



Research article

Classification with automatic detection of unknown classes based on SVM and fuzzy MBF: Application to motor diagnosis

Romain Breuneval^{1,2}, Guy Clerc^{2,*}, Babak Nahid-Mobarakeh³ and Badr Mansouri¹

¹ Safran Electronics & Defense, Avionics Division, Center of Product Excellence in Avionic Systems and Actuation (CEP-SAA) F-91300, Massy, France

² University of Lyon, UCB Lyon 1, CNRS, AMPERE, F-69100, Villeurbanne, France

³ Groupe de Recherche en Electrotechnique et Electronique de Nancy (GREEN), University of Lorraine, F-54510 Nancy, France

* **Correspondence:** Email: guy.clerc@univ-lyon1.fr; Tel: +33472432833; Fax: +33472431193.

Abstract: Classification algorithms based on data mining tools show good performances for the automatic diagnosis of systems. However, these performances degrade quickly when the database is not exhaustive. This happens, for example, when a new class appears. This class could correspond to a previous unknown fault or to an unknown combination of simultaneous faults. Described algorithm in this paper proposes a solution to this issue. It combines Support Vector Machine (SVM), fuzzy membership functions (mbf) and fuzzy information fusion. It results in the construction of a matrix of memberships to known classes U_{class} and a vector of membership to unknown classes U_{others} . Then, from these values, indicators of distance and ambiguity of the observations can be computed. These indicators allow setting a simple rejection rule with a threshold classifier. The algorithm is validated by using Cross-Validation (CV) on experimental data on an induction motor faults supplied by a voltage-source inverter. The results show the good performances of the proposed algorithms and its suitability for transportation systems like aircrafts.

Keywords: health-monitoring; diagnosis; SVM; fuzzy membership function; information fusion; triangular norm; unknown classes; rejection; electromechanical actuator; reject option

Nomenclature

| | | | |
|--------------|--|------------|--|
| X | Matrix [m, d] of observations | Y | Vector [m, 1] of labels |
| x | Observation | y | Label of x |
| α | Lagrange Multipliers | s | Support Vector (SV) |
| K | Kernel function | θ_K | Parameters of K |
| b | Bias term | d | Distance to the separating hyperplane |
| c | Number of classes | u | Membership function |
| mbd | Membership degree | γ | Membership degree of SV |
| φ | T or S-Norm | ψ | Fusion operator based on φ |
| u_i | Membership to class for which $y = -1$ | u_2 | Membership to class for which $y = +1$ |
| g_k | Center of gravity of class k | ID | Indicator of Distance |
| IA | Indicator of Ambiguity | IA_{th1} | First threshold on IA (for decision rules) |
| IA_{th2} | Second threshold on IA (for decision rules) | D | Defined classes (in learning base) |
| ND | Undefined classes | P_k | Power related fault features |
| H_k | Harmonics related fault features | S_k | Fault features based on stator currents |
| MDL | Fuzzy SVM binary model | | |
| $MLFZ_S$ | SVM-MBF multi class model (trained and set) | | |
| U_{class} | Matrix [m, c] of membership to the classes | | |
| U_{others} | Vector [m, 1] of membership to unknown classes | | |
| ID_{th1} | First threshold on ID (for decision rules) | | |
| ID_{th2} | Second threshold on ID (for decision rules) | | |

1. Introduction

Maintenance costs are a limiting factor to the economical performances of industrial processes. All industries are impacted by this and more particularly transportation industries: maritime [1], railway [2] and aeronautics [3]. Among the existing methods, predictive maintenance is a promising family of tools to help reducing the cost while increasing the availability of industrial systems. These methods consist in analyzing measurements on the system/process to estimate the state of degradation of the system and, if there is a fault, to detect its localization. It allows reducing costs up to 30% in the power generation and in the oil/gas industry [4]. It might also help to increase the availability of aircrafts by over 35% [4]. Most of these methods suppose that a

representative database on the system operating states has been built. This database should contain observations for the healthy case and main faulty cases, i.e., the most frequent and the most critical faults.

However, if a new type of fault appears, the algorithm will not be able to detect it. It may classify it in one of the defined/identified faults or even as healthy operating. The proposed algorithm in this paper provides a solution to this problem. It is a two-level classifier. The first one is based on support vector machines (SVM), fuzzy membership functions (mbf) and fuzzy fusion (FF). Its outputs allow computing rejection and ambiguity features. These features feed a simple threshold classifier in order to determine if the points belong to a single fault mode, to a combined fault mode (ambiguity) or to an unknown fault mode (rejection). Other SVM algorithms with rejection option exist in the literature [5–7]. However, they do not distinguish rejection in distance from rejection in ambiguity. Moreover, there is no degree level associated to the rejection. Here, the proposed algorithm solves these two issues. Indeed, other versions of fuzzy SVM exist [8–10]. Unlike these methods, the proposed algorithm can classify an observation to several classes and to unknown classes, i.e., those not defined in the training database.

In this paper, to assess performances of the proposed algorithm, it is applied to an induction motor. This type of motor is widely used in industry including transportation systems [11–13]. Recent investigations are made on the diagnosis of induction motors [14–17,19,20], but none of them consider the case of unknown faults.

Here is the organization of this paper: a short reminder on support vector machine is given first (Section 2). Then, the proposed multi-classes fuzzy SVM-MBF is explained in Section 3. It allows ambiguity and distance rejection options. The principle of the automatic detection of unknown classes is also presented in detail in Section 4. The general architecture is detailed in Section 5. Then, Section 6 describes the experimental tests emulating induction motor faults. These tests are used to create a database, which is used to validate the algorithm. The optimal selection of signature of fault is realized in Section 7. Experimental results validate the approach in Section 8. A comparison with classical classifier is made in Section 9.

2. Support vector machine

2.1. Binary case classifier

The aim of training a SVM model is to find a hyperplane allowing separating two classes of observations [18]. It is computed by maximizing the margin between the hyperplane and the nearest observations and so, generating support vectors (SV). The classification of a new observation is made by computing the distance between this observation and the hyperplane. If the distance is positive, then the label of the observation is +1. If not, its label is -1. The distance is computed for an observation $x_{new} \in \mathbb{R}^p$ using the following equation [21]:

$$d(x_{new}) = \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x_{new}) + b \quad (1)$$

where $s_i \in \{1, \dots, N_s\}$ is a support vector (observation for which $d(s_i) = \pm 1$), α_i : the Lagrange

multipliers, y_i : label of s_i , b : the bias and K the kernel function.

The Lagrange multipliers are computed by maximizing the following cost function:

$$L_D = \sum_{i=1}^l \alpha_i - \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (2)$$

under the following constraints:

$$C \geq \alpha_i \geq 0 \quad (3)$$

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (4)$$

with C as the regularization parameter allowing us to set the bias/variance trade-off of the SVM model and m the number of observations in the training database.

