Expert Systems With Applications xxx (xxxx) xxx

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An experimental methodology to evaluate machine learning methods for fault diagnosis based on vibration signals

Thomas Walter Rauber [a,*], Antonio Luiz da Silva Loca [b], Francisco de Assis Boldt [c], Alexandre Loureiros Rodrigues [d], Flávio Miguel Varejão [a]

[a] Departamento de Informática, Centro Tecnológico, Universidade Federal do Espírito Santo, 29075-910 Vitória, Brazil
[b] Coordenadoria de Informática, Instituto Federal do Espírito Santo, 29173-087 Serra, Brazil
[c] Coordenadoria de Informática, Instituto Federal do Espírito Santo, 29520-000 Alegre, Brazil
[d] Departamento de Estatística, Centro de Ciências Exatas, Universidade Federal do Espírito Santo, 29075-910 Vitória, Brazil

ARTICLE INFO

Keywords:
Fault detection
CWRU bearing fault database
Performance criteria
Classification
Pattern recognition
Machine learning

ABSTRACT

This paper presents a systematic procedure to fairly compare experimental performance scores for machine learning methods for fault diagnosis based on vibration signals. In the vast majority of related scientific publications, the estimated accuracy and similar performance criteria are the sole quality parameter presented. However, the experimental design giving rise to these results is mostly biased, based on unacceptably simple validation methods and on recycling identical patterns in test data sets, previously used for training. Moreover, the methods in general overfit their hyperparameters, introducing additional overoptimistic results. In order to remedy this defect, we critically analyse the usual training-validation-test division and propose an algorithmic guideline in the form of a validation framework. This allows a well defined comparison of experimental results. In order to illustrate the ideas of the paper, the Case Western Reserve University Bearing Data benchmark is used as a case study. Four distinct classifiers are experimentally compared, under gradually more difficult generalization tasks using the proposed evaluation framework: K-Nearest-Neighbor, Support Vector Machine, Random Forest and One-Dimensional Convolutional Neural Network. An extensive literature review suggests that most vibration based research papers, particularly for the Case Western Reserve University Bearing Data, use similar patterns for training and testing, making their classification an easy task.

1. Introduction

Software based fault diagnosis is an essential tool to guarantee the safety and maintainability of dynamic processes (Gao, Cecati, & Ding, 2015; Chiang, Braatz, & Russell, 2001). A principal distinction of employable methods is model-based fault diagnosis, c.f. for instance (Varga, 2017; Gertler, 2017; Ding, 2012), and model-free diagnosis, e.g. (Ding, 2016; McMillan & Vegas, 2019). Vibration based fault diagnosis focuses on analysing vibration signals in order to identify possible equipment faults. Some research works try to find signal characteristics or measures for identifying faults (Smith & Randall, 2015; Diaz et al., 2015; Diaz et al., 2017). Other research works focus on machine learning techniques that use the signals for training and testing classifiers for identifying faults. However, this work focuses on the latter approach.

The access to real world, well documented benchmark data is limited. A limited set of repositories with real vibration signals, obtained from existing mechanical systems, is available (Lee, Qiu, Yu, & Lin, 2007; Nectoux et al., 2012; Bechhoefer, 2016; Paderborn, 2020; MaFaulDa, 2016). The Case Western Reserve University (CWRU) Bearing Data (CWRU, 2014) is probably referred to the most in scientific literature. Some research papers use non publicly available vibration data sets (Liao, Gao, Yang, & Guo, 2019; Lei, Jia, Lin, Xing, & Ding, 2016; Verstraete, Ferrada, Droguett, Meruane, & Modarres, 2017).

An extensive amount of research works apply model-free, machine learning methods to classify distinct operational states of the process. A robust fault diagnosis system must be able to generalize well. This means that a trained classifier should be able to recognize as many faults as possible, even when there are variations of the machine conditions. Presented results must be reproducible and must be statistically

* Corresponding author.
E-mail addresses: thomas@inf.ufes.br (T.W. Rauber), antonio.loca@ifes.edu.br (A.L. da Silva Loca), franciscoa@ifes.edu.br (F.A. Boldt), alexandre.rodrigues@ufes.br (A.L. Rodrigues), fvarejao@inf.ufes.br (F.M. Varejão).

https://doi.org/10.1016/j.eswa.2020.114022
Received 6 April 2020; Received in revised form 22 August 2020; Accepted 14 September 2020
Available online 30 September 2020

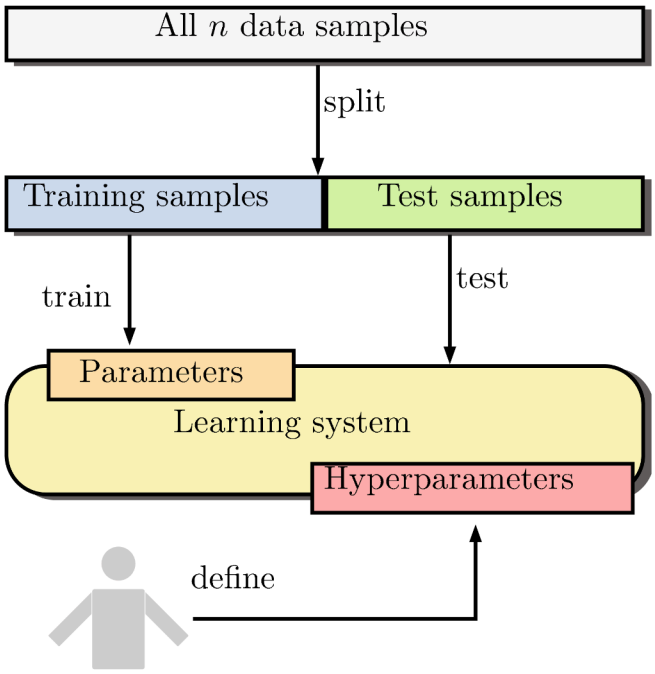


Fig. 1. Simple train-test division for a machine learning based fault diagnosis system.

significant; otherwise overoptimistic results are obtained. For instance, when a limited amount of samples are available, the data must be separated into non-overlapping sets. Each set may not be visible to the other set during the parametrization of the diagnosis system. It must be avoided that the same data is used both for tuning the hyperparameters of a classifier model and for testing the resultant classifier. Otherwise the results will probably be biased towards optimism. A typical example are the kernel and its associated variables in a Support Vector Machine (SVM). Which kernel is best for a C-SVM, RBF or Polynomial? And then, when using, for instance, the RBF kernel, use a grid search to find the best combination of the C regularization value and of the spread γ. Another example are deep learning structures of artificial neural networks where the layout is sometimes adjusted until it delivers the best results for the same data set. A more justified way of presenting performance scores is to isolate a test set completely. With the rest of the data, the hyperparameter tuning and the training can be done. When the final adjustments have been made, the test set is then used to estimate, for instance, an accuracy score. This procedure can be repeated several times, however the test set needs to always be kept apart. This hierarchy will be denoted as the inner and outer loop. Even then, when there is only a small amount of samples available for training, the performance may surpass theoretical Bayes limits.

Another common practice with vibration based machine learning research consists of defining a class, based on the chunks of a single chopped signal, even sometimes overlapping. Then, chunks of the same signal are used both on the training and testing data sets (Lei et al., 2016; Zhang, Peng, Wu, Yao, & Guan, 2017; Verstraete et al., 2017; Liao et al., 2019). We call this the similarity bias problem. The patterns used for testing are almost indistinguishable from patterns used in the training data set. This fact may lead to an oversimplified model of real world fault diagnosis problems. A robust system must comprehend different machine conditions, and still be able to provide reasonable diagnostic information.

An important aspect when experimentally comparing fault diagnosis approaches consists of statistically verifying if the results are significantly different. However, this is seldom the case in the vibration based fault diagnosis research works.

The concern with the reproducibility of scientific research has

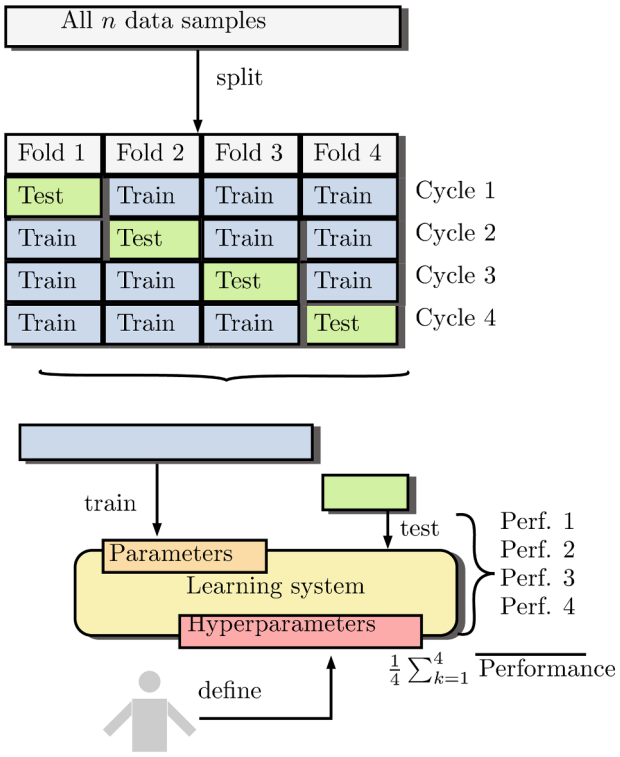


Fig. 2. K-fold cross validation.

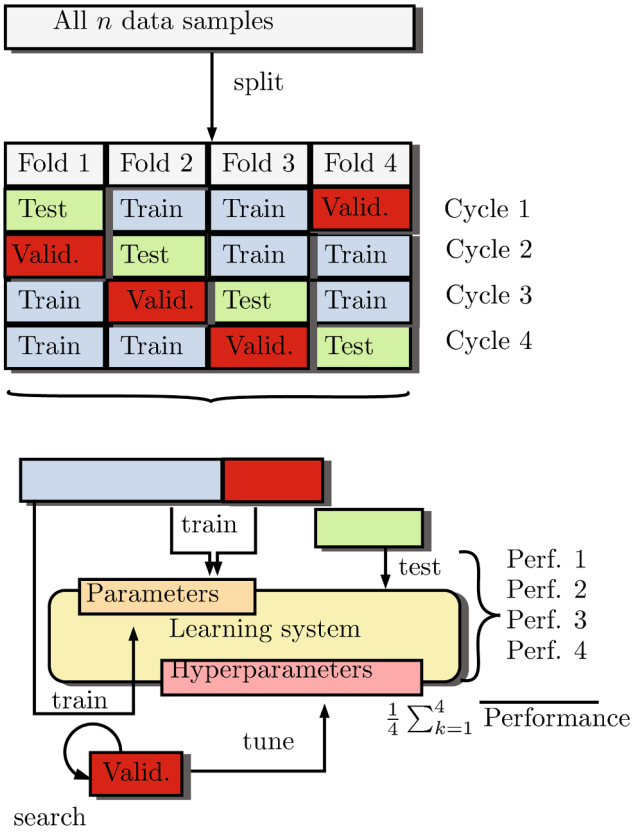


Fig. 3. Nested K-fold cross validation with inner and outer loops. The outer loop has $K = 4$ folds.

Table 1
Parameters of Gaussian data sets from Fukunaga (1990), plus "null" data set from Varma and Simon (2006).

data set	Class	Mean Vector μ	Covariance Matrix Σ	Bayes Accuracy Limit
"I-I"	$\mathscr{C}_a$ $\mathscr{C}_b$	$[0,0,0,0,0,0,0,0]$ $[2.56,0,0,0,0,0,0,0]$	$\mathrm{diag}(1,1,1,1,1,1,1,1)$ $\mathrm{diag}(1,1,1,1,1,1,1,1)$	90.0%
"I-4I"	$\mathscr{C}_a$ $\mathscr{C}_b$	$[0,0,0,0,0,0,0,0]$ $[0,0,0,0,0,0,0,0]$	$\mathrm{diag}(1,1,1,1,1,1,1,1)$ $\mathrm{diag}(4,4,4,4,4,4,4,4)$	$\approx 91.0\%$
"I-Λ"	$\mathscr{C}_a$ $\mathscr{C}_b$	$[0,0,0,0,0,0,0,0]$ $[3.86,3.10,0.84,0.84,1.64,1.08,0.26,0.01]$	$\mathrm{diag}(1,1,1,1,1,1,1,1)$ $\mathrm{diag}(8.41,12.06,0.12,0.22,1.49,1.77,0.35,2.73)$	$\approx 98.1\%$
"null"	$\mathscr{C}_a$ $\mathscr{C}_b$	$[0,0,0,0,0,0,0,0]$ $[0,0,0,0,0,0,0,0]$	$\mathrm{diag}(1,1,1,1,1,1,1,1)$ $\mathrm{diag}(1,1,1,1,1,1,1,1)$	50.0%

Table 2
Hyperparameters used for each classifier architecture. Experiments relative to the data sets of Table 1.

Model	Hyperparameters	Range
KNN	Number of neighbors	$\{1,3,5\}$
	Distance metric	{Euclidean, Weighted}
SVM	C-SVM C, $\log_{10}(.)$	$\{-3,-2,-1,0,1\}$
	RBF kernel γ $\log_{10}(.)$	$\{-3,-2,-1,0\}$
RF	Number of trees (estimators)	$\{10,20,50\}$
	Number of features for split	$\{1,\ldots,8\}$
MLP	Hidden nodes	$\{2,3,4,5,10,15,20\}$
	Hidden layer activation	{Relu, Sigmoid}
	Max iter	{100, 1000}

steadily increased recently. Reproducibility is defined as obtaining consistent results using the same data and computational code as the original study. It has been found that many scientific studies are difficult or impossible to reproduce. The vibration based fault diagnosis works often have a lack of reproducibility.

