A STUDY OF MACHINE LEARNING ALGORITHMS FOR ANOMALY DETECTION IN ELECTRIC MOTORS BY VIBRATION AND AUDIO SIGNALS

Klaus Jürgen Folz[1], Herbert Martins Gomes[2]

Universidade Federal do Rio Grande do Sul, Departament of Mechanical Engineering, Av. Sarmento Leite, 425, sala 202, 2º. Andar, 90050-170, Porto Alegre, RS, Brazil

[1] klausjurgenfolz@gmail.com , [2] herbert@mecanica.ufrgs.br

Abstract. Machine maintenance requires continuous improvement of techniques and equipment for monitoring equipment operating parameters. The financial gains in avoiding catastrophic and cascading problems in industrial plants due to failures far outweigh the expenses with monitoring investments in new technologies. The use of machine learning techniques has become an area of intense research, largely due to the relative success of these methodologies in defect classification, lifetime predictions, and the possibility of an online and real-time monitoring. The objective of this article is to evaluate and compare the performance of two machine learning algorithms, Support Vector Machine (SVM) and Random Forests (RF), when classifying 7 states of operation of an electric motor using the Mel Frequency Cepstral Coefficients (MFCCs) as features, calculated using the motor's vibration and audio signals separately. After the training, the SVM model obtained a mean accuracy of 100 % for the MFCCs obtained from the vibration signals and 69.6% for the audio signal. The RF had a mean accuracy of 99.15% for the MFCCs from the vibration signals and 63.82% for the audio signal.

Keywords: Machine learning, support vector machine, random forests, Mel frequency cepstral coefficients, predictive maintenance.

1 INTRODUCTION

Modern electric motors are found in the most diverse activities, in which force and torque are required to move turbines, boilers, conveyors and other components. Problems such as misalignment, imbalance and coupling issues can lead to dangerous conditions or even to the equipment's premature failure. The precise identification of malfunctions makes a premature correction possible and, therefore, an extension of the machine's lifetime, or even avoid more catastrophic events or frequent stops for repair.

The efficiency and reliability of electric motors are of paramount importance in many industrial applications. The failure of such equipment can be a serious problem in terms of safety or costs. To achieve adequate results, predictive maintenance methods can be used, which, through constant condition monitoring, enable the prediction and detection of failures. Predictive maintenance practices, when compared to traditional maintenance methods, increase the average time between each revision, reduce the stock of spare parts and minimize downtime for corrective maintenance [3]. Condition monitoring techniques generate large amounts of data, requiring the automation of diagnostic methods. If properly trained, machine learning algorithms provide an alternative for autonomously interpreting data.

The objective of this article consists of comparing the performance of the Support Vector Machine (SVM) and Random Forest (RF) algorithms, when classifying 7 states of operation of a ¼ hp electric motor (1 normal and 6 anomalous). For that reason, the Mel Frequency Cepstral Coefficients (MFCCs) will be used as features. The coefficients are calculated from the vibration signals of two 3-channel accelerometers installed in a shaft bearing and using the operating motor's audio signal. The states are vertical misalignment, horizontal misalignment, cage fault, ball fault, outer race, imbalance, and the normal state of operation.

Furthermore, the results obtained for both algorithms with the coefficients calculated from the audio and vibration signals separately will be compared, to determine if it is possible to use only the audio signal, since it's a less costly and complex way to acquire the data. The performance evaluation will be based on the Receiver Operating Characteristic Curve (ROC Curve), Mean Accuracy, F1 Score and Confusion Matrices. The data was made available by the Laboratory of Signals, Multimedia and Telecommunications (SMT) of the Federal University of Rio de Janeiro (UFRJ) in the Machinery Fault Database [9].

Machine Learning (ML) is the generic term commonly used to designate a series of techniques and algorithms that can classify and acquire information from data. [20] evaluated the performance of the Support Vector Machines and Fast Clustering Algorithm, combined with the variational mode decomposition method and principal component analysis, in predictive maintenance of imbalanced rotating machines. The authors concluded that the model would be able to effectively diagnose failures caused by imbalance in rotating machines.

A prediction model for proactively preventing failures in a gravitational accelerator was proposed by [7], using a 4-channel accelerometer coupled to the bearing housing. The algorithm was trained with a Deep Learning model, converting the vibration signal into a short-term Fourier transform (STFT) and into a spectrogram of Mel Frequency Cepstral Coefficients (MFCCs) converting the result into a two-dimensional image. As a result, it was possible to quantify the severity of the imbalance by observing the defect areas that cannot be seen with one-dimensional images.

[13] investigated different statistical methods of machine learning and control graphics for the automatic detection of anomalies in a rotating bearing from a commercial semiconductor manufacturing machine. The developers showed that both the control charts and the fully supervised classification methods performed very similarly, directly conditioned to the quality of the training data.

An unsupervised machine learning model for condition monitoring of rotating machines using an anomaly detection approach was studied by [1]. Effectiveness was evaluated by comparing the automatically generated results with a manual dataset by training an Isolation Forest model achieving an average F1-score of 99.6%.

2 THEORETICAL BACKGROUND

2.1 Support Vector Machine (SVM)

SVM is a supervised learning classifier, used both in classification and for regression analysis, clustering and other machine learning applications using a geometric concept called hyperplane (decision limit). The purpose of the SVM algorithm is to compute a hyperplane in an N-dimensional space (where N is the number of features) to distinctly classify (separate) the data points. However, for a given dataset, there are many different hyperplanes that can successfully separate the classes. To determine the best possible plane, the algorithm will maximize the separation of the hyperplane and its closest points (features), also called margin. The robustness of the algorithm is directly linked to the need for the separation between the points and the hyperplane being as large as possible for all classes. Therefore, the objective becomes to calculate the plane that has the maximum distance between the data points of the classes.

2.1.1 Algorithm

If a vector $\vec{w}$ is always perpendicular to the hyperplane, it is possible to calculate the distance of a sample from the decision limit (margin) using the projection of the sample vector $\vec{u}$ onto the vector $\vec{w}$. If this projection is greater than a certain value c, the sample vector $\vec{u}$ will be a positive value, as shown in Equation 1, otherwise, it will be a negative value. The projection is proportional to the inner product of the first vector and the second. Subsequently, a constant b is introduced in Equation 1, where b=-c, resulting in Equation 2. The next step is simply to force the decision function to return a value greater than or equal to 1 for positive samples, according to Equation 3 and 4, and -1 for negative samples.

$$\vec{w}\,\vec{u} \geq c \tag{1}$$

$$\vec{w}\,\vec{u} + b \geq 0 \tag{2}$$

$$\vec{w}\,\vec{x} + b \geq 1 \tag{3}$$

$$\vec{w}\,\vec{x} + b \geq -1 \tag{4}$$

The samples located between the decision limit and the support vectors will not be considered, in other words, if the vector $\vec{w}$ is equal to a support vector, the value on the left side of the equation will be 0. To consolidate Equations 3 and 4, a variable y is added, resulting in Equation 5, where y is equal to 1 for positive samples and -1 for negative samples, as in Equation 6.