This formulation of the SVM learning problem is a quadratic optimization problem. Therefore, a quadratic optimization solver has to be applied to solve it [22]. The kernel function is a kind of similarity measure. The function must verify some properties like continuity, symmetry and semi-definite positive [18]. This function and its parameters have to be chosen regarding the database. Some examples of kernel functions are given in Table 1.

Table 1. Kernel functions.

| Name | Parameters (θ_K) | Formula |
|-----------------------------|------------------------------|--|
| Linear | | $K(x_i, x_j) = x_i \cdot x_j$ |
| Polynomial | p | $K(x_i, x_j) = (x_i \cdot x_j + 1)^p$ |
| Radial Basis Function (RBF) | σ | $K(x_i, x_j) = e^{-\frac{\ x_i - x_j\ ^2}{2\sigma^2}}$ |
| Sigmoidal | $a ; b$ | $K(x_i, x_j) = \tanh(a * x_i \cdot x_j + b)$ |

2.2. Multi-class case classifier

SVM can be generalized to multi-class cases by combining several binary SVM. Two main techniques exist in the literature.

2.2.1. One vs All

The “One vs All” method is illustrated in Figure 1.

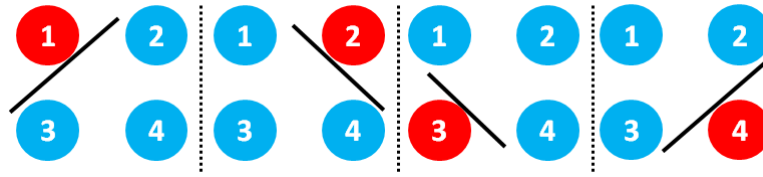


Figure 1. “One vs All” method.

This method consists in splitting classes in two groups. One of them is made of only one class. The other one gathers all the other ones. The considered class is labeled as +1 and the other ones are labeled as -1. Then, all the classes are successively labeled as +1. Therefore, for a c class problem, the method will result in the combination of c hyperplanes.

2.2.2. One vs One

The “One vs One” method is illustrated in Figure 2.

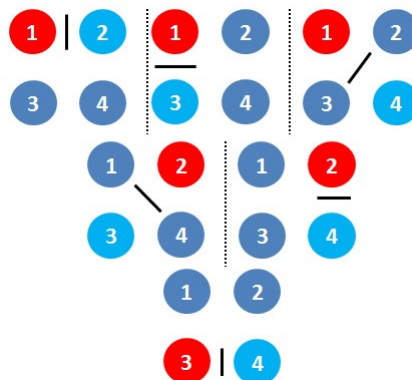


Figure 2. “One vs One” method.

This method consists in processing successively each pair of classes, without considering the other classes. Therefore, for a c class problem, the method will count the combination of $c(c - 1)/2$ hyperplanes. It is generally more accurate than the “One vs All”. However, its computation costs are also higher, for both training and test phases. For this reason, the “One vs All” is used in the rest of the paper.

3. Presentation of the new fuzzy SVM classifier: SVM-MBF

A method to combine SVM with fuzzy membership functions is now introduced.

3.1 Binary case

First, for the binary case of SVM, the membership degree (mbd) of an observation to a class is

introduced. We remind that the “binary case” means classification between two classes. The notion of membership degree is based on Zadeh’s fuzzy set theory [23]. A membership function (mbf) is attributed to each class. For a given observation, the mbf are calculated from the distance between the observation and the separating hyperplane d_{sv} as illustrated in Figure 3.

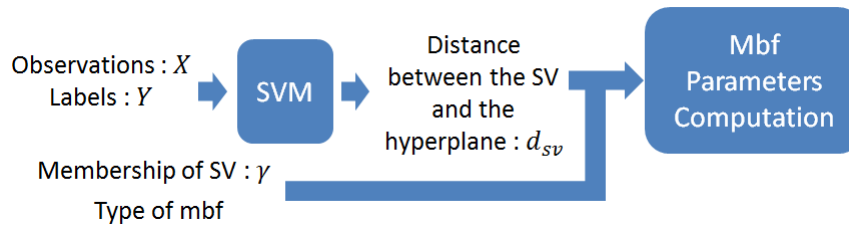


Figure 3. Process to compute fuzzy mbf parameters (SV stands for Support Vector).

Membership functions can have different forms: trapezoidal, polynomial, sigmoidal, etc. They must meet the following requirements:

- The mbf corresponding to the positive class u_2 should be symmetric around the hyperplane to the one corresponding to the negative class u_1 .
- The two mbf represent a strong fuzzy partition of the distance set. This means that $u_1(d_x) + u_2(d_x) = 1, \forall d_x$, with d_x : distance to the hyperplane of an observation x .
- The mbd of the support vector represents the confidence in the quality of the training data. The higher it is, the better the training data is. This level of confidence is noted $\gamma \in [0.5, 1]$.

An example of fuzzy partitions of the distance, for polynomials mbf, is given in Figure 4.

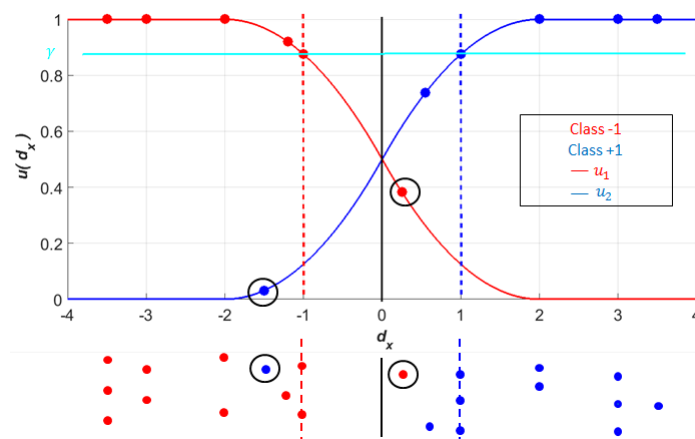


Figure 4. Membership functions (top) and the SVM (bottom).

Two misclassified points by the classical SVM are encircled in black in Figure 4. Though these points have non-zero memberships for both classes, their affiliation to a class could avoid some classification error by considering a threshold $u_{th} \in [0.5, \gamma]$ in the case of the fuzzy SVM.

Clearly speaking, if $u(x) > u_{th}$, then the observation is classified. Otherwise, further investigations are required for this observation.

3.2. Multi-class case

This is the same principle as for classical multi-class SVM. Several binary fuzzy SVM models are trained using one of the multi-class methods. Then, during the classification phase, the computed membership degrees (mbd) for binary case are combined by using triangular operators [24] and complement to one [25], as explained later in this paper. This allows building a matrix U_{class} , of m rows and c columns. It represents the mbd of the observations to the classes. It also allows building a vector U_{others} , representing the mbd of the observation to unknown classes.