This work proposes an experimental methodology for evaluating machine learning approaches for fault diagnosis based on vibration signals aiming to cope with all previously mentioned problems, i.e. completely isolating the test set, avoiding the similarity bias, verifying statistically significant differences and also allowing reproducibility. The main contributions of the paper are

1. A methodology for machine learning applied to vibration signal analysis, integrating nested cross validation, reproducibility, statistical analysis and avoiding similarity bias. Except for the latter, all these topics have already been applied in an isolated manner in the context to fault diagnosis;
2. An experimental study with synthetic data sets suggests superiority of the nested cross validation approach, especially for a small number of training samples;
3. The identification of common problems of research papers in the area of model-free fault diagnosis;
4. Identification of the overoptimistic bias, due to very similar training and test samples of the same class of machine conditions;
5. A study of the CWRU database, the principal real vibration signal source used in the scientific literature of the area.

The rest of the paper is organized in the following manner: Section 2 reviews conventional performance estimation methods in supervised learning problems. Furthermore, the common cross validation techniques are improved by a two-level evaluation hierarchy that isolates performance estimation from hyperparameter tuning. Synthetic data sets with known Bayesian error bounds are used to juxtapose the conventional and improved evaluation methods, suggesting the mostly overoptimistic results in research papers. Section 3 focuses on the important aspects of the proposed methodology for avoiding the similarity bias, for checking statistical significance of result differences, and also how to guarantee reproducibility. Section 4 critically reviews research works that use the vibration signals on machine learning approaches for fault diagnosis. Special focus is given to the CWRU mechanical data set, emphasizing the applied methods, and highlighting, if applicable, the defects that motivate the elaboration of this study. Moreover, the paper shows how the methodology may be customized for a specific data set. Section 5 and Section 6 present the case study of the CWRU Bearing Data Set. Section 5 defines different experimental modes for fault diagnosis, with gradually increasing difficulties. Section 6 provides an application of the proposed framework for the CWRU data, with four different classifier models, varying the generalization difficulty of the fault diagnosis, and finally the conclusions are drawn in Section 7.

2. Supervised learning performance evaluation techniques

In this section, a fair performance estimation framework with an outer validation loop is defined and introduced by data sets with analytically known Bayesian error rates.

2.1. Single level performance evaluation

A completely inappropriate performance evaluation technique is *resubstitution* where the same data is used to training and test. A one time split into training and test (*holdout*), depicted in Fig. 1, bears the risk of a high bias towards optimism when a favorable training–test division occurs. This evaluation is denominated as *Single train-test split* in the literature review presented in Table 5. The bias can be reduced by *repeated holdout*. A commonly employed technique is *K-Fold Cross Validation* where the whole data set is divided into K non overlapping parts. The training–test cycle is done K times, each of the K parts is held out from the training once, and the remaining $(K-1)$ parts are used to train the system. The held out fold is submitted as the test set, and the K results, are simply averaged to obtain the final estimate. Fig. 2 shows the procedure for the example of $K = 4$. If the number of folds equals the number of all available samples, i.e. $K = n$, the result is a *Leave-One-Out* cross validation. The hyperparameters are generally manually defined once and remain unchanged during the particular evaluation procedure.

2.2. Nested performance evaluation

A better performance evaluation is to tune the hyperparameters automatically (Cawley & Talbot, 2010). It is important to note that for obtaining a just score, there is not such a thing as the best set of hyperparameters. This set would only be available for an infinitely large number of data samples.

2.2.1. K-fold CV outer loop and tuning inner loop

Fig. 3 presents the structure of a performance evaluation that retains a test set in the same fashion as the conventional K-Fold CV. However, there is an inner loop that takes the $(K-1)$ training folds, usually kept for training the system and splits off a part as the *validation set*. In general, the inner loop data set can be split into any number $1 < L < n_{\text{inner}}$ of folds, where n_{inner} is the number of samples passed to the inner loop. In order to facilitate the presentation of the theory, $L = K-1$ for the rest of the paper and additionally, the division of the folds is done only once. The validation set can repeatedly be used to make affirmations about the quality of the current hyperparameters. Bear in mind that the test set of the outer loop is left untouched by the inner loop. The inner loop only uses the training data and the validation data. If an improvement of the result can be achieved by variation of the hyperparameters, then this set of hyperparameters is frozen to report the result of the Kth fold of the outer loop. Once the best hyperparameters have been defined in that inner loop, the final performance score of the Kth fold uses the left-out test set of the outer loop for testing and the union of both the training and validation sets for training. For each of the K folds of the outer loop, different best sets of hyperparameters are generally obtained. Hence the nested evaluation is unable to deliver a defined final best set of hyperparameters.

Consider the example of a K-Nearest Neighbors classifier[1] Cover et al. (Jan. 1967). Assuming that all other hyperparameters are fixed, for instance, the metric of the distance (Euclidean, weighted, etc.), the only hyperparameter is the number K of neighbors. Suppose that only one and three nearest neighbors are considered, i.e. the hyperparameter can assume the values $K = 1$ and $K = 3$. Further suppose a minimal illustrative division of the data into only four folds, denoted as F1, F2, F3, and F4. Hence there are four cycles of the outer loop. In the first cycle of the outer loop, F1 is retained as the test fold. F2, F3 and F4 are passed to the inner loops. It can be noted that in general the data that is passed to the inner loops can be divided into new arbitrary folds, but here, the fold division inherited from the outer loop is not changed. Since we defined that in the inner loop there is one fold less than in the outer loop, we have three folds for three inner cycles. The hyperparameters are tuned using F2, F3 and F4. In the first inner cycle, the union of F2 and F3 is used for training and F4 is used for test. In the second inner cycle, the union of F2 and F4 is used for training and F3 is used for test. In the third inner cycle, the union of F3 and F4 is used for training and F2 is used for test. The average performance of these three inner cycles is calculated for each possible combination of the hyperparameters, in this case for the different possible numbers $\{1, 3\}$ of neighbors of the K-Nearest Neighbors classifier. Suppose that $K = 1$ gave the best result. Consequently, the hyperparameter is tuned to $K = 1$. Now the first outer cycle is ready to calculate the performance. With the best hyperparameter, the training set is the union of F2, F3 and F4. The test fold is F1 which had no influence on the choice of the best hyperparameter. The performance is recorded. For the remaining three outer cycles, the procedure is repeated and the final estimated performance is the average value of all four performances of the outer loop. In the second outer cycle, F2 is the retained test set and F1, F3, F4 are passed to the inner loop. The best hyperparameter might be different from the first outer cycle. In the third outer cycle, F3 is the retained test set and F1, F2, F4 are passed to the inner loop. In the fourth outer cycle, F4 is the retained test set and F1, F2, F3 are passed to the inner loop. The inner loop works in the same way as explained for the first outer cycle.

2.2.2. Repeated nested evaluation

The idea of tuning the hyperparameters and using an independent test in each cycle of the K-Fold CV delivers K results, one for each cycle of the outer loop. In the conventional form of the K-Fold CV, these results are combined by taking the average of the folds. Assume, e.g., that the above K-Nearest Neighbors classifier has a total of $n = 150$ data samples, and that $K = 4$ folds have twice 38 and twice 37 samples. Assume that the classifier in the first cycle misclassified 4 samples, in the second cycle 2 samples, in the third 5 samples, and 3 in the fourth. A possible result is then an estimated accuracy $1-1/4[4/38 + 2/38 + 5/37 + 3/37] = 90.65\%$. The four results could also be combined in a different manner, for instance, to be submitted to a statistical analysis. Remember that the four results are usually obtained by different best sets of hyperparameters.

One may take this idea further by introducing an additional level of the nested evaluation, repeating the experiment for several *rounds* in order to obtain a more robust model assessment score since folds have different composition in each round. The resulting performance evaluation framework is outlined in Algorithm 1. The following symbols are used:

$\mathscr{C}$	classifier
$\mathscr{D}$	total data set of n patterns
$\mathscr{T}$	training set
$\mathscr{V}$	test set
$\mathscr{P}$	hyperparameter candidate set
$\mathscr{P}^*$	best hyperparameter set
R	number of rounds
K	number of folds of the outer CV loop for overall performance estimation
$f_{r,k}$	kth fold of rth round
PM	$R \times K$ performance matrix
L	number of folds of the inner CV loop for tuning
f_ℓ	ℓth fold of one tuning loop with current hyperparameter candidate set $\mathscr{P}$
crit	performance criterion, e.g. accuracy, precision, recall, F-measure, ROC-AUC

[1] Note that the letter "K" is generally used for both the number K of folds in a K-Fold cross validation and the number of K of neighbors of a K-Nearest Neighbors classifier.

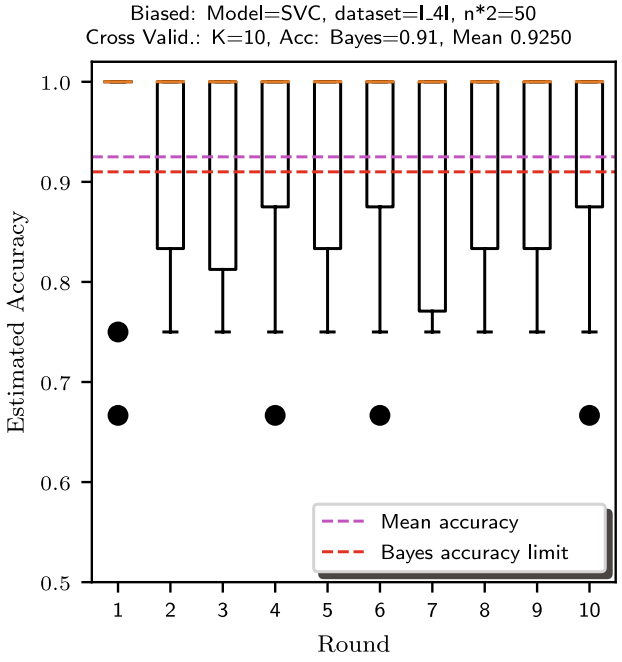
Algorithm 1: Performance evaluation framework.

Function $perf \leftarrow$ *ModelPerformance*($\mathcal{D}$,R,K)
- **Input:** n labelled patterns $\mathcal{D}$ from c classes, R number of rounds, number of folds K (outer loop), L (inner loop)
- **Output:** For each round $r \leftarrow 1, \ldots, R$ make K stratified folds. One fold is fixed as test, remaining folds are train and tune. Calculate performance criterion for each combination of round and fold and put into $R \times K$ performance matrix PM. Process PM, for instance its average as final performance.
- **for** $r = 1$ **to** R **do** // all rounds
  - // generate K stratified folds
  - $f_{r,k}, k = 1, \ldots, K$;
  - **for** $k = 1$ **to** K **do** // all folds
    - // training set of k-th fold of r-th round
    - $\mathcal{T} \leftarrow \{f_{r,1} \cup \ldots \cup f_{r,K}\} \setminus f_{r,k}$;
    - // test set of k-th fold of r-th round
    - $\mathcal{V} \leftarrow f_{r,k}$;
    - $\mathcal{P}_k^* \leftarrow$ Tuning($\mathcal{T}$, L);
    - $\mathcal{C} \leftarrow$ trainclassifier($\mathcal{T}, \mathcal{P}_k^*$);
    - $crit \leftarrow$ testclassifier($\mathcal{C}, \mathcal{V}$);
    - PM(r,k)$\leftarrow crit$
  - **end**
- **end**
- $perf \leftarrow \overline{PM}$;

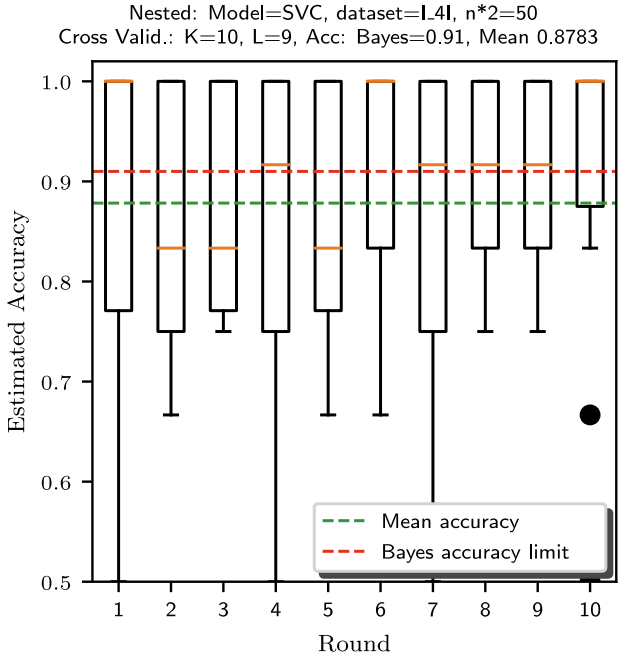
end

Function $\mathcal{P}^* \leftarrow$ *Tuning*($\tilde{\mathcal{D}}$,L)
- **Input:** Data set $\tilde{\mathcal{D}}$, number of tuning folds L
- **Output:** Optimal hyperparameter set $\mathcal{P}^*$ of classifier model
- initialize best criterion $maxcrit \leftarrow 0$;
- // generate L stratified folds
- $\tilde{f}_k, k = 1, \ldots, L$;
- **repeat** // grid search for best hyperparameter set $\mathcal{P}^*$
  - generate hyperparameter candidate set $\mathcal{P}$;
  - **for** $k = 1$ **to** L **do** // all folds
    - // training set of k-th fold
    - $\mathcal{T} \leftarrow \left\{\tilde{f}_1 \cup \ldots \cup \tilde{f}_L\right\} \setminus \tilde{f}_k$;
    - // test set of k-th fold
    - $\mathcal{V} \leftarrow \tilde{f}_k$;
    - $\mathcal{C} \leftarrow$ trainclassifier($\mathcal{T}, \mathcal{P}$);
    - $crit_k \leftarrow$ testclassifier($\mathcal{C}, \mathcal{V}$);
  - **end**
  - $\overline{crit} \leftarrow mean(crit_k)$;
  - **if** $\overline{crit} > maxcrit$ **then**
    - $maxcrit \leftarrow \overline{crit}$; $\mathcal{P}^* \leftarrow \mathcal{P}$
  - **end**
- **until** *search completed*;

end



(a) Simple K-Nearest Neighbors cross validation depicted in Fig. 2 with $n = 2 * 25 = 50$ samples, C-SVM classifier. Data set "I_4I" of Tab. 1



(b) Nested K-Fold cross validation of Fig. 3.

