construct the tree, a metric determines the information degree contributed for each feature with regard to the classes. Commonly, the metric used to build the tree is the entropy. Random Forest (RF) is an algorithm based on k decision trees that are built and trained with bootstrap samples versions of the original training data. Then, given a new input, the predicted class is obtained from a voting process that is executed over the result of each tree [29]. RF is suitable for obtaining the most representative features, since each decision tree computes the information degree contributed by each feature to the classes; therefore, in RF the information of each feature over all the trees can be averaged in order to rank the variables [30].
- Linear Discriminant Analysis (LDA) is a well-known supervised method for dimensionality reduction. It performs a projection from high-dimensional data to a lower dimensional space; this is achieved by maximizing the separation of data points from different classes and minimizing the dispersion of data from the same class at the same time. It constructs a small number of features by applying a linear transformation $G^T \in \mathcal{R}^{m\times l}$ that maps each data point $X=[x_1,..,x_n] \in \mathcal{R}^{m\times n}$ with m dimensions to a lower

dimensional space l using $z = G^T X$. G^T is obtained through an objective function that maximizes the between-class distance and minimize the within-class distance in the dimensionality-reduced space [31].

2.2.2. Feature selection by unsupervised learning approaches

- Principal Component Analysis (PCA) is one of the most popular unsupervised techniques for dimensionality reduction. PCA tries to find a space with a lower dimension in which the data is projected, the so-called principal components that explain the maximum possible data variability. More concretely, PCA is a linear orthogonal transformation that transforms a data set $X \in \mathcal{R}^{m \times n}$ into a new space $z \in \mathcal{R}^{k \times n}$. Generally speaking, to reduce the n-dimensional data into a k-dimensional space, the vectors $U = [u_1, \ldots, u_k]$ need to be found in order to project the data into them. These vectors are the eigenvectors of the covariance matrix of the normalized data.
- Non-negative matrix factorization (NMF) is a decomposition method which represents a non-negative matrix in two low-rank non-negative matrices using the expression $X \approx A \times Y$. Let $X \in \mathcal{R}^{m \times n}$ be a non-negative matrix with m features (columns) and n samples (rows). The feature selection objective is to find the non-negative factors $A \in \mathcal{R}^{m \times k}$ and $Y \in \mathcal{R}^{k \times n}$, decomposing the training data into k meta-samples. That is accomplished through an optimization process. Specifically for feature selection, the training data (with m features and n samples) can be decomposed into k meta-samples; where Y represents the reduced matrix and stores the k new attributes with n samples [32].
- Most neighbors first Most Neighbors First (MNF) is an unsupervised algorithm for data reduction based on attribute clustering. The objective is finding k clusters that represent the most representative features, where a cluster is a set of features that are represented by a medoid as its most important feature. MNF splits the feature space into k clusters following the k-medoid principle. The process starts randomly selecting k features as the initial cluster medoids. Next, for each feature a dissimilarity metric based on the relative dependency concept is computed between the feature and each medoid. All non-center features are assigned to their nearest centroid and a reconfiguration of the clusters takes place. The process is repeated until the stopping criterion is met [33].

3. Experimental framework

Fig. 1 presents an activity diagram describing the steps required to achieve a multicomparison between several ANNs. Generally speaking, the process starts with vibration signal acquisition from sensors (accelerometers). From these, time domain, frequency domain, and frequency-time domain features are computed. The exact features are described in the sections below.

When the data extraction is finished, a data reduction procedure is executed in order to delete redundant features and thus to improve both the classifier performance and the time required for training. By deleting correlated features the database $data_1$ is obtained; $data_1$ is then further processed to get the most significant features through supervised and unsupervised methods for feature selection $data_2$,.., $data_n$. The data sets are evaluated by ANN classifiers, that are basically divided into three types; (1) classic multi-output ANN classifiers, with supervised and unsupervised learning, (2) binary ANN classifiers using the One Vs All approach and (3) ANN with GA, the optimized structure of the ANN and the most significant features are used.

ANNs are sensitive to initial conditions (e.g., initial weights), and each time we run an ANN classifier we may end up with different results. For that reason, we evaluate the ANN k times and obtain a confusion matrix in each run. Once the classification phase finishes, we get as a result k*n confusion matrices with n the amount of classifiers established for the experiment. Metrics such as accuracy (ACC) and AUC are computed now. As a final step, statistical tests are used to compare the results obtained with the ANNs. The classification and comparison phases (A) and (B) are going to be detailed in the following items.

3.1. Classification phase

The ANNs, A_i with $i = \{1, .., n\}$, are trained with the data sets $data_1$,…,$data_m$ obtained, respectively. In order to deal with the bias/variance tradeoff in ANNs, we used the cross-validation technique to train the models with random partitions of 10-fold

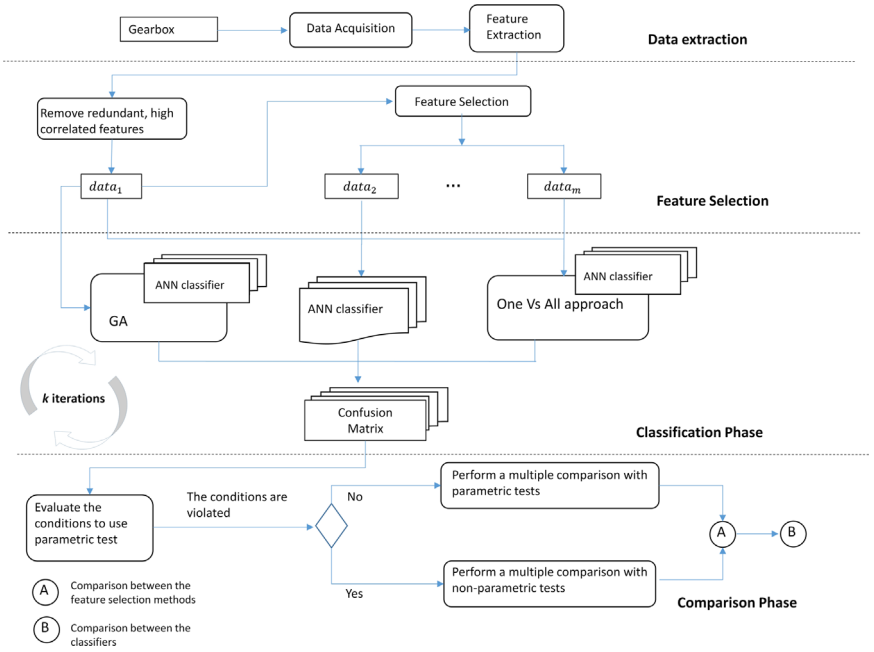


Fig. 1. Activity Diagram for the experimental framework.

over the observations. Each experiment is repeated 30 times; a confusion matrix being computed each time. The accuracy (ACC) and the area under the curve (AUC) of the Receiver Operating Characteristic (ROC) curve is obtained, for each experiment.

3.2. Comparison phase

This phase is in charge of performing multiple comparisons between the results obtained in the classification phase. Statistical tests are incorporated to perform a robust comparison among the classifiers. Basically, these procedures define a null hypothesis stating that the results under evaluating come from the same distribution. Statistical tests can be either parametric or non-parametric. The parametric tests are commonly used in computer science, but usually they are not suitable when underlying assumptions are violated; in that case non-parametric test should be used.

The aim of this phase is (a) to identify which feature selection method provides the best results in term of ACC and AUC for the ANNs and (b) to detect the ANNs that get the best ACC and AUC using the previous results to the comparison. These objectives will be achieved through the activities (A) and (B) respectively, in Fig. 1. The activities in (A), for the multicomparison between the feature selection methods, are listed as follows:

- Let O_{ij} be a vector with the results of the classifier i using the method M_j for data reduction.
- As a first step, for each classifier A_i the data reduction that gets the best ACC and AUC is going to be detected using either parametric or nonparametric tests. This is achieved using the procedure detailed in the activity diagram in Fig. 2.
- A statistical test for multicomparison is going to be applied for each A_i, where the means of comparison will be the data reduction methods M_j; if there is a statistical difference between them, a post hoc procedure is applied to detect the pair(s) of methods that are different $P_{ik} = \{M_j, M_m | j \neq m\}$, k is the number of dissimilar pairs of A_i.
- P_{ik} goes through a pairwise comparison allowing the winner method in the pair(s) to be detected. A ranking process is executed in order to know how many times a method M_j was a winner; the method that got the highest score is considered as the best M_{ibest}, which in turn is associated to a set of observation O_{ibest} for the classifier i.

Once M_{ibest} is detected for each classifier, that information feeds back the next activities in (B) to identify the best ANN. This task is addressed using the activities in Fig. 3, and the steps described in Fig. 2 are mostly repeated with the difference that the observations O_{iBest} are used by the comparison and the results given by GA-ANN approach; this last one assuming that the best data reduction was chosen to the ANN with the GA process. Finally, only one result is obtained A_{Best}, either for ACC or AUC.

The activities proposed can be used with parametric or non-parametric tests depending on the conditions of the study problem. The methods needed by the activities in Figs. 2 and 3; i.e, multicomparison tests, pairwise comparison test and the post-hoc procedures; are briefly presented in the following section.

4. Data extraction and analysis

4.1. Test rig and experiment description

To generate as realistic as possible experimental conditions, the test rig presented in Fig. 4 was set up. A three phase motor generates the rotation motion. The motor was used with an adjustable-speed drive to generate different speeds. The motion is transmitted to the gearbox, which in turn is produced and transmitted to a pulley. The pulley is part of a magnetic brake control, which has under its control the load regulation. An accelerometer (Digivive MX 300) is placed vertically in the gearbox case; and a data acquisition (DAQ) device collects and sends the vibration signals to a computer.

