Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/130519

Three-Phase Induction Motors Faults Classification using Audio Signals and Decision Trees

Beatriz Cristina Flamia de Azevedo [1] , *Glaucia Maria Bressan*[2,*], *Cristiano Marcos Agulhari* [1], *Herman Lucas dos Santos* [1] *and Wagner Endo* [1]

[1] Department of Electrical Engineering, Federal University of Technology of Paraná, Cornélio Procópio - PR, Brazil.
[2] Department of Mathematics, Federal University of Technology of Paraná, Cornélio Procópio - PR, Brazil.

Received: 27 Dec. 2018, Revised: 27 Mar. 2019, Accepted: 3 Apr. 2019
Published online: 1 Sep. 2019

Abstract: The study of intelligent systems, being able to learn and to generalize patterns, has become an area highly explored in diverse fields of science. In the industry area, systems are able to diagnose faults that have been widely studied. The electric motors present fundamental role in industry, since much of the production process depends on its good working. Therefore, avoiding unscheduled stoppage and faults is important in the production process. This paper presents supervised learning approaches to classify three-phase induction motors faults, applying *Decision Trees* and *Random Forest* algorithms. The great advantage of using intelligent systems in the motor faults classification is the fact that data collection can be done without interrupting the production process. The input to the proposed classifiers is the audio generated from motor noises, which are obtained using an experimental setup with defective real motors. From Decision Trees structure, we can generate understandable "IF-THEN" linguistic rules, which facilitate the understanding of the results and allow the evaluation between the developed models.

Keywords: Classification, Motor Faults, Audio Signals, Decision Trees

1 Introduction

The advancement of computing, along with the ease of data acquisition and storage, requires faster and efficient pattern recognition and classification systems. Thus, the study of intelligent systems, which learn and generalize patterns, has become an area highly explored in diverse fields of science. Currently, the data storage has become easier and cheaper, and the accessibility and abundance of this information has made data mining and pattern recognition be a necessity to deal with the exponential growth of information which is not only challenging but computationally demanding [15,17,27].

In the industrial context, pattern recognition methods are widely applied, such as in production planning, quality control, financial analysis and equipments monitoring. Many industrial activities depend on the operation of the so-called *Three-phase Induction Motors* (TIMs), as compressions, power machines, elevators, lathes, and others. For this reason, the development of recognition and classification methods for induction motors failures has become an important area of research and interest in the industrial area, since the occurrence of a fault can lead to damages for productive process, such as: complete stop of production process, costly machinery repair and process downtime extension [1]. Due to the primordial role of the TIMs in industry tasks, increased reliability and machine availability are the main reasons for the high demand for predictive maintenance and early identification of incipient faults, since preventing faults can avoid secondary effects such as overheating, vibration, current and voltage unbalance, torque losses, reduced efficiency and large financial losses [3,4].

In addition, TIMs are broadly used in industry due to its low acquisition and maintenance cost characteristics, constructive simplicity, robust operation, application diversity and reliability [28]. In general, faults affecting TIMs result from progressive degenerations of certain components, rather than being random or unpredictable [3]. Therefore, it is possible to detect faults using information such as as temperature rise and non-standard variations in vibration and equipment noise, for example.

An induction motor can develop either internal fault or external fault, which can be originated either from

* Corresponding author e-mail: glauciabressan@utfpr.edu.br

mechanical or electrical elements [5]. Internal faults could have mechanic or electric origin and they are caused by the manufacturing errors and by the deterioration of materials, while external faults are caused by interactions with the environment, the power supply and the load. It is estimated that faults distribution in electric motors originate 45% from the bearings, 35% in the stator, 10% in the rotor and the remaining 10% in other categories [3,5].

Electric faults are associated with problems in stator, rotor windings, broken rotor bars and rings and their connections. This type of faults usually promotes changes in torque, electric field flow, stator currents and others [2, 3,4,6]. Mechanic faults are related to eccentricity, misalignment, unbalance, asymmetries of the rotor, internal and external rings or in the rotating elements of the bearing. The progression of this kind of faults occur due to vibration, wear, friction for example [3,4]. Finally, the environment faults are related to humidity, temperature and cleanliness [5]. The lack of cleanliness leads to contamination from dust and particles, chemicals products, which can clog filters, ventilation elements and overheat the machine.

In the context of electric motors faults recognition and classification, the state-of-the-art is very broad and several techniques have been applied in order to deal with this kind of problem [1,3,4,5,23]. The great advantage of using intelligent systems in the motor faults classification is the fact that data collection can be done without interrupting the production process. Moreover, intelligent systems present an easy implementation and the results can be obtained without requiring complex mathematical models. Thus, in this paper, we present supervised learning approaches to classify three-phase induction motors faults, applying *Decision Trees* and *Random Forest* algorithms. Decision Trees can be named as "White Box" methods, which consists of a system where the inner components are available for inspection. In other words, White Box algorithms reveal the structure, allowing users to assemble algorithms from algorithm building blocks. Random Forest is an ensemble of Decision Trees and its algorithm integrates the decisions obtained from each tree that composes the forest [7]. In this paper, the input to the proposed classifiers is the audio generated from motor noises, which are obtained using an experimental setup with defective real motors. In addition, from Decision Trees structure, we can generate understandable "IF-THEN" rules. So, users can understand each split, see the impact of that split and even compare it to alternative splits [21].

In supervised learning, there must be a relationship between the input attributes and the output classes, so that the method used can map the behavior of the system and provide a result that describes the reality [15]. For the motor faults evaluation, for example, attributes can be collected by using devices that extract motors characteristics, such as microphones or sensors.

This paper is organized as follows: in Section 2 is presented an overview of the state-of-art involving diagnoses and classification of TIMs using Decision Trees and Random Forest algorithm. Section 3 describes essential properties of a Decision Tree model and the main characteristic of each one of the algorithms used in this paper, as well as the statistic measures used to evaluate the model performance. The database composed by audio signals is presented in Section 4, together with the pre-processing data methods. The results are presented in Section 5 and, finally, the analysis and the conclusion are exposed in Section 6.

2 Related Works

Due to the versatility of the TIMs, systems capable of diagnosing motor faults have been widely studied, since eliminating unscheduled stops is a challenge of great interest to the industry. Different methodologies can be applied in the motor faults diagnostic, since each method has its own characteristics and the performance varies according to the constructive process and to the data set used. Intelligent systems are powerful tools to improve the efficiency of fault diagnosis in electrical machines. The advantage of using intelligent systems, as Decision Trees, is the fact that the data acquisition can be done without damage to the motors and, in many cases, without productive process interruption. Besides, some Decision Trees algorithms provide access to the constructive process and the tree structure can generate linguistic rules of easy interpretation.