Fig. 4. Comparison of simple K-Fold and nested cross validation. Two hyperparameters are tuned, C and γ of the C-SVM.

Table 3
Estimated accuracy [%] as a function of the number n of samples per class. Scores above the theoretical Bayes limit are underlined.

			n per class (= 2 ∗ n samples)										
Classifier model	data set	Cross Validation	5	10	12	15	20	25	50	100	300	500	1000
K-Nearest Neighbor	I_I	NESTED	90.00	88.00	94.75	99.75	88.25	89.83	83.10	90.15	85.72	86.35	87.61
		BIASED	90.00	90.00	97.25	99.75	91.75	90.42	86.60	91.65	85.97	86.43	87.67
	I_4I	NESTED	60.00	61.50	65.75	75.50	61.50	67.17	59.70	71.00	76.60	76.35	81.19
		BIASED	60.00	61.50	65.75	75.50	63.00	68.58	62.30	71.00	76.83	76.74	81.31
	I_Λ	NESTED	90.00	100.0	96.50	100.0	99.50	90.50	91.70	97.35	95.60	96.34	96.75
		BIASED	90.00	100.0	98.50	100.0	99.75	92.33	92.60	97.90	95.95	96.57	96.84
	null	NESTED	50.00	55.00	65.75	68.50	62.00	57.58	46.60	50.40	48.98	49.19	49.05
		BIASED	50.00	61.00	71.25	75.50	63.75	62.00	49.70	52.75	50.12	50.14	49.89
C-Support Vector Machine	I_I	NESTED	0.00	86.50	91.00	94.00	83.25	88.33	89.60	92.50	88.25	87.83	89.53
		BIASED	100.0	88.00	95.75	96.50	88.25	91.83	90.10	93.90	88.90	88.46	90.01
	I_4I	NESTED	100.0	94.00	87.75	93.50	89.75	87.83	92.40	90.20	92.08	90.07	90.29
		BIASED	100.0	95.00	90.75	94.25	91.50	92.50	92.40	91.35	92.17	90.33	90.47
	I_Λ	NESTED	90.00	99.00	93.00	97.50	97.75	92.67	92.70	97.15	96.08	96.75	97.59
		BIASED	100.0	100.0	100.0	100.0	98.25	94.25	94.80	97.95	96.67	97.17	97.73
	null	NESTED	20.00	66.00	53.00	52.50	51.75	57.92	44.90	51.95	46.90	49.30	49.22
		BIASED	60.00	77.00	67.75	69.50	59.50	63.58	52.00	57.75	50.63	51.60	51.32
Random Forest	I_I	NESTED	90.00	81.00	88.50	97.00	89.25	86.83	88.80	91.15	87.10	87.13	89.49
		BIASED	93.00	79.00	87.75	96.00	88.25	87.25	89.50	90.90	87.75	87.34	89.48
	I_4I	NESTED	92.00	69.50	67.25	77.50	79.75	81.42	83.90	84.90	86.78	87.21	88.08
		BIASED	88.00	75.00	72.25	78.50	79.50	80.83	84.10	86.45	86.77	86.99	88.08
	I_Λ	NESTED	80.00	92.00	95.00	94.00	92.50	90.08	93.20	95.80	96.65	97.24	97.57
		BIASED	83.00	90.00	93.00	96.25	93.75	90.58	92.10	95.80	96.88	97.22	97.45
	null	NESTED	88.00	48.50	56.00	59.25	51.50	50.67	48.60	52.00	50.77	47.98	50.14
		BIASED	84.00	49.00	57.75	63.50	49.50	52.50	48.30	51.80	51.28	48.52	49.82
Multilayer Perceptron	I_I	NESTED	98.00	78.00	89.75	89.50	80.50	89.17	86.40	88.70	87.95	87.72	89.22
		BIASED	100.00	79.50	93.25	91.75	82.50	87.25	85.80	88.75	87.40	87.98	89.49
	I_4I	NESTED	52.00	63.00	70.00	72.75	69.50	75.00	75.40	83.10	86.33	87.64	89.05
		BIASED	50.00	63.00	71.25	72.25	71.75	76.75	74.80	83.55	86.47	87.91	88.87
	I_Λ	NESTED	92.00	85.50	93.50	95.25	89.75	86.08	91.30	93.70	94.97	95.01	96.47
		BIASED	99.00	91.50	96.00	98.75	90.25	88.25	91.00	93.85	95.27	95.55	96.52
	null	NESTED	41.00	47.50	60.00	62.25	48.50	54.67	47.40	53.75	49.17	49.28	49.89
		BIASED	49.00	48.50	61.75	59.50	54.00	56.92	49.30	54.80	49.18	50.70	50.18

The performance scores of R rounds and K folds are stored in a $R \times K$ matrix which is post processed to obtain the final evaluation result. The simplest approach is drawing a conclusion, considering the average overall scores, however inference based on this simple strategy is misleading since it does not account for the uncertainty related to experimental procedures. It is also possible to process the matrix in a more sophisticated analysis to draw conclusions based on statistical hypothesis testing (Oliveira-Santos et al., 2018). Remember that the three results are usually obtained by different best sets of hyper-parameters. This is expressed in Algorithm 1 by the fact that the "Tuning" function returns a particular best set $\mathscr{P}_k^*$ in the kth of a total of K folds of the outer loop. In any case, the global result is much less biased than a conventional validation technique of Figs. 1 and 2. In the next section, this fact will plausibly be shown with the help of data with known densities and analytically tractable performance scores.

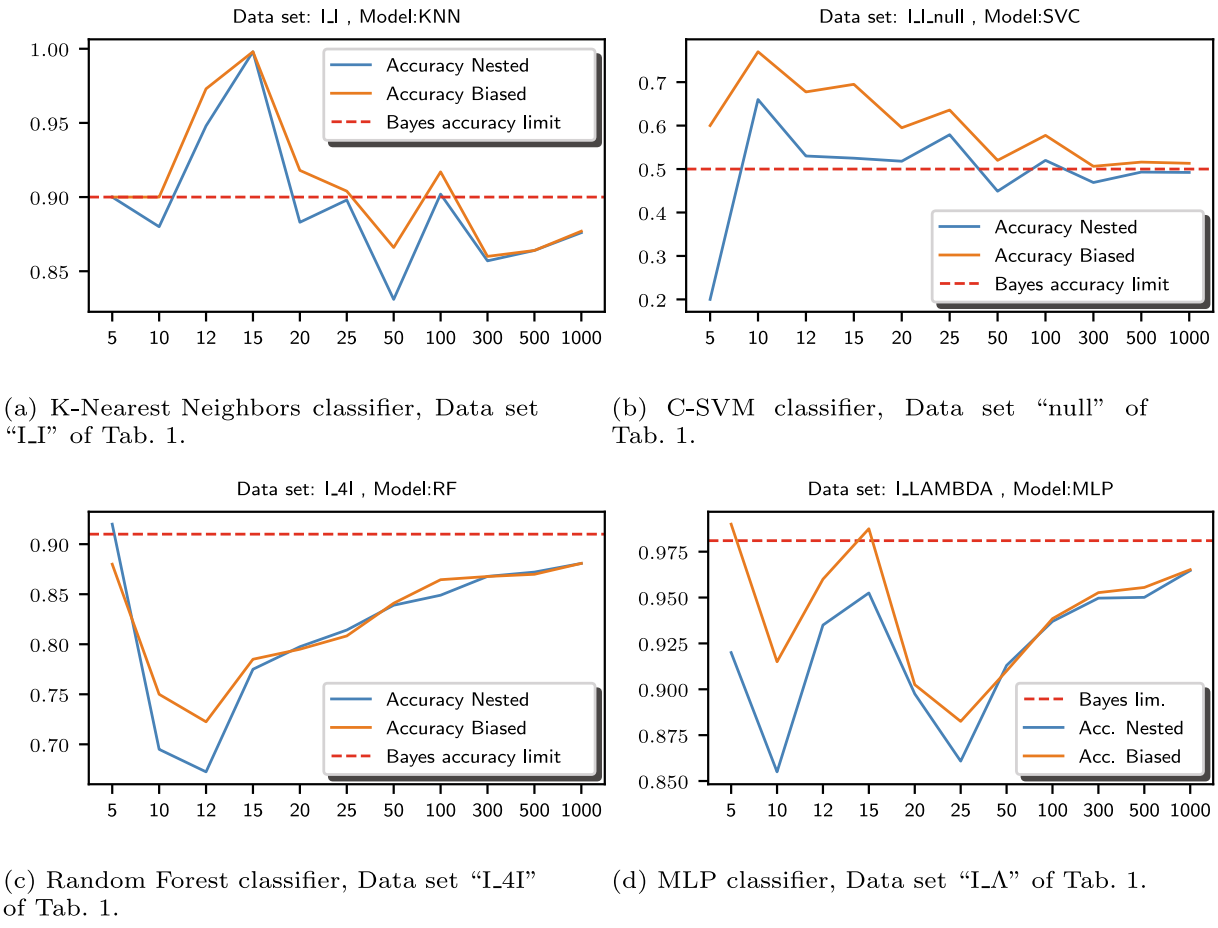
2.3. Performance experiments with known Bayesian limits

The pioneering work of Fukunaga (1990) in statistical pattern recognition proposed synthetic multivariate Gaussian data sets that allow to establish the theoretical Bayesian limit (Duda, Hart, & Stork, 2012), beyond which results are certainly biased towards overoptimism. These data sets can be used as a *meta-validator*. This means that a researcher can submit these data sets to its performance validation algorithm and classifier model, for instance, K-Fold cross validation together with a Random Forest. If a particular application in fault diagnosis has n data samples, exactly these amount of n random samples can be generated for the Fukunaga data sets, and then the performance estimation procedure proposed by a researcher could suggest an estimation of a priori overestimated performance scores, if the Fukunaga data sets are used as an input for the model proposed by the researcher.

2.3.1. Synthetic multivariate Gaussian data sets

Three data sets of 8-variate Gaussians are defined. Two classes $\mathscr{C}_a$ and $\mathscr{C}_b$ are characterized uniquely by their mean vectors $\boldsymbol{\mu}_a, \boldsymbol{\mu}_b$ and covariance matrices Σ_a, Σ_b, respectively. There is no covariance, i.e. the off-diagonal elements σ_{ij}, $i, j = 1, \ldots, 8$ in both Σ_a and Σ_b are zero. This is justifiable, because any two non-diagonal covariance matrices can be simultaneously diagonalized by a linear transformation. Moreover, a non-zero mean vector can be shifted to the origin without harm to the general case which is defined by non zero mean vectors and non zero covariances (Fukunaga, 1990). Therefore, only 32 parameters have to be specified to define the two probability density functions, two 8-dimensional mean vectors and twice eight variances on the two diagonals of the covariance matrices. The identical a priori probability $P(\mathscr{C}_a) = P(\mathscr{C}_b) = 50\%$ of each class is simulated by an equal number of n samples for each class. In order to ensure reproducibility by any researcher, for a random generation of n samples for each class, i.e. a total of $n * 2$ samples, the seed of the random number generator was set to 66649. The Python class `numpy.random` with method `numpy.random.normal` was used[2] to generate the samples. Table 1 resumes the specification of each of the three data sets defined by Fukunaga. Additionally the so called "null" data set (Varma & Simon, 2006) is included, where the two Gaussians have zero means, and unit variances. Hence the two classes overlap perfectly which gives rise to a 50% classification error, because the decision to which class a sample belongs to is equivalent to coin throwing.

[2] For instance, `numpy.random.normal(loc=numpy.zeros(8), scale=numpy.sqrt(numpy.diag(4*numpy.eye(8))), size=(100,8))` generates $n = 100$ samples of the second class of the "I-4I" data set, with zero mean vector $\boldsymbol{\mu} = \mathbf{0}$ and variances $\sigma_i^2 = 4.0$ on the diagonal, $i = 1, \ldots, 8$.