$$y_i(\vec{w}\,\vec{x_i} + b) - 1 \geq 0 \tag{5}$$

$$y_i = \begin{cases} +1, & samples + \\ -1, & samples - \end{cases} \tag{6}$$

To maximize the margin and determine an equation for its width, a positive support vector $\overrightarrow{x_+}$ is subtracted from a negative $\overrightarrow{x_-}$ and multiplied by the unit vector $\overrightarrow{w_u}$, calculated by dividing the vector $\vec{w}$ by its magnitude (Equation 7).

$$width = \frac{(\overrightarrow{x_+} - \overrightarrow{x_-})\vec{w}}{||\vec{w}||} \tag{7}$$

Introducing Equations 3 and 4 in Equation 7, Equations 8 and 9 are obtained. To select the optimal decision limit, Equation 9 must be maximized. The maximum width is proportional to the inverse of the maximum magnitude of the vector $\vec{w}$, being equivalent to the minimum of the same, as shown by Equation 10. Equation 10 can also be calculated using Equation 11.

$$width = \frac{((1-b) - (-1-b))}{||\vec{w}||} \tag{8}$$

$$width = \frac{2}{||\vec{w}||} \tag{9}$$

$$max\frac{2}{||\vec{w}||} \propto max\frac{1}{||\vec{w}||} \equiv \min||\vec{w}|| \tag{10}$$

$$\min||\vec{w}|| = \min\frac{1}{2}||\vec{w}||^2 \because \frac{d}{dx}\frac{1}{2}x^2 = x \tag{11}$$

The objective function is convex and the points that satisfy the constraints form a convex set, so this problem has a unique global minimum. Therefore, it can be solved with the introduction of a Lagrangian function, Equation 12, which encompasses the constraints of the objective function, associated with parameters called Lagrange multipliers (α_i).

$$L = \frac{1}{2}||\vec{w}||^2 - \sum_{i}^{n} \alpha_i[y_i(\vec{w}\,\vec{x_i} + b) - 1] \tag{12}$$

Differentiating with respect to $\vec{w}$ and equating Equation 12 to zero (Equation 13), we conclude that $\vec{w}$ is a linear combination of the $\vec{x_i}$ vectors. The value of α_i will not be

0 only for the points that lie on the hyperplanes, that is, the samples that lie closer to the hyperplane. Consequently, the other points do not participate in the calculation of $\vec{w}$. Points where $\alpha_i > 0$ are called support vectors and configure the most informative samples of the training set. Introducing Equation 13 to 12, Equation 14 is obtained. Equation 14 shows that the decision rule depends only on the inner product of the sample vectors by the unknown vector. Using the same procedure for the constant b, Equation 15 is derived.

$$\vec{w} = \sum_{i}^{n} \alpha_i y_i \vec{x_i} \tag{13}$$

$$\vec{w} = \sum_{i}^{n} \alpha_i y_i \vec{x_i}\, \vec{u} + b \geq 0 \tag{14}$$

$$\sum_{i}^{n} \alpha_i y_i = 0 \tag{15}$$

Equation 16 is obtained by introducing Equations 13, 14 and 15 in Equation 12. Equation 16 is relevant for the SVM algorithm. Since the optimization depends on the inner product of pairs of samples, the use of the "Kernel Trick" is possible. The "Kernel Trick" allows the original space of the data to be mapped into a high-dimensional scalar product space, called feature space, where the data can then be linearly separable. Table 1 shows SVM training pseudocode.

$$L = \sum_{i}^{n} \alpha_i - \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} \alpha_i \alpha_j y_i y_j \vec{x_i}\, \vec{x_j} \tag{16}$$

Table 1 - Pseudocode of the support vector machine algorithm.

```
Algorithm 1, training SVM

Request: x and y loaded with labeled training data, an $\alpha \Leftarrow 0$ or $\alpha \Leftarrow$ partially trained SVM

1:    C ⇐ some value (10 for example)

2:    Repeat

3:            Loop: (all) $\{x_i, y_i\}$, $\{x_j, y_j\}$ do
```

```
4:                optimize $\alpha_i$ and $\alpha_j$
5:          **End of the loop**
6:    Until no change in either resource restriction or criteria are met
Ensure: Retain only support vectors ($\alpha_i > 0$)
```

Adapted from [11]

2.1.2 Kernel

Some machine learning algorithms assume that the input data is linearly separable, like SVM, however many real problems do not have this linear characteristic. Nevertheless, as Equation 16 depends exclusively on inner products, it is possible to use a non-trivial and arbitrary function to project the input data of dimension N into a space of dimension M, where: M>N. Any linear model can be transformed into a non-linear model by applying the "Kernel Trick" to the model: replacing its attributes (predictors) with a Kernel function (Equation 17).

$$K(x,x') = \langle \varphi(x), \varphi(x') \rangle, \tag{17}$$

Where x and x' are the input data and 〈.〉 is the inner product. Among the various existing functions, the kernel linear function (Equation 18) is the most used. The polynomial function is shown in Equation 19. The degree of the polynomial is d and r is a constant, both must be determined during the construction of the model. The radial basis Kernel function (RBF) is shown in Equation 20.

$$K(x,x') = \langle x, x' \rangle \tag{18}$$

$$K(x,x') = (\gamma \langle x, x' \rangle + r)^d \tag{19}$$

$$K(x,x') = exp(\gamma \|x - x'\|^2) \tag{20}$$

2.1.3 Regularization parameter (C)

The regularization parameter (c) represents the flexibility in relation to outliers. It allows control over the trade-off between classification accuracy and the generalization model. This parameter configures the SVM optimization regarding the error tolerance when building the machine learning model. For high values of C, a lower margin hyperplane is used, obtaining more accurate results with respect to the training points. In the same way, for low values of C, the optimizer calculates a hyperplane of separation of larger margin, with bigger errors, however with considerable gains in performance.

2.2 Random Forests

[5] devised and developed the first Random Forests (RF) algorithm using a random subspace method. Later the model was improved by Leo Breiman and Adele Cutler [4]. The random forest algorithm is defined by [4] as a combination of decision tree predictors, where each tree depends solely on the values of an independent input dataset with the same distribution for all trees in the set (forest). In this way, a random subset of a certain size is produced from the space of possible attributes (input data) of division. Therefore, the best split is the deterministically selected feature of this subset.

A pseudocode for the random forest algorithm is shown in Table 2. For the classification task, the algorithm classifies the instance by combining all the results from each of the trees in the forest. In other words, the method used to combine the results can be as simple as predicting the class obtained from the largest number of trees, as in a majority vote. [21] highlights the ease of parameterization of the model, so that different values for the parameters have little influence on performance and accuracy. However, different values for the total number of trees used in the forest and the maximum depth of each tree are commonly tested for in-depth assessments of the model's accuracy.