Different faults are established for the experimental test, each defining a gearbox state; The different states are summarized in Table 1, and Fig. 5 presents some of the faults set up for the experiment. Table 2 shows the test experimental settings, the setting established allows for obtaining a variety of scenarios in the test bed. Basically, the experiments were performed with three loads and six velocities, in each of them five samples were collected; the experiments were symmetric in all the scenarios. This is going to allow balancing the bias/variance in the classifiers tested.

Experimental data is acquired for each one of the gearbox conditions described in Table 1 resulting in a data set with 900 vibration signals.

4.2. Feature extraction

The vibration signals are processed in order to extract the features that describe the condition states.

1. *Time domain*: Root mean square (RMS), crest factor, mean, standard deviation, variance and skewness were obtained over all the signal length. The mean of a vibration signal ($i = 1, \ldots, 900$) is defined as the sum of all its amplitudes x_{ij} ($j = 1, \ldots, N$) divided into the number of samples N, as described in Eq. (3). Following the same idea, the variance is computed

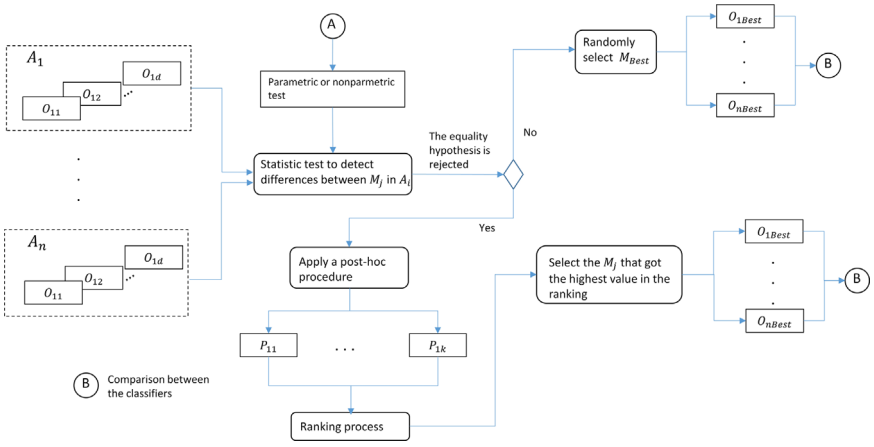


Fig. 2. Activity Diagram for the multicomparison regarding the data reduction methods.

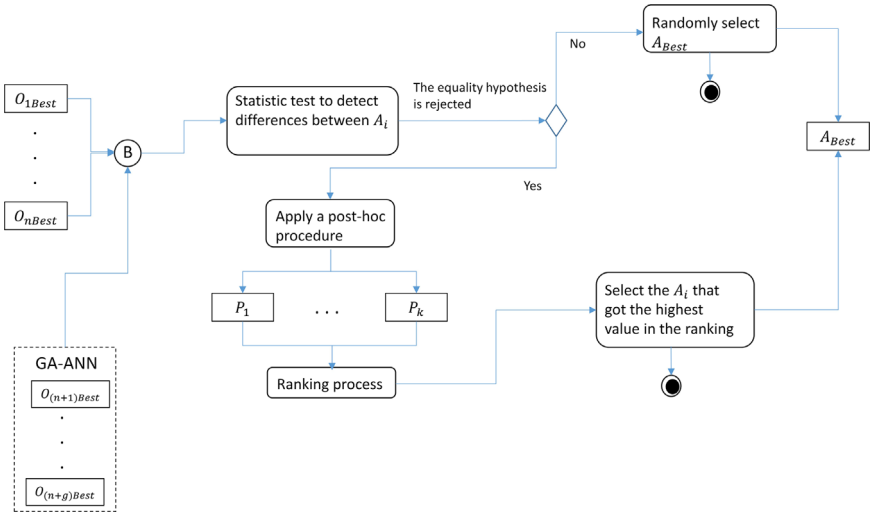


Fig. 3. Activity Diagram for the multicomparison regarding the ANNs.

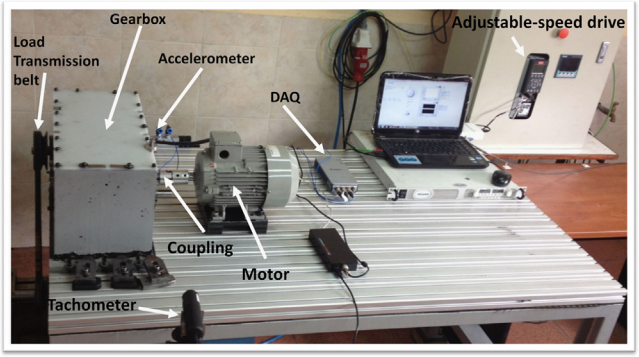


Fig. 4. Experimental test rig.

Table 1
Gear faults conditions.

Label	Description
f1	Healthy pinion, healthy gear
f2	Tooth pinion chaffing, healthy gear
f3	Tooth pinion wear, healthy gear
f4	25% tooth pinion breakage, healthy gear
f5	50% tooth pinion breakage, healthy gear
f6	100% tooth pinion breakage, healthy gear
f7	Healthy pinion, 25% gear crack
f8	Healthy pinion, 100% gear crack
f9	Healthy pinion, 50% gear chaffing
f10	25% tooth pinion breakage, 25% gear crack

using Eq. (4); and the standard deviation is defined by square root of the variance:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} x_{ij} \tag{3}$$

$$\rho_i^2 = \frac{1}{N}\sum_{j=1}^{N} (x_{ij}-\mu_i)^2 \tag{4}$$

RMS is defined by Eq. (5). The crest factor (cf) is the maximum peak of the signal divided by RMS. The skewness metric allows detecting if there is not symmetry between the signal samples with a center point as reference, and is given by Eq. (6).

$$RMS_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N} (x_{ij})^2} \tag{5}$$

$$S_i = \frac{N\sum_{j=1}^{N}(x_{ij}-\mu_i)^3}{\rho^3} \tag{6}$$

We also included as indicators of the machinery state the kurtosis k defined as,

$$k[x_i] = \frac{N\sum_{j=1}^{N}(x_{ij}-\mu_i)^4}{\left[\sum_{j=1}^{N}(x_{ij}-\mu_i)^2\right]^4} \tag{7}$$

and the kurtosis of the derivative of the acceleration of the signal $a(x)$, that is,

$$kda[x_i] = k\left\{\frac{d}{dt}a(x_i(t))\right\} = k\left\{\frac{d^3}{dt^3}x_i(t)\right\} \tag{8}$$

2. *Frequency domain*. The vibration signals are transformed into frequency signals using Fast Fourier Transform (FFT). The results are divided into bands and the RMS, mean, standard deviation, and kurtosis are calculated for each band. 730 features were obtained with this analysis.
3. Time-frequency domain Wavelet analysis in gear fault detection and diagnosis are widely used in this field [34]. The time raw vibration signals are decomposed using the Wavelet Packet Transform (WPT). Five mother wavelets are considered for this analysis: Daubechies (db7), Symlet (sym), Coifier (coif4), Biorthogonal (bior6.8) and Reverse Biorthogonal (rbior6.8). The wavelet decomposition was performed up to four levels for each mother wavelet, then 2^4 coefficients are obtained for each one. Finally, 80 features associated to the energy parameter are extracted.

We get as a result 817 features and 900 samples from the feature extraction process.

5. Classification phase

The experimental setup is divided into three approaches to train some of the ANNs chosen by our proposal. The first one is training and creating ANNs, where it is previously defined all the parameters needed to obtain their models such as hidden layers and training parameters. The second one is using One Vs All strategy to get a multi-classification process with single output NN. Finally, we use GA to automatically define parameters to create an ANN and reduce features at the same time given a database. The parameters established for each strategy are described in the following items.

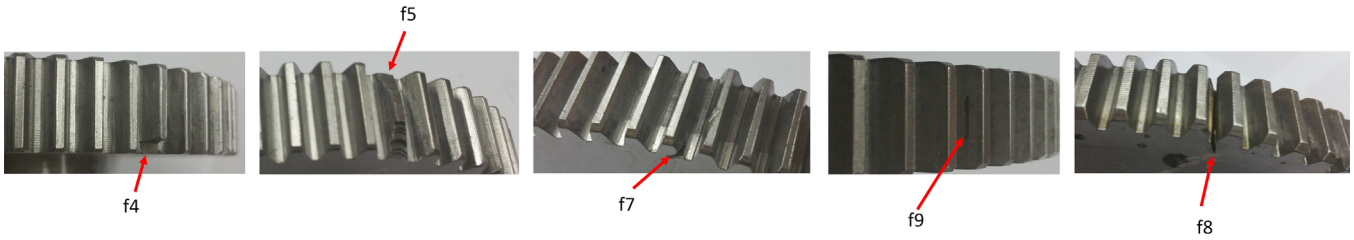


Fig. 5. Gear with the faults $f4$, $f5$, $f7$, $f8$ and $f9$ induced to the test rig.

Table 2
Test rig's experimental settings.

Parameter	Value
Sampling frequency (kHz)	50
Length of each sample (s)	10
Number of tests	5
Rotation Frequency (constant speed) (Hz)	8, 12, 15
Range Frequency (variable speed) (Hz)	5–12 , 12–18, 8–15
Load	No Load, 10 V, 30 V

5.1. ANN

This is the simplest way to train an ANN, where all its parameters are previously defined, e.g., the number of hidden layers, the activation function, the learning process, among others. Table A1, Appendix A shows the ANNs chosen for this phase and the parameters defined. Pattern recognition network, denoted as FFPN, is feedforward network that can be trained to classify inputs according to target classes. It is important to mention that the hidden layers of FFNN, CFN and FFPN were selected empirically through several tests until the configuration gave one of the best results for the dataset. FFNN, CFN and FFPN will be trained by two algorithm, that is gradient descent (GD) and scaled conjugate gradient (SGD). On the other hand, RBFN was selected with its simplest structure. A PNN with a radial basis function to compute the distance is used and it will be denoted as RBFN-PNN. Finally, a SOM is also tested.