(a) K-Nearest Neighbors classifier, Data set "I_I" of Tab. 1.

(b) C-SVM classifier, Data set "null" of Tab. 1.

(c) Random Forest classifier, Data set "I_4I" of Tab. 1.

(d) MLP classifier, Data set "I_Λ" of Tab. 1.

Fig. 5. Comparison of simple K-Fold and nested cross validation for different classifier architectures and data sets. Number of samples per class is *x*-axis, estimated accuracy is *y*-axis.

2.3.2. Performance estimation experimental design

The data sets of Table 1 are submitted to performance estimation experiments to investigate especially the influence of the number n of samples and the effect of a nested cross validation. Although more sophisticated performance metrics could be used, the estimated accuracy of a two-class classifier will be measured. The objective of these tests is to suggest that few training samples in fault diagnosis are a severe problem, and that a nested cross validation can at least provide a less biased idea, compared to simple CV techniques. One of the main hypotheses of this work is to show that a nested cross validation, although not prone to overoptimistic performance scores, is always better than conventional cross validation.

Obviously a Quadratic Discriminant Function (Duda et al., 2012) is the ideal classifier, since it exactly models Gaussian densities. Nevertheless, other models, like e.g. a simple K-Nearest Neighbors classifier (Duda et al., 2012) will qualitatively accompany the scores of the ideal classifier, i.e. in case of biased, overoptimistic performance values, these will be visible in all classifier architectures. In this work, four quite distinct classifiers[3] are used, K-Nearest Neighbors (Cover et al., Jan. 1967), Support Vector Machine (C-SVM) (Vapnik, 2013) with RBF kernel, Random Forest (Breiman, 2001) and a Multilayer Perceptron (MLP) (Rumelhart, Hinton, & Williams, 1985) with one hidden layer and logistic activation function in the output layer. Later, in the context of fault diagnosis with the CWRU benchmark data, the MLP classifier is replaced by a One-dimensional Convolutional Neural Network (CNN) (LeCun & Bengio, 1995; Bengio, Goodfellow, & Courville, 2017). The Fukunaga Gaussian distributions do not have any temporal or structural relationships among the eight features. Therefore, it does not make sense to submit them to a CNN which performs the convolution operator in its first layer, and this operator does assume such a relationship, which is the case for a vibration signal. Table 2 shows the used architectures, together with all combinations of hyperparameters that are tuned during the nested cross validation, c.f. Fig. 3. For instance, the C-SVM with RBF kernel can vary two different hyperparameters, the penalty parameter C and the spread parameter γ of the Gaussian kernel. Each hyperparameter can have five, respectively four different values. Consequently, the Cartesian product creates 20 different (C, γ) options during the hyperparameter tuning, namely (0.001, 0.001), (0.001, 0.01), …, (10.0, 0.1), (10.0, 1.0).

2.3.3. Experimental comparison of simple and nested cross validation

In the following experiments, two CV techniques are compared. The first is the proposed framework of Fig. 3. The second is the conventional CV method. The simple CV of Fig. 2 is done in the following manner: For all possible combinations of the hyperparameter values, also used in the tuning stage of the framework, perform a conventional cross validation, here K-Fold. Report only the best result, i.e. that set of hyperparameters which were responsible for the highest performance score. This emulates the behaviour of a human that repeatedly tries to push the result towards

[3] The classes `sklearn.neighbors.KNeighborsClassifier`, `sklearn.sv.SVC`, `sklearn.ensemble.RandomForestClassifier` and `sklearn.neural_network.MLPClassifier` from the Python Machine Learning library scikit-learn were used to implement the four classifiers. Later, for the CWRU data, MLP was replaced by CNN, using the classes `keras.models.Sequential` and `keras.layers.Conv1D` from the the Python Deep Learning library Keras.

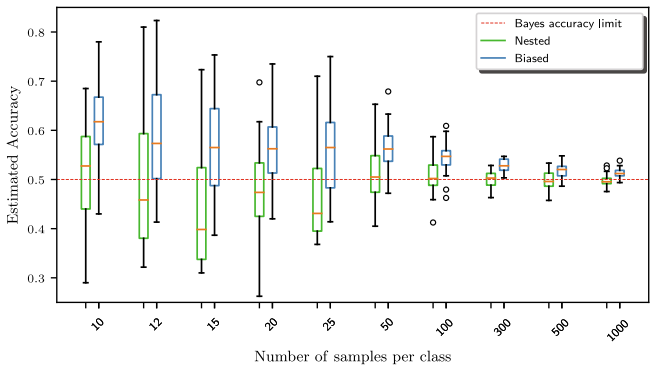


Fig. 6. Boxplots for nested cross validation compared to K-Fold cross validation.The Support Vector Machine classifier with the "null" data set is repeatedly submitted to the framework, using 30 different seeds.

optimality by varying the values of the hyperparameters, using the same train-test splits over and over again. Fig. 4 juxtaposes the boxplots of the simple and nested CV for a C-Support Vector Machine classifier, K-Fold cross validation $K = 10, L = 9$ and $n = 2 * 25 = 50$ samples on the "I_4I" data set. A total of 20 different combinations of the hyperparameter search, varying the pair (C, γ), are performed for tuning, c.f. Table 2. The simple CV called "Biased" in the figure, surpasses with 92.50% the theoretical barrier of Bayes accuracy of 91%. The nested CV called "Nested" in the figure, stays with 87.83% below the limit. Of course, the nested approach in general can exhibit better scores than theoretically possible, especially for a small number of samples.

In the next experiment, the extremely important number n of available samples is analyzed. For the synthetic data, the scores are based on $R = 10$ rounds, $K = 10$ folds, and $L = 9$ tuning folds. For the K-Nearest Neighbors classifier, the maximum number of neighbors was limited to five, since for a small number of samples, the number of neighbors cannot be higher than the number of training samples, hence the number of neighbors are selected from the set $\{1, 3, 5\}$. Table 3 shows the evolution of the estimated accuracy for the four different classifiers and the four different data sets of Table 1. The two approaches of performance validation, nested and conventional are compared to each other, gradually increasing the number n of randomly generated samples. Four typical examples are depicted in Fig. 5. The performance scores of this experiment suggest

1. For a small number n of samples in a supervised learning task, the performance estimation is almost always overoptimistic, independent of the cross validation method;
2. For a large number n of samples, simple and more sophisticated cross validation techniques asymptotically deliver the same results;
3. For a small number n of samples, the nested cross validation commonly delivers more conservative scores.

Fig. 6 focuses on Fig. 5b, repeating the experiment with different seeds in order to statistically compare the conventional and nested CV approaches as described in the next paragraph. A total of $S = 30$ seeds [4] is generated, based on a "metaseed". The 30 estimated accuracies are used to generate a pair of boxplots for each of the possible number of samples of Table 3, except for five samples which had a very high variance. Each pair compares the nested evaluation based on Algorithm 1 against the conventional K-Fold cross validation. It can be observed that the estimated accuracies approach the theoretical Bayes accuracy of 50% when the number of samples is increased. Simultaneously, the variance of the 30 values diminishes, but the overoptimistic bias of the conventional cross validation is clearly distinguishable from the more realistic estimation of the nested cross validation.

In order to draw final conclusions from the simulation study, two sets of statistical tests were conducted. In the first set of tests, the traditional version of the paired Student t-test (Casella & Berger, 2002) is employed to compare the average performance of both CV approaches for each sample size. The second set of tests investigates the bias of the cross validation methodologies by employing the one sample t-test to compare the results from each method with the theoretical Bayes accuracy (50% in this case). The results confirm the findings from the previous paragraph. Indeed, the first set of tests are highly significant with the maximum p-value being 1.36E−13, which was achieved for the test with the results from a sample size equal to $n = 12$. This means that the nested cross validation is a more conservative approach to report performances. With that conclusion in hand, one can access the overoptimistic bias of the conventional CV by applying the second set of tests. For example, for $n = 2000$, the t-test leads to a conclusion that the mean accuracy of the conventional CV is greater than the theoretical Bayes accuracy (p-value of 6.19E−08), whereas the same test does not reject the hypothesis that the average accuracy of the nested cross validation is the theoretical Bayes error rate (p-value = 0.257).

3. Important additional aspects of the proposed methodology

This section describes three important aspects of the proposed methodology: requirement of reproducibility, avoiding similarity bias, and verifying the statistical significance of the difference in the results.

3.1. Reproducibility

Missing or even contradictory information for the experimental design make it impossible for a researcher to compare the results of a paper to his own work, or verify the presented performance scores. A good quality paper should deliver a meticulous description of the experimental design to permit the scientific community to check the veracity of the performance criteria. Vibration signal data sets are usually composed of a set of files containing a single signal acquisition. Since this signal fundamentally is a unique, single sample of a certain machine condition, machine learning approaches often chop the signal into various chunks, in order to form the data samples used for training and testing the classifiers. The way in which this single file is transformed into a set of many distinct patterns is a fundamental question that strongly influences the performance of a diagnosis system. In order to allow reproducibility of the experiments, it is required that the paper describes how the chunks are extracted from the signals. It is important

[4] The seeds $\{6038151, 82966818, \ldots, 3997341\}$ were generated by invocation of `metaseed=66649; numpy.random.seed(metaseed); seeds=numpy.random.randint(0,100000000,30)`. Note also that the quantity of the seeds $S = 30$ is different from the number n of the generated samples. In the field of statistics, it is commonsense that sample sizes $S \geqslant 30$ are considered to be large enough for asymptotic normality.

to state the size of the signal chunks, if there is an overlapping of these segments or a gap in between, and all the necessary information required for other researchers to generate the same feature vectors from the files. Another aspect to be detailed is how the patterns are classified, i.e. what are the classes of the problem and how they are associated with the chunks extracted from the acquisition files. Furthermore, details must be provided, describing which patterns were used to train, validate and test the classifiers along with how many folds were used in the inner loop and outer loop, how many rounds and also which seeds of the random number generators were used for dividing the patterns in each fold of each round. The choice of the hyperparameters is essential for the performance results. This issue was discussed in Section 2, highlighting the fact that the values are manually tuned, erroneously based on the same cross validation of the available data. A similar problem is the frequently observed use of the default values of the classifier models of a software toolbox, for instance, the Matlab neural network toolbox or the Python scikit-learn toolbox, and then to affirm, for instance, "The accuracy of the SVM was 99.8%", without mentioning which kernel and further associated hyperparameters were employed. If the hyperparameters were automatically tuned, the method used for tuning, e.g. grid search, should be specified along with the values of the hyperparameter combinations that were tried. Machine learning approaches seldom use the raw vibration signal. They often extract features from the signals to compose the feature vector describing the patterns. The methods used for extracting the features and their eventual parameter values should also be explicitly described. Recently, there is a growing practice of publicizing both the data and the code used for performing the experiments. This is the ultimate approach for allowing reproducibility and verifiability. Thus, it is strongly recommended that well documented data and codes should be exposed in some public repository.

3.2. Avoiding similarity bias

Once several patterns are extracted from files containing a single signal acquisition, these patterns are typically very similar. If a machine learning experiment uses patterns extracted from the same file for both training and testing the classifiers, the task of classification becomes relatively trivial, almost an exercise of consulting identical patterns in a database table, resembling a memorization process. This work denominates this issue as the similarity bias problem. In order to avoid similarity bias, researchers should guarantee that the division of folds for training and testing does not allow that patterns extracted from one file be present in more than one fold. Probably, there are different ways how to distribute the patterns in the folds, in a manner that avoids the similarity bias. Therefore, the division should be carefully designed, trying to keep the folds stratified and with equivalent size. In order to circumvent the similarity bias in the hyperparameter tuning process, it is recommended that the fold division used in the outer loop should also be maintained in the inner loop and that the value of L of inner loop folds should be equal to $(K-1)$ in Algorithm 1, where K is the number of outer loop folds.

However, it is best to keep in mind that depending on the way vibration data signals are collected, it is not possible to avoid the similarity bias. If the data signals representing one machine condition is collected in a single acquisition session, there is no way of separating patterns from different acquisition sources in the training and testing subsets.