Sugumaran et. al (2007) used Decision Tree to identify the best feature from a given set of roller bearing samples. The bearing is an essential component in TIMs and its operation influences the whole machinery. Vibration signals were used in the model and the most representative features of the Decision Tree are used in a Support Vector Machine classifier for fault detection [29].

Based on a boundary analysis for feature extraction and on a Fuzzy Decision Tree for classification, Aydin et. al (2014), propose a system for TIMs faults diagnosis. Authors consider data samples of healthy motors, broken rotor bars and broken connect faults. The fault-related features are used to construct the Decision Tree and extract a set of linguistic rules used in the Fuzzy model. Comparing to the other methodologies, as Gaussian Mixture Model and Artificial Neural Networks, the Decision Tree model presented excellent results [1]. Bazan (2016) presents an approach to diagnose stator short-circuit faults in induction motors driven directly from a supply line. Decision Trees and Artificial Neural Networks methodologies are used to predict classes and to classify patterns. Both methodologies presented satisfactory results, however Artificial Neural Networks showed a slightly superior performance. On the other hand, the complexity elaboration of a Decision Tree

model is much lower than an Artificial Neural Network model [2].

Using Random Forest models, which can be understood as an extension of Decision Trees, Patel and Giri (2016) investigated the multi-class mechanical faults diagnosis in bearing of an induction motor. Bearing vibration records were selected and divided into four classes (normal, inner raceway fault, bearing ball and raceway fault). These information were processed and used to feed the classifier. The results of Random Forest models stand out when compared with Artificial Neural Networks. Authors state that the conventional classifiers, as Neural Networks, are affected when the number of classes is bigger and the answer is slower, while these problems do not occur in Random Forest model [23].

Another interesting approach is presented in Yang et al. (2008), which investigated machine fault diagnosis combining Random Forest and Genetic Algorithm. The Genetic Algorithm is used to strengthen the Random Forest model, evaluating the best parameters. Three-direction vibration signals were collected; a number of feature parameters in time and frequency domains and regression coefficients are calculated to extract helpful information and remove the background noise of the data. The faults classification model is elaborated according to these features, achieving a high accuracy [30]. Panigrahy and Chattopadhyay (2018) also studied and explored machine learning techniques, including Random Forest, in the stator faults problem. Authors concluded that, when compared to the others techniques, the Random Forest shows extraordinary learning ability for very less number of training samples even in the noisy feature space because of its distributive features model [22].

According to the literature and considering the interest in the electric motors faults diagnostic, this paper proposes the design and the application of a three-phase induction motor faults classification system, with the main contribution of predicting faults, aiming to reduce the maintenance cost and damages to the motor and production. The two most recent algorithms of Decision Trees and a Random Forest algorithm are used in this approach, since Decision Trees algorithms are "White Box" and then they provide a graphic structure that allows the access to the construction method and run efficiently using big data, with no overfitting problem. Forests, in turn, consist of an ensemble of Decision Trees [7]; the Random Forest algorithm does not generate rules, but it consists of a collection of trees that contains linguistic rules. Another important contribution of this work is the use of audio signals from motor noises as input attributes to the classifiers, which is not frequent in the literature, since most of the works use current, voltage and vibration signals to classify faults [3,5].

3 Decision Trees

Decision Trees are widely used in pattern recognition and classification, since the classification mechanism needs to be transparent for legal reasons or the results need to be shared in order to facilitate decision making [20].

The mathematical model is constructed in a top-down way, partitioning the data set in groups, starting in a general group and refining the data set in more specific subgroups, according to the characteristic of the attribute. Each group is called *node* and it is connected by branches (arcs). In general, a node is labeled by an attribute name, and an arc by a valid value of the attribute associated with the node from which the arc originates [17]. Each decision outcome at a node is called a *split*, since it corresponds to splitting a subset of the training data [16]. The most top node is called *root node* where is passed through the various decisions in the tree according to the values of its features [20]. Under the root node, we find the internal nodes, which are originated from the data set partition. They constitute the *tree branches*. At the end of each branch are the terminal nodes of the partition process, which are named *leaf nodes* and determine the class. It is important to observe that the attributes in the upper parts of the tree have stronger influence on the value of the target variable than the nodes in the lower parts of the tree, since they have a larger number of data [17].

The greatest difficulty in developing a Decision Tree algorithm is the construction of its topology. There are many algorithms and constructive methods that build distinctive Decision Trees. These algorithms normally use a greedy strategy to build the tree structure by the most informative attribute of each step and do not allow regression [15,17]. The most informative attribute divided the data set in the best way, aiming to find the most homogeneous subsets.

The formulation of a Decision Tree model usually is done in two stages. In the first stage, the data set is partitioning recursively, using a *divide-and-conquer* strategy [15,20]. As a result of the recursive partitioning of the data, the number of examples that end up in each node decreases steadily, so the reliability of the chosen attributes decreases with increasing depths of the tree; thereby high complexity model are generated, which explain the training data but do not generalize well to unseen data [17]. This fact is known as overfitting. In order to eliminate the overfitting, the second stage, named *prune*, is applied.

The pruning process consists of eliminating the branches and nodes near the leaves, replacing some of the internal nodes with a new leaf, thereby removing the subtree that was rooted at this node. The leaf nodes of the new tree are no longer pure nodes, containing only training examples of the same class labeling the leaf, however the leaf will bear the label of the most frequent class at the leaf [17]. Then, the pruning process optimizes

the Decision Tree model, since it reduces the tree complexity, the size and the number of leaves of the tree. This process also eliminates noise and redundant information that decrease the efficiency of the classification.

An important characteristic of Decision Tree model is the possibility of extracting linguistic "IF-THEN" rules, which are closer to the way human can interpret information, making the classification more comprehensible by the user. One rule is composed by each terminal node, plus the internal nodes that belongs to one specific branch and the root node. An "IF-THEN" rule is an expression as follows:

"IF <first condition> AND/OR <second condition> AND/OR ... THEN <conclusion>".

The "IF" part of the rule is known as *antecedent* and it is composed by one or more input attributes of the data set, combined using a logical connectors of the type "AND" or "OR". The "THEN" part is called as *consequent* and it indicates the prescribed class for the rule [18].