Table 2 – Random forests' pseudocode.

```
Algorithm 2, Random forests' pseudocode

To create classifiers c:
1:  Loop: from i = 1 to c do
2:          Random sample of training data D, replace to generate $D_i$
```

```
3:          Create a root node, $N_i$ containing $D_i$
4:          Call Tree ($N_i$)
5:    End of the loop
6:    Tree (N):
7:    If N contains instances of only one class, then
8:          Return
9:    Else
10:         Randomly select x% of possible N-splitting features
11:         Select feature F with the highest information gain to split
12:         Create f nodes for $N, N_1, N_2, \dots, N_f$, where F has f possible values ($F_1, \dots, F_f$)
13:   Loop: from i = 1 to f do
14:         Define the content of $N_i$ for $D_i$, where $D_i$are all the instances of N respective to $F_i$
15:         Call Tree ($N_i$)
16:   End of the loop
```

Adapted from [17].

2.3 Mel Frequency Cepstral Coefficients (MFCCs)

Mel Frequency Cepstral Coefficients (MFCCs) are attributes extracted from a wave signal, widely applied in speech recognition algorithms. [8] interprets the coefficients as a compact description of the shape of a spectral envelope of a signal. The MFCC feature extraction technique, according to [6] consists of splitting the signal into windows, calculating the Discrete Fourier Transform (DFT), obtaining the logarithm of magnitude and the distortion of frequencies in a Mel scale, followed by the Inverse Discrete Cosine Transform (DCT). The detailed description of the various steps involved in extracting the MFCC is explained below.

1. **Splitting the signal into windows:** The analysis should always be performed in short segments in which the signal is approximately stationary.
2. **Frequency Spectrum (DFT):** Each segment is converted into a magnitude spectrum by applying DFT.
3. **Mel's Spectrum:** The Spectrum is calculated by passing the Fourier transformed signal through a set of bandpass filters known as the Mel filter bank. Mel's approximation to the physical frequency is expressed according to Equation 21, where f denotes the frequency in Hz and f_{mel} the frequency on the Mel scale.

$$f_{mel} = 2595 \log_{10}(1 + \frac{f}{700}) \tag{21}$$

4. **Filter Bank**: The filter center frequencies are normally evenly spaced on the frequency axis. The Mel spectrum obtained from the magnitude X(k) spectrum in Equation 22 is calculated by multiplying the magnitude spectrum by each of Mel's weighted triangular filters according to Equation 23, where M is the total number of weighted triangular filters of Mel and $H_m(k)$ is the weight given to the k-th energy spectrum that contributes to the m-th output band, expressed in Equation 24.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi nk}{N}} \tag{22}$$

$$s(m) = \sum_{k=0}^{N-1} [|X(k)^2| H_m(k)] \tag{23}$$

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \frac{2\big(k - f(m-1)\big)}{f(m++1) - f(m)}, & f(m-1) \le k \le f(m) \\ \frac{2\big(k - f(m-1)\big)}{f(m++1) - f(m)}, & f(m) < k \le f(m+1) \\ 0, & k > f(m+1) \end{cases} \tag{24}$$

5. **Discrete cosine transform:** The DCT is applied to the transformed Mel frequency coefficients and produces a set of cepstral coefficients, the MFCCs, calculated according to Equation 25, where c (n) are the cepstral coefficients and C is the number of MFCCs. Traditional MFCC systems only use 8 to 13 cepstral coefficients. The zero

coefficient is often excluded as it represents the average logarithmic energy of the input signal, which carries only little specific information.

$$c(n) = \sum_{m=0}^{M-1} \log_{10}\big(s(m)\big) \cos(\frac{\pi n(m - 0.5)}{M}) \tag{25}$$

3 METHODOLOGY

3.1 Accuracy

The accuracy of an algorithm can be defined as the ratio between the correct predictions and the total number of predictions made. For a problem with only two classes, Equation 26 can be used to calculate the accuracy.

$$Accuracy = \frac{T_{positives} + T_{negatives}}{(T_{positives} + T_{negatives} + F_{positives} + F_{negativoe})} \tag{26}$$

Where $T_{positives}$ are the positive occurrences (belonging to the class) predicted correctly, $T_{negatives}$ are the correct negative predictions, $F_{negatives}$ are the negative predictions misclassified by the model, likewise $F_{positives}$ represents the incorrect positive predictions. Equation 26 calculate the accuracy for a single class. Since the proposed problem is not binary, it is necessary to generalize Equation 26 to n-classes. To this end, we simply calculate the accuracy for each class separately, followed by calculating the arithmetic mean for all categories, returning average accuracy as a result.

$$Average\ Accuracy = \frac{1}{N}\sum_{i=1}^{N} Accuracy_i \tag{27}$$

3.2 F1 Score, Precision and Recall

Precision is calculated using Equation 28, which represents the ratio of the number of true positive results to the number of all positive results, including false positives. Equation 29 represents the recall, with the number of true positive results divided by the number of true positives plus the number of false negatives. The F1 score in Equation 30 is simply the harmonic mean of precision and recall.

$$Precision = \frac{T_{positives}}{(T_{positives} + F_{positives})} \tag{28}$$

$$Recall = \frac{T_{positives}}{(T_{positives} + F_{negatives})} \tag{29}$$

$$F1\ Score = \frac{2}{(Precision^{-1} + Recall^{-1})} \tag{30}$$

3.3 Receiver Operating Characteristic Curve (ROC)

The Receiver Operating Characteristic Curve (ROC) represents the performance of a classifier model, whose points are defined by Recall (Equation 29) on the y-axis and by the false positive rate (FPR) on the x-axis, defined by the ratio between false positives and the total number of negatives, for various values of the classification threshold. High-performance classifiers have an area under the curve (AUC) and Recall values close to (or equal to) 1.

3.4 Confusion Matrix

A confusion matrix is a table with N rows and N columns, where N is the number of classes assigned to the model, therefore a confusion matrix will always be square. The table displays the number of false positives, false negatives, true positives and true negatives, allowing a detailed analysis of the model behavior for all classes and types of errors. In the matrix, the lines represent the real classes, while the columns represent the classifications performed by the model. Thus, diagonal elements represent all correct classifications, while off-diagonal elements represent errors. This metric can be used to evaluate the performance of binary classifier models, as well as for problems involving several classes. Table 3 shows a confusion matrix model for a binary classification problem, in this example the positive value is assigned to class 2.

Table 3 – Confusion matrix for a binary model.