3.3. Statistical analysis

Analysis based on only averaged results may guide the researcher to erroneous conclusions since variability is not considered in this simple strategy. A more reliable approach is to verify the statistical significance of the difference in the results via statistical hypothesis testing. A traditional procedure to statically detect difference between two groups in a paired experiment is the paired Student t-test (Casella & Berger, 2002). In the context of performance evaluation of two classifiers, the t-test is a possible alternative to detect differences only if the performance evaluation is carried out over several data sets, since one of the assumptions of the t-test is independence among samples. This is not the case for comparisons based on the performance scores of a cross validation study over a single data set, since the intersection of the training sets is not the empty set. Note that non-parametric alternatives to the t-test, as the Wilcoxon technique, also suffer from the same problem, since they require the independence among samples. Nadeau and Bengio (2000) showed that the violation of the independence yields underestimation of the standard deviation. In order to overcome this shortcoming, the authors use the ratio of the size of training and test sets to estimate the correlation derived from the overlapping training sets. Finally, the authors correct the usual t-test statistics to propose the statistic in Eq. (1) to conduct the test. In that equation $\overline{d}$ and s^2 are the mean and variance of the differences in performance of the two classifiers. The values n_{TE} and n_{TR} are the size of the training and test sets, respectively. The total number of test sets used is J, therefore considering the cross validation in Section 2.2.2 results in $J = K \times R$. Small values of t suggest similar performance between the classifiers, whereas large absolute values indicate significant difference in performances. The p-value of the test is given by $2{\cdot}P(T > |t|)$, where T is a random variable following a Student's t-distribution with $(J-1)$ degrees of freedom. It is worthwhile to note that in order to avoid similarity bias, the size of the folds may differ, and therefore the ratio $n_{\mathrm{TE}}/n_{\mathrm{TR}}$ is not constant across the evaluation process. For this case we suggest replacing n_{TR} and n_{TE} by the total number of samples used as training and test sets, respectively. Under these circumstances the ratio $n_{\mathrm{TE}}/n_{\mathrm{TR}}$ is equal to $1/K$.

$$t = \frac{\overline{d}}{\sqrt{s^2\left(\frac{1}{J} + \frac{n_{\mathrm{TE}}}{n_{\mathrm{TR}}}\right)}}. \tag{1}$$

In applications where performance comparison among three or more methods is required, one can perform the same statistical analysis for each pair of classifiers, however adjusting the p-values to control the family-wise error, i.e. the cumulative Type I error. In order to control the family-wise error, several statistical solutions have been proposed. Oliveira-Santos et al. (2018) have employed Holm's step-down procedure (Holm, 1979) to compare the results of twenty different classifiers. In this work, we suggest the use of the Benjamini-Hochberg procedure (Benjamini & Hochberg, 1995). This solution is more effective and powerful than Holms' approach under non-negative dependence, which is the case in this work, since the overlapping in the training sets leads to positive correlations. Benjamini-Hochberg's procedure starts by increasingly ordering the p-values associated with each pairwise test, $p_{(1)},\ldots,p_{(m)}$, where m is the total amount of tests required, i.e the number of pairwise comparisons. Then, for a given significance level α, let k be the largest i such that

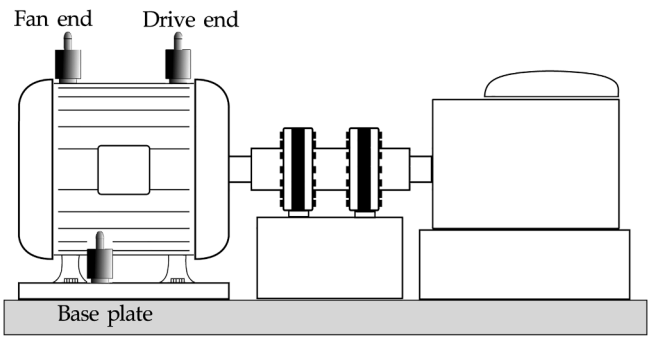


Fig. 7. CWRU test bed for bearing faults diagnosis.

$$k = \underset{i}{\operatorname{argmax}} \left\{ p_{(i)} \leqslant \frac{i}{m}\alpha \right\}. \quad (2)$$

Finally, it rejects the hypotheses of similarity in performance of comparisons associated with the p-values $p_{(1)}, \ldots, p_{(k)}$.

Another well-known assumption of the paired t-test or its corrected version described in this article is that samples are drawn from a population that is known to be normally distributed. If this requirement is not satisfied, a large sample size is necessary to assure asymptotic normality of the sample mean, as established by the central limit theorem (Casella & Berger, 2002). In the field of statistics, it is commonsense that sample sizes equal to or greater than 30 are considered to be large enough for asymptotic normality. Therefore, in order to satisfy this condition, it is suggested to develop a experimental framework so that the total number of test sets is $J = K \times R \geqslant 30$ to perform the statistical comparison. In situations where this arrangement $(J \geqslant 30)$ is impossible, the methodology described in this subsection to perform statistical comparison should not be conducted, unless the differences between pairs are known to be normally distributed.

4. Related research in vibration based fault diagnosis

Section 2 and Section 3 present an experimental methodology for evaluating model-free machine learning methods for fault diagnosis based on vibration signals. This methodology requires that experiments should use nested cross validation as their evaluation method, be completely described to allow their reproducibility, avoid the similarity bias problem and apply appropriate statistical tests for showing significant differences between the proposed methods and other state of the art methods from the research literature.

In this section, we claim that no related research work has achieved these requirements. Indeed, except for one paper, none has achieved at least two of these four requirements. This review shows research work that used many different vibration data sets, however it spends more effort reviewing works that use the CWRU data set, by far, the most used vibration based data set in machine learning fault diagnosis research. Section 4.1 reviews research works using other data sets, while Section 4.2 reviews works that use CWRU data.

4.1. Non CWRU vibration based research in fault diagnosis

Vibration signal based model-free research works use different data sets for evaluating their approaches. Some works use both the CWRU data set and an additional data set in their experiments.

The second most used data set is the publicly available bearing data from the center for Intelligent Maintenance Systems (IMS), University of Cincinnati (Lee, Qiu, Yu, & Lin, 2007). The IMS data was collected in run-to-failure tests under constant loads on a specially designed test rig. Although the experimental data was designed for fault prognosis, it has been extensively used for evaluating fault diagnosis methods. Ali, Fnaiech, Saidi, Chebel-Morello, and Fnaiech (2015) use empirical mode decomposition (EMD) and artificial neural networks for automatic bearing fault diagnosis based on the IMS vibration signals. A total of 3911 samples from seven class conditions are evaluated, using 5-fold conventional cross validation. Zhang et al. (2017) use deep neural networks for detecting faults from IMS raw time domain vibration data. The set of 10,880 samples are divided according to a single train-test split for evaluating the proposed method. The paper does not reveal the number of samples used for training and for testing. Berredjem and Benidir (2018) apply fuzzy rules, automatically induced from numerical data for bearing faults diagnosis. For automatically inducing the fuzzy rules, the similarity partition method is proposed. Experiments were performed using a single train-test split strategy, where 50 samples are used for training and 50 samples for testing. Zhang, Li, Gao, and Li (2018) propose a subset based auto-encoder model feature learning method for fault diagnosis. A set of 1400 samples, representing seven health conditions, use a single train-test split with 560 samples for training and the remaining 840 samples for testing. A total of 20 rounds of training and test are performed. Li et al. (2020) propose a deep learning-based method to transfer fault diagnosis knowledge learned from one machine to different, but related machines. The work only considers the health condition and outer race faults at different levels (incipient, medium e severe), therefore, using four classes, and 600 samples for training and 100 for testing in a 5-round single train-test split strategy.

Verstraete et al. (2017) propose a deep learning enabled featureless methodology to automatically extract the features from the data. Time-frequency representations of the raw data are used to generate image representations of the raw signal, which are then fed into a deep 2-D convolutional neural network (CNN) architecture for classification and fault diagnosis. The paper describes experiments for a publicly available vibration data set provided by the Machinery Failure Prevention Technology (MFPT) Society (Bechhoefer, 2016). From vibration signals acquired from 18 different conditions, the authors generate 3423 images from the health condition, 1981 images from the inner race fault condition and 5404 images of the outer race fault condition. Each image is used as a sample of their fault diagnosis data set, with a single train-test split, using 70% of the data set for training and the remaining 30% for test. Zhang et al. (2018) also perform fault diagnosis experiments for a publicly available vibration signals data set collected from a self-priming centrifugal pump (Lu, Wang, Ragulskis, & Cheng, 2016). Experiments consist of 20 rounds for a single train-test split, where 200 of the 500 data set samples are used for training, while the remaining 300 samples are used for testing. Lei et al. (2016) use deep learning to directly learn features from mechanical vibration signals and classify the health conditions based on the extracted features. The proposed method is evaluated on a non publicly available locomotive bearing data set, with 7644 samples obtained from the vibration signals. For training the model, 764 samples (10%) are used, while the remaining samples (90%) are used for testing. Thus, they also use a single train-test split strategy. Liao et al. (2019) use an one-dimensional convolutional neural network and gated recurrent unit in the hydroelectric generating unit fault diagnosis. The non publicly available data set consists of vibration data collected from the Xianghu hydropower station, Fujian, China. Vibration signals from different time periods were picked as faulty (540 samples) and healthy data (540 samples), respectively. The authors use a single train-test split, although they separate a part of the training set for validation. The training set, the validation set, and the test set are randomly split with the ratio 8:1:1, therefore, 864 samples are used for training, 108 for validation and 108 for testing.

In all of the previously mentioned experiments, similarity bias can be stated, since many samples are extracted from the same acquisition files, no statistical tests were performed, and reproduction is unviable due to the lack of information of how the samples were generated, how they were divided into the training and testing data sets, or how the hyper-parameter tuning was performed.

Marins, Ribeiro, Netto, and da Silva (2018) use a similarity-based modeling method for the classification of rotating-machine failures from vibration signals. The vibration data set is publicly available (MaFaulDa, 2016). Vibration data is collected from 1951 different acquisitions scenarios. Each scenario is used as a different sample in the experiments. Thus, in this case, there is no similarity bias. For training, 90% of the samples are used, and the remaining 10% are used for testing in a single train-test split strategy. Except for not publishing the random seeds that were employed, the experiments are well described. Hence, the reproducibility requirement is fulfilled. Nested cross validation is not used and no statistical tests are performed.

4.2. CWRU related research in fault diagnosis

The CWRU Bearing Data Center (CWRU, 2014) is a publicly available vibratory repository widely cited in the scientific literature (Rauber, de Assis Boldt, & Varejão, 2014). The philosophy of providing such a data

Table 4
CWRU Articles categorized with respect to number of data points per sample, number of samples per file, sample overlap and position of the accelerometers.

Article	Data points per sample	Samples per data file	Overlap	Acquisition position
Sreejith et al. (2008)	6050	20	None	DE
Zhao et al. (2011)	Unreported	76	Unreported	Unreported
Yiakopoulos et al. (2011)	Unreported	Unreported	Overlap	DE
Yu (2011)	Unreported	30	Overlap	DE
Wu et al. (2012)	2048	Varied	Overlap	DE – FE
Wu et al. (2013)	2048	Varied	Overlap	DE
Rodriguez et al. (2013)	1024	Varied	Unreported	DE
Shen et al. (2013)	1024	90	None	FE
Xu and Chen (2013)	1024	70	None	DE
de Assis Boldt et al. (2014)	Varied	Varied	None	DE – FE
Zhao et al. (2014)	Unreported	80	Unreported	DE
Rauber et al. (2014)	Varied	15	None	DE – FE
de Assis Boldt et al. (2015)	Varied	Unreported	None	DE – FE
Yu et al. (2015)	2000	60	None	DE
Li et al. (2015)	1024	250	Overlap	DE
Lei et al. (2016)	1200	400	Overlap	DE
Han and Jiang (2016)	Unreported	100	Unreported	Unreported
Kavathekar et al. (2016)	Unreported	55	Unreported	DE
Guo et al. (2016)	1024	1000	Overlap	DE
Gan et al. (2016)	Unreported	50	Overlap	DE
Zhang et al. (2017)	100	1210	None	DE
Zhang et al. (2017)	400	Varied	Overlap	DE
Zhang et al. (2017)	2048	Varied	Unreported	DE
Razavi-Far et al. (2017)	1024	Varied	Overlap	DE - FE
Pan et al. (2017)	Unreported	400	Overlap	DE
Wen et al. (2017)	1024	300	Overlap	DE
Verstraete et al. (2017)	1024	Unreported	Unreported	DE - FE
Ding and He (2017)	512	100	Overlap	DE
Shao et al. (2017)	Unreported	20	Unreported	DE
Marins et al. (2018)	Varied	15	Overlap	DE - FE
Zhang et al. (2018)	2048	Varied	Unreported	Unreported
Yang et al. (2018)	Varied	Varied	None	DE - FE
Berredjem and Benidir (2018)	Unreported	59	None	DE
Li et al. (2019)	1024	150	Overlap	DE
Li et al. (2019)	2048	Varied	None	DE - FE
Hoang and Kang (2019)	200	606	None	DE
Eren et al. (2019)	512	Varied	Overlap	DE
Piltan et al. (2019)	Unreported	Unreported	Unreported	DE
Sun et al. (2019)	Unreported	Varied	Overlap	DE
Li et al. (2020)	1024	Varied	Overlap	DE
Yin et al. (2020)	Unreported	Varied	Overlap	DE

set is to allow the research community to test their methods in order to find a sophisticated system that allows the correct diagnosis of the machine condition. Machine learning techniques are used here as the basic tool to analyze the provided signal data. Unfortunately, the data set is not accompanied by a manual of how to perform classification experiments, which leaves too much room to interpretation. For instance, a single Matlab file contains raw vibration data from two or three accelerometers, with a duration of approximately ten seconds. A classifier based on the supervised learning paradigm, must have training and test sets, but there is no recommendation of how to split the raw signal. How many patterns can be generated by the whole signal, chopping it into parts? May the parts be overlapping? What is the minimum length of a signal chunk before any method degenerates too much to provide reliable results? How must the data be divided into different sets, in order to avoid biased results? In the following, a description of the data set is presented, and how different research work is handling these questions.