The most current Decision Trees (CART and C4.5) and Random Forest algorithms, used in this paper for TIMs faults classification, are described in the next section.

3.1 CART Algorithm

The Classification and Regression Tree (CART) algorithm was developed by Breiman [8]. An important characteristic of CART is that it generates only binary trees, in other words, each internal node has exactly two branches. The Decision Tree partition splits the data set into smaller subsets, aiming to find the a subset with samples of the same category label. In this case, the subset is called *pure*, and the correspondent class can be determined [16,27]. However, there is a mixture of labels in each subset, and thus for each branch we will have to decide either to stop splitting and accept an imperfect decision, or instead select another property and grow the tree further, which suggests a recursive tree-growing process [16].

The tree is created by minimizing the impurity of the subsets. Let $i(N)$ be the impurity of a node N. To be pure the $i(N)$ must be zero, which means that all patterns belong to the same class. The most popular measure is the *entropy impurity*, given by (1):

$$i(N) = -\sum_j P(\omega_j) log_2 P(\omega_j), \tag{1}$$

where $P(\omega_j)$ is the fraction of patterns at node N that are in category ω_j. By the properties of entropy, if all the patterns are of the same category, the impurity is 0; otherwise it is positive, with the greatest value occurring when the different classes are equally likely. The log is used to the base 2 since the information is encoded in bits.

Another definition of impurity consists of using the variance impurity, which is related to the variance of a distribution associated with the two categories. It is particularly useful in the two-category case. Therefore, a generalization of the variance impurity, applicable to two or more categories, is written as (2), and it is called *Gini impurity*:

$$i(N) = \sum_{i \neq j} P(\omega_i) P(\omega_j) = 1 - \sum_j P^2(\omega_j). \tag{2}$$

Therefore, the Gini measures the divergences between the probability distributions of the target attributes values [27].

The Gini index may encounter problems when the domain of the target attribute is wide. In this case, in multiclass binary tree creation, is recommended to employ a binary criterion called *Twoing*. The overall goal is to find the split that best splits groups of the c categories. Let the class of categories be $C = \{\omega_1, \omega_2, ..., \omega_c\}$. At each node, the decision splits the categories into $C_1 = \{\omega_{i_1}, \omega_{i_2}, ..., \omega_{i_k}\}$ and $C_2 = C - C_1$. For every candidate split s, the change in impurity $\Delta i(s, C_1)$ is computed as though it corresponded to a standard two-class problem. That is, the split $s^*(C_1)$ that maximizes the change in impurity, is found. Finally, the supercategory C_1^* which maximizes $\Delta i(s^*(C_1), C_1)$ is obtained.

As a consequence of computational cost, the entropy impurity is more used, although the Gini impurity receives significant attention as well [16]. However, due to the mixture of labels in each subset, not always the pure class is achieved, and it is necessary to decide if stop splitting, and accept the imperfect decision, or instead selected another property and further grow the tree [16].

If we continue to grow the tree fully until each leaf node corresponds to the lowest impurity, then the data has typically been overfitting. The principal alternative approach to stop splitting is *pruning*. In pruning, a tree is grown fully, until leaf nodes have minimum impurity. Then, all pairs of neighboring leaf nodes are considered for elimination. Any pair whose elimination yields a satisfactory increase in impurity is eliminated, and the common antecedent node declared a leaf [16].

One of the pruning processes that can be applied in the tree generated by CART algorithm is the *Cost Complexity Pruning* (CCP) method. In the first stage, a sequence of trees $T_0, T_1, ..., T_k$ is built on the training data where T_0 is the original tree before pruning and T_k is the root tree [27].

In the second stage, one of these trees is chosen as the pruned tree, based on its generalization error estimation. Tree T_{i+1} is obtained by replacing one or more of the sub-trees in the predecessor tree T_i with suitable leaves. The pruned sub-trees are those that obtain the lowest increase in apparent error rate per pruned leaf, given by (3).

$$ErrorRate = \frac{\varepsilon(pruned(T,t),S) - \varepsilon(T,S)}{|leaves(T)| - |leaves(pruned(T,t))|} \tag{3}$$

where:

I) $\varepsilon(T,S)$ indicates the error rate of tree T over the sample S;

II) $|leaves(T)|$ indicates the number of leaves in T;

III) $pruned(T,t)$ indicates the tree obtained by replacing the node t in T with a suitable leaf.

In the second stage, the generalization error of each pruned tree is estimated and the best one is then selected [9,27]. The Cost Complexity Pruning algorithm used in CART is an example of the post-pruning approach. This approach considers the cost complexity of a tree to be a function of the number of leaves in the tree and the error rate of the tree (where the error rate is the percentage of tuples misclassified by the tree). It starts from the bottom of the tree. For each internal node, N, it computes the cost complexity of the subtree at N, and the cost complexity of the subtree at N if it were to be pruned (i.e., replaced by a leaf node). The two values are compared. If pruning the subtree at node N would result in a smaller cost complexity, then the subtree is pruned. Otherwise, it is kept [18].

3.2 C4.5 Algorithm

The C4.5 algorithm is an evolution of the Iterative Dichotomiser (ID3) algorithm, since C4.5 can run categorical or numerical attributes and it can run with unknown values. Both algorithms are developed by Quinlan [25]. C4.5 uses the so called *Gain Ratio* as splitting criteria, in which the element with highest ratio is taken as the root node and data set is then split based on the root element values [27]. When the number of data is less than a threshold, the split process is stopped.

In order to define Gain Ratio, the Entropy and the Information Gain definitions must be introduced, which are used by ID3 algorithm as attribute-selection measures [18]. The notation used here is as follows. Let D, the data partition, be a training set of class-labeled tuples. Suppose the class label attribute has m distinct values defining m distinct classes, C_i (for $i = 1,...,m$). Let $C_{i,D}$ be the set of tuples of class C_i in D. Let $\mid D \mid$ and $\mid C_{i,D} \mid$ denote the number of tuples in D and $C_{i,D}$, respectively [18].

Let node N represent or hold the tuples of partition D. The attribute with the highest information gain is chosen as the splitting attribute for node N.

The expected information needed to classify a tuple in D is also called *Entropy* and is given by [18]

$$Entropy(D) = -\sum_{i=1}^{m} P_i log_2(P_i), \tag{4}$$

where P_i is the nonzero probability that a tuple in D belongs to class C_i and is estimated by $\mid C_{i,D} \mid / \mid D \mid$.