3.2.1. Fusion of membership degrees

To merge the membership degrees during the different steps of the fuzzy SVM process, triangular operators are used. T-Norm (equivalent to intersection) and T-Conorm (or S-Norm, equivalent to union) operators must be commutative, associative and monotone [24]. Zero is the neutral element for the S-norm. Similarly, one is the neutral element for the T-Norm. Usual triangular operators are gathered in Table 2, with $a \in [0, 1]$, $b \in [0, 1]$.

Table 2. T-Norm and S-Norm.

| Name | T-Norm | S-Norm |
|---------------|---------------------------------------|---------------------------|
| Max/Min | $\min(a, b)$ | $\max(a, b)$ |
| Probabilistic | $a * b$ | $a + b - a * b$ |
| Lukasiewicz | $\max(0, (a + b - 1))$ | $\min(1, (a + b))$ |
| Einstein | $\frac{a * b}{1 + (1 - a) * (1 - b)}$ | $\frac{a + b}{1 + a * b}$ |

Let's consider an observation x belonging to class a with a degree $u_a(x)$, to class b with a degree $u_b(x)$ and to class c with a degree $u_c(x)$. Then, the membership degree of x to known classes (a , b or c) is given by:

$$u_{all}(x) = \perp (u_a(x), u_b(x), u_c(x)) \quad (5)$$

where \perp stands for an S-Norm [24].

Because of the operator properties (commutability and associativity), this operation is equivalent to:

$$u_{all}(x_{new}) = \perp [u_a(x), \perp (u_b(x), u_c(x))] \quad (6)$$

In the same way, suppose several membership degrees of x to a class d are computed: $u_{d1}(x)$, $u_{d2}(x)$ and $u_{d3}(x)$. The combination of all these degrees will be obtained by:

$$u_d(x) = \top(u_{d1}(x), u_{d2}(x), u_{d3}(x)) \quad (7)$$

with \top as a T-Norm.

Likewise, because of the operator properties, this can be written as:

$$u_d(x) = \top[u_{d1}(x), \top(u_{d2}(x), u_{d3}(x))] \quad (8)$$

Then, an information fusion operator Ψ , based on a triangular operator φ (T or S-Norm) is defined.

$$M_{m,c}(\mathbb{R}) \mapsto M_{m,1}(\mathbb{R})$$

$$\Psi : \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1c} \\ \vdots & \ddots & \cdots & \vdots \\ u_{m1} & \cdots & \cdots & u_{mc} \end{pmatrix} \mapsto \begin{pmatrix} \varphi(u_{11}, \varphi(u_{12}, \varphi(u_{13}, \dots))) \\ \cdots \\ \varphi(u_{m1}, \varphi(u_{m2}, \varphi(u_{m3}, \dots))) \end{pmatrix} \quad (9)$$

where $M_{m,c}(\mathbb{R})$ is a matrix of m rows and c columns with elements u_{ij} in \mathbb{R} ($i \in \{1, \dots, m\}$, $j \in \{1, \dots, c\}$).

The application of this fusion operator will be explained in the following paragraphs. Let's consider that a multi-class SVM model MDL is set and trained on a representative dataset. Each binary SVM model composing this multi-class model is noted MDL_k ($k \in \{1, \dots\}$).

3.2.2. One vs All

The algorithm, which computes U_{class} and U_{others} for a test set of m observations, is given in Table 3. It uses a "One vs All" model. The fusion operator Ψ used here is based on a T-Norm φ .

Table 3. Membership computations for "One vs All" SVM Model.

| Step | Actions during the step |
|------|--|
| 1 | <p>Initializations</p> <p>Initialize U_{class} and U_{all} as (m, c) matrix of zeros</p> <p>Initialize U_{others} as $(m, 1)$ vector of zeros</p> <p>Membership Computations</p> |
| 2 | <p>For k from 1 to c</p> <p>Compute u_1 and u_2 from MDL_k</p> <p>Replace the k^{th} column of U_{class} by u_2</p> <p>Replace the k^{th} column of U_{all} by u_1</p> <p>End For</p> <p>U_{others} Computation</p> |
| 3 | <p>$U_{others} = \Psi(U_{all})$</p> |

An intermediate matrix U_{all} is also used. This matrix represents the membership of the observations to the “All” side of the “One vs All”. Applying the information fusion to this matrix, one obtains the membership degrees of unknown classes U_{others} .

3.2.3. One vs One

The algorithm, which computes U_{class} and U_{others} with a “One vs One” model is given in Table 4. Two fusion operators are used: Ψ_1 based on a T-Norm φ_1 and Ψ_2 based on a S-Norm φ_2 . For a binary model MDL_k , we note k_+ the class considered as +1, respectively k_- for the class considered as -1. The matrix $U_{class,q}$ ($q \in \{1, \dots, c\}$) represents all the mbd of the observations to classes. The vector U_{rel} represents the relevance of the observations (membership degree of the observation to the group of known classes).

Table 4. Membership computations for “One vs One” SVM Model.

| Step | Actions during the step |
|------|---|
| 1 | <p>Initializations</p> <p>Initialize U_{class} as (m, c) matrix of zeros</p> <p>Initialize U_{others} and U_{rel} as $(m, 1)$ vector of zeros</p> |
| 2 | <p>Membership Computations</p> <p>For k form 1 to $\frac{c(c-1)}{2}$</p> <p>Compute u_1 and u_2 from MDL_k</p> <p>Save u_2 in $U_{class_{k_+}}$</p> <p>Save u_1 in $U_{class_{k_-}}$</p> <p>End For</p> |
| 3 | <p>Fusion for each class</p> <p>For l form 1 to c</p> <p>Replace the l^{th} column of U_{class} by $\Psi_1(U_{class_l})$</p> <p>End For</p> |
| 4 | <p>Fusion on all the class (relevance vector)</p> <p>$U_{rel} = \Psi_2(U_{class})$</p> |
| 5 | <p>U_{others} Computation</p> <p>$U_{others} = 1 - U_{rel}$</p> <p>(with respect to dimensions)</p> |

4. Automatic detection of new classes

From the matrix U_{class} and the vector U_{others} , a method to detect new classes is presented in this section. It is based on indicators of novelty which are attributed for a second-class classifier.

4.1. Indicators of novelty

4.1.1 Detection of a whole new class (new fault)

The hypothesis is made that a new class is going to present high U_{others} values. Therefore, an indicator of distance, for an observation x , can be computed by:

$$ID(x) = \frac{u_{others}(x)}{\max_{k \in \{1, \dots, c\}} \{K(g_1, x); \dots; K(g_k, x); \dots; K(g_c, x)\}} \quad (10)$$

with $u_{others}(x)$: the membership degree of x to unknown classes, $K(g_k, x)$: the kernel between the center of gravity g_k of class k and the observation. $u_{others}(x)$ is computed in the same way as presented in Table 3.