5.2. One Vs All approach with ANN

The One Vs All strategy in classification problems consists of creating one model per class existent in the database. Each model m_i is trained with all the examples belonging to the class c_i, they are labeled as positive and all the remaining examples are labeled as negative in the training phase.

When a new example x needs to be evaluated by this strategy, the class given for each model m_i is obtained; then a voting process defines to which class belongs the input as it is described in Eq. (9).

$$\hat{Y}(x) = majority\ vote\{m_i(x) | i = 1, 2, \ldots, k\} \tag{9}$$

This particular test uses One Vs All approach with ANNs as base classifiers. We use the FFNN, CFN and FFPN as base classifiers for this study with the architectures described in Table A1 for each one. The training method for all the cases is SGD.

5.3. ANN-GA based model

The experiments addressed with GA process are applied to build and select features using a FFNN, CFN and FFPN. Additionally, One Vs All approach is implemented with GA using FFNN. Some of the parameters defined for each test are detailed as follows:

- Fitness function: the ANN either FFNN, FFPN and CFN; or the ANNs defined for One Vs All approach.
- Fitness: Performance of the ANN.
- Population type: It is a string describing the data type of the population and it is defined as *Bit string*.
- Population size: 50.
- Creation function: It handles the function that creates the initial population, this was defined as an uniform distribution.
- Population initial range: It is the range of the uniform function defined previously. The range was established between [0;1].
- Mutation function: It produces mutation children, and it was set as uniform.

6. Parametric and nonparametric tests for multicomparison

When a new classifier is presented, it is important to show its goodness against classic and novel algorithms. One way to compare different algorithms is contrasting their results in terms of accuracy or error, but sometimes these measures are not adequate. Statistical tests have showed to improve the evaluation of new algorithms, and they are commonly included in the experimental results to underline and analyze which algorithm is better than another.

Statistical tests are divided into two main groups denoted as parametric and non-parametric tests, and their usage depends on the problem conditions. Parametric tests are the most common techniques in computer science, but they are based on several assumptions that must be fulfilled in order to use them. This trend is not a good option for those algorithms leaded by random processes. On the other hand, non-parametric tests are used when parametric assumptions are violated [35].

6.1. Parametric tests

Parametric tests are comprehensive tools to contrast a set of observations in a study problem, but they are usually subject to three specific conditions: (1) Independence, that is two events are independent if the occurrence of one of them does not modify the probability of the occurrence of the other one, (2) Normality, which is achieved when a collected data follows a normal distribution with mean μ and variance ρ, and (3) Homoscedasticity, i.e., equal variance for distributions in analysis.

For comparing multiple distributions, one of the most popular parametric procedure is the Analysis of Variance (ANOVA). The difference between two or more related sample means is obtained with the repeated measures ANOVA. In ANOVA a null-hypothesis is defined, that states all the algorithms are equivalent; so a rejection of this hypothesis implies differences among the algorithm performances [36].

6.2. Nonparametric tests

Nonparametric tests, for performing comparisons between a set of observations, have been recently incorporated to several works in order to present a robust comparison within new-classic or classic-classic algorithms [36–39]. In [28] a complete study of non-parametric tests for pairwise and multiple comparisons is reported for multi-problem analysis. The work aims to give some guidelines to:

(1) Use nonparametric tests through several examples and (2) Select correctly the adequate procedure regarding the problem study. There are two types of comparisons that can be performed through this approach: (1) pairwise comparison, this is an intrinsic comparative statistical procedure between two algorithms and (2) multiple comparison, that is either a contrast within an algorithm against a group of algorithms *N*, or a comparison of all versus all the algorithms.

The pairwise comparison result is easier to analyze. On the other hand, in multiple comparison the result analysis is more complex as a post hoc procedure is applied in order to discriminate the differences among the algorithms evaluated. The diagram in Fig. 6 shows the types of comparisons performed with non-parameter testing as used in this study [40].

The post hoc procedures are posteriori tests used to detect pairwise differences, that were not detected by the multi comparison method. For example, the multiple comparison with nonparametric test (e.g., performed by the Friedman test) can only detect a significant difference between the observations through the null hypothesis rejection, but it does not contribute with more information [28].

Particularly, our approach verifies if the conditions to use parametric tests are fulfilled with Shapiro–Wilk test for data normality, and Levene's test for homoscedasticity. If the conditions to use parametric tests are violated, the Friedman test is applied; and the post-hoc procedures selected are Holm and Shaffer tests.

7. Results

This section present the results in the following order: Section 7.1 reports the feature selection performed over the original data for fault diagnosis in gearboxes. In Section 7.2 the classification results of the ANNs, evaluated over the feature selections are shown. Section 7.3 details the conditions needed to use parametric tests to compare the results obtained in the previous section. Finally, if those conditions are violated, in Section 7.4 the nonparametric tests are evaluated to detect the best feature selection methods to each ANNs; and the best algorithm is identified considering the previous results.

7.1. Feature selection

7.1.1. Redundant highly correlated features

The database has 817 features obtained in the feature extraction phase. In Section 4, it was established that the feature extraction is performed computing statistics, that come from vibration analysis on time domain, frequency domain and time-frequency domain. Therefore, it is likely to find redundant high correlated features. The correlation analysis was executed over the original dataset getting as a result a high correlation within more than 400 features. Each and every feature exhibiting a correlation higher than 95% with another one is remove, and this analysis ended up with a database $data_1$ with 330 features and 900 samples.

7.1.2. Feature selection by supervised learning approaches

1. *Random forest*: An RF with 500 trees was trained with the noncorrelated database $data_1$, and the information degree was computed using the entropy metric. The average information contributed by each feature is obtained and ranked. The feature selection is executed fixing two thresholds on higher than 40% and 50%. We obtained two new databases, $data_2$ with 29 features and $data_3$ with 14 features.
2. *Linear Discriminant Analysis*: LDA was applied over $data_1$, getting as a result a new dataset $data_4$ with 9 features.

7.1.3. Feature selection by unsupervised learning approaches

1. *Principal Component Analysis*: The result after using PCA over the 330 features leads to obtain 14 principal components $data_5$, with 98% of data variability.
2. *Non-negative matrix factorization*: It was applied over the noncorrelated dataset converging into a reduced dataset $data_6$ with 35 features.
3. *Most neighbors first*: The entire feature space is split into *N* disjoints sets A_i. Then, MNF is applied over each subset and a group of features are selected from each subset, these last ones become a partial result for the following level of the hierarchy. This process is repeated until a desired hierarchy level is reached N_{it}. This modification to the original MNF algorithm, is proposed in order to improve the search procedure of the

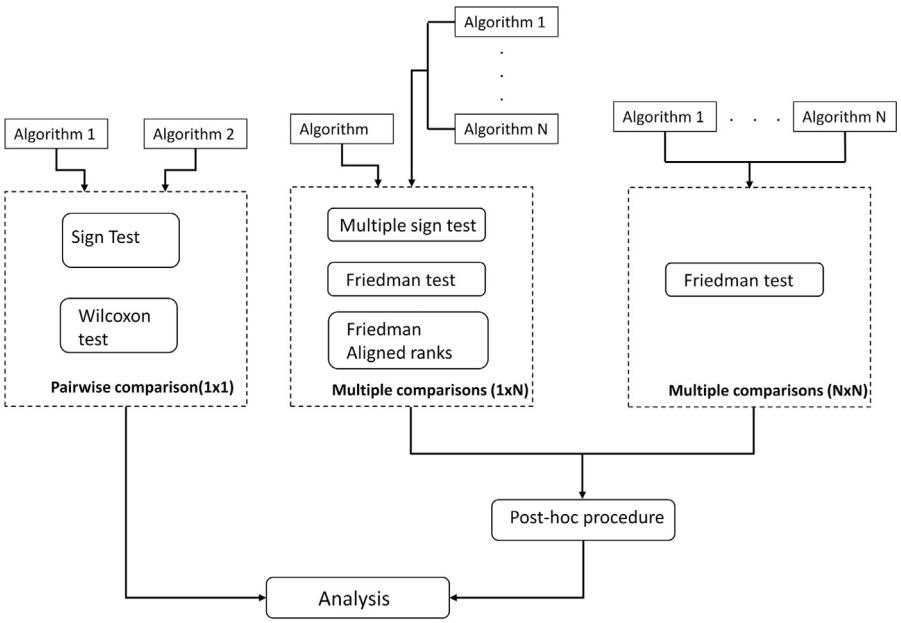


Fig. 6. Nonparametric procedures for comparing algorithms.

significant features regarding the original set [41]. For this particular case study, MNF was executed by the hierarchical approach with six levels, using the noncorrelated dataset with 330 features to start the reduction. The number of features NA_i obtained for each level were $NA_1 = 144$, $NA_2 = 82$, $NA_3 = 62$, $NA_4 = 54$, $NA_5 = 42$ and $NA_6 = 24$. The reductions chosen by this proposal correspond to level 2 and level 5, $data_7$ with 82 features and $data_8$ with 42 respectively.