4.2.1. CWRU data description

The data set is composed of vibratory signals of normal and defective bearings, extracted from a 2 hp reliance electric motor. The faults are introduced at a specific position of the bearing, using an electro-discharging machining with fault diameters of 0.007, 0.014, 0.021 and 0.028 in. A dynamometer induces loads of 0, 1, 2 and 3 hp, as an effect changing the shaft rotation from 1797 to 1720 rpm. One model of bearing is used on the drive end (6205-2RS JEM SKF) and the other is used on the fan end (6203-2RS JEM SKF). Three accelerometers, placed on the drive end, fan end and the base of the motor, collected the vibration signal, as shown in Fig. 7. The collected data are available in the form of Matlab files (*.mat). Outer raceway faults are sensitive to their load zone. Thus, there are acquisitions for both positions, fan end, and drive end bearings, with outer raceway faults located at 3 o'clock (directly in the load zone), at 6 o'clock (orthogonal to the load zone), and at 12 o'clock. The identified classes can be labeled according to the number of bearings, the bearing state (normal or defective), fault location (when the bearing is faulty), fault severity (depth) and motor load. For outer raceway faults, it is also possible to identify the load zone. Usually, works found in the literature identify ten conditions. In Rauber et al. (2014), 19 conditions are identified. The present work identifies only four distinct conditions by merging samples from different CWRU files, c.f. Section 5.1.

4.2.2. Review of CWRU fault diagnosis research

The following compilation reviews existing research papers, based on a machine learning approach. All works present quantitative performance scores, mainly the estimated accuracy of the learned fault classifier. Many results are questionable since they employ a simplistic one-time training-test split, reuse the same data in several stages of the classifier learning, or present irreproducible, and thus unverifiable methods. In Table 4,5, a compilation of representative scientific publications that work with the CWRU data is listed. The papers are preliminarily categorized with respect to number of data points that

Table 5
Related CWRU Articles - Experimentation methodology.

Article	Evaluation Method	Number of Classes	Samples total	Samples train	Samples test	Rounds	Tuning	Similarity Bias	Accuracy
Sreejith et al. (2008)	Single train-test split	4	80	48	32	One	None	Yes	100
Zhao et al. (2011)	Single train-test split	10	760	600	160	Many	Automatic	Yes	96.00
Yiakopoulos et al. (2011)	Resubstitution	4	8192	8192	8192	One	None	Yes	100
Yu (2011)	Single train-test split	4	296	150	146	One	None	Yes	Unreported
Wu et al. (2012)	Single train-test split	4	3181	Varied	Varied	Many	None	Yes	99.55
Wu et al. (2013)	Single train-test split	6	10776	Varied	Varied	Many	None	Yes	98.81
Rodriguez et al. (2013)	Single train-test split	4	40	20	20	Many	Automatic	No	90.00
Shen et al. (2013)	Single train-test split	4	720	240	480	One	None	Yes	100
Xu and Chen (2013)	Single train-test split	6	420	240	180	One	None	Yes	Unreported
de Assis Boldt et al. (2014)	Leave-one-out	21	2295	2294	1	One	None	Yes	99.96
Zhao et al. (2014)	Single train-test split	10	800	Varied	Varied	Many	None	Yes	97.65
Rauber et al. (2014)	Leave-one-out	19	2295	2294	1	One	None	Yes	99.97
de Assis Boldt et al. (2015)	Leave-one-out	4	Unreported	Unreported	1	One	None	Yes	98.27
Yu et al. (2015)	Single train-test split	11	660	220	440	One	None	Yes	99.64
Li et al. (2015)	Single train-test split	4	1000	500	500	One	None	Yes	100
Lei et al. (2016)	Single train-test split	10	4000	400	3600	One	Manual	No	99.66
Han and Jiang (2016)	Single train-test split	4	400	Varied	Varied	Many	None	Yes	99.29
Kavathekar et al. (2016)	Resubstitution	4	440	440	440	One	None	Yes	73.21
Guo et al. (2016)	Cross validation	4	1000	900	100	One	None	Yes	97.90
Gan et al. (2016)	Single train-test split	10	1000	500	500	Many	Manual	Yes	99.93
Zhang et al. (2017)	Single train-test split	6	7260	Unreported	Unreported	One	Manual	Yes	94.40
Zhang et al. (2017)	Single train-test split	6	6040	4832	1208	One	Manual	Yes	84.20
Zhang et al. (2017)	Single train-test split	10	6850	6600	250	One	None	Yes	100
Razavi-Far et al. (2017)	Nested cross-validation	7	3060	2754	306	One	Automatic	Yes	Unreported
Pan et al. (2017)	Single train-test split	4	1600	800	800	One	None	Yes	99.63
Wen et al. (2017)	Single train-test split	7	2400	2000	400	Many	None	Yes	99.79
Verstraete et al. (2017)	Single train-test split	4	11753	8227	3526	One	None	Yes	100
Ding and He (2017)	Single train-test split	10	1000	500	500	One	None	Yes	99.80
Shao et al. (2017)	Single train-test split	16	320	160	160	One	None	Yes	100
Marins et al. (2018)	Cross validation	19	2295	2065	230	One	None	Yes	98.95
Zhang et al. (2018)	Single train-test split	10	6850	6600	250	One	None	Yes	99.50
Yang et al. (2018)	Single train-test split	8	2612	2090	522	One	None	Yes	99.77
Berredjem and Benidir (2018)	Single train-test split	3	1008	504	504	Many	Manual	Yes	98.31
Li et al. (2019)	Single train-test split	4	600	300	300	One	None	Yes	99.74
Li et al. (2019)	Single train-test split	4	840	420	420	One	None	Yes	100
Hoang and Kang (2019)	Single train-test split	4	2424	2024	400	One	Manual	Yes	99.83
Eren et al. (2019)	Cross validation	6	3630	3267	363	One	None	Yes	99.93
Piltan et al. (2019)	Resubstitution	4	Unreported	Unreported	Unreported	One	Manual	Yes	100
Sun et al. (2019)	Single train-test split	10	6000	Varied	Varied	Many	None	Yes	98.03
Li et al. (2020)	Single train-test split	4	700	600	100	Many	Manual	Yes	100
Yin et al. (2020)	Single train-test split	10	4100	Varied	Varied	Many	None	Yes	99.00

compose a single sample, the number of samples extracted from one file, if the extraction came from overlapping signal chunks, and which accelerometers data were used. As mentioned earlier, each machine condition is characterized by a single file. If more than one sample is required for machine learning approaches, the division of this single signal has to be defined. i.e. how the time intervals are defined. It is also important to know if there was an overlap during this division, since overlapping implicitly means using knowledge more than once. This can also be a source of overoptimism. Finally, some papers use only the front end (FE) accelerometer, others also use the drive end (DE). Table 4 shows that many articles do not report these aspects, making reproducibility unviable. It can still be observed that there is a very large variation on how the works use the CWRU data.

Table 5 characterizes the works of Table 4 with respect to the method to estimate the performance of the fault diagnosis system, how many machine condition classes were defined, how many samples are in the whole data set, how the samples were divided into training and test data sets, how many rounds (c.f. Section 2.2.2) were executed to calculate the performance scores, how the hyperparameter tuning was performed, if patterns of the same data source were used for training and performance estimation, and finally, the reported performance which for all reviewed papers was always the estimated accuracy. As one may see, the vast majority of publications used the simple *single train-test split* method to estimate the performance of their proposed fault diagnosis system. Only one work used *nested cross-validation*, however with only one round, and similarity bias. Most works only perform one round of training and testing. The hyperparameters tuning is rarely done, with few works using an automatic procedure. It is most likely that the hyperparameters of the system are manually defined a priori, but this is not even mentioned in the papers. Finally, except for one, all research papers share the similarity bias problem defined in Section 3.2.

5. Proposal for CWRU systematic performance comparison

This section initially describes how the CWRU signal files were used to define the classes of the fault diagnosis problem and generate the patterns used for training and testing the classifiers. It also describes which feature extraction models were applied on each chunk of data. The second part of the section presents three proposals for the distribution of patterns with different degrees of difficulty. The first form of division is proposed without concerning the effects of the similarity bias problem. The second and third forms of division take into account the similarity bias, but with a gradual increase in difficulty.

Table 6
Fold separation in eight rounds for the load and severity generalization experiment.

	(Load, Severity)			
Round	Fold 1	Fold 2	Fold 3	Fold 4
1	(0, 7)	(1, 14)	(2, 21)	(3, 28)
2	(3, 7)	(0, 14)	(1, 21)	(2, 28)
3	(2, 7)	(3, 14)	(0, 21)	(1, 28)
4	(1, 7)	(2, 14)	(3, 21)	(0, 28)
5	(0, 14)	(1, 7)	(2, 21)	(3, 28)
6	(3, 14)	(0, 7)	(1, 21)	(2, 28)
7	(2, 14)	(3, 7)	(0, 21)	(1, 28)
8	(1, 14)	(2, 7)	(3, 21)	(0, 28)

5.1. Class definition, feature model and sample acquisition

The idea of this paper is not the proposal of yet another feature model or classifier architecture to improve nearly 100% accuracies observed in the reviewed literature. Two feature models from Rauber et al. (2014) are used: Classical statistical features from the time and frequency domain are merged with a feature vector of wavelet package coefficients. All signals are divided into non-overlapping 512 point chunks to generate a single pattern by feature extraction. As an example, the file `234.mat` stores the CWRU signal that contains an outer race defect with a severity of the drilled hole with a 0.021 in diameter. It contains 122,426 signal points, corresponding to a duration of $10.2021\overline{6}s$. Segmented into chunks of 512 points, it delivers 239 samples, discarding the last 58 points. Only signals acquired at 12 kHz are used in this study. Further, signals collected on different positions from where the fault occurs are discarded, e.g. signals collected in the Fan End (FE) and Base Plate (BP) are not used in the study when the fault occurs in the Drive End (DE). Overall, 113 signals acquisitions from 109 files generated 31521 samples that were used in the case study.

Each sample is subjected to the feature extraction model defined in Rauber et al. (2014) to form the feature vector used in the machine learning methods. For each bearing state, fault location, fault severity, motor load and load zone, there is only one acquisition file. This is a drawback when the CWRU data set is used in machine learning experiments, because the supervised learning approach needs a considerable amount of training and test samples to provide unbiased results. Often, the works of Table 4 define as a class the signal contained in a single CWRU file, like the example `234.mat` mentioned above, subsequently dividing the whole signal into contiguous non-overlapping or overlapping intervals. In this paper, a class is defined in more general terms, merging accelerometer positions (DE and FE), simulated loads (0 hp, 1 hp, 2 hp, 3 hp), and severities (0.007 in, 0.014 in, 0.021 in, 0.028 in). This approach syntonizes more with real fault situations, since in case of the occurrence of a problem, the severity, position, or other characteristics are inaccessible. Hence, a priori, there are only four classes: Normal, inner race, outer race and ball. The particular compositions of the files depend on the experimental modalities, described in the following.

Table 7
Hyperparameters used for each classifier architecture, CWRU data.

Method	Hyperparameters	Range
KNN	Number of neighbors	$\{1,3,5,\ldots,15\}$
SVM	C-SVM C, $\log_{10}(.)$	$\{-3,-2,-1,0,1\}$
	RBF Kernel γ $\log_{10}(.)$	$\{-3,-2,-1,0\}$
RF	Number of trees (estimators)	$\{10,20,50,100,200,500\}$
	Number of features for split	$\{1,5,10,15,20\}$
CNN-1-D	Filters	$\{16,32\}$
	Kernel size	$\{16,32\}$
	Number of convolutional layers	$\{1,2\}$
	1-D Pooling window size	$\{8\}$

5.2. Experimental modes

The evaluation framework proposed in Algorithm 1 bears an intrinsic problem. When an acquisition signal is split into many samples, one sample carries information of the acquisition itself, not only failure information. Thus, the nested cross-validation may still produce over-optimistic results. The memorization experimental mode applies the framework without worrying about this problem. One way to have more reliable results is not to allow samples originated from the same acquisition in the same fold. Therefore, in the generalization by load experimental mode, the folds are divided by load to have a 4-fold cross-validation where the samples of some load are used as the test fold while the remaining are joined in the training fold. This evaluation approach reduces the influence of the acquisition in the samples. The generalization by load and fault severity experimental mode provides a scenario where the performance scores drop considerably. It also verifies the capacity of the machine learning method to adapt to different conditions, because the operational point of the test samples is not present in the training data set. Besides, in real industrial processes there is no guarantee that the training data set has all operational points, which would produce a more realistic feedback about the diagnosis system.

5.2.1. Memorization

This is the usual categorization of the data samples found for model-free machine learning fault diagnosis based on vibration signals. This type of division is called memorization because the generated classifiers are similar to memorizers capable of remembering patterns previously seen without great generalization capacity. Frequently, in the works compiled in Table 4, one CWRU file is equivalent to one class, i.e., one class is defined by samples belonging to the CWRU signals. In this proposal we merge many machine conditions into a single class, for example, the "inner race" class is fed by the samples from all[5] 28 files severities, sensor positions and motor loads associated to the inner race defect. The experiments are executed as an instance of the K-Fold nested performance validation framework of Fig. 3. Exactly, like in the figure, four folds are separated. This choice of $K = 4$ is motivated by the subsequent experiments where the folds are related to the four distinct motor loads. Moreover, the number of rounds was set to $R = 8$ in order to obtain a total number of statistical tests $J = 4 \times 8 = 32 \geqslant 30$, aiming to achieve asymptotic normality as discussed in Section Section 3.3.

5.2.2. Generalization by load

Considering Fig. 3, in each cycle, the test round contains a value of the motor load that is not present in the training and validation folds. For instance, in the first cycle, the 0 hp load is only present in the test fold, the training and validation folds contain loads from 1 hp, 2 hp and 3 hp. The objective of this experimental mode is to validate the generalization capability with respect to a single variable operational state of

[5] In the CWRU files, not all combinations of the Cartesian product of all conditions are present.