	Class 1	**Class 2**
Class 1	True negatives	False positives
Class 1	False negatives	True positives

4 MATERIAL AND METHODS

The classification using the accelerometers' signals will be compared with the results obtained using only the audio signal to calculate the MFCCs for both SVM and Random Forests algorithms. For that purpose, the same method, features, and training/test segregation with fixed random state will be used. The procedure lies in treating the data from the vibration and audio signal separately and calculating the MFCCs for a time interval of 5 seconds. Then, divide the data into two categories, training (70% of the data) and test (30% of the data), randomly (with a fixed random state for reproducibility of the results), train each of the algorithms with the training data and evaluate the performance of each algorithm based on the metrics of mean accuracy, ROC curve and confusion matrix.

4.1 Database

The data was made available by the Laboratory of Signals, Multimedia and Telecommunications (SMT) of the Federal University of Rio de Janeiro (UFRJ) in the Machinery Fault Database [9]. This database is composed of 1951 files containing multivariate vibration and audio signals in the time domain, acquired by sensors in a Machinery Failure Simulator (MFS) from SpectraQuest, Alignment-Balance-Vibration (ABVT), shown in Figure 1 - Machinery Failure Simulator (MFS) from SpectraQuest, Alignment-Balance-Vibration ABVT. This equipment allows the simulation of the main types of failures present in rotating equipment. Table 4 shows the main characteristics of the motor used in the simulator, as well as the characteristics of the bearings coupled to its shaft.

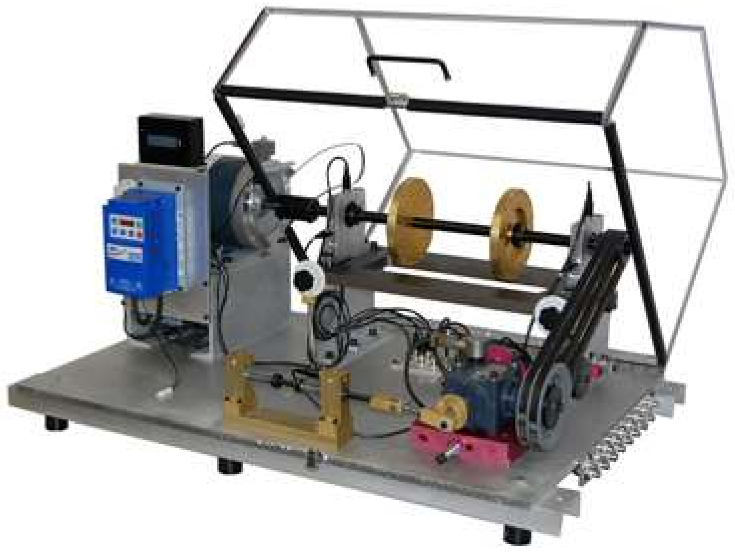


Figure 1 - Machinery Failure Simulator (MFS) from SpectraQuest, Alignment-Balance-Vibration ABVT. Adapted from [18]

4.2 Data processing

A single measurement represents 5 seconds for each class, representing 250,000 points in time. Thus, the 1951 files with their 487.75 million rows and 8 columns were iteratively imported to calculate the MFCCs. To save RAM memory, the vibration signal data in the time domain is then discarded, only the MFCCs are stored in a dictionary type variable. The feature.mfcc function of the Librosa package for Python was used to obtain the MFCCs, with a number of coefficients equal to 40, resolution of the fast Fourier transform equal to 2048, and each data set was segmented into 5 equal parts, resulting in a subset of 1 second per frame superimposed by 512 measurements or 0.01024 seconds.

The resulting set of MFCCs is a two-dimensional 40-row by 98-column array (numpy array in Python) for each 5-second dataset from a single measurement. The application of the SVM and Random Forests algorithms requires a one-dimensional set of features. In order to circumvent this limitation, the resulting matrix of the MFCCs undergoes a juxtaposition of all its lines, transforming the original 40 x 98 matrix into a one-dimensional vector with 3920 elements for each measurement using the numpy.flatten method of the Numpy package for Python. Finally, for a single vibration measurement, the result is a matrix of 1951 rows and 3921 columns, one of which contains the description of the fault or faultless normal state and the other 3920 the respective MFCCs juxtaposed to a single axis (axial, tangential or radial) or for the audio signal.

To use all measured axes and encompass both accelerometers (upper and lower) and not lose any information from the original signals, the procedure of juxtaposing the lines is repeated. However, in this step the juxtaposition is applied on the cepstral coefficients calculated for each axis (axial, tangential and radial) for each accelerometer. The result is a one-dimensional vector with 23,520 features, representing all MFCCs calculated for the signals of all axes. The procedure is not repeated for the audio signal, as this is just one measurement.

Finally, the data are randomly divided into two subsets: test and training. The random state is fixed, allowing reproducibility of tests and validations and comparison of results for different algorithms and different iterations. Table 6 shows the data division, stating that the training data represents 70% of the original set, that is, 1,366 samples containing 32,128,320 coefficients calculated from the vibration signals and 5,354,720 for the audio signal. The test data contains 585 different samples, 30% of the original set, with 13,759,200 MFCCs for vibration signals and 2,293,200 for the sound wave.

Table 6 – Segregation of the original set into training and test subsets for algorithm validation.

Data Acquisition Method	Subclass	Relative quantity	Samples	Number of MFCCs
Acceleration	Train	70%	1.366	32.128.320
Acceleration	Test	30%	585	13.759.200
Audio	Train	70%	1.366	5.354.720
Audio	Test	30%	585	2.293.200

4.3 Model selection and Parameterization

The selection of support vector machines and random forests algorithms was influenced by the effectiveness in applications with datasets with a high number of dimensions, ease of parameterization and computational performance of the models. Random forests are not very sensitive to the parameters used and, in general, it is easy to determine which parameters to use [4]. In addition, [21] highlights the high precision, ease of parameterization and accuracy of the random forest model. The SVM Algorithm combines the training efficiency and simplicity of linear algorithms with the precision of the best non-linear techniques. In many practical applications, the model can tolerate high-dimensional and/or incomplete data.

4.3.1 Grid Search for Parameter Optimization

Testing different combinations of the most influential parameters for each model is known as Grid Search. To obtain the best possible performance, different values for the regularization parameter (C), Kernel, degree of the polynomial Kernel function (d) and parameterization constant (r) were tested. For this procedure, the comparison metric used will be the average accuracy for each iteration. In Table 7 the values of each parameter are listed, all possible combinations were tested, however, distinct variations of degree (d) and constant (r) were applied only to the polynomial kernel, being kept fixed for the other Kernel functions with the default values, respectively 3 and 0.

Table 7 - Values for each parameter tested for the SVM model

Parameter	Tested Values
Regularization parameter (C)	[0.01, 1, 10, 100, 1000]
Kernel	[Polynomial, Linear, RBF]
Degree (d)	[3, 4, 5, 6]
Constant (r)	[2, 3, 6, 12, 18, 24]

Although the Random Forests algorithm is known for its easy parameterization, when compared to SVM models, different combinations of parameters were tested in the grid search process: the number of trees in the forest and maximum depth.