7.2. Classification performance

The ANNs defined in Section 5 are trained with the feature selection performed. Consequently, each algorithm has 30 confusion matrices related to the iteration k, for the 8 databases. The ACC and AUC are computed with the confusion matrices, getting as a result a set of vectors that are going to be denoted as O_{ij}; for the algorithm A_i with the feature selection method M_j.

The ACC results, of each ANN over only one database, are represented by a Q–Q plot; in order to see how the 30 tests are distributed in contrast with an expected normal distribution. A Q–Q plot is a graphical method used to find differences between the probability distribution of a population with random samples, and another known distribution.

The database used was the noncorrelated data with 330 features, due to the rest of the reductions originally come from this one in particular. Fig. 7 shows the Q–Q plots of FFNN, CFN and FFPN with both GD and SGD for training, Fig. 8 RBFN, RBFN-PNN and SOM results, and finally Fig. 9 the ANNs-GA based model results. One Vs All approach, with three ANNs as base classifiers, reports similar results to Fig. 7. These graphs allow detecting the differences between an expected normal distribution with the resulting observations in 30 iterations. According to Fig. 7(c), CFN-SGD results are not normally distributed, because the observations do not follow the expected normal line; in contrast, in Fig. 7(d), CFN-GD results are more likely to be normally distributed due to the distribution of the observations are similar to the normal line. This behavior is repeated for the rest of the cases; subsequently, expecting to perform a comparison regarding the ACC means of the results is incorrect for this particular case, because they do not contribute with important information about the differences between the ANNs.

In the following sections, the conditions to use parametric tests are evaluated and if they are not fulfilled, the nonparametric tests for multiple comparison are applied over the results.

7.3. Multiple comparison of the classifiers: parametric tests

As was detailed in Section 3 Fig. 1, the study of conditions to use parametric tests must be satisfied in order to compare the ANN algorithms; to accomplish that, we present the normality and the homoscedasticity tests applied over the classification phase results.

7.3.1. Normality

In order to test if the data is normally distributed, the Shapiro–Wilk test is evaluated. This test compares the variance of the data, which is estimated, with the variance expected by a normal distribution. The null hypothesis states the equally of variances, when it is rejected the normality condition fails [42].

This test was applied over the accuracy obtained in the 30 runs, the results are depicted in Table 3. The Shapiro–Wilk test shows that the majority of the experiments performed with the ANNs selected do not fulfill the normality condition, because they are lower than the level of significance $\alpha = 0.05$. In Table 3, the algorithms that do not violated the normality condition are standing out in blue and italic text. Classic ANNs (FFNN, CFN and FFPN) trained with GD fulfilled the normality condition in almost all the

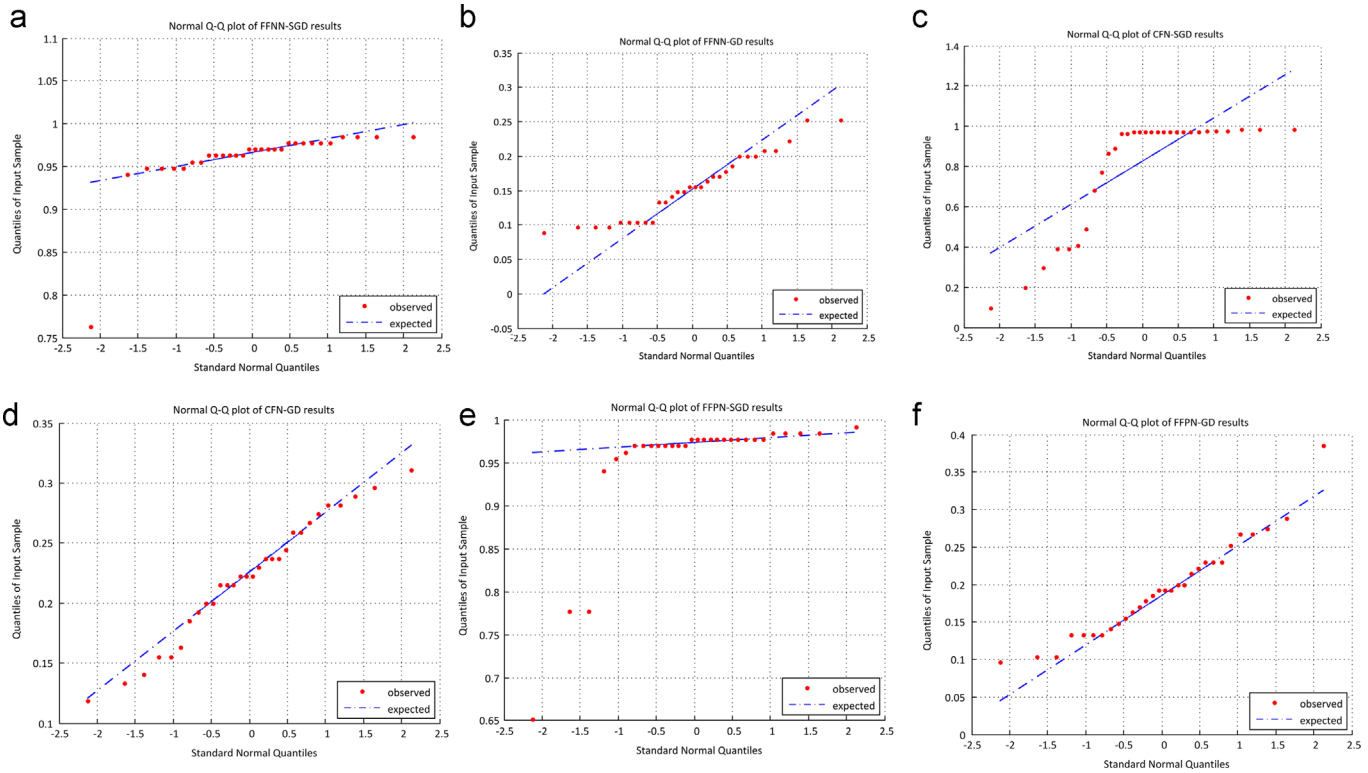


Fig. 7. (a) Q–Q plot of FFNN-SGD results using ACC.(b) Q–Q plot of FFNN-GD results using ACC. (c) Q–Q plot of CFN-SGD results using ACC. (d) Q–Q plot of CFN-GD results using ACC. (e) Q–Q plot of FFPN-SGD results using ACC. (f) Q–Q plot of FFPN-GD results using ACC.

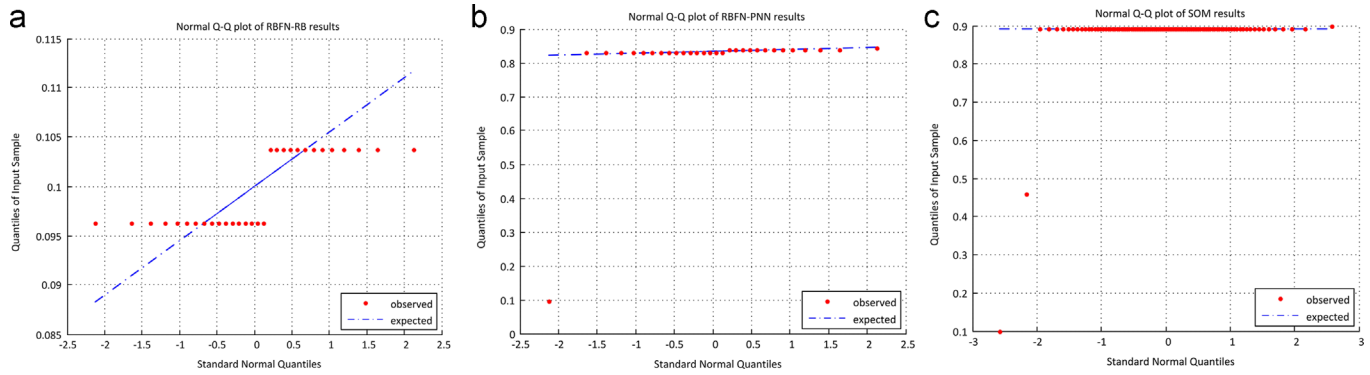


Fig. 8. (a) Q–Q plot of RBFN-RB results with ACC. (b) Q–Q plot of RBFN-PNN results with ACC. (c) Q–Q plot of SOM results with ACC.