4.1.2. Detection of an ambiguous class (combined fault)

An ambiguous class can correspond to combined fault cases. The hypothesis is made that this type of new classes is going to present close membership degrees to two (or more) classes. Therefore, an indicator of ambiguity, for an observation x , can be computed by:

$$IA(x) = \frac{\max\{K(g_{k_1}, x); K(g_{k_2}, x)\}}{\Delta u_{classk_1k_2}(x)} \quad (11)$$

with k_1 : class for which $u(x)$ is maximum, k_2 : class for which $u(x)$ reaches its second maximum value, $u_{classk_1k_2}(x) = u_{classk_1}(x) - u_{classk_2}(x)$.

4.1.3. Indicators of references

It is supposed that the training base X_a has been correctly pre-processed and that it is cleaned, i.e., no outliers, no misclassified observations ... Then, the reference values for the indicators are defined by:

$$ID_{ref} = \max_{x \in X_a} \{ID(x)\} \quad (12)$$

$$IA_{ref} = \max_{x \in X_a} \{IA(x)\} \quad (13)$$

As presented in the next subsection, the above references are used in the rest of the paper to normalize corresponding indicators before making decision on rejection/acceptance of a new observation to a known class.

4.2. Decision map

A decision rulebase can be expressed by selecting four thresholds on the normalized indicators (ID/ID_{ref} and IA/IA_{ref}). This allows representing these rules by a map composed of nine areas

such as shown in Figure 5 where zones are limited by thresholds ID_{th1} , ID_{th2} , IA_{th1} and IA_{th2} .

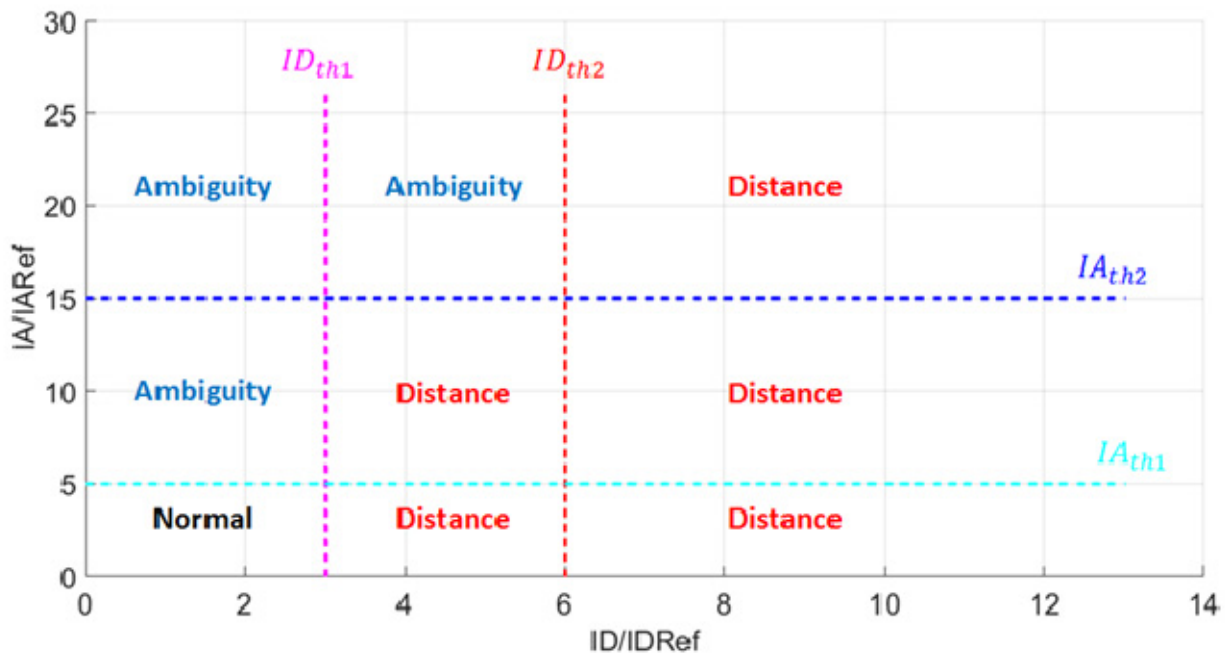


Figure 5. Representation of the rejection rulebase.

The Normal zone corresponds to observations belonging to known classes. For the points in this area, the algorithm classifies the observation in the class for which its mbd is the highest and gives this degree as a level of confidence. The Ambiguity zones correspond to combined fault modes. For these points, the algorithm associates the observation to the two classes with the highest mbd and derives a level of confidence from the two mbds. The Distance zones correspond to unknown fault modes. Points in these zones are given to experts for further analysis, possibly leading to new classes. The choice has been made, for safety issues, to make the distance prior to the ambiguity. Thus, there are five zones for distance and only three for ambiguity.

The different thresholds are defined empirically according to the application and the level of confidence.

5. Architecture of the classification algorithm

The global architecture of the proposed classification algorithm can now be introduced. The training phase is shown in Figure 6.

The parameterized and trained model is noted $MLFZ_S$. The kernel function is set here as for the classical SVM. Then, the type of membership functions is chosen regarding the classifier in charge of novelty detection. Therefore, the SVM kernel is first designed (Choice of the kernel, of the Lagrange parameter C). Then, the fuzzy part is defined (Membership function, Method of fusion, ID and IA thresholds).

The test procedure on a base X_t is given in Figure 7.

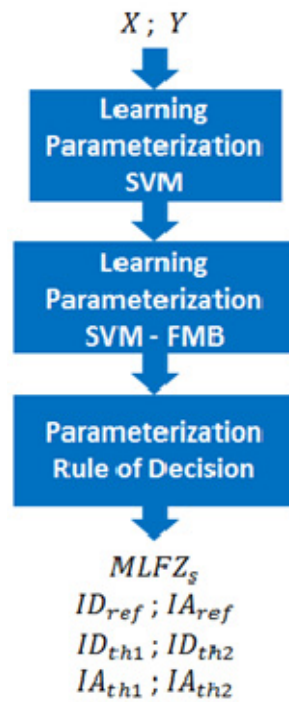


Figure 6. Flowchart for the training and parameterization procedure.