Suppose the tuples in D are partitioned on some attribute A having v distinct values, $\{a_1,a_2,...,a_v\}$. If A is discrete-valued, so these values will correspond directly to the v outcomes of a test on A. Therefore, the attribute A can be used to split D into v partitions or subsets, $\{D_1,D_2,...,D_v\}$, where D_j contains those tuples of D, that have outcomes a_j of A. These partitions would correspond to the branches grown from node N. The ideal would be that each partition was pure, which means that each partition would produce an exact classification of the tuples. Since it is quite likely that the partitions are impure, a greater amount of information is required. This amount is measured by

$$Entropy_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} EntropyD_j, \tag{5}$$

the term $\frac{|D_j|}{|D|}$ acts as the weight of the jth partition and the $Entropy_A(D)$ is the expected information required to classify a tuple from D based on the partitioning by A. The smaller the $Entropy_A(D)$, the greater the purity of the partitions. Thus, the Information Gain is defined as the difference between the original information requirement, based on just the proportion of classes, and the new requirement, obtained after partitioning on A. That is:

$$Gain(A) = Entropy(D) - Entropy_A(D). \tag{6}$$

The attribute A with the highest $Gain(A)$ is chosen as the splitting attribute at node N [18].

The Information Gain measure is biased toward tests with many outcomes; it prioritizes the selection of attributes having a large number of values. Then, C4.5 algorithm, as an evolution of ID3, uses an extension to Information Gain known as *Gain Ratio*, which attempts to overcome this bias. The Gain Ratio applies a kind of normalization to Information Gain using a *split information* value. This value represents the potential information generated by splitting the training data set, D into v partitions, corresponding to the v outcomes of a test on attribute A. The split information (*SplitInfo*) is defined as

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{D_j}{D} log_2\left(\frac{|D_j|}{|D|}\right). \tag{7}$$

For each outcome, it considers the number of tuples having that outcome with respect to the total number of tuples in D. It differs from information gain, which measures the information with respect to classification that is acquired based on the same partitioning. The Gain Ratio is defined as

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}. \tag{8}$$

The attribute with the maximum $GainRatio(A)$ is selected as the splitting attribute. Note, however, that as the split information approaches 0, the ratio becomes unstable. A constraint is added to avoid this, whereby the information gain of the test selected must be large - at

least as great as the average gain over all tests examined [18].

About the pruning process, traditionally, C4.5 uses the so called *Error Based Pruning* (EBP) method to optimize the model generated in the growing phase. In C4.5 also is possible to use the *Reduce Error Pruning* (REP) method as an alternative pruning method, both described as follows.

In EBP, the error rate is estimated using the upper bound of the statistical confidence interval for proportions, as in (9):

$$\varepsilon_{UB}(T,S) = \varepsilon(T,S) + Z_{1-\alpha}\sqrt{\frac{\varepsilon(T,S)(1-\varepsilon(T,S))}{|S|}} \quad (9)$$

where $\varepsilon(T,S)$ denotes the misclassification rate of the tree T on the training set S; Z is the inverse of the standard normal cumulative distribution; and α is the desired significance level [9].

Let $subtree(T,t)$ denote the subtree rooted by the node t. Let $maxchild(T,t)$ denote the most frequent child node of t (namely most of the instances in S reach this particular child) and let S_t denote all instances in S that reach the node t. The procedure traverses bottom-up all nodes and compares the following values:

I) $\varepsilon_{UB}(subtree(T,t),S_t)$

II) $\varepsilon_{UB}(pruned(subtree(T,t),t),S_t)$

III) $\varepsilon_{UB}(subtree(T,maxchild(T,t)),Smaxchild(T,t))$

According to the lowest value, the procedure either leaves the tree as is; prune the node t; or replaces the node t with the subtree rooted by $maxchild(T,t)$, [9,27].

The REP method, proposed by [24], uses a hold out set for error estimates. While traversing over the internal nodes from the bottom to the top, the procedure checks each internal node to determine whether replacing it with the most frequent class does not reduce the trees accuracy. So, the node is pruned if accuracy is not reduced and the procedure continues until any further pruning would decrease the accuracy [9,27].

3.3 Random Forest

Random Forest is an ensemble learning method for classification, created by Breiman (2001). The Random Forest uses CART methodology to grow trees with maximum size and without pruning [7]. Due to their power, versatility, and ease of use, Random Forest is quickly becoming one of the most popular machine learning methods [20]. A Random Forest is composed of Decision Trees, where each Decision Tree is considered as an element of this ensemble, denominated forest. Random Forest algorithm integrates the decisions obtained from each tree that composes the forest. Thus, ensemble classification methods train several classifiers and combine the decision of a set of classifiers by weighted or unweighted voting process to classify unknown examples [13,30].

For the kth tree, a random vector Θ_k is generated, independent of the past random vectors $\Theta_1,...,\Theta_{k-1}$ but with the same distribution; and a tree is grown using the training set and Θ_k, resulting in a classifier $h(\bar{x},\Theta_k)$ where $\bar{x}$ is an input vector [7].

Formally, an Random Forest is defined by [7] as follows:

Definition: A Random Forest is a classifier consisting of a collection of tree structured classifiers $\{h(\bar{x},\Theta_k),k=1,...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\bar{x}$.

An ensemble classifier is generally found to be more accurate than any of the individual classifiers making up the ensemble [13]. In order to grow ensembles, often random vectors are generated that govern the growth of each tree in the ensemble. Bagging, Random split and Random subspace are some examples of split selectors [7]. In Bagging, a random selection is made from the examples in the training set, this selection is done without replacement. In Random split method, each node split is selected at random from among the K best splits. Finally, in Random Subspace does a random selection of a subset of features to use to grow each tree.

Random Forests consider many fewer attributes for each split, for this reason, it can efficiently handle with extremely large datasets [18,20]. The generalization error for a forest converges as long as the number of trees in the forest is large. Thus, overfitting is not a problem.

The accuracy of a Random Forest depends on the strength of the individual classifiers and a measure of the dependence between them. The idea is to maintain the strength of individual classifiers without increasing their correlation. A Random Forest is insensitive to the number of attributes selected for consideration at each split [18]. The major limitation of the Random Forest algorithm is that a large amount of trees can become the algorithm slow and inefficient for predictions after training.

3.4 Statistical Measures to Evaluate the Classification

In order to evaluate the performance of the classifiers and search for the best mathematical model to describe the TIMs faults in different aspects, some statistical measures are applied to analyze the efficiency of each classification model. In this work, besides evaluating the Decision Tree models considering the number of leaves and branches, the following concepts are applied: Confusion Matrix, Accuracy, Receiver Operating Characteristic (ROC) and Kappa Index, which are described below.