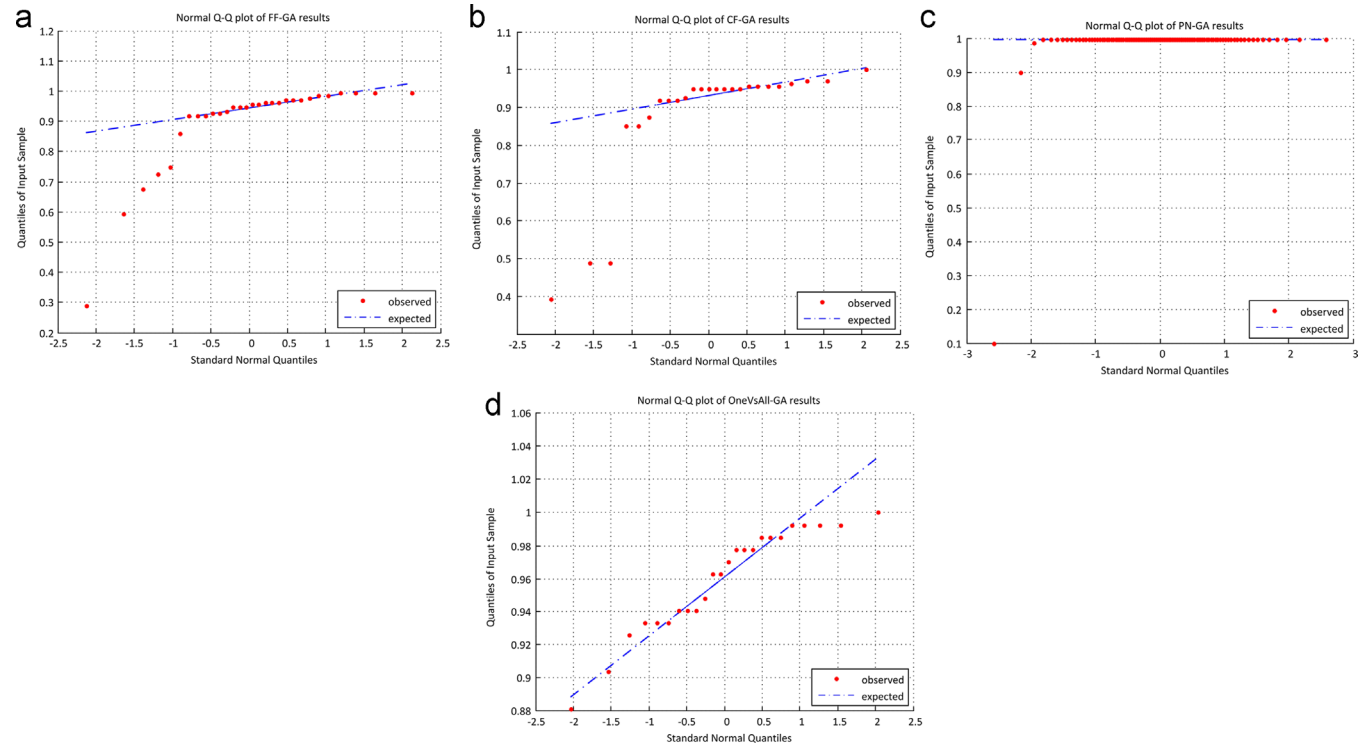


Fig. 9. (a) Q–Q plot of GA results using FFNN-SGD. (b)Q–Q plot of GA results using CFN-SGD. (c) Q–Q plot of GA results using FFPN-SGD. (d) Q–Q plot of GA results using One Vs All approach and FFNN-SGD as base classifier.

Table 3
Normality condition using Shapiro-Wilk test, with $\alpha = 0.05$.

Classifier	Corr	RF		LDA	PCA	NMF	MNF	
	330Att	29Att	14Att	9Att	14Att	35Att	82Att	42Att
FFNN-SGD	7.3750e−09	4.8380e−09	2.3355e−09	9.7249e−08	1.9972e−05	4.6807e−08	2.3363e−07	2.3633e−09
FFNN-GD	*0.0774*	*0.0532*	*0.1219*	*0.0664*	*0.1388*	*0.3801*	*0.9226*	0.0178
CFN-SGD	2.4093e−05	2.3186e−07	1.2751e−07	9.7249e−08	1.1821e−04	0.0423	*0.3128*	0.0011
CFN-GD	*0.6060*	*0.4138*	*0.0860*	*0.1698*	*0.1963*	*0.7606*	0.0194	0.0214
FFPN-SGD	2.0208e−08	2.0085e−06	3.5361e−06	1.8416e−04	8.4631e−05	5.2787e−07	2.2164e−08	1.8772e−06
FFPN-GD	*0.1231*	0.0131	*0.0836*	*0.6447*	*0.4629*	0.0477	0.0010	0.0098
One Vs All(FFNN)	1.1024e−08	1.9925e−09	1.3318e−08	1.3563e−08	3.9365e−07	8.4101e−07	1.8939e−08	1.8939e−08
One Vs All(FFPN)	9.9939e−09	2.0085e−06	3.5361e−06	1.0095e−06	8.4631e−05	6.1254e−07	6.3246e−08	2.7017e−08
One Vs All(CFN)	1.1024e−08	1.9925e−09	1.3318e−08	1.2353e−08	5.0470e−08	0.0188	*0.0925*	1.1689e−04
RBFN-PNN	0.0000	0.0000	*0.0987*	0.0000	0.0020	0.0002	0.0000	0.0000
RBFN-RB	0.0000	*0.5888*	*0.6924*	0.0000	*0.3821*	*0.2775*	*0.0674*	*0.0558*
SOM	*0.6326*	*0.4578*	*0.0063*	8.1406e−04	2.5807e−10	*0.9777*	*0.1491*	*0.9572*

databases used. The SOM algorithm presents five cases where the hypothesis is accepted, with different databases for each case; in contrast the hypothesis was rejected with LDA, PCA and RF with 9, 14 and 14 features respectively. On the other hand, the hypothesis is rejected with the noncorrelated data and the data obtained with LDA using RBFN-RB as classifier.

Table 4
Normality condition using Shapiro-Wilk test for ANN-GA based models, with $\alpha = 0.05$.

	FFNN-GA	FFPN-GA	CFN-GA	One Vs All(FFPN)-GA
p-value	2.8708e−08	6.9934e−07	1.2810e−06	0.0427

Table 5
p-values of the Levene's Test applied in the case study.

330 Att/Corr	29 Att/Entropy	14 Att/Entropy	9 Att/LDA	14 Att/PCA	35 Att/NMF	42 Att/MNF	GA
∗(0.0000)	∗(0.0000)	∗(0.0000)	∗(0.0000)	∗(0.0000)	∗(0.0000)	∗(0.0000)	∗(0.0000)

Table 6
Friedman's test of each ANNs regarding the ACC and AUC results with the different data reduction methods.

Algorithm	*p*−value	
	ACC	AUC
FFNN-SGD	4.758e−11	8.192e−11
FFNN-GD	5.989e−6	5.911e−7
FFPN-SGD	4.604e−11	5.123e−11
FFPN-GD	8.910e−11	4.637e−10
CFN-SGD	1.703e−9	2.325e−7
CFN-GD	9.473e−11	1.232e−10
One Vs All(FFNN)	3.234e−4	1.364e−4
One Vs All(FFPN)	8.505e−1	6.219e−11
One Vs All(CFN)	5.260e−4	5.260e−4
RBFN-PNN	6.285e−11	6.285e−11
RBFN-RB	5.186e−11	7.038e−11
SOM	1.042e−10	1.177e−10

The normality condition was also tested for the ANNs optimized by GA. The results in Table 4 report that none of the GA approaches follows a normal distribution; this can be verified graphically in the *Q*–*Q* plots in the previous section.

7.3.2. Homoscedasticity

Most parametric tests are based on the assumption that samples are drawn from normally distributed populations with equal variance (homoscedasticity). These assumptions can be verified by a number of tests such as the Shapiro–Wilk test for normality and the Levene's test for homoscedasticity. Applying the median version of the Levene's test to the $p_{\text{Leven}} = 0$ was obtained revealing that no significant difference exists among variances. However, applying the Shapiro–Wilk test to the same data reveals that data are not normally distributed ($p_{\text{Shapiro-Wilk}} = 0$).

The Levene's test results applied over the case study are depicted in Table 5. The null hypothesis was rejected in all of the cases. Consequently, there is a difference between the variances in the population, and the parametric tests cannot be used in the following items to perform a comparison between the algorithms.

7.4. Multiple comparison of the classifiers: non-parametric tests

7.4.1. Comparison between the feature reduction methods

In order to find the best feature selection methods for each ANNs proposed, the steps described in Section 3.2 are executed, and their results are summarized as follows:

- *Statistic test to detect differences between the methods*: Friedman test is applied over each one of the 6 feature selection methods described in Section 1.5 as the sole varying factor. The *p*-values obtained when comparing 30 independent runs of each method are presented in Table 6. The Friedman test gave as results the *p*-values in Table 6. It is important to mention that this test was evaluated using both ACC and AUC. Table 6 shows that the hypothesis of equality is rejected for all the data reduction methods with a *p*-value lower than $p_F = 0.05$.
- *Post hoc procedure*: The next step is to detect which of the feature selection methods present difference between them. A post hoc procedure is adequate for this task, and allows obtaining the pair(s) different. Likewise, the post-hoc procedure is applied over the ACC and AUC results for each ANNs. Appendix B shows some of the results after being applied Holm's and Shaffer's test. The pairs of methods, with a *p*-value lower than the adjusted *p*-value given by Holm's test, are taken into account to the next step and they standout in black text. As an example, in Appendix B Table B1, the pairs of feature selection methods studied to FFNN-SGD using ACC, where 16 null hypothesis were rejected and are used to the next phase; likewise Table B2 shows the results using AUC with 11 null hypothesis rejected.
- *Ranking process*: The resulting pairs of the previous step are evaluated by a pairwise nonparametric test, specifically the Wilcoxon test. Thus allows detecting which method (winner) in the pair outperforms the other one. A ranking process is executed, counting the number of times that a method was a winner in the pairwise comparison. This process was repeated for all the ANNs with ACC and AUC, the results are depicted in Tables 7 and 8. The method M_{iBest} that got the highest ranking in the table is selected. In case that more than one method gets the highest value in the raking, these are all listed in Table 9. As an example in Table 7, for FFNN-SGD its best feature selection methods were RF, LDA and PCA methods and the noncorrelated dataset with 330 features; on the contrary, for One Vs All approach using FFNN as base classifier the best method is RF with 29 features.

7.4.2. Comparison between the ANNs

Each ANNs A_i, with its best data reduction observations O_{iBest}, is organized in order to perform a multiple comparison, as was detailed in Section 3.2, Fig. 3. It is important to mention that the GA-ANN results are included in this comparison, as on the GA process attempts to simultaneously accomplish feature reduction and ANN structure optimization. The Friedman test is evaluated and the resulting *p*-values for both ACC and AUC are presented in Table 10. The hypothesis of equality is rejected in both cases.