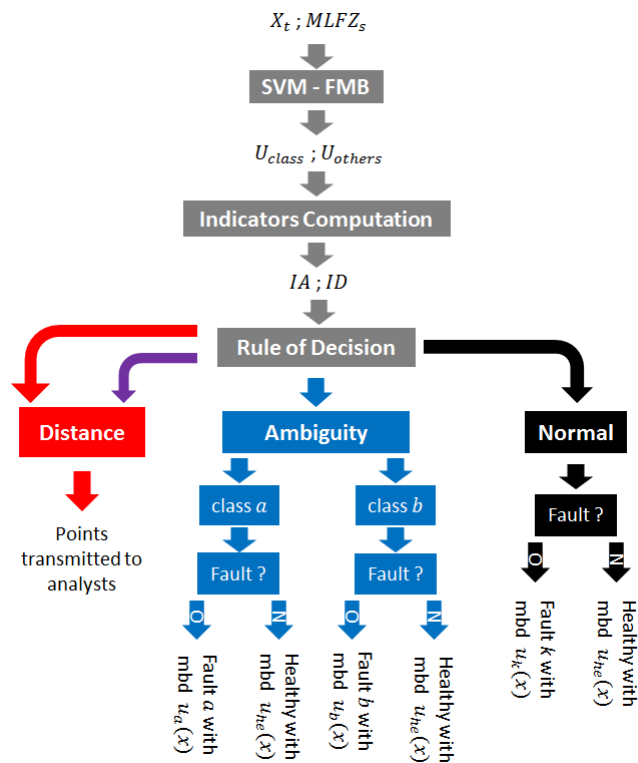


Figure 7. Flowchart for the test procedure.

With new samples, we first compute the membership functions and ID and IA indicators. According to the results, we define the health state and the membership values or send the rejected point to an expert.

6. Experimental tests

6.1. Test campaign

Faults were physically realized on a 5.5 kW induction motor powered by an inverter and loaded by a powder brake. The database comes from a previous work [26]. Figure 8 shows the experimental test bench.

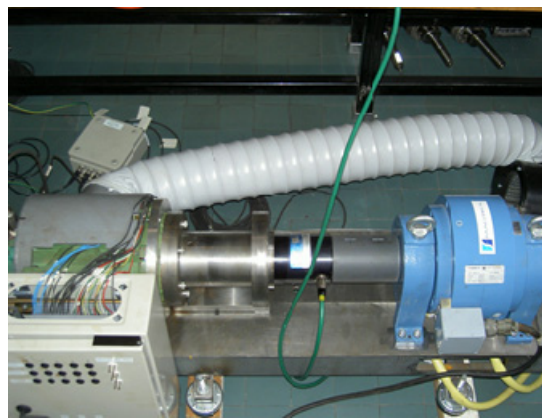


Figure 8. Experimental test bench.

The induction machine is on the left and the brake on the right. Acquisition has been realized with an Odyssey recorder from Nicolet. This bench was equipped with temperature, vibration, current and voltage sensors. The motor parameters are given in Table 5.

Table 5. Parameters of the induction motor.

| Parameter (Unit) | Value |
|------------------------------------|-------|
| Nominal Voltage between Phases (V) | 400 |
| Power Frequency (Hz) | 50 |
| Nominal Speed (r/min) | 1440 |
| Nominal Useful Power (kW) | 5.5 |
| Power Factor | 0.84 |
| Pair of Poles | 4 |
| Number of rotor slots | 28 |
| Number of stator slots | 48 |

Three faults were considered from a previous work performed by the authors [27]. The first one is related to the roller bearings [27,28]. It was realized by removing the healthy bearings of the

machine and replacing them by pre-aged bearings. The second one is the rotor broken bars [27,29]. It was created by drilling holes in the conductors of the squirrel cage as illustrated in Figure 9. The last fault concerns the electrolytic capacitor bank of the inverter [30]. It was emulated by replacing a healthy bank by a pre-aged one. The healthy set of inverter-machine and the three cases of fault were tested at nominal speed for no load, half load and full load. The motor's speed, currents and voltages were acquired at a sampling frequency of 20 kHz.



Figure 9. Emulation of the broken bar fault.

6.2. Features of faults

Features of faults are computed from the measurements. These features are based on specific harmonics [31,32], harmonics of the fundamental, active and reactive powers [33] and Park's current [34]. They are all given in [27]. Please note that all the signals are normalized by the technique of the first harmonic: each signal is divided by the amplitude of its fundamental. This allows gathering in one class observations of the same fault, but with different load levels.

6.3. Databases

From the computed features, two databases are created. They are detailed in Table 6 and Table 7.

Table 6. First database (Fault Detection).

| N° | Class | Number of Observations |
|--------|---|------------------------|
| 1 (D) | Healthy motor and inverter | 30 |
| 2 (D) | Damaged bearings and healthy inverter | 30 |
| 3 (D) | Broken bars (1,3,4 bars) and healthy inverter | 30 (10 for each) |
| 4 (ND) | Healthy motor and damaged capacitor | 30 |

For the second database, the inverter is always healthy. During the validation process, some classes are used for both training and test and are called as “*Known*” or “*Defined*” (D). For the test phase, an extra class, called “*Unknown*” or “*Not Defined*” (ND), is added. For the first database, the ND-class should be located in the distance zones of the decision map in Figure 5. For the second database, it should be detected in ambiguity zones.

Table 7. Second database (Fault Severity).

| N° | Class | Number of Observations |
|--------|-------------------|------------------------|
| 1 (D) | Healthy motor | 30 |
| 2 (D) | One broken bar | 30 |
| 3 (ND) | Three broken bars | 30 |
| 4 (D) | Four broken bars | 30 |

For each database, all the observations of the parameters are normalized between 0 and 1.

7. Feature vector

7.1. Feature selection

In both databases, 48 features are computed [27]. All are based on electrical, mechanical or thermal measurements. Among all the features, some are particularly sensitive to faults. To identify them, a “Sequential Backward Selection” (SBS) is performed on the observations in D-classes [35]. This method allows selecting a set of parameters d' optimizing a given criterion. The criterion used here is the Fisher criterion given by:

$$J = \sum_{k=1}^c \sum_{i=1}^{c-1} \frac{g_k - g_i}{n_k \sigma_k^2 - n_i \sigma_i^2} \quad (14)$$

with c : number of classes, g_k : center of gravity of class k , n_k : number of observations in class k and σ_k : variance of observations in class k given by:

$$\sigma_k = \frac{\sum_{j=1}^{n_k} (x_{kj} - g_k)^2}{n_k} \quad (15)$$

where x_{kj} is the j^{th} observation belonging to class k .

The numerator of the criterion deals with the objective of maximizing the separation between classes. The denominator is related to the objective of maximizing the compactness of each class. The application of the SBS to the first database is illustrated in Figure 10. The x -axis represents the iteration of the method and the y -axis the features. For each iteration, the least influent parameter, regarding to the criterion, is eliminated. The algorithm iterates from forty-seven parameters to one.

The variation of the criterion with the number of parameters is given in the following Figures (Figures 11 and 12).