A *Confusion Matrix* is a square matrix, with dimension greater than or equal to two, that categorizes

predictions according to whether they match the actual value in the data [20]. Four terms are necessary to comprehend the confusion matrix, according the possible class classification:

- –True Positive (TP): Correctly classified as the class of interest
- –True Negative (TN): Correctly classified as not the class of interest
- –False Positive (FP): Incorrectly classified as the class of interest
- –False Negative (FN): Incorrectly classified as not the class of interest.

When the predicted value is the same as the actual value, this is a correct classification. Correct predictions are observed in the matrix main diagonal, while the incorrect ones are located in the off-diagonal. The performance measures for classification models are based on the counts of predictions falling on and off the diagonal in the confusion matrix.

One of the measures derived from the confusion matrix concepts is *Accuracy*. The Accuracy is the proportion that represents the number of true positives and true negatives divided by the total number of predictions, given by (10)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \tag{10}$$

This measure indicates how well the classifier is able to recognize elements of different classes [20].

Another important measure used in this paper is the *Receiver Operating Characteristic* (ROC) or ROC curve. This measure is used to examine the tradeoff between the detection of true positive, while avoiding the false positives.

A typical ROC curve diagram is illustrated in Figure 1. These curves are defined on a plot with the proportion of true positives on the vertical axis, and the proportion of false positives on the horizontal axis.

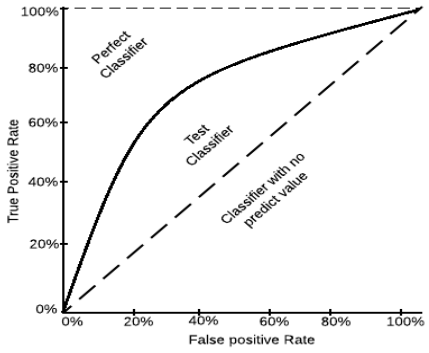


Fig. 1: ROC curve

To create the curves, a classifier's predictions are sorted by the model's estimated probability of the positive class, starting with the largest values. Beginning at the origin, each prediction's impact on the true positive rate and false positive rate will result in a curve tracing vertically (for a correct prediction), or horizontally (for an incorrect prediction). The closer the curve is to the perfect classifier, the better it is at identifying positive values. This can be measured using a statistic known as the *area under the ROC curve*, which measures the total area under the ROC curve, with the area ranging from 0.5 (for a classifier with no predictive value), to 1.0 (for a perfect classifier) [20].

Finally, it is also used the *Kappa Statistic Index* [11]. This index adjusts accuracy by accounting for the possibility of a correct prediction by chance alone. The Kappa values ranges from 0 to 1, the closer to 1 the better the rating is. The Kappa Index can be obtained by (11)

$$k = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}, \tag{11}$$

where Pr refers to the proportion of actual (a) and expected (e) agreement between the classifier and the true values.

4 Obtaining the Data Set: Audio Signals

In order to classify three-phase induction motors faults using audio signals, an experimental setup has been assembled at the Intelligent Systems Laboratory (ISL), together with Signal Processing and Applications Laboratory (LPSA), located in the Federal University of Technology of Paraná, in Cornélio Procópio city, in the state of Paraná, Brazil.

The data set is then composed by acoustic emission recorded by two Behringer ECM8000 condenser type omni-directional microphone, being the acoustic emission sensors running to a *Focusrite Scarlett 2i2* audio interface, which is responsible for the data acquisition. The microphones were positioned close to the electric motor, with 9.53 cm between them, as showed in Figure 2, in order to avoid any spatial ambiguity. The best distance has been obtained experimentally.

Since the classification models considered in this paper, which use CART, C4.5 and Random Forest algorithms, are *supervised* models, the data set is composed by input attributes and outputs. In order to detect faults and to classify patterns, it is necessary to extract attributes that represent acoustic emission signals. Time domain and some frequencies characteristic are used in this paper. The frequency characteristics are very reliable for fault detection, so the time signals are divided in portions of XYZ samples, without overlaps.

The input attributes for the classification models consist of the peak magnitudes of the signal frequency spectrum at predetermined frequencies and the total signal power. The *frequency peaks* were extract using the *Fast Fourier Transformer* algorithm and *Hanning* windowing, considering the frequencies of 30, 60, 120 and 2500 Hz.

Fig. 2: Equipment used for obtaining the data set.

From the literature, electrical unbalance generates vibration in the power supply frequency first harmonic [10]. Broken rotor bars fault causes vibration around the harmonics of the rotor speed, related to supply frequency [12]. Bearing faults are commonly detected by modulations in high frequency [26]. At last, winding faults vibrations are related to the power supply frequency harmonics and slip [14].

Therefore, we have 5 numerical input attributes: frequency peak of 30 Hz, frequency peak of 60 Hz, frequency peak of 120 Hz, frequency peak of 2500 Hz and the *signal power*. In this work, we have estimated the signal power using the autocorrelation of the signals in point 0. This feature is obtained using the crosscorrelation function of the signal with itself. Equation (12) describes the crosscorrelation of two signals x_1 and x_2 with length N.

$$\hat{R}_{x_1x_2}(m) = \sum_{n=0}^{N-m-1} x_1(n+m)x_2(n) \qquad (12)$$

with $m = 0,1,\ldots,N-1$ and $x_1(n)$ and $x_2(n)$ are the signals value at the sample n. The signal power features are calculated as $\hat{R}_{x_1x_1}(m)$ and $\hat{R}_{x_2x_2}(m)$.

The output data set of the supervised classification system is divided into three classes, called Mechanical Conditions: healthy motor, motor with bearing outer race fault and motor with 2 broken rotor bars. Each one of these three Mechanical Conditions are subdivided into five other subclasses, called Electrical Conditions, which correspond to the unbalancing of the supply voltage phases, as shown in Table 1. Each one of the 15 rows of Table 1 corresponds to an output of the classification system, as shown in the last column of the table. The unbalance generates an excessive current flow in one or more phases, in this way the motor suffers the voltage unbalance, causing abnormal behavior in the rotor and irregular vibration. The voltage difference affects the magnetic balance in the stator, resulting in vibrations at the first harmonic of power supply frequency. In this work, we have tested four unbalance configuration, as follows: (a) 2% higher voltage in phase A, (b) 2% higher voltage in phase B and less voltage in phase C, (c) 4% higher voltage in phase A, and (d) 4% higher voltage in phase B and 4% less voltage in phase C.