Table 8
Memorization – Accuracy and F1-Score

Round	KNN	SVM	RF	CNN-1-D
		Accuracy		
1	0.9761	0.9849	0.9884	0.9839
2	0.9761	0.9849	0.9882	0.9898
3	0.9762	0.9851	0.9882	0.9799
4	0.9759	0.9849	0.9884	0.9744
5	0.9764	0.9846	0.9884	0.9819
6	0.9763	0.9848	0.9880	0.9899
7	0.9760	0.9849	0.9883	0.9843
8	0.9755	0.9850	0.9883	0.9767
Mean	0.9761	0.9849	0.9883	0.9826
		F1 – Score		
1	0.9757	0.9851	0.9885	0.9843
2	0.9756	0.9850	0.9883	0.9899
3	0.9758	0.9852	0.9884	0.9799
4	0.9755	0.9851	0.9885	0.9756
5	0.9760	0.9848	0.9885	0.9823
6	0.9759	0.9850	0.9881	0.9899
7	0.9756	0.9850	0.9884	0.9842
8	0.9751	0.9852	0.9884	0.9773
Mean	0.9756	0.9851	0.9884	0.9829

Table 9
Matrix of results of the pairwise comparison between classifier models for the memorization mode. The upper triangle presents the corresponding *p*-values of the corrected *t*-test for each pair of methods when accuracy is assumed as the performance criterion. The lower triangle shows similar results based on the F1 score.

		p-value for accuracy		
p-value for F1-Score	KNN	4.8E−09	2.4E−12	0.3573
	2.4E−09	SVM	0.0001	0.7473
	6.6E−13	0.0001	RF	0.4216
	0.2873	0.7527	0.4197	CNN

the process, namely the motor load in this case.

Therefore, data is split into four folds ($K = 4$), one fold for each type of motor load, in order to avoid the similarity bias. Note that any other composition of folds would lead to examples of the same motor load in the training and test folds, and therefore, the execution of several rounds would not produce extra variability, since it can only be done by repeatedly shuffling the samples within each fold. For this reason, the number of rounds is set to $R = 1$ for this generalization mode, with the consequence that no statistical test can be performed.

5.2.3. Generalization by load and fault severity

This data division is the most challenging classification experiment. It resembles more a realistic situation where one wants to diagnose a fault, even if the machine condition has never occurred before. For instance, the severity of the fault is a characteristic that in the industrial environment is not so neatly specified as in the CWRU data. In a supervised learning context, it can be expected that the fault diagnosis accuracy degrades. This can be attributed to the absence in the training set of patterns representing exactly that machine condition. Again, the framework of Fig. 3 is used. Each fold is a pair (motor load – fault severity) that only occurs once in the test fold. The division is depicted in Table 6. Each of the $R = 8$ rounds additionally has different combinations of the load–severity pair. For instance in round three, if the pair (3, 14) is defined as the test fold, the remaining folds (2, 7), (0, 21) and (1, 28) are used for training and hyperparameter tuning. The motivation to have a pair is that the normal machine condition is only distinguishable by the different loads. Note that the second part of the pair, the files for a certain severity are composed by all four load values.

6. Experiments

In the following we report classification experiment results with two principal objectives. Firstly, the tests are a practical application of the proposed nested performance evaluation. Secondly, a more realistic scenario for fault diagnosis is created by the fusion of several machine conditions into a single class. Moreover, in order to evaluate their generalization capabilities, the classifiers are subjected to data that has never been seen during training. We qualitatively show that the high performance scores reported in the majority of publications are unrealistic for a real world fault diagnosis environment.

Three of the four different classifier architectures presented in Section 2.3.2 are tested for the three modes described in Section 5.2. The MLP classifier was replaced by a One-dimensional Convolutional Neural Network (CNN) (LeCun & Bengio, 1995; Bengio et al., 2017). The replacement aims to show the proposed framework works both with traditional feature extraction, selection and classification machine learning approaches and also with deep machine learning approaches where these steps are all embedded in one unique processing pipeline.

Due to the different nature of the synthetic data set of Fukunaga and the CWRU signals, the modified set of hyperparameters in Table 7 was used. For instance, the synthetic data start with a very low number n of samples to analyse the influence of a limited data set. Then it is not possible to use many neighbors for the K-Nearest Neighbors classifier. Moreover, the CNN is computationally very expensive, especially for many patterns. Therefore less combinations of hyperparameters were searched. Besides the accuracy performance metric, the F1 score is calculated, a performance metric often encountered in the context of fault diagnosis.

The performance comparison is initially based on the average scores but the statistical framework described in Section 3.3 is employed to conduct the inference process. The results for each experimental model are presented sequentially. First, the results for the accuracies and F1 scores for the Memorization mode and the corresponding *p*-values of the corrected *t*-test for all combinations of pairs of classifier methods are listed in Table 8 and Table 9 respectively. In sequence, the results for the Generalization by Load mode are listed in Table 10. Finally, Table 11 and Table 12 exhibit the average scores and the *p*-values for the pairwise comparisons for the Generalization by Load and Fault Severity mode. The results of the tables corresponding to the pairwise comparisons are organized as matrices. The principal diagonal indicates the method. The upper triangle part corresponds to the *p*-values when considering accuracy as performance criteria, whereas the lower triangle part presents

Table 10
Load generalization – accuracy and F1-score.

Round	KNN	SVM	RF	CNN-1-D
		Accuracy		
1	0.9471	0.9525	0.9580	0.9325
		F1 – Score		
1	0.9476	0.9561	0.9594	0.9367

Table 11
Severity generalization – accuracy and F1-Score.

Round	KNN	SVM	RF	CNN-1-D
		Accuracy		
1	0.5162	0.5294	0.5277	0.5427
2	0.4836	0.4618	0.5082	0.4939
3	0.5115	0.4923	0.5353	0.5431
4	0.5076	0.4918	0.5178	0.5383
5	0.5079	0.4928	0.5373	0.5216
6	0.4836	0.4618	0.5131	0.5164
7	0.5107	0.4913	0.5411	0.5510
8	0.5168	0.5294	0.5245	0.5536
Mean	0.5047	0.4938	0.5256	0.5326
		F1 - Score		
1	0.4636	0.4342	0.4571	0.4926
2	0.4635	0.4379	0.4617	0.4791
3	0.4636	0.4406	0.5431	0.5019
4	0.4617	0.4402	0.4518	0.5051
5	0.4620	0.4379	0.4644	0.4814
6	0.4635	0.4394	0.4634	0.5004
7	0.4620	0.4394	0.4638	0.5081
8	0.4649	0.4352	0.4565	0.4898
Mean	0.4631	0.4381	0.4702	0.4948

Table 12
Matrix of results of the pairwise comparison between classifier models for the Generalization by Load and Fault Severity mode.

		p-value for accuracy		
p-value for F1-Score	KNN	0.7169	0.2849	0.5420
	0.3166	SVM	0.4259	0.1739
	0.9244	0.5709	RF	0.8893
	0.4666	0.0170	0.5093	CNN

the *p*-values for the F1 scores. The Benjamini-Hochberg procedure was used to highlight *p*-values corresponding to hypotheses that can be rejected at a significance level of 5% ($\alpha = 5\%$).

High performance scores can be observed in the Memorization mode (Table 8), due to the simultaneous occurrence of very similar signal patterns in the training and test sets which constitutes an intrinsic bias of this mode of evaluation. The fundamental problem for the practical applicability of this learning procedure is the absence of training patterns that cover all possible fault conditions. All four classifier architectures exhibit relatively high performance scores, with lowest scores on average for the KNN. The statistical procedure confirms the inferior performance of the KNN and indicates that the Random forest and the SVM outperform the KNN architecture. In order to clarify the application of the Benjamini-Hochberg approach to control the family wise error, the analysis of the upper triangle part of Table 9, which corresponds to the *p*-values when considering accuracy as performance criteria, is detailed next. The highest *p*-value in the ordered sequence that satisfies the constraint of Eq. (2), $p_{(i)} \leqslant \frac{i}{m}\alpha$ is $p_{(k=3)} = 0.0001 < \left(\frac{k}{m}\right)\alpha = \left(\frac{3}{6}\right)0.05 = 0.025$ for the pairwise comparison of SVM against Random Forest. Consequently, the hypothesis corresponding to the pairwise comparison of KNN, SVM and RF against to CNN are not rejected. We conjecture that those comparisons are not statistically significant due to the high variances of the CNN architecture. Indeed, note that the highest and lowest accuracies of Table 8 are originating from the CNN method. Except for the CNN, in which the high variance impacts on the statistical results, we conclude that KNN delivers the worst performance, followed by SVM, and Random Forest delivers the best performance.

The accuracy and F1 score for the Generalization by Load mode of Section 5.2.2 (Table 10) suggests that this mode is more difficult than Memorization mode, because in the test stage, the trained fault diagnosis system might encounter a signal, originating from load values not seen before. High performance values were obtained for the SVM and RF classifiers, even for diverse loads. This could be attributed to the presence of very similar fault signatures, differentiated only by the magnitude of the different loads, in both the training and test folds. The KNN and CNN-1-D classifiers performed a little worse, but they still obtained high classification performance. Since the requirement $J = K \times R \geqslant 30$ was not satisfied, no statistical tests were performed. Therefore, a more detailed investigation must be employed to detect significant performance differences.

Finally, for the most difficult fault diagnosis task of Section 5.2.3, which accuracies and F1 scores are listed in Table 11, it can be observed that the performance scores drop dramatically. Now, a real world situation is simulated with higher fidelity. A fault diagnosis system has to be trained with the available data. During the operational stage of the system, under a different set point of the load, and with a higher or lower degree of the defect severity of the bearings, a signal must be diagnosed by the trained system. A noteworthy observation is the superior performance of the convolutional neural network. At the current stage of our knowledge, we can only speculate that the generalization capability of this kind of learning architecture for more difficult classification problems is better than alternative methods. The considerable amount of successful research related to this deep learning approach might be a hint that also for machine learning based fault diagnosis, CNN is a good technique. However, the difference in accuracies were not sufficient to reject the hypothesis of similar performances among classifier methods.

7. Conclusion

This work presents a performance evaluation framework for machine learning approaches for fault diagnosis from vibration signals. One experimental study is performed showing that nested cross validation is more reliable than conventional cross validation techniques. The work also identifies common methodological evaluation drawbacks on machine learning approaches for fault diagnosis. Special attention is given to the identification of the similarity bias problem and how it impacts the folds division used in cross validation. Another important aspect when comparing fault diagnosis approaches is to account for the experimental variability to verify the statistical significance of the results. Even though the proposal of using repeated nested cross validation, statistical tests for comparing results and making the research work reproducible is not new, integrating these recommendations on the proposed methodology is important for the vibration based machine learning fault diagnosis research community once it clearly has not been integrally following these recommendations on the research works. This paper restricts its claims to vibration analysis based research, but we conjecture that the paper claims are even more generally applicable to signal based machine learning approaches for fault diagnosis.

The work applied the framework to the CWRU data set, showing different scenarios by defining three operational modes. The first mode does not avoid the bias similarity effect and resembles the vast majority of research works that use this data set. Accuracy and F1-score results are extremely high and overoptimistic. Even though the second mode does not share patterns from the same acquisition file in the training and testing data set, it is still not a very difficult diagnosis task because the magnitude of the signals is the only difference between patterns in the training and test data sets. Results are high and optimistic yet. The third

mode avoids similarity bias and defines a more challenging diagnostic task. Accuracy and F1-score results drop significantly. The statistical results of the pairwise comparisons indicate the decrease in the performance is not the unique consequence of avoiding similarity bias. It also may increase the variability of the scores, decreasing the power to detect statistically significant differences among classifiers methods. These results show that avoiding similarity bias is important and makes the diagnostic task closer to reality. However, there are different ways to distribute the patterns in the folds, with different degrees of difficulty. Therefore, a meticulous choice is required when designing the experiments. Moreover, some vibration data sets may not allow avoiding the similarity bias. If this is the case, authors should acknowledge that their results may be overoptimistic due to the similarity bias problem.

One should be aware that the proposed framework is computationally demanding. Applying the framework together with highly computational demanding technologies, such as deep learning, will probably require high performance parallel computing resources like GPUs, supercomputers or large computer clusters. In addition to showing that research results in machine learning fault diagnosis based on vibration signals are overoptimistic, the proposed framework serves as a fairer benchmark for the comparison of different fault diagnosis approaches applicable to the CWRU data set. The results transmit an idea of the difficulties related to the machine learning based approach to fault diagnosis. Bear in mind, that even with the most difficult generalization mode experimented in this work, real life fault diagnosis scenarios are more difficult to build. The whole machinery can suffer degradation which causes alternating patterns, sensors do not always deliver information and can also degrade. New set points are introduced, or formerly unknown faults appear suddenly during the production phase of the process. In a supervised learning context, samples cannot always reliably be labeled by the human specialist, which introduces an a priori source of degrading performance.

8. Computational framework

In order to facilitate experimental comparisons of methods involving the CWRU data, Python source code and the full experimental results of this work are provided at `http://bit.ly/2S0Dnhj`. The programming framework allows to evaluate classification, feature extraction and feature selection methods. It is implemented using Numpy, Scikit-learn and Keras, following the design pattern of these libraries. The python source code of the experiments with the synthetic Fukunaga data can be found at `https://nuvem.ufes.br/index.php/s/gsJnYLgcA6wKgn9`.