Table 8 - Values for each parameter tested for the Random Forests model

Parameter	Tested Values
Number of trees in the forest	[100, 1000, 10000]
Max Depth	[10, 50, 100, None]

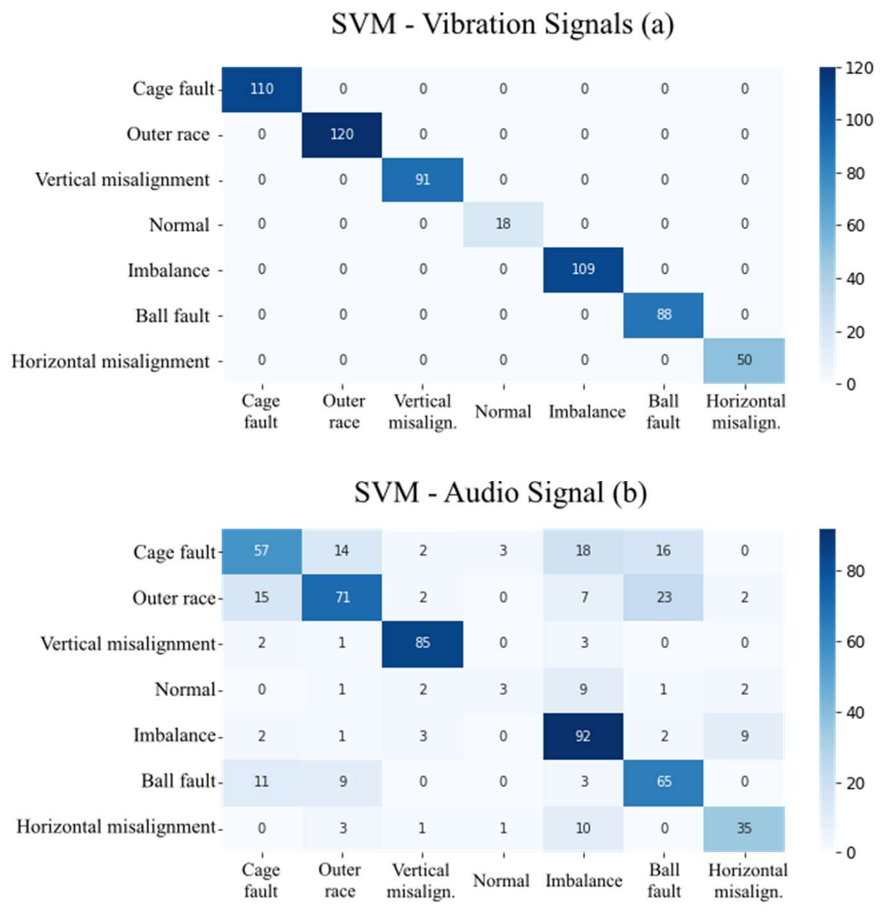
5 RESULTS

5.1 Support Vector Machine

The SVM parameter settings that showed the best performance were the polynomial Kernel function, with degree (d) and constant (r) equal to 3 and regularization parameter (C) equal to 10. For the group of MFCCs obtained from the vibration signals the mean accuracy was 100%. Figure 2 (a) displays the confusion matrix, Figure 3 (a) shows the ROC curve, Table 9 contains the F1 Score for the vibration signals MFCCs. Additionally, the model training lasted approximately 51 seconds. The mean accuracy was 69.6% for the MFCCs calculated using the audio signal. For this set of coefficients, the confusion matrix is shown in Figure 2 (b), the ROC curve in Figure 3 (b) and the F1 scores in Table 9. The model training lasted 33 seconds.

Table 9 – F1 score for the SVM model using MFCCs calculated from the Vibration Signals (Accelerometers) and Audio Signal (Microphone).

Class	**F1 Score (Vibration signal)**	**F1 Score (Audio signal)**
Vertical misalignment	100,0%	91,4%
Normal	100,0%	24,0%
Outer race	100,0%	64,6%
Cage fault	100,0%	57,9%
Imbalance	100,0%	73,3%
Ball fault	100,0%	66,7%
Horizontal misalignment	100,0%	71,4%

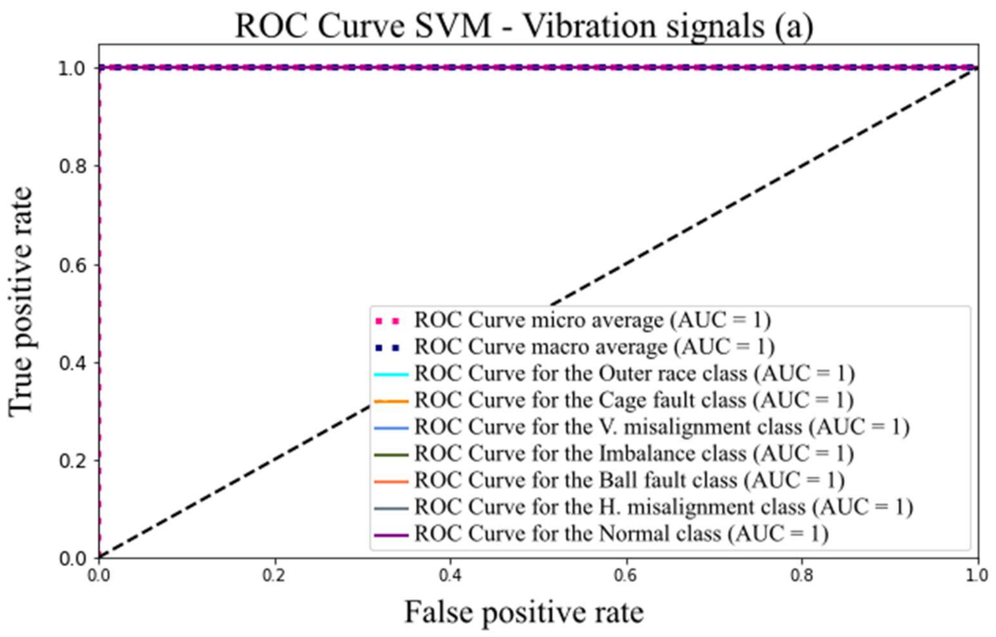

SVM - Vibration Signals (a)

	Cage fault	Outer race	Vertical misalign.	Normal	Imbalance	Ball fault	Horizontal misalign.
Cage fault	110	0	0	0	0	0	0
Outer race	0	120	0	0	0	0	0
Vertical misalignment	0	0	91	0	0	0	0
Normal	0	0	0	18	0	0	0
Imbalance	0	0	0	0	109	0	0
Ball fault	0	0	0	0	0	88	0
Horizontal misalignment	0	0	0	0	0	0	50

SVM - Audio Signal (b)

	Cage fault	Outer race	Vertical misalign.	Normal	Imbalance	Ball fault	Horizontal misalign.
Cage fault	57	14	2	3	18	16	0
Outer race	15	71	2	0	7	23	2
Vertical misalignment	2	1	85	0	3	0	0
Normal	0	1	2	3	9	1	2
Imbalance	2	1	3	0	92	2	9
Ball fault	11	9	0	0	3	65	0
Horizontal misalignment	0	3	1	1	10	0	35

Figure 2 - SVM Confusion matrix with MFCCs calculated using vibration (a) and audio signals (b).