Afterward, a post hoc procedure is applied in order to detect the pair of algorithms that are different. The Holm's and Shaffer's results are obtained, for ACC and AUC, respectively. 52 pairs do not fulfill the equality hypothesis using ACC, while 62 hypothesis were rejected using AUC. In the final analysis, a ranking process is executed after applying a pairwise comparison between the pairs that differ. In Table 11, the number of

Table 7
Rank values achieved by the multiple pairwise comparisons of the feature selection methods that are statistically different, using ACC.

Classifier	Corr	RF		LDA	PCA	NMF	MNF	
	330Att	29Att	14Att	9Att	14Att	35Att	82Att	42Att
FFNN-SGD	*2*	*2*	*2*	*2*	*2*	0	0	0
FFNN-GD	1	3	*4*	1	*4*	0	0	0
CFN-SGD	0	0	*3*	2	*3*	0	0	2
CFN-GD	*5*	*5*	*5*	1	3	0	0	0
FFPN-SGD	*5*	1	0	*5*	0	0	0	0
FFPN-GD	2	0	2	*4*	1	0	0	0
One Vs All (FFNN)	0	*2*	1	0	0	0	0	1
One Vs All (FFPN)	*5*	1	*5*	*5*	1	0	0	1
One Vs All (CFN)	0	*1*	*1*	0	0	0	0	1
RBFN-PNN	2	*6*	2	1	4	0	0	1
RBFN-RB	1	2	*4*	0	3	1	0	3
SOM	0	4	*5*	0	*5*	0	2	3
Average	1.9166	2.25	2.8333	1.75	2.1666	0.08333	0.1666	1

Table 8
Rank values achieved by the multiple pairwise comparisons of the feature selection methods that are statistically different, using AUC.

Classifier	Corr	RF		LDA	PCA	NMF	MNF	
	330Att	29Att	14Att	9Att	14Att	35Att	82Att	42Att
FFNN-SGD	*2*	*2*	*2*	*2*	*2*	0	0	0
FFNN-GD	2	2	*3*	1	2	0	0	0
CFN-SGD	0	0	*2*	*2*	1	0	0	2
CFN-GD	0	0	0	*5*	*5*	3	3	3
FFPN-SGD	*3*	1	0	*3*	0	0	*3*	*3*
FFPN-GD	*4*	1	1	*4*	0	0	0	0
One Vs All (FFNN)	1	*2*	0	1	0	0	0	*2*
One Vs All (FFPN)	*6*	1	*6*	1	1	0	0	1
One Vs All (CFN)	0	*1*	*1*	0	0	0	0	*1*
RBFN-PNN	2	*6*	2	1	4	0	0	1
RBFN-RB	0	2	*4*	0	*4*	1	1	3
SOM	0	*5*	*5*	2	*5*	0	2	2
Average	1.6666	1.9166	2.1666	1.8333	2	0.3333	0.75	1.5

Table 9
Best feature selection methods for each classifier.

Algorithm (A_i)	M_{iBest}	
	ACC	AUC
FFNN-SGD	Corr-330Att, RF-14Att, RF-29Att, LDA-14Att and PCA-14Att	Corr-330Att, RF-14Att, RF-29Att, LDA-14Att and PCA-14Att
FFNN-GD	RF-14Att and PCA-14Att	RF-14Att
CFN-SGD	RF-14Att and PCA-14Att	RF-14Att, PCA-14Att and MNF-42Att
CFN-GD	Corr-330Att, RF-14Att and RF-29Att	LDA-9Att and PCA-14Att
FFPN-SGD	Corr-330Att and LDA-9Att	Corr-330Att, LDA-9Att, MNF-82Att and MNF-42Att
FFPN-GD	LDA-9Att	Corr-330Att and LDA-9Att
One Vs All (FFNN)	RF-29Att	RF-29Att and MNF-42Att
One Vs All (FFPN)	Corr-330Att, RF-14Att and LDA-9Att	Corr-330Att and RF-14Att
One Vs All (CFN)	RF-29Att and RF-14Att	RF-14Att, RF-29Att and MNF-42Att
RBFN-PNN	RF-29Att	RF-29Att and PCA-14Att
RBFN-RB	RF-14Att	RF-14Att and PCA-14Att
SOM	RF-14Att and PCA-14Att	RF-14Att, RF-29Att and PCA-14Att

Table 10
Ranks achieved by Friedman test and p-values for ACC and AUC.

Algorithm	Ranking	
	ACC	AUC
FFNN-SGD	7.7	3.56
FFNN-GD	15.3	14.7
CFN-SGD	5.55	4.83
CFN-GD	13.23	14.7
FFPN-SGD	5.85	2.46
FFPN-GD	15.53	15.0
OA-FFNN	7.93	8.35
OA-CFN	6.44	7.71
OA-FFPN	3.31	4.28
RBFN-PNN	7.23	7.65
RBFN-RB	13.03	12.73
SOM	3.98	9.68
GA-FFNN	7.78	8.01
GA-CFN	9.26	8.01
GA-FFPN	7.78	8.01
GA-OA	6.04	6.26
p-value	1.167e − 10	1.707e − 10

Table 11
Rank values achieved by the multi pairwise comparisons of the ANNs, using ACC and AUC.

Algorithm	Ranking	
	ACC	AUC
FFNN-SGD	4	***10***
FFNN-GD	0	0
CFN-SGD	4	5
CFN-GD	0	0
FFPN-SGD	4	*11*
FFPN-GD	0	0
OA-FFNN	4	4
OA-CFN	4	4
OA-FFPN	*9*	5
RBFN-PNN	4	4
RBFN-RB	0	0
SOM	***5***	3
GA-FFNN	4	4
GA-CFN	2	4
GA-FFPN	4	4
GA-OA	4	4

times that an ANN was winner is counted. The best ANN is standing out in blue and italic text. On the other hand, the second best is in bold and italic text.

8. Discussion

Several ANNs were selected, and their performance were evaluated and compared in a realistic fault classification problem. A gear fault data set was extracted and preprocessed, then a feature selection phase was performed and evaluated. The results presented, after the databases were evaluated by the classifiers, represent a set of observations given by the ACC and AUC metrics for each ANN. In Section 7.2, it is graphically noticeable that with a mere comparison of the means and standard deviations of the results, we cannot conclude about the performance of the ANNs; actually most of the pairs (ANN, learning method) yield results that are not normally distributed. Comparisons are carried out using statistic tests following the activities proposed in Section 3. In Section 7.3 the conditions to use parametric test are evaluated. Only a very limited number of pairs, i.e, FFNN-GD, CFN-GD, FFPN-GD, SOM and RBFN-RB, showed to be normally distributed with some specific data sets. On the other

hand, when the homoscedasticity condition was evaluated none of the pairs fulfilled it. In consequence, nonparametric comparisons allowed identifying the best data reduction for each classifier. In Section 7.3 it is evident that the supervised based feature selection methods (RF and LDA) outperformed the unsupervised ones, RF being the technique that was selected as a winner 8 times out of 12. However, the unsupervised based method PCA showed good results in the ranking process.

Finally, in Section 7.4, the multicomparison between the ANNs was performed using the information obtained in the previous section. The nonparametric process proposed to detect the best algorithm was applied as a result of two best algorithms: (1) FFPN-SGD using ACC and (2) One Vs All-FFPN using AUC.

All the considered ANNs are universal approximators. This means that they should all be able to perform equivalently when only either ACC or AUC are considered in the comparison. As this is not the case, we are forced to conclude that the selected hyper-parameters (e.g., number of layers; number of neurons) are not the more suitable for the problem. Notice that in general no fully systematic procedure exists for select these hyper-parameters. Even when the GA stochastic process is used for this purpose, the results are not as equivalent as theoretically anticipated. A discussion on this is issue is presented below.

- Feature selection phase, this a very important factor that can change the results of the ANNs. The gear fault diagnosis problem initially had a database with 817 features; after being applied several feature selection methods, it was apparent that with only 14 features obtained with RF the initial problem is characterized.
- The learning process, plays a crucial role in the classification problem. The classic ANNs trained by two different learning algorithm differ completely in their results. On the other hand, one might expect to find the GA approaches as the best classifiers; as they attempt to find the best structure and features for a specific ANN. This occurs when the algorithm does not find the global solution and goes into a local solution. It is very likely that with a particular tuning of each ANN for the case study the nonparametric results change.
- The evaluation metric, the multicomparison performed with ACC observations gave different results to the AUC ones. Therefore, it is important to know what needs to be measured regarding the classification problem. AUC is equivalent to the probability that a randomly chosen negative example has a smaller estimated probability of belonging to the positive class than a randomly chosen positive example [43]. On the other hand, ACC measures the predictive ability of the classifier on the testing examples.

9. Conclusions

The work presents a comprehensive process to perform a multicomparison between several artificial neural networks, learning procedures, and feature selection methods over a realistic case study. The case study is the gearbox fault classification problem. This has been shown to be both economically relevant and technically challenging attracting the attention of many researchers both in the academia and in the industry. Both parametric and nonparametric statistical tests were considered. A criterion to rank (i) feature selection methods and (ii) pair neural networks and learning procedures was established. This process aims to detect the best algorithm with the multicomparison test results through multiple pairwise comparisons and ranking procedures.

A study of several feature reduction methods with supervised and unsupervised learning is completed. The classifiers were evaluated using ACC and AUC metrics.

The results show that the overall best feature selection method using ACC and AUC was obtained with Random Forest, a boosted entropy based selection method. On the other hand, the best ANN varies depending on the evaluation metric. When ACC is used the method ranked first is FFPN while when AUC is considered the best ANN is One Vs All-FFPN. In general, FFPN networks yield the best results.