Table 1: Output data set

Mechanical Conditions	Electrical Conditions	Outputs
Healthy	normal	output 1
	2% for (-A)	output 2
	2% for (+B, -C)	output 3
	4% for (-C)	output 4
	4% for (+B, -C)	output 5
Bearing outer race fault	normal	output 6
	2% for (-A)	output 7
	2% for (+B, -C)	output 8
	4% for (-C)	output 9
	4% for (+B, -C)	output 10
Broken rotor bar	normal	output 11
	2% for (-A)	output 12
	2% for (+B, -C)	output 13
	4% for (-C)	output 14
	4% for (+B, -C)	output 15

The healthy motor data is obtained from a three-phase, 1HP and 4 pole electric motor with 220/380 V - 60 Hz power supply, considering nominal speed of approximately 1700 RPM.

In order to obtain the bearing outer race faults data, a corrosive slurry is placed in the outer race of the bearing and after the action of the pulp the bearing is inserted into the motor after the action of the pulp the bearing was cleaned and properly lubricated with greasy in order to simulate normal operation. Finally, the last class consists of signal samples provided by motor with 2 broken rotor bars. The bars are damaged using a drilling machine in two adjacent bars to emulate the fault.

5 Results

The data set described in the previous section is divided into two other mutually exclusive sets: training set and test set. The objective is to classify mechanical and electrical TIMs faults and compare the performance of supervised classification models using Decision Trees algorithms (CART and C4.5) and the Random Forest algorithm, also considering their pruning methods. The algorithms are implemented using the WEKA software [1], developed by the University of Waikato, New Zealand. The hardware used in computational experiments is composed by an Intel Core I7 processor, RAM memory 16GB executing operational system Microsoft Windows 10.

[1] Waikato Environment of Knowledge Analysis - http://www.cs.waikato.ac.nz

The training set is used for the knowledge acquisition, i.e., to "learn" the possible patterns extracted from the data and to generate a mathematical model able to generalize the acquired knowledge for samples of unknown data. After that, from the classification model generated, the samples that comprise the test set are inserted in the model and a classification is obtained for these data. Since the classification models are supervised, both training and test sets are composed by the input attributes (frequency peaks of 30, 60, 120 and 2500 Hz and signal power) and output (one of the 15 possibilities described in Table 1). Then, the results provided from the test set are analyzed. If the model correctly classified the sample of the test set (comparing it with the known output), then one success is counted; otherwise, one error is considered. This method is known in the literature as *holdout* [19]. For that, the data set is randomly divided in proportion 80/20, in which the training set consists of 80% of the data and the test set is composed of the remaining 20%. Since the total data set consists of 570 rows (instances) and 6 columns (the 5 input attributes and the output), 456 instances correspond to the training set and 114 instances correspond to the test set.

For performance comparison reasons, as described in Section 4, data are provided from 2 microphones, named as Microphone 1 and Microphone 2. Therefore, the classification methods are applied separately. For each one of the microphones, 6 classification models are developed: first, by using CART algorithm with unpruned method and using the CCP method. Second, by using C4.5 algorithm with unpruned method and using the two pruning methods described in Section 3.2, the EBP and REP methods. Finally, the Random Forest algorithm is also applied to both microphones.

Tables 2 and 3 present, respectively, results obtained from the Microphone 1 and Microphone 2.

Table 2: Results from Microphone 1

Microphone 1						
Algorithms	Prune	Leave	Size	Kappa Index	ROC area	Accuracy
CART	CCP	47	93	0.872	0.952	88.330
CART	unpruned	49	97	0.872	0.952	88.330
C4.5	EBP	46	91	0.964	0.999	96.667
C4.5	REP	34	67	0.935	0.955	85.000
C4.5	unpruned	46	91	0.964	0.999	96.667
Random Forest	-	-	-	0.872	0.996	88.333

The columns of Tables 2 and 3 show the algorithms used for the faults classification and their respective methods of pruning. In order to evaluate the complexity of the Decision Trees, the number of leaves and the size of the trees generated for each classification model are shown in columns 3 and 4. Besides, in order to evaluate

Table 3: Results from Microphone 2

Microphone 2						
Algorithms	Prune	Leaves	Size	Kappa Index	ROC area	Accuracy
CART	CCP	41	81	0.817	0.929	83.333
CART	unpruned	52	103	0.817	0.920	83.333
C4.5	EBP	48	95	0.936	0.958	94.167
C4.5	REP	37	73	0.863	0.963	87.500
C4.5	unpruned	49	97	0.936	0.986	94.167
Random Forest	-	-	-	0.8812	0.997	89.167

the performance of the classification models, the statistical measures described in Section 3.4 are calculated and shown in Tables 2 and 3. They are: Kappa Index, area under the ROC curve and the Accuracy.

Since the data were collected simultaneously and with few divergences, a high correlation between the results of the two microphones was already expected. From the evaluation of the presented results, for Microphone 1 (Table 2), we can observe that the model generated by using C4.5 algorithm together with EBP pruning method and with unpruned method, providing the best performance, since these two models present the best and the same accuracy, area under the ROC curve and Kappa Index. In addition, they also present the smallest size and the smallest number of leaves, which prove that these two models are less complex than the others.

Analyzing the Microphone 2 (Table 3), the model generated by using C4.5 algorithm together with EBP pruning method also provided the best performance. Although this model has an area under the ROC curve slightly smaller than unpruned method, it presents the smallest size and the smallest number of leaves, which reduce the complexity of the model. Therefore, the classification model that presented the best performance, for both microphones, is the model generated using C4.5 algorithm together with EBP pruning method. For this reason, the Confusion Matrices are obtained for this model, using the test set, for Microphone 1 and for Microphone 2, as can be seen in Figures 3 and 4, respectively.