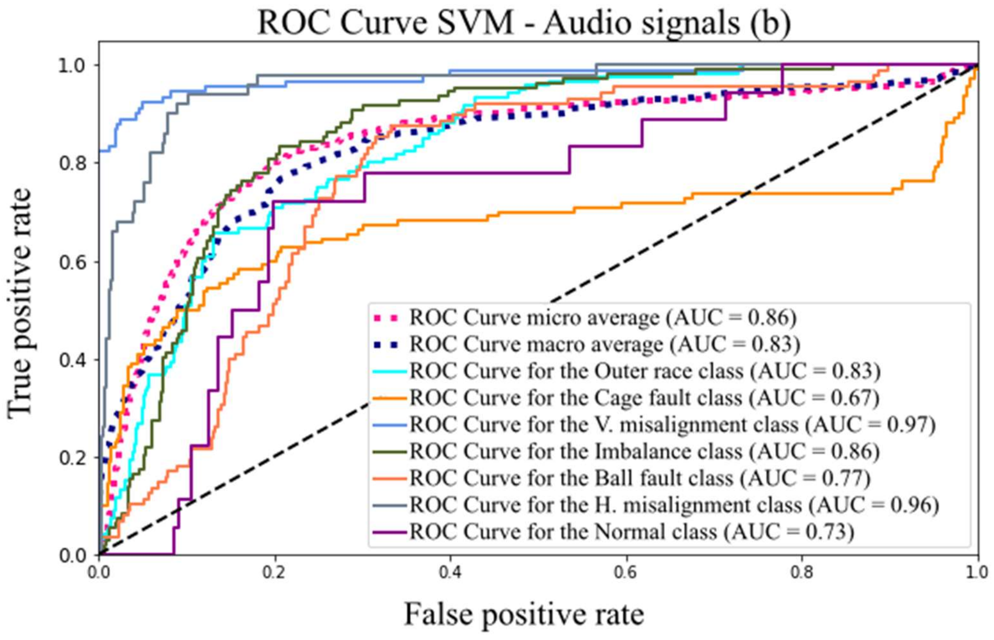


Figure 3 - ROC curve for the SVM model with MFCCs calculated from the accelerometer vibration signals (a) and audio signal (b).

5.2 Random Forests

The best results were obtained when applying the Random Forest algorithm for a number of trees equal to 1000 and maximum depth equal to 10. The total duration of the training stage was 43 seconds for the audio signal and 1 minute and 17 seconds for coefficients calculated using the vibration signals. The mean accuracy for the MFCCs calculated using the vibration signals was 99.15%. The confusion matrix in Figure 4 (a) shows all classifications performed by the model. All errors are represented by off-diagonal elements. The ROC curve for the set of MFFCs calculated using vibration signals is shown in Figure 5 (a). The mean accuracy for the MFCCs calculated from the audio signal was 63.82%. For this group of MFCCs, the confusion matrix is shown in Figure 4 (b), the ROC curve in Figure 5 (b) and the F1 scores are shown in Table 10, for both audio and vibration signals.

Table 10 - F1 score for the Random Forests model using MFCCs calculated using the Vibration Signals (Accelerometers) and Audio Signal (Microphone).

Class	**F1 Score (Vibration signal)**	**F1 Score (Audio signal)**

Vertical misalignment	100,0%	88,3%
Outer race	100,0%	52,4%
Normal	87,5%	0,0%
Cage fault	99,1%	64,6%
Ball fault	100,0%	54,8%
Horizontal misalignment	99,0%	42,1%
Imbalance	98,6%	75,8%

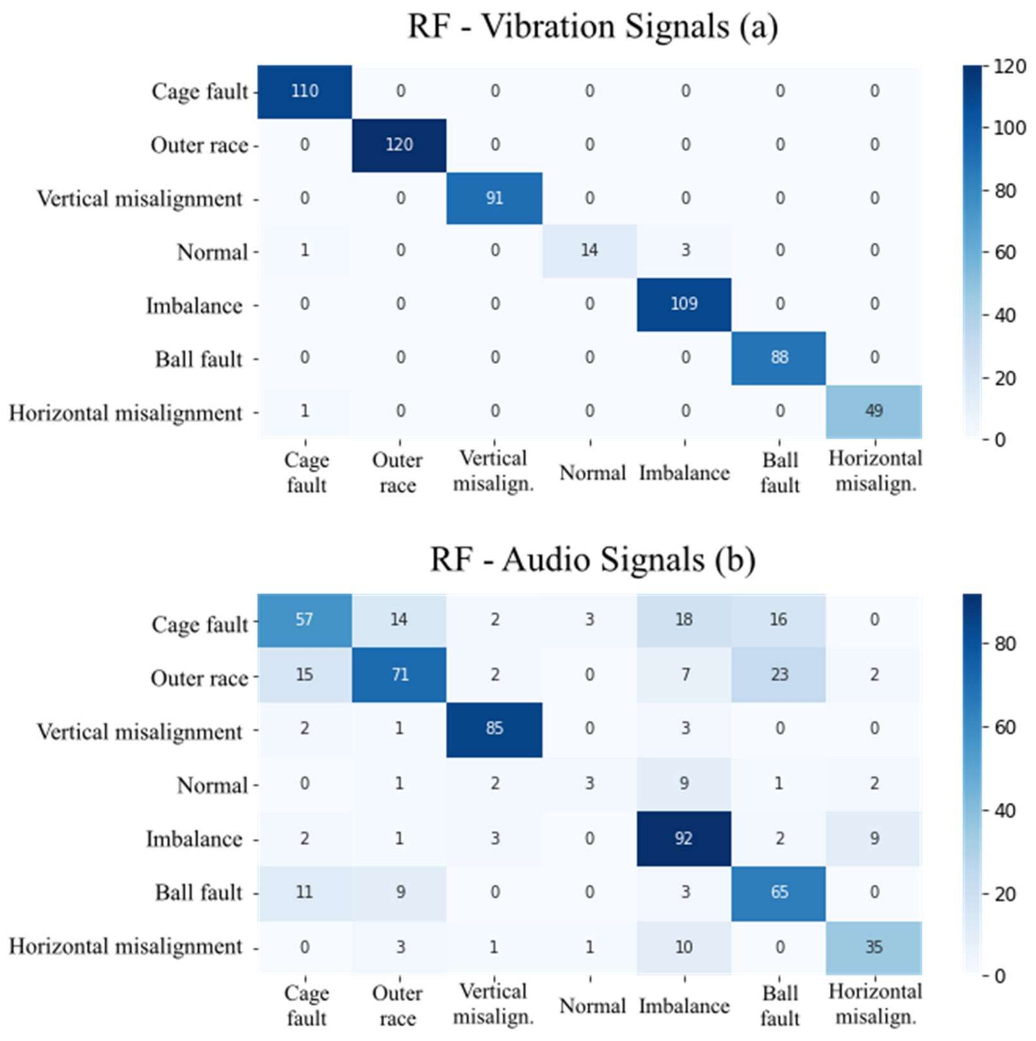


Figure 4 - Confusion matrix of the Random Forests model with MFCCs calculated using the vibration signals of the accelerometers (a) and audio signal (b).