Acknowledgements

The work was sponsored in part by the GIDTEC project called “Desarrollo de una herramienta computacional basada en modelos de computación inteligente para el monitoreo en maquinaria rotativa” No. 017-007-2015-11-05, and the Prometeo Project of the Secretariat for Higher Education, Science, Technology and Innovation (SENESCYT) of the Republic of Ecuador. The experimental work was developed at the GIDTEC research group lab of the Universidad Politécnica Salesiana de Cuenca, Ecuador.

Appendix A. Parameter settings for the ANN used

Table A1.

Table A1
ANNs selected and parameters used.

Classifier	Training method	Architecture
Feed Forward (FFNN)		
	• SGD • GD	• Hidden layers: 3 • Size of each layer: 40, 40 and 30 • Transfer function: Hyperbolic tangent sigmoid • Perform function: Mean squared error • Maximum number of epochs to train: 1000 with 10-fold cross validation
Cascade Forward (CFN)		
	• SGD • GD	• Hidden layers: 2 • Size of each layer: 40 and 50 • Transfer function: Hyperbolic tangent sigmoid • Perform function: Mean squared error • Maximum number of epochs to train: 1000 with 10-fold cross validation
Pattern recognition network (FFPN)		
	• SGD • GD	• Hidden layers: 1 • Size of each layer: 45 • Transfer function: Hyperbolic tangent sigmoid • Perform function: Mean squared error • Maximum number of epochs to train: 1000 with 10-fold cross validation
Radial Basis Function Network (RBFN)		
	• PNN • RB	• Spread: random value between 0 and 1 • 75% of the data for training and 35% for test
Self Organizing Map (SOM)		
		• Initial neighborhood size: 10 • Distance Function: Euclidean distance • Topology Function: Hexagonal layer topology function • Number of training steps for initial covering of the input space: 500

Appendix B. Post hoc procedure for the data reduction comparisons

Table B1.
Table B2.

Table B1
Adjusted *p*-values for multiple comparisons among the data reduction method results given by a FFNN-SGD and ACC as evaluation metric.

Pair	*p*−value	Holm	Shaffer
14Att/RF vs 35Att/NMF	3.2911×10^{-15}	0.0018	0.0018
14Att/RF vs 82Att/MNF	7.2063×10^{-14}	0.0019	0.0024
14Att/PCA vs 35Att/NMF	1.1477×10^{-9}	0.0019	0.0024
14Att/PCA vs 82Att/MNF	1.2549×10^{-8}	0.0020	0.0024
14Att/RF vs 9Att/LDA	2.3927×10^{-6}	0.0021	0.0024
29Att/RF vs 35Att/NMF	4.5333×10^{-6}	0.0022	0.0024
330Att vs 35Att/NMF	1.5478×10^{-5}	0.0023	0.0024
29Att/RF vs 82Att/MNF	2.7893×10^{-5}	0.0024	0.0024
14Att/RF vs 42Att/MNF	7.7227×10^{-5}	0.0025	0.0031
330Att vs 82Att/MNF	8.6193×10^{-5}	0.0026	0.0031
35Att/NMF vs 42Att/MNF	8.6193×10^{-5}	0.0028	0.0031
330Att vs 14Att/RF	3.7431×10^{-4}	0.0029	0.0031
82Att/MNF vs 42Att/MNF	4.1366×10^{-4}	0.0031	0.0031
29Att/RF vs 14Att/RF	9.8759×10^{-4}	0.0033	0.0033
9Att/LDA vs 35Att/NMF	0:0016	0.0036	0.0038
9Att/LDA vs 14Att/PCA	0:0034	0.0038	0.0038
9Att/LDA vs 82Att/MNF	0:0057	0.0042	0.0042
14Att/PCA vs 42Att/MNF	0:030	7 0.0045	0.0045
14Att/RF vs 14Att/PCA	0:0731	0.0050	0.0050
330Att vs 14Att/PCA	0:0775	0.0056	0.0056
29Att/RF vs 14Att/PCA	0:1331	0.0063	0.0063
29Att/RF vs 9Att/LDA	0:1547	0.0071	0.0071
330Att vs 9Att/LDA	0:2463	0.0083	0.0083
9Att/LDA vs 42Att/MNF	0:4447	0.0100	0.0100
29Att/RF vs 42Att/MNF	0:5100	0.0125	0.0125
330Att vs 42Att/MNF	0:6926	0.0167	0.0167
35Att/NMF vs 82Att/MNF	0:6926	0.0250	0.0250
330Att vs 29Att/RF	0:792	10.0500	0.0500

Table B2
Adjusted *p*-values for multiple comparisons among the data reduction method results given by a FFNN-SGD and AUC as evaluation metric.

Pair	*p*−value	Holm	Shaffer
14Att/PCA vs 35Att/NMF	3.5713×10^{-9}	0.0018	0.0018
330Att vs 35Att/NMF	7.6213×10^{-8}	0.0019	0.0024
29Att/RF vs 35Att/NMF	7.6213×10^{-8}	0.0019	0.0024
14Att/RF vs 35Att/NMF	7.6213×10^{-8}	0.0020	0.0024
9Att/LDA vs 35Att/NMF	7.6213×10^{-8}	0.0021	0.0024
35Att/NMF vs 42Att/MNF	4.5333×10^{-6}	0.0022	0.0024
14Att/PCA vs 82Att/MNF	3.1318×10^{-5}	0.0023	0.0024
330Att vs 82Att/MNF	2.7624×10^{-4}	0.0024	0.0024
29Att/RF vs 82Att/MNF	2.7624×10^{-4}	0.0025	0.0031
14Att/RF vs 82Att/MNF	2.7624×10^{-4}	0.0026	0.0031
9Att/LDA vs 82Att/MNF	2.7624×10^{-4}	0.0028	0.0031
82Att/MNF vs 42Att/MNF	0:0044	0.0029	0.0031
35Att/NMF vs 82Att/MNF	0:0820	0.0031	0.0031
14Att/PCA vs 42Att/MNF	0:1876	0.0033	0.0033
330Att vs 42Att/MNF	0:4292	0.0036	0.0036
29Att/RF vs 42Att/MNF	0:4292	0.0038	0.0038
14Att/RF vs 42Att/MNF	0:4292	0.0042	0.0042
9Att/LDA vs 42Att/MNF	0:4292	0.0045	0.0045
330Att vs 14Att/PCA	0:5982	0.0050	0.0050
29Att/RF vs 14Att/PCA	0:5982	0.0056	0.0056
14Att/RF vs 14Att/PCA	0:5982	0.0063	0.0063
9Att/LDA vs 14Att/PCA	0:5982	0.0071	0.0071
330Att vs 29Att/RF	1:0000	0.0083	0.0083
330Att vs 14Att/RF	1:0000	0.0100	0.0100
330Att vs 9Att/LDA	1:0000	0.0125	0.0125
29Att/RF vs 14Att/RF	1:0000	0.0167	0.0167
29Att/RF vs 9Att/LDA	1:0000	0.0250	0.0250
14Att/RF vs 9Att/LDA	1:0000	0.0500	0.0500