Once the Decision Tree is generated using the C4.5 algorithm together with EBP method for the Microphone 1, each branch provides a classification linguistic rule. For example, from a particular branch of the tree, the following rule can be extracted:

"IF signal power is $>$ 17754.01 AND $\leq$ 26008.19 AND frequency peak of 120Hz is $\leq$ 0.095915 AND $\geq$ 0.084671 THEN output 1 (motor with healthy mechanical condition and balanced electrical condition)"

Similarly, for the Microphone 2, from a particular branch of the tree, the following rule can be extracted:

"IF signal power is $\leq$ 34750.85 AND frequency peak of 30Hz is $\leq$ 0.191952 AND signal power is $>$ 21516.17

Microphone 1 - C4.5 - EBP														
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	classified as
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = output 1
1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	b = output 2
0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	c = output 3
0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	d = output 4
0	0	0	1	13	0	0	0	0	0	0	0	0	0	0	e = output 5
0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	f = output 6
0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	g = output 7
0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	h = output 8
0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	i = output 9
0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	j = output 10
0	0	0	0	0	0	0	0	0	0	12	2	0	0	0	k = output 11
0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	l = output 12
0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	m = output 13
0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	n = output 14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	o = output 15

Fig. 3: Confusion Matrix for Microphone 1

Microphone 2 - C4.5 - EBP														
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	classified as
13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	a = output 1
2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	b = output 2
0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	c = output 3
1	0	0	13	0	0	0	0	0	0	0	0	0	0	0	d = output 4
0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	e = output 5
0	0	0	0	0	12	0	0	0	2	0	0	0	0	0	f = output 6
0	0	0	0	0	0	4	0	0	1	0	0	0	0	0	g = output 7
0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	h = output 8
0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	i = output 9
0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	j = output 10
0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	k = output 11
0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	l = output 12
0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	m = output 13
0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	n = output 14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	o = output 15

Fig. 4: Confusion Matrix for Microphone 2

AND signal power is $\leq$ 24184.34 AND signal power is $\leq$ 21616.64 THEN output 15 (motor with Broken rotor bar mechanical condition and unbalanced electrical condition of 4% for +B and -C)"

The diagonals of the Confusion Matrices illustrated in Figures 3 and 4 present the highest numerical values of the matrix. Since all correct classifications are located in the diagonal of the Confusion Matrix (Section 3.4), the test sets are correctly classified using C4.5 with EBP, with high accuracy.

6 Conclusion

Due to the great use of TIMs, it is necessary to develop efficient techniques capable of recognizing and classifying faults in order to avoid the high maintenance costs and the stops in the production process. In this paper, supervised learning approaches are proposed in order to classify three-phase induction motors faults, applying *Decision Trees* and *Random Forest* algorithms. These methods are applied since they are supervised methods and the structure of Decision Trees allows access to the parts of the classification process from algorithm building blocks. In addition, from Decision Trees branches, it is possible to generate understandable "IF-THEN" linguistic rules, which facilitates the understanding of the results by a non-expert user. Numerical data referring to the motor audio signals are considered as input attributes for the classifiers and it is a great contribution to the research area.

Although the execution time of the algorithms is not a parameter of comparison of their performances, the time is mentioned here so that readers can have an idea of how long these problems take to be solved. The Decision Trees algorithms, C4.5 and CART, took about 10 seconds to read data and generate the tree. Random Forest algorithm took about 15 seconds.

From results presented in the previous section, we can conclude that all the classification models yield very satisfactory results and can be used to obtain the required classification with high accuracy. Additionally, Tables 2 and 3 present values of statistical measures so that it is possible to choose the model that presents the best classification in relation to the others.

The classification models generated by the CART algorithm, for both microphones, present the same accuracy and Kappa Index; the areas under the ROCs curves are also very close. However, there is a reduction in the complexity of the models (size and number of leaves), when a pruning method is applied, without changing the statistical measures. This fact evidences the robustness of pruning in Decision Tree models, since the technique allows the optimization of the model and avoids overfitting without affecting the performance of the classifier.

In relation to the classification models generated by the algorithm C4.5, analyzing Tables 2 and 3, the model pruned by the REP method has shown a greater reduction of the trees complexity. On the other hand, this method significantly reduces the accuracy of the classification compared to the others. When comparing the unpruned with those pruned by the EBP method, there are no changes in Microphone 1, neither in the complexity of the trees nor in the statistical measures; while in Microphone 2, there is a reduction in the complexity of the tree pruned by the EBP method. This occurred since the unpruned trees are very close to the most optimized model according to the constructive metrics of the EBP.

Analyzing the classification models generated by Random Forest, the results are very close to the results presented by the trees generated by the CART algorithm for the Microphone 1, whereas for the Microphone 2, Random Forest present better classification performance than the CART algorithm. However, Random Forest performance is lower than the C4.5 algorithm for both microphones. Moreover, another difficulty found in

Random Forest classification models is that their structure does not provide the generation of linguistic rules, which makes the interpretation of the results more difficult.

When the data classification performances of microphones 1 and 2 are compared, it is possible to notice that the results of Microphone 1 are better. This may be justified due to the position of the microphones, since Microphone 2 is closer to the rotor and then it captures more noise than Microphone 1. These noises pollute the signals audio compromise the results.

Although the statistical measures presented by all the models are very satisfactory, Confusion Matrices shown in Figures 3 and 4 were generated using the model that presented the best performance (C4.5 with EBP). Some other models presented difficulties in differentiating some classes of data that are close, as the case of electrical faults caused by electrical unbalance. This confusion of classes is not desired when implementing the model in a real system that works with a critical component, which occurs with TIMs on a production process.

Therefore, by analyzing the statistical measures and confusion matrices, the classification models generated by the C4.5 algorithm together with the EBP method are the models that best classified the data set and, consequently, would present the best performance when implemented in a real system. The mechanical conditions are better classified than electrical conditions for the test set used in this work. Classifying electrical conditions from noise is a difficult task, but even in this case the classifier was efficient. The confusion matrices presented in Figures 3 and 4 show the presence of the well-defined main diagonal, evidencing the excellent performance of the classifiers, even for very close data classes.

References

[1] I. Aydin, M. Karakose, and E. Akin. An approach for automated fault diagnosis based on a fuzzy decision tree and boundary analysis of a reconstructed phase space. *ISA Transactions*, **53**:220–229, (2014).

[2] G. H. Bazan. *Medidas de Informação e Sistemas Inteligentes Aplicados no Diagnóstico de Curto-Circuito do Estator de Motores de Indução Trifásicos*. Master's thesis, M.A. thesis. Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brazil, (2016).

[3] A. J. Bazurto, E. C. Quispe, and R. C. Mendoza. Causes and failures classification of industrial electric motor. *IEEE ANDESCON*, (2016).

[4] A. Bellini, F. Filippetti, C. Tassoni, and G. Capolino. Advances in diagnostic techniques for induction machines. *IEEE Transactions on Industrial Eletronics*, **55**(10):4109–2008, (2008).