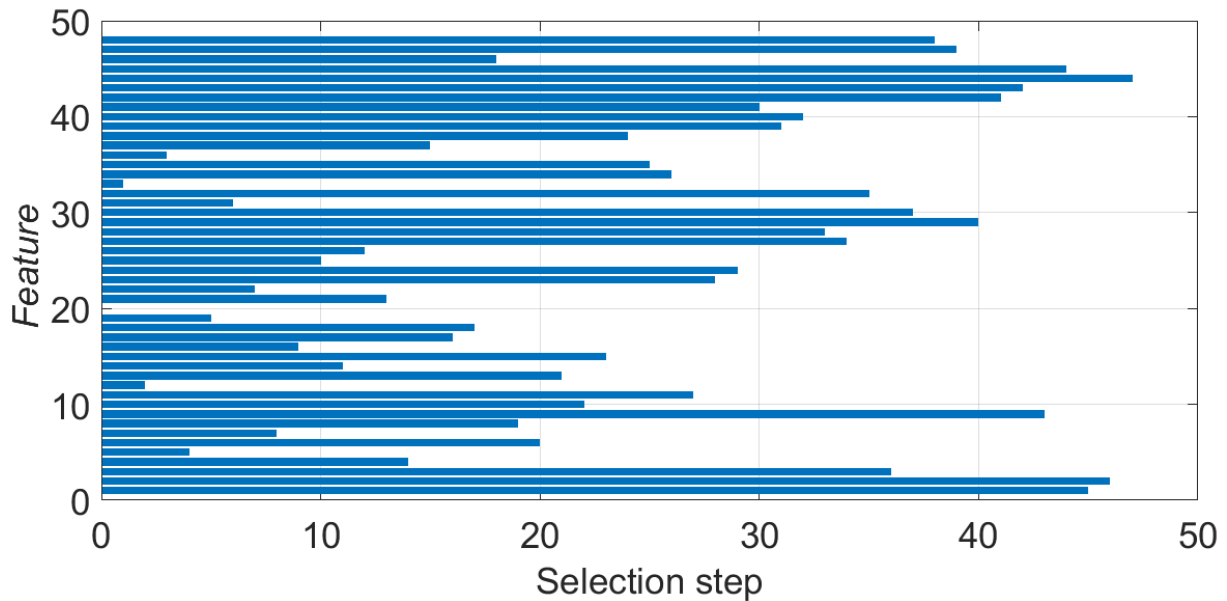


Figure 10. SBS on the first database.

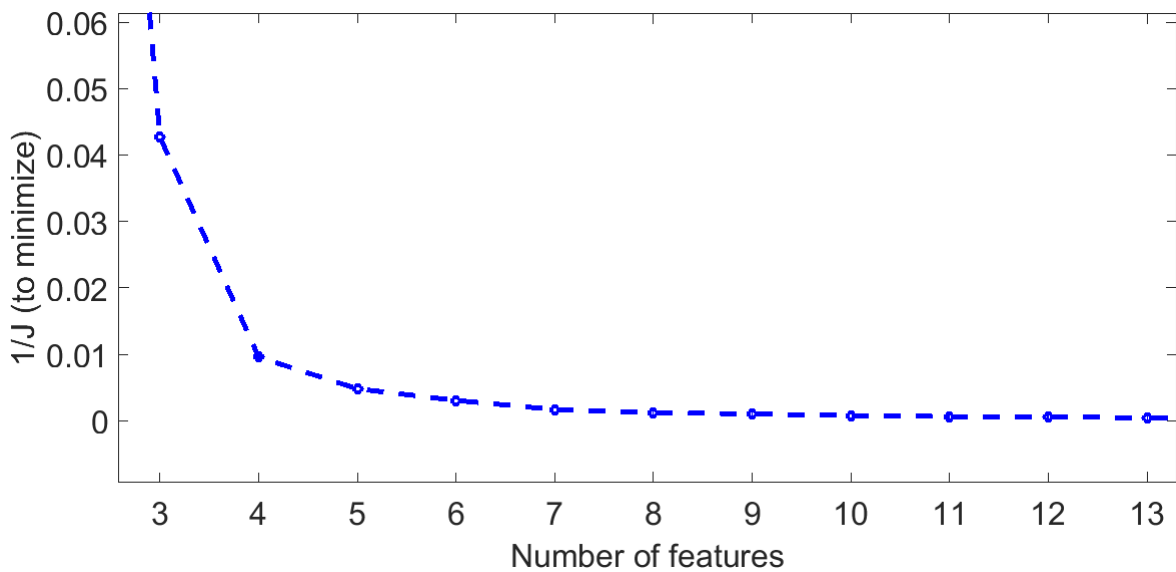


Figure 11. Cost regarding the number of selected features – Database 1 (zoom).

The optimal number of parameters depends on the significant slope change on the curve. Here, this number is between 4 and 13 features. During the setting of the SVM parameters, all the combination of features between 4 and 13 were tested. The best results were obtained for a number of parameters $d_{opt1} = 11$. The same study was performed for the second database and resulted in $d_{opt2} = 4$.

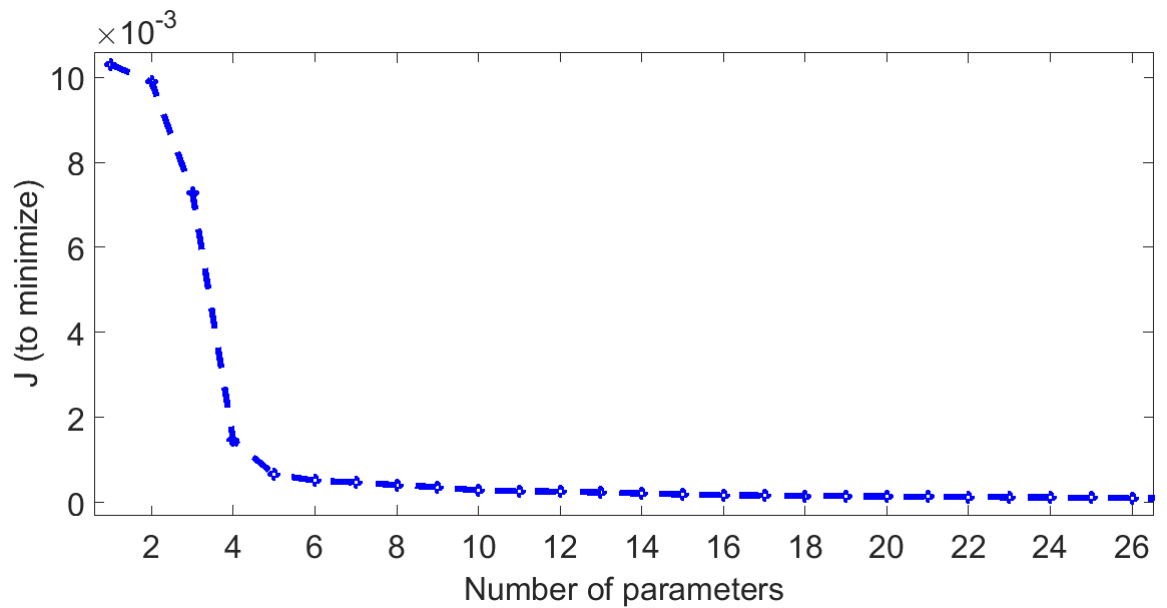


Figure 12. Cost regarding the number of selected features – Database 2 (zoom).