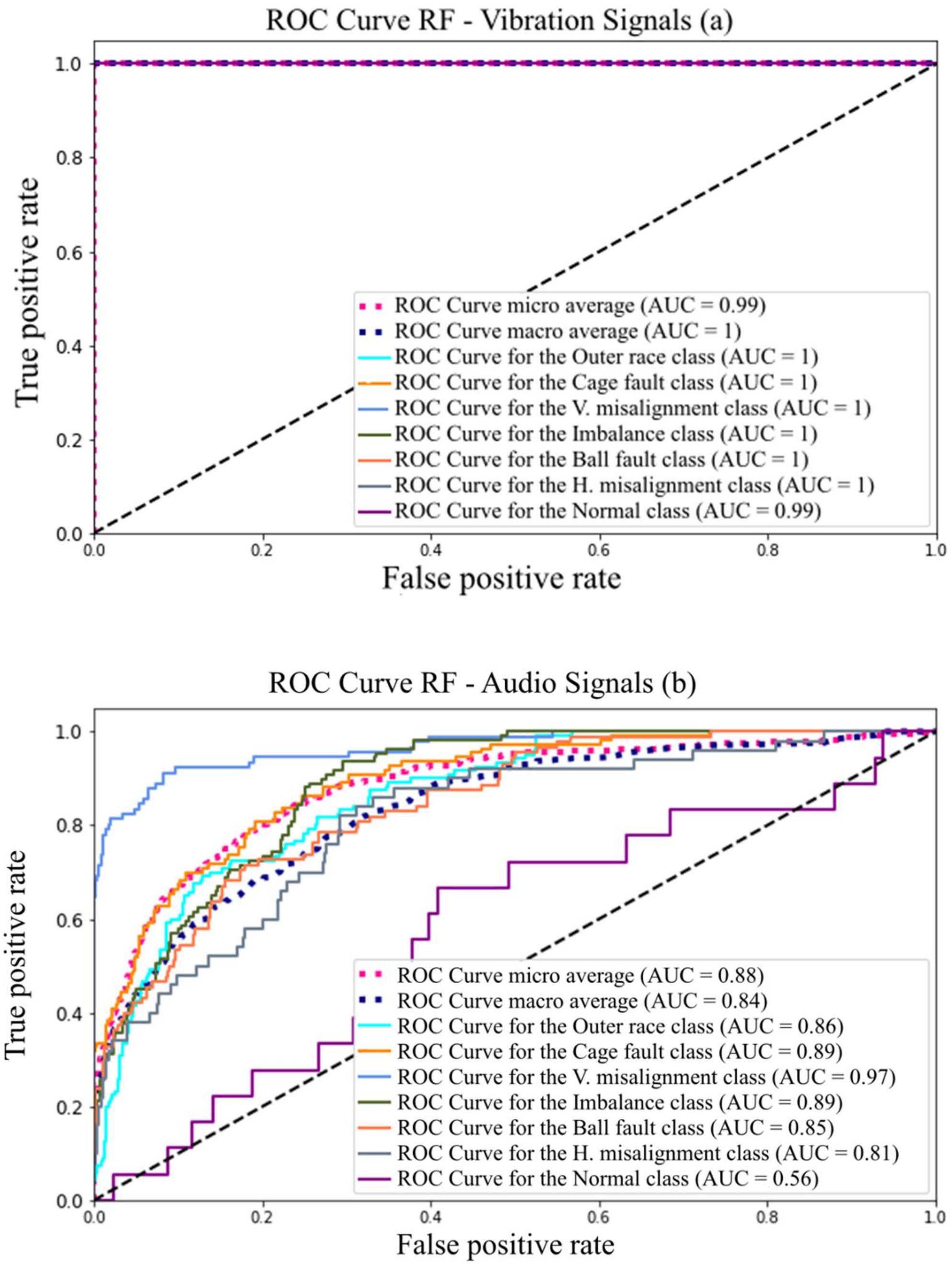


Figure 5 - ROC curve for the Random Forests model with MFCCs calculated from the vibration signals of the accelerometers (a) and audio signal (b).

6 DISCUSSION

The SVM model with the polynomial Kernel function, degree (d) and constant (r) equal to 3 and regularization parameter (C) equal to 10 provided an error-free classification with 100% average accuracy for the MFCCs calculated from the vibration signals. The confusion matrix, shown in Figure 2 - SVM Confusion matrix with MFCCs calculated using vibration (a) and audio signals (b). (a), does not have a single off-diagonal element, which is characteristic of a perfect classifier. Likewise, the ROC curve with parameter AUC equal to 1, shown in Figure 3 (a), is identical to the curve of a perfect classifier.

For the set of MFCCs obtained from the audio signal, the F1 scores for all classes of anomalies were close to a random classifier, except for the vertical misalignment class, with 91.4% accuracy. The lowest F1 score, belonging to the normal class (24%), is due to the low number of samples for this class, with only 49 measurements. The model training lasted 33 seconds, approximately 37% faster when compared to the training performed by the SVM with MFCCs from the vibration signals. The higher speed in the training stage is influenced by the lower number of attributes for the model built with the audio signal, with 3920 coefficients against 23520 coefficients for the model built from the vibration signal.

The RF's 99.15% mean accuracy shows that, even with the best performance of the SVM algorithm, the RF algorithm still is an excellent classifier. Likewise, the model's ROC curve approaches the curve of a perfect classifier, being compromised only by the errors made in the normal class, with AUC of 0.9996.

For both algorithms, the performance for the set of MFCCs generated from the vibration signals and the audio signal differ substantially. While the vibration data and their respective cepstral coefficients provide the creation of machine learning models very close to perfect classifiers, the model trained with the coefficients calculated using the audio signals have a behavior like random classifiers. Furthermore, all metrics presented in this article point to the same result, with the SVM algorithm being the best classifier in terms of classification performance and processing time in the model training stage.