[5] P. S. Bhowmik, S. Pradhan, and M. Prakash. Fault diagnostic and monitoring methods of induction motor : A review. *Int. J. Appl. Control. Electr. Electron. Eng.*, **1**(1):1–18, (2013).

[6] A. Bonnett and C. Yung. Increased efficiency versus increased reliability. *IEEE Industry Applications Magazine*, **14**(1):29–36, (2008).

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

[8] L. Breiman, J. Friedman, and R. A. Olshen. *Classification and Regression Trees*. Chapman and Hall, (1984).

[9] G. M. Bressan, B. C. F. Azevedo, and E. A. S. Lizzi. A decision tree approach for the musical genres classification. *Appied Mathematics and Information Sciences*, **17**(6):1703–1713, (2017).

[10] M. Campbell and G. Arce. Effect of motor voltage unbalance on motor vibration: Test and evaluation. In *2016 Petroleum and Chemical Industry Technical Conference (PCIC)*, pages 1–7, Philadelphia, Pennsylvania, USA, Sept (2016). Institute of Electrical and Electronics Engineers (IEEE).

[11] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, **20**(1):37–46, (1960).

[12] P. A. Delgado-Arredondo et al. Methodology for fault detection in induction motors via sound and vibration signals. *Mechanical Systems and Signal Processing*, **83**(Supplement C):568 – 589, (2017).

[13] T. G Dietterich. *Ensemble Learing: The Handbook of Brain Theory and Neural Networks*. MIT Press, (2002).

[14] S. Djurović, D. S. Vilchis-Rodriguez, and A. C. Smith. Investigation of wound rotor induction machine vibration signal under stator electrical fault conditions. *The Journal of Engineering*, **2014**(5):248–258, (2014).

[15] G. Dougherty. *Pattern Recognition and Classification, An Introduction*. Springer, California - USA, (2013).

[16] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, (2000).

[17] J. Furnkranz, D. Gamberger, and Nada Lavrac. *Foundations of Rule Learning (Cognitive Technologies)*. Springer, London - UK, (2012).

[18] J. Han, M. Kamber, and J. Pei. *Data mining: Concepts and Techniques*. Morgan Kaufmann, (2012).

[19] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Appers in the International Joint Conference on Artificial Intelligence*, **2**:1137–1143, (1995).

[20] B. Lantz. *Machine learning with R*. Packt Publishing, (2013).

[21] H. Lee and S. Kim. Black-box classifier interpretation using Decision Tree and Fuzzy Logic-based classifier implementation. *International Journal of Fuzzy Logic and Intelligent Systems*, **16**(1):27–35, (2016).

[22] P. S. Panigrahy and P. Chattopadhyay. Improved classificaion by non iterative and ensemble classifiers in motor fault diagnosis. *Advances in Electrical and Computer Engineering*, pages 95–104, (2018).

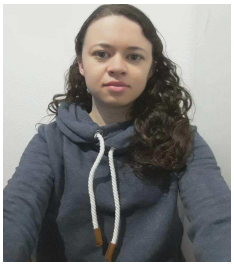
[23] R. K. Patel and V. K. Giri. Feature selection and classification of mechanical fault of an induction motor using random forest classifier. *Perspective in Science*, **8**(1):334–337, (2016).

[24] J. R. Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, **27**(3):221–234, (1987).

[25] J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, (1993).

[26] V. K. Rai and A. R. Mohanty. Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert-Huang transform. *Mechanical Systems and Signal Processing*, **21**(6):2607 – 2615, (2007).

[27] L. Rokach and O. Maimon. *Data mining with decision trees: theory and applications*. World Scientific, (2008).
[28] S. P. Santos and J. A. F. Costa. Application of multiple decision tree for condition monitoring in induction motors. *International Joint Conference on Neural Networks (IJCNN)*, pages 3736–3741, (2008).
[29] V. Sugumaran, V. Muralidharan, and K. I. Ramachandran. Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. *Mechanical Systems and Signal Processing*, **21**:930–942, (2007).
[30] B. Yang, X. Di, and T. Han. Random forest classifier for machine fault diagnosis. *Journal of Mechanical Science and Tecnology*, **22**(9):1716–1725, (2008).



Beatriz C. Flamia de Azevedo is bachelor's degree in Control and Automation Engineering from Federal University of Technology of Paraná (UTFPR), Brazil. Currently, she is a master's degree student in Industrial Engineering at Polytechnic Institute of Braganca (IPB), Portugal. Her interests relate to Optimization, Mathematics Modeling and Machine Learning.

Glaucia Maria Bressan received a teaching degree in Mathematics with emphasis in computation from Federal University of Sao Carlos (UFSCar), Brazil, in 2000, a Master's degree in Computational and Applied Mathematics from the University of Sao Paulo (USP) in 2003 and her PhD in Electrical Engineering at University of Sao Paulo (USP) in 2007. She did a post-doctoral research in Electrical Engineering at University of Sao Paulo (USP) in 2008. She works as a professor and researcher at the Federal University of Technology of Parana (UTFPR) in the Department of Mathematics, where she has been a professor since 2012. Her interests relate to Operational Research, machine learning, Bayesian Networks and Fuzzy Logic.

Cristiano Marcos Agulhari received the B.S. degree in Computer Engineering from Universidade Estadual de Campinas (2006), master's at Electric Engineering (2009) and doctorate at Electric Engineering (2013), both from Universidade Estadual de Campinas. Doctorate was done in cotutelle with the Institut National des Sciences Appliquees (INSA), Toulouse. Currently teaching at Universidade Tecnologica Federal do Paraná, acting on the following subjects: Signal processing, wavelets, data compression, biomedical signal compression, control systems, robust control, control of LTV systems, linear matrix inequalities (LMIs).

Herman L. dos Santos received a Master's degree in electrical engineering at the Technologic Federal University of Paraná (UTFPR) at Cornélio Procópio, Brazil, in 2018, where he also received the B.S. degree in Electrical Engineering in 2016. His current research interests are electric machine fault detection via acoustic emission signals, and beamforming.

Wagner Endo received the B.S. degree in Electrical Engineering in 2004, the M.Sc. degree in Electrical Engineering in 2006 both from the State University of Londrina, Londrina, Brazil and the Ph.D. degree in electrical engineering from the University of Sao Paulo (USP), Sao Carlos, Brazil, in 2014. Currently, he is an Associate Professor with the Federal Technological University of Paraná, Cornélio Procópio , Brazil. His research interests are within the fields of signal processing and automation of systems.