7.2. Details of features

The details of the optimal features are given in the following Tables. They are based on the type of physical values. Features given in Table 8 correspond to power related features.

Table 8. Features based on powers.

| Numeration | Denomination (Unit) |
|------------|-----------------------------|
| P_1 | Average Active Power (W) |
| P_2 | Average Reactive Power (VA) |
| P_3 | Loss Power (W) |
| P_4 | Heating of the armature (K) |

Features in Table 9 are related to specific current harmonics. We noted f_s the frequency of the stator currents (Hz) and f_r the frequency of the rotor (Hz).

Table 9. Features based on current harmonics.

| Numeration | Denomination (Unit) |
|------------|-------------------------------------|
| H_1 | Amplitude of $f_s + f_r$ signal (A) |
| H_2 | Amplitude of $5f_s$ (A) |
| H_3 | Amplitude of $7f_s$ (A) |

In Table 10, features correspond to characteristics of the currents in the stator reference frame. We note $I_{s\alpha}$ the stator current component in the α axis and $I_{s\beta}$ the stator current component in the

β axis. The complex stator current I_s is computed by:

$$I_s = I_{s\alpha} + j * I_{s\beta} \quad (16)$$

Table 10. Features based on the stator current components.

| Numeration | Denomination (Unit) |
|------------|--|
| S_1 | Deformation $I_{s\beta}(I_{s\alpha})$ in α axis |
| S_2 | Standard deviation of $I_{s\alpha}$ |
| S_3 | Standard deviation of $I_{s\beta}$ |
| S_4 | Standard deviation of $\ I_s\ $ |

Finally, the direct impedance Z_d is also a sensitive feature.

7.3. Feature vector

Table 11 summarizes selected features for each database.

Table 11. Selected features for the two databases.

| Database | Selected Features |
|----------|---|
| 1 | $P_1 ; P_2 ; P_3 ; P_4 ; H_1 ; H_2 ; H_3 ; S_2 ; S_3 ; S_4 ; Z_d$ |
| 2 | $P_1 ; P_2 ; S_1 ; S_3$ |

8. Experimental results: validation of the diagnosis process with the induction machine data base

8.1. Validation procedure

To validate the robustness of the proposed algorithm, a 5-fold out cross validation (CV) is used [36,37]. These types of method lead to a pessimistic estimation of the generalization error [37–39]. Then, for both databases, the training phase uses 4/5 of the observations of the D-classes. The test phase uses 1/5 of the observations of the D-classes and the entire ND-class. The precision of the classification is computed by:

$$P = 1 - \frac{\sum_{j=1}^n I(y_j, \hat{y}_j)}{n} \quad (17)$$

where n is the number of observations; y_j is the class of observation j ; \hat{y}_j is the predicted class of observation j (class which maximizes $u(x)$); $I(a, b) = 1$ if $a \neq b$ and 0 otherwise.

Please note that the average precision on all the CV iterations will be given throughout the

paper. For each validation test, 90 points belong to D-classes and 30 belong to the ND-class.

The validation procedure was performed on a laptop with an Intel i5 CPU, 4 cores, computing at 2.7 GHz with 8 GB RAM.

8.2. Setting the parameters

The Radial Basis Function (RBF) kernel and the “One vs All” method are used for both databases. In order to set the parameters, the model is trained on the training database and then tested on the test database. Values of the parameters are modified by trial and error, until suitable performances are reached in terms of rate of well classified points. So, these parameters are optimized in the framework of the studied application (Table 12).

Table 12. One vs All parameters.

| Parameter | Value |
|---------------------|-----------|
| C | 100 |
| Type of Kernel | RBF |
| Parameter of Kernel | 0.3 |
| Type of FA | Sigmoidal |
| Parameter of FA | 0.8 |
| Type of Norm | Max / Min |

8.3. Diagnosis of an induction motor

The obtained results by the proposed algorithm applied to the first database are given in Table 13.

Table 13. Results for rejection (Database 1).

| Type of observations | Number of well Classified/Rejected observations | Total number of observations | Rate of well classified (%) |
|----------------------|---|------------------------------|-----------------------------|
| D – classes | 87 | 90 | 96.7 |
| ND – class | 30 | 30 | 100 |

With the rejection rule and the threshold defined in Figure 5, three points of D-classes have been rejected (3 points over 90). This makes a precision of 96.7% on the D-classes (87 points on 90 well classified). Furthermore, all the unknown observations are classed in the ND-class and more specifically, in the distance zones. So, they are all rejected (see Figures 13 and 14). Therefore, the overall precision is 97.5%: 87/90 D-classes and 30/30 ND-class giving 117/120 points. This result is quite satisfactory considering that one class is unknown. Without the rejection rule, the precision is 75% (90/120 points), because in this case, all 90 known points are well detected and put in the D-classes while 30 unknown points are wrongly classed in the D-classes. It should be noted that when you look closer, the three rejected points with the proposed algorithm are very noisy.

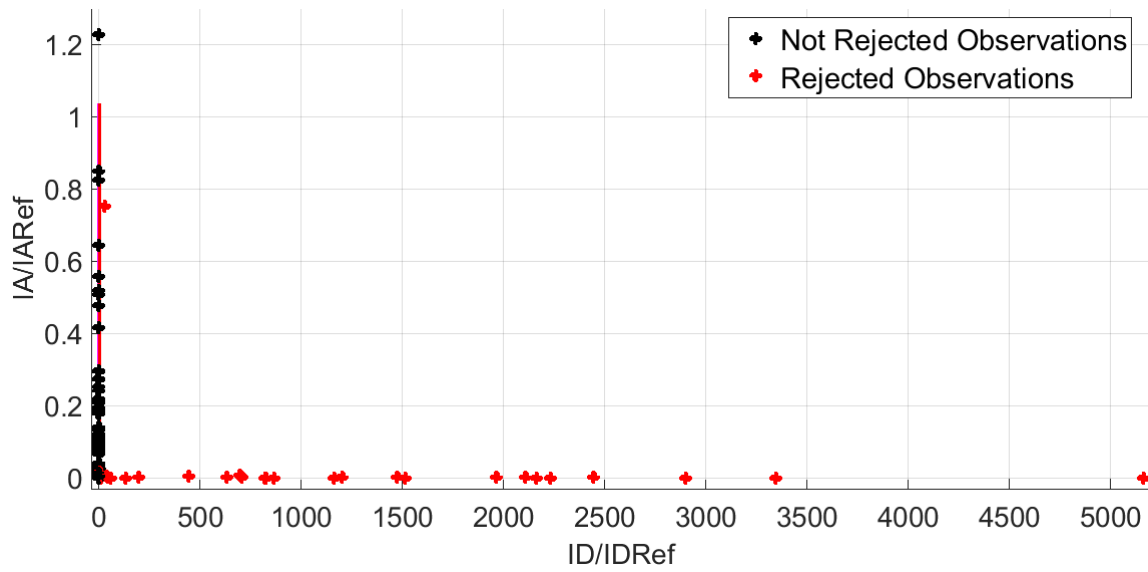


Figure 13. Decision map for the first database.