7 CONCLUSION

The Support Vector Machine (SVM) using the polynomial Kernel function, with degree (d) and constant (r) equal to 3 is the best option for classifying problems and anomalies in electric motors using the Mel frequencies cepstral coefficients as features. The Algorithm showed average accuracy and F1 score of 100% for all classes, for MFCCs obtained via vibration signals and average accuracy of 69.6% for coefficients calculated from the sound signal. In addition, the SVM model proved to be faster in the training phase when compared to the Random Forests model, with a duration of 51 seconds against

1 minute and 17 seconds of the concurrent algorithm. This result is due to the high number of attributes, MFCCs, and the correct choice of parameters to obtain the best results.

The MFCCs were excellent attributes for applications of fault detection in electric motors using vibration signals. However, applying the same methods and parameters, with the coefficients calculated from the motor's audio signal (very similar application for which MFCCs were initially developed), both models did not perform well, with a better mean accuracy of 69.6% from the SVM model. Although the results obtained via audio signal were not adequate, the application of models of this type in real situations in the industry would be less complex and of lower cost, when compared to the application using accelerometers, due to the cost of this equipment and the complexity of their configuration in a machine.

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.

Data availability statement

The data, models, and algorithms that support the findings of this study are available from the corresponding author upon request.

CRediT authorship contribution statement

Klaus Jürgen Folz: Writing, Data accumulation, Visualization, Validation, Data curation, Software, Writing – review & editing. **Herbert Martins Gomes**: Conceptualization, Supervision. Investigation, Writing – review & editing.

REFERENCES

[1] AHMADs S.; STYP-REKOWSKI, K.; NEDELKOSKI, S; O. KAO, **Autoencoder-based Condition Monitoring and Anomaly Detection Method for Rotating Machines**. 2020 IEEE International Conference on Big Data (Big Data), 2020, p. 4093-4102. https://doi.org/10.48550/arXiv.2101.11539

[2] ANTONI, L.; ZIEMOWIT, D.; PIOTR, C. **An anomaly detection method for rotating machinery monitoring based on the most representative data**. Journal of

Vibroengineering Vol. 23, Issue 4, 2021, p. 861-876. https://doi.org/10.21595/jve.2021.21622

[3] ARALTO, A. **Manutenção Preditiva: Usando Análise de Vibrações**. São Paulo: Manole, 2004.

[4] BREIMAN, L. **Random Forests.** Machine Learning, v. 45, p. 5–32, 2001. https://doi.org/10.1023/A:1010933404324

[5] HO, TIN KAM (1995). **Random Decision Forests**. Proceedings of the 3rd International Conference on Document Analysis and Recognition. Montreal, QC, 14–16 August 1995. pp. 278–282. https://doi.org/10.1109/ICDAR.1995.598994

[6] K.S. RAO AND MANJUNATH K.E., **Speech Recognition Using Articulatory and Excitation Source Features**. Springer Briefs in Speech Technology, 2017. https://doi.org/10.1007/978-3-319-49220-9

[7] LEE, S.; YU, H.; YANG, H.; SONG, I.; CHOI, J.; YANG, J.; LIM, G.; KIM, K.-S.; CHOI, B. Kwon, J. **A Study on Deep Learning Application of Vibration Data and Visualization of Defects for Predictive Maintenance of Gravity Acceleration Equipment**. Appl. Sci. vol. 11, p. 1564, 2021. https://doi.org/10.3390/app11041564

[8] LERCH, A. **An introduction to audio content analysis: Applications in signal processing and music informatics**. [s.l.] Wiley-IEEE Press, 2012. https://doi.org/10.1002/9781118393550

[9] MAFAULDA. COPPE/Poli/UFRJ. (2014). **Machinery Fault Database – MAFAULDA**. Available: http://www02.smt.ufrj.br/~offshore/mfs/page_01.html. Access in: 21/09/2021.

[10] MCFEE, B. et al. librosa/librosa: 0.8.0. 22 July 2020. https://doi.org/10.25080/Majora-7b98e3ed-003

[11] PEDERSEN, R.; SCHOEBERL, M. (2006**). An Embedded Support Vector Machine. In Proceedings of the Fourth Workshop on Intelligent Solutions in Embedded Systems**. WISES, 2006, pp. 79-89. http://hdl.handle.net/20.500.12708/51591

[12] PEDREGOSA ET AL. **Scikit-learn: Machine Learning in Python**. Journal of Machine Learning Research, v. 12, p. 2825–2830, 2011. http://scikit-learn.sourceforge.net

[13] PITTINO, F., PUGGL M., MOLDASCHL, T., HIRSCHL, C. **Automatic Anomaly Detection on In-Production Manufacturing Machines Using Statistical Learning Methods**. Sensors. v. 20, p. 8: 2344, 2020. https://doi.org/10.3390/s20082344

[14] PRATI, R. C.; BATISTA, G. E. A. P. A; MONARD, M. C., **Evaluating Classifiers Using ROC Curves**. IEEE Latin America Transactions, vol. 6, no. 2, pp. 215-222, June 2008. https://doi.org/10.1109/TLA.2008.4609920

[15] PYTHON. **Support Vector Machine Python Example. Towards data science**. Access in: 15/07/2021. Available: https://towardsdatascience.com/ support-vector-machine-python-example-d67d9b63f1c8

[16] SCIKIT**.** **Support Vector Machines – Scikit-Learn Documentation**. Available: https://scikit-learn.org/stable/modules/svm.html#mathematical-formulation. Access in: 11/10/2021.

[17] SIRIKULVIRIYA, NAPHAPORN AND SUKREE SINTHUPINYO. **Integration of Rules from a Random Forest**. International Conference on Information and Electronics Engineering IPCSIT vol.6, 2011.

[18] SpectraQuest. **SpectraQuest, Inc**. Available: https://spectraquest.com/ Acess in: 21/09/2021.

[19] VAPNIK, V. N. **The Nature of Statistical Learning Theory**. Springer, NY, 1995. https://doi.org/10.1007/978-1-4757-2440-0

[20] ZHANG, X.; JIANG, D.; HAN, T.; WANG, N.; YANG, W.; YANG, Y. **Rotating Machinery Fault Diagnosis for Imbalanced Data Based on Fast Clustering Algorithm and Support Vector Machine**. Journal of Sensors, Beijing, China, v. 2017, p.0-15, 2017. https://doi.org/10.1155/2017/8092691

[21] HORNING, N. **Random Forests: An algorithm for image classification and generation of continuous fields data sets**, 2010.

[22] GAVRISHCHAKA, V.; GANGULI, S. **Support vector machine as an efficient tool for high-dimensional data processing: Application to substorm forecasting**. Journal of Geophysical Research. 106. 29911-29914, 2001. https://doi.org/10.1029/2001JA900118

[23] SCIKIT. **Metrics and scoring: quantifying the quality of predictions**. Available: https://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics. Access in: 07/11/2021.