CRediT authorship contribution statement

Thomas Walter Rauber: Writing - original draft, Writing - review & editing. **Antonio Luiz da Silva Loca:** Software, Resources, Investigation. **de Francisco Assis Boldt:** Software, Data curation, Resources. **Alexandre Loureiros Rodrigues:** Writing - review & editing, Software. **Flávio Miguel Varejão:** Conceptualization, Methodology, Project administration, Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ali, J. B., Fnaiech, N., Saidi, L., Chebel-Morello, B., & Fnaiech, F. (2015). Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Applied Acoustics, 89*, 16–27.

Bechhoefer, E. (2016). A quick introduction to bearing envelope analysis. MFPT Data, http://www. mfpt. org/FaultData/Fault-Data. htm.

Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1,, Citeseer.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological), 57*(1), 289–300.

Berredjem, T., & Benidir, M. (2018). Bearing faults diagnosis using fuzzy expert system relying on an improved range overlaps and similarity method. *Expert Systems with Applications, 108*, 134–142.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Casella, G., & Berger, R. L. (2002). *Statistical inference* (Vol. 2). CA: Duxbury Pacific Grove.

Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research, 11* (Jul), 2079–2107.

Chiang, L., Braatz, R., & Russell, E. (2001). *Fault detection and diagnosis in industrial systems. Advanced textbooks in control and signal processing*. London: Springer. URL: http://web.mit.edu/braatzgroup/links.html.

Cover, T., & Hart, P. (Jan. 1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on, 13*(1), 21–27.

CWRU (2014). Case Western Reserve University, Bearing Data Center. http://csegroups.case.edu/bearingdatacenter, accessed: 2020-06-10.

de Assis Boldt, F., Rauber, T. W., & Varejão, F. M. (2014). A fast feature selection algorithm applied to automatic faults diagnosis of rotating machinery. *Journal of Applied Computing Research, 3*(2), 78–86.

de Assis Boldt, F., Rauber, T. W., Varejão, F. M. & Ribeiro, M. P. (2015). Fast feature selection using hybrid ranking and wrapper approach for automatic fault diagnosis of motorpumps based on vibration signals. In: 2015 IEEE 13th international conference on industrial informatics (INDIN) (pp. 127–132). IEEE.

Diaz, M., Henríquez, P., Ferrer, M. A., Alonso, J. B., Pirlo, G. & Impedovo, D. (2015). Novel method for early bearing fault detection based on dynamic stability measure. In 2015 International Carnahan conference on security technology (ICCST) (pp. 43–47). IEEE.

Diaz, M., Henriquez, P., Ferrer, M. A., Pirlo, G., Alonso, J. B., Carmona-Duarte, C., & Impedovo, D. (2017). Stability-based system for bearing fault early detection. *Expert Systems with Applications, 79*, 65–75.

Ding, S. (2012). *Model-based fault diagnosis techniques: Design schemes, algorithms and tools. Advances in industrial control*. London: Springer.

Ding, S. X. (2016). *Data-driven design of fault diagnosis and fault-tolerant control systems*. Springer.

Ding, X., & He, Q. (2017). Energy-fluctuated multiscale feature learning with deep convnet for intelligent spindle bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement, 66*(8), 1926–1935.

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Eren, L., Ince, T., & Kiranyaz, S. (2019). A generic intelligent bearing fault diagnosis system using compact adaptive 1d cnn classifier. *Journal of Signal Processing Systems, 91*(2), 179–189.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). San Diego: Academic Press.

Gan, M., Wang, C., et al. (2016). Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mechanical Systems and Signal Processing, 72*, 92–104.

Gao, Z., Cecati, C., & Ding, S. X. (2015). A survey of fault diagnosis and fault-tolerant techniques (part i): Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics, 62*(6), 3757–3767.

Gertler, J. (2017). *Fault detection and diagnosis in engineering systems*. Routledge.

Guo, X., Chen, L., & Shen, C. (2016). Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. *Measurement, 93*, 490–502.

Han, T. & Jiang, D. (2016). Rolling bearing fault diagnostic method based on vmd-ar model and random forest classifier. *Shock and Vibration* 2016.

Hoang, D.-T., & Kang, H.-J. (2019). Rolling element bearing fault diagnosis using convolutional neural network and vibration image. *Cognitive Systems Research, 53*, 42–50.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 65–70.

Kavathekar, S., Upadhyay, N., & Kankar, P. (2016). Fault classification of ball bearing by rotation forest technique. *Procedia Technology, 23*, 187–192.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks, 3361*(10), 1995.

Lee, J., Qiu, H., Yu, G., & Lin, J. (2007). *Bearing data set, nasa ames prognostics data repository*.

Lee, J., Qiu, H., Yu, G., Lin, J. & et al. (2007). Bearing data set. IMS, University of Cincinnati, NASA Ames Prognostics Data Repository, Rexnord Technical Services.

Lei, Y., Jia, F., Lin, J., Xing, S., & Ding, S. X. (2016). An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data. *IEEE Transactions on Industrial Electronics, 63*(5), 3137–3147.

Li, X., Jia, X.-D., Zhang, W., Ma, H., Luo, Z., & Li, X. (2020). Intelligent cross-machine fault diagnosis approach with deep auto-encoder and domain adaptation. *Neurocomputing, 383*, 235–247.

Li, X., Zhang, W., & Ding, Q. (2019). Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism. *Signal Processing, 161*, 136–154.

Li, Y., Wang, X., Si, S., & Huang, S. (2019). Entropy based fault classification using the case western reserve university data: A benchmark study. *IEEE Transactions on Reliability*.

Li, Z., Fang, H., & Huang, M. (2015). Diversified learning for continuous hidden markov models with application to fault diagnosis. *Expert Systems with Applications, 42*(23), 9165–9173.

Liao, G.-P., Gao, W., Yang, G.-J., & Guo, M.-F. (2019). Hydroelectric generating unit fault diagnosis using 1-d convolutional neural network and gated recurrent unit in small hydro. *IEEE Sensors Journal, 19*(20), 9352–9363.

Lu, C., Wang, Y., Ragulskis, M., & Cheng, Y. (2016). Fault diagnosis for rotating machinery: A method based on image processing. *PloS One, 11*(10), Article e0164111.

MaFaulDa (2016). MaFaulDa – Machinery Fault Database [online]. http://www02.smt.ufrj.br/ offshore/mfs/page_01.html, accessed: 2020-01-20.

Marins, M. A., Ribeiro, F. M., Netto, S. L., & da Silva, E. A. (2018). Improved similarity-based modeling for the classification of rotating-machine failures. *Journal of the Franklin Institute, 355*(4), 1913–1930.

McMillan, G., & Vegas, P. (2019). *Process/industrial instruments and controls handbook*. McGraw-Hill Education. Sixth Edition.

Nadeau, C. & Bengio, Y. (2000). Inference for the generalization error. In Advances in neural information processing systems (pp. 307–313).

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N. & Varnier, C. (2012). Pronostia: An experimental platform for bearings accelerated degradation tests. In IEEE international conference on prognostics and health management, PHM'12, Denver, Colorado, United States (pp. 1–9).

Oliveira-Santos, T., Rodrigues, A., Rocha, V. F., Rauber, T. W., Varejão, F. M., & Ribeiro, M. P. (2018). Combining classifiers with decision templates for automatic fault diagnosis of electrical submersible pumps. *Integrated Computer-Aided Engineering, 25*(4), 381–396.

Paderborn (2020). Bearing DataCenter, Paderborn University. https://mb.uni-paderborn.de/kat/forschung/datacenter/bearing-datacenter, accessed: 2020-06-10.

Pan, J., Zi, Y., Chen, J., Zhou, Z., & Wang, B. (2017). Liftingnet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification. *IEEE Transactions on Industrial Electronics, 65*(6), 4973–4982.

Piltan, F., Prosvirin, A. E., Jeong, I., Im, K., & Kim, J.-M. (2019). Rolling-element bearing fault diagnosis using advanced machine learning-based observer. *Applied Sciences, 9*(24), 5404.

Rauber, T. W., de Assis Boldt, F., & Varejão, F. M. (2014). Heterogeneous feature models and feature selection applied to bearing fault diagnosis. *IEEE Transactions on Industrial Electronics, 62*(1), 637–646.

Razavi-Far, R., Farajzadeh-Zanjani, M., & Saif, M. (2017). An integrated class-imbalanced learning scheme for diagnosing bearing defects in induction motors. *IEEE Transactions on Industrial Informatics, 13*(6), 2758–2769.

Rodriguez, P. H., Alonso, J. B., Ferrer, M. A., & Travieso, C. M. (2013). Application of the teager–kaiser energy operator in bearing fault diagnosis. *ISA Transactions, 52*(2), 278–284.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1985). Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science.

Shao, H., Jiang, H., Wang, F., & Wang, Y. (2017). Rolling bearing fault diagnosis using adaptive deep belief network with dual-tree complex wavelet packet. *ISA Transactions, 69*, 187–201.

Shen, C., Wang, D., Kong, F., & Peter, W. T. (2013). Fault diagnosis of rotating machinery based on the statistical parameters of wavelet packet paving and a generic support vector regressive classifier. *Measurement, 46*(4), 1551–1564.

Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical Systems and Signal Processing, 64*, 100–131.

Sreejith, B., Verma, A. & Srividya, A. (2008). Fault diagnosis of rolling element bearing using time-domain features and neural networks. In 2008 IEEE region 10 and the third international conference on industrial and information systems (pp. 1–6). IEEE.

Sun, G., Gao, Y., Lin, K. & Hu, Y. (2019). Fine-grained fault diagnosis method of rolling bearing combining multisynchrosqueezing transform and sparse feature coding based on dictionary learning. *Shock and Vibration* 2019.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Varga, A. (2017). *Solving fault diagnosis problems: Linear synthesis techniques. Studies in systems, decision and control*. Springer International Publishing.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics, 7*(1), 91.

Verstraete, D., Ferrada, A., Droguett, E.L., Meruane, V. & Modarres, M. (2017). Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings. *Shock and Vibration* 2017.

Wen, L., Li, X., Gao, L., & Zhang, Y. (2017). A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Transactions on Industrial Electronics, 65*(7), 5990–5998.

Wu, S.-D., Wu, C.-W., Wu, T.-Y., & Wang, C.-C. (2013). Multi-scale analysis based ball bearing defect diagnostics using mahalanobis distance and support vector machine. *Entropy, 15*(2), 416–433.

Wu, S.-D., Wu, P.-H., Wu, C.-W., Ding, J.-J., & Wang, C.-C. (2012). Bearing fault diagnosis based on multiscale permutation entropy and support vector machine. *Entropy, 14*(8), 1343–1356.

Xu, H., & Chen, G. (2013). An intelligent fault identification method of rolling bearings based on lssvm optimized by improved pso. *Mechanical Systems and Signal Processing, 35*(1–2), 167–175.

Yang, Y., Fu, P., & He, Y. (2018). Bearing fault automatic classification based on deep learning. *IEEE Access, 6*, 71540–71554.

Yiakopoulos, C., Gryllias, K. C., & Antoniadis, I. A. (2011). Rolling element bearing fault detection in industrial environments based on a k-means clustering approach. *Expert Systems with Applications, 38*(3), 2888–2911.

Yin, H., Li, Z., Zuo, J., Liu, H., Yang, K. & Li, F. (2020). Wasserstein generative adversarial network and convolutional neural network (wg-cnn) for bearing fault diagnosis. *Mathematical Problems in Engineering* 2020.

Yu, J.-B. (2011). Bearing performance degradation assessment using locality preserving projections. *Expert Systems with Applications, 38*(6), 7440–7450.

Yu, X., Ding, E., Chen, C., Liu, X., & Li, L. (2015). A novel characteristic frequency bands extraction method for automatic bearing fault diagnosis based on hilbert huang transform. *Sensors, 15*(11), 27869–27893.

Zhang, R., Peng, Z., Wu, L., Yao, B., & Guan, Y. (2017). Fault diagnosis from raw sensor data using deep neural networks considering temporal coherence. *Sensors, 17*(3), 549.

Zhang, R., Tao, H., Wu, L., & Guan, Y. (2017). Transfer learning with neural networks for bearing fault diagnosis in changing working conditions. *IEEE Access, 5*, 14347–14357.

Zhang, W., Li, C., Peng, G., Chen, Y., & Zhang, Z. (2018). A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mechanical Systems and Signal Processing, 100*, 439–453.

Zhang, W., Peng, G., Li, C., Chen, Y., & Zhang, Z. (2017). A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors, 17*(2), 425.

Zhang, Y., Li, X., Gao, L., & Li, P. (2018). A new subset based deep feature learning method for intelligent fault diagnosis of bearing. *Expert Systems with Applications, 110*, 125–142.

Zhao, M., Jin, X., Zhang, Z., & Li, B. (2014). Fault diagnosis of rolling element bearings via discriminative subspace learning: Visualization and classification. *Expert Systems with Applications, 41*(7), 3391–3401.

Zhao, X., Li, M., Xu, J., & Song, G. (2011). An effective procedure exploiting unlabeled data to build monitoring system. *Expert Systems with Applications, 38*(8), 10199–10204.