References

[1] N. Bayar, S. Darmoul, S. Hajri-Gabouj, H. Pierreval, Fault detection, diagnosis and recovery using artificial immune systems: a review, Eng. Appl. Artif. Intell. 46 (Part A) (2015) 43–57.
[2] M.S. Kan, A.C. Tan, J. Mathew, A review on prognostic techniques for non-stationary and non-linear rotating systems, Mech. Syst. Signal Process. (62–63) (2015) 1–20.
[3] S. Yin, D. S, X. Xie, H. Luo, A review on basic data-driven approaches for industrial process monitoring, IEEE Trans. Ind. Electron. 61 (11) (2014) 6418–6428.
[4] B. Xiao, M. Huo, X. Yang, Y. Zhang, Fault-tolerant attitude stabilization for satellites without rate sensor, IEEE Trans. Ind. Electron. 62 (11) (2015) 7191–7202.
[5] S. Yin, X. Zhu, K. O, Improved pls focused on key-performanceindicator-related fault diagnosis, IEEE Trns. Ind. Electron. 62 (3) (2015) 1651–1658.
[6] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, D. Siegel, Prognostics and health management design for rotary machinery systems- Reviews, methodology and applications, Mech. Syst. Signal Process. 42 (1–2) (2014) 314–334.
[7] N. Saravanan, V. Kumar, K. Ramachandran, Fault diagnosis of spur bevel gear box using artificial neural network (ANN), and proximal support vector machine (PSVM), Appl. Soft Comput. 10 (2010) 344–360.
[8] R. Mesquita, J. Beleza, F. Pires, Neural networks for condition monitoring of wind turbines gearbox, J. Energy Power Eng. 6 (2012) 638–644.
[9] P. Bangalore, L. Bertling, An artificial neural network approach for early fault detection of gearbox bearings, IEEE Trans. Smart Grid 6 (2) (2010) 980–987.
[10] N. Saravanan, K. Ramachandran, Incipient gear box fault diagnosis using discrete wavelet transform (DWT) for feature extraction and classification using artificial neural network (ANN), Expert Syst. Appl. 37 (2010) 4168–4181.
[11] Y. Wang, Q. Li, M. Chang, H. Chen, G. Zang, Research on fault diagnosis expert system based on the neural network and the fault tree technology, Procedia Eng. 31 (2012) 1206–1210.
[12] J. Sanz, R. Perera, C. Huerta, Gear dynamics monitoring using discrete wavelet transformation and multi-layer perceptron neural networks, Appl. Soft Comput. 12 (2012) 2867–2878.
[13] T. Jia-li, L. Yi-jun, W. Fang-sheng, Levenberg-marquardt neural network for gear fault diagnosis, Networking and Digital Society (ICNDS), 2010 2nd International Conference 1 (2010) pp. 134–137.
[14] W. Chai, J. Qiao, Passive robust fault detection using RBF neural modeling based on set membership identification, Eng. Appl. Artif. Intell. 24 (2010) (2014) 1–12.
[15] G. Rameshkumar, B. Rao, K. Ramachandran, Use of radial basis function neural networks for analysis of unbalance in rotating machinery, Int. J. Innov. Technol. Explor. Eng. 1 (2) (2012) 168–171.
[16] J. Ben, L. Saidi, A. Mouelhi, B. Chebel-Morello, F. Fnaiech, Linear feature selection and classification using PNN and SFAM neural networks for a nearly online diagnosis of bearing naturally progressing degradations, Eng. Appl. Artif. Intell. 42 (2015) 67–81.
[17] J.-D. Wu, J.-J. Chan, Faulted gear identification of a rotating machinery based on wavelet transform and artificial neural network, Expert Syst. Appl. 36 (2009) 8862–8875.
[18] K. Li, P. Chen, S. Wang, An intelligent diagnosis method for rotating machinery using least squares mapping and a fuzzy neural network, Sensor 12 (2012) 5919–5939.
[19] W. Wang, An enhanced diagnostic system for gear system monitoring, Syst. Man Cybern. Part B: Cybern. 38 (2008) 102–112.
[20] H. Mok, C. Chan, Online fault detection and isolation of nonlinear systems based on neurofuzzy networks, Eng. Appl. Artif. Intell. 21 (2008) 171–181.
[21] W. Wang, D. Kanneg, An integrated classifier for gear system monitoring, Mech. Syst. Signal Process. 23 (4) (2009) 1298–1312.
[22] W. Li, L. Zhang, Y. Xu, Gearbox pitting detection using linear discriminant analysis and distance preserving self-organizing map, in: Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International, 2012, pp. 2225–2229.
[23] J. Yu, L. Zhixiong, G. Yuancheng, Research on AR modeling method with SOFM-based classifier applied to gear multi-faults diagnosis, Informat. Control Autom. Robot. 2 (2010) 488–491.
[24] M. Cerrada, G. Zurita, D. Cabrera, R.-V. Sánchez, M. Artés, C. Li, Fault diagnosis in spur gears based on genetic algorithm and random forest, Mech. Syst. Signal Process. (70–71) (2016) 87–103.
[25] Z. Yang, W.I. Hoi, J. Zhong, Gearbox fault diagnosis based on artificial neural network and genetic algorithms, in: 2011 International Conference on System Science and Engineering (ICSSE), 2011, pp. 37–42.
[26] M. Cerrada, R.V. Sánchez, D. Cabrera, G. Zurita, C. Li, Multi-stage feature selection by using genetic algorithms for fault diagnosis in gearboxes based on vibration signal, Sensors 15 (9) (2015) 23903–23926.
[27] T. Xie, H. Yu, B. Wilamowski, Comparison between traditional neural networks and radial basis function networks, in: 2011 IEEE International Symposium on Industrial Electronics (ISIE), 2011, pp. 1194–1199.
[28] J. Derrac, S. García, D. Molina, H. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18.
[29] R. Genuer, J. Poggi, C. TuleauMalot, Variable selection using random forests, Pattern Recognit. Lett. 14 (31) (2010) 2225–2236.
[30] D. Cabrera, F. Sancho, R.V. Sánchez, G. Zurita, M. Cerrada, C. Li, R. Vásquez, Fault diagnosis of spur gearbox based on random forest and wavelet packet decomposition, Front. Mech. Eng. 10 (2015) 1–10.

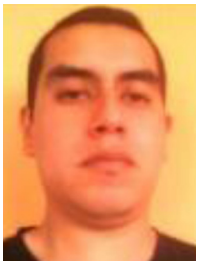
[31] S. Ji, J. Ye, Generalized linear discriminant analysis: a unified framework and efficient model selection, IEEE Trans. Neural Netw. 19 (10) (2008) 1768–1782.
[32] Y. Li, A. Ngom, The non-negative matrix factorization toolbox for biological data mining, Source Code Biol. Med. 10 (8) (2013).
[33] T.-P. Hong, Y.-L. Liou, S.-L. Wang, B. Vo, Feature selection and replacement by clustering attributes, Vietnam J. Comput. Sci. 1 (1) (2014) 47–55.
[34] R. Yan, R.X. Gao, X. Chen, Wavelets for fault diagnosis of rotary machines: a review with applications, Signal Process. 96 (Part A) (2014) 1–15 time-frequency methods for condition based maintenance and modal analysis.
[35] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, CRC Press, Western Connecticut, USA, 2004.
[36] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[37] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, J. Mach. Learn. Res. 9 (2006) 2677–2694.
[38] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, J. Heuristics 9 (2009) 2677–2694.
[39] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (2010) 2044–2064.
[40] J. Luengo, S. García, F. Herrera, A study on the use of statistical tests for experimentation with neural networks: analysis of parametric test conditions and non-parametric tests, Expert Syst. Appl. 36 (2009) 7798–7808.
[41] M. Cerrada, R.-V. Sánchez, F. Pacheco, D. Cabrera, G. Zurita, C. Li, Hierarchical feature selection based on relative dependency for gear fault diagnosis, Appl. Intell. (2015) 1–17, http://dx.doi.org/10.1007/s10489-015-0725-3.
[42] H.-M. Kaltenbach, A Concise Guide to Statistics, Springer, ETH Zurich, Switzerland, 2012.
[43] D. Hand, R. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems, Mach. Learn. 45 (2) (2001) 171–186.

Fannia Pacheco received her M.Sc. Degree in Computer Science from the Universidad de Los Andes, Venezuela, 2015. She joined the GIDTEC research team at the Universidad Politécnica Salesiana (UPS), Ecuador. Her research interests cover novelty detection, data analysis and intelligent systems.

José Valente de Oliveira received the Ph.D. (1996), M.Sc. (1992), and the "Licenciado" (five-years) degree in Electrical and Computer Engineering, all from the IST, Technical University of Lisbon. Currently he is a Faculty at the University of Algarve and a member of CEOT with research interests in interdisciplinary areas of computational intelligence and machine learning. He is an Associated Editor of the Journal of Intelligent & Fuzzy Systems, IOS Press, and co-editor of the books Advances in Fuzzy Clustering and Its Applications, Wiley 2007; and Human-Computer Interaction: The Agency Perspective, Springer, 2012. He was a visiting Faculty at University of Alberta (2005), Universidade Nacional de Timor Loro Sae (2005), Carnegie Mellon University (2012), Universidade Federal do Ceará (2013), and an Ecuador Prometeo Researcher at UPS (2015).

René-Vinicio Sanchez received the B.S. in Mechanical Engineering in 2004, from the Universidad Politécnica Salesiana (UPS), Ecuador. He got his master in management audit quality in 2008 at the UTPL, Ecuador, and the master degree in industrial technologies research in 2012 at the UNED, Spain. Currently, he is Professor of the Department of Mechanical Engineering in the UPS. His Research interests are in machinery health maintenance, pneumatic and hydraulic systems, artificial intelligence and engineering education.

Mariela Cerrada received her Ph.D. degree in Automatic Systems in 2003 from the INSA Toulouse-France. She is currently a full dedicated time titular professor in the Department of Control Systems and associate member of the Studies Center on Microcomputers and Distributed Systems (CEMISID) at the Engineering Faculty in the Universidad de Los Andes of Venezuela. She was Prometeo Researcher at the Universidad Politécnica Salesiana of Ecuador. Her main research area is on fault diagnosis, supervision and intelligent control systems.



Diego Cabrera received his M.Sc. degree at the Sevilla University in 2014. Currently, he is a lecturer at the Universidad Politécnica Salesiana (UPS), Ecuador. He is a member of the research group of innovation and development at UPS, and a member of the research group of complex systems modelling at the Universidad Central, Ecuador. His research areas are condition-based maintenance, complex systems modelling, and intelligence systems.

Chuan Li received his Ph.D. degree from the Chongqing University, China, in 2007. He has been successively a Postdoctoral Fellow with the University of Ottawa, Canada, a Research Professor with the Korea University, South Korea, and a Senior Research Associate with the City University of Hong Kong, China. He is currently a Professor with the Chongqing Technology and Business University, China, and a Prometeo Researcher with the Universidad Politécnica Salesiana, Ecuador. His research interests include machinery healthy maintenance, and intelligent systems.

Grover Zurita received his Ph.D. degree from Luleå University of Technology, Sweden, in 2001. He was a Postdoctoral Fellow at the University of New South Wales, Australia, in 2002. Currently, he is a Professor at the Private University of Bolivia, and he was Prometeo Researcher at the Universidad Politécnica Salesiana of Ecuador. His research interests are machine diagnosis, optimization and control of internal combustion engines.

Mariano Artés obtained his degree in Mechanical Engineering from the Polytechnic University of Madrid, Madrid, Spain, and his Ph.D. degree from the same University. Professor Artés currently serves as Full Professor at the Department of Mechanics at the Universidad Nacional de Educacion a Distancia (UNED), Madrid, Spain. He also served as Visiting Professor at the State University of New York at Buffalo during one academic year. His research has been mainly carried out in the broad field of machine design and machine behavior under vibration including machine failure diagnosis and prognosis using vibrational methods. He has also participated in more than twenty research projects both as a principal investigator and as a member of the research team. As a result of his research, Dr. Artés is the author of more than eighty technical publications in specialized journals and conferences.