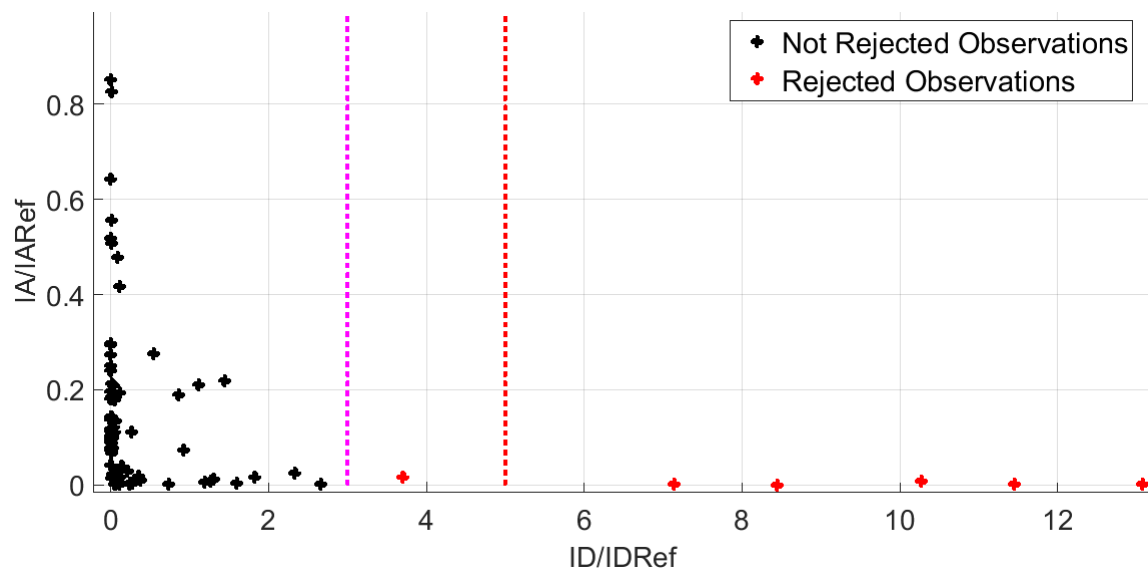


Figure 14. Decision map for the first database (zoom).

Finally, the computation costs for training and test phases are given in the following table (Table 14). The presented characteristics (minimum, average, maximum) are obtained from the computation times on all CV iterations.

Table 14. Computation costs (Database 1).

| Step | Min | Average | Max |
|--------------|------|---------|------|
| Training (s) | 1.67 | 1.93 | 2.27 |
| Test (ms) | 6.58 | 8.18 | 12.2 |

Please note that the test phase computation costs are in milliseconds. These results have to be linked to the number of observations trained, which is 96, and those tested, which is 24, at each iteration. This low computation times could make the algorithm suitable for transportation systems. These results must be confirmed on more complex problems with higher number of classes and features.

8.4. Detection of combined faults

The application of the proposed algorithm to the second database gives the results shown in Table 15.

Table 15. Results for classification and rejection (Database 2).

| Type of observations | Number of well Classified/Rejected observations | Total number of observations | Rate (%) |
|----------------------|---|------------------------------|----------|
| D – classes | 85 | 90 | 85.44 |
| ND – class | 26 | 30 | 86.67 |

The “Rate” in this table is defined by the ratio of well classified (without taking into account the rejected samples) over the total number of observations.

Like the previous case, when we do not use the rejection rule all of the known observations are correctly classified and all of the unknown points are misclassified. With the rejection rule and the threshold defined in Figure 5, five known points are rejected and the others are well classified in their corresponding D-classes. This provides a precision of 94.4% (85 well classified points over 90). Then, 26 unknown observations over 30 are detected as ND-class in the ambiguity zones (see Figures 15 and 16). Therefore, the precision is around 92.5 % for the whole process (D-classes and ND-class: 111 well classified points over 120). This precision should be compared to the case that no rejection rule exists. Indeed, in the latter case, the precision is 75% (90/120 points).

It can be seen in Figures 14 and 15 that suitable thresholds are difficult to determine. One explanation is that the ND-class, which is used here, is not really a combination of the two other faulty classes; it is not a case of combined faults. Another is related to the fact that threshold classifiers are very basic. A more evolved classifier could improve the accuracy.

For the second database, the computation times are given in Table 16. They are obtained from the computation times on all CV iterations.

Table 16. Computation costs (Database 2).

| Step | Min | Average | Max |
|--------------|------|---------|------|
| Training (s) | 1.76 | 1.97 | 2.51 |
| Test (ms) | 6.06 | 9.29 | 16.6 |

Once again, it should be noticed that these results depend to the number of observations trained and tested. These numbers are the same as those used in the first database. They confirm the suitability of the method for transportation applications.

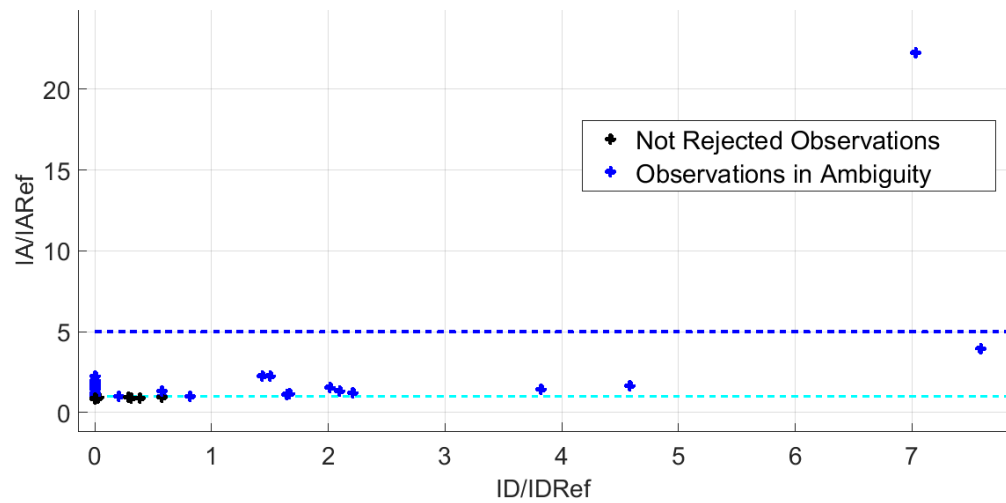


Figure 15. Decision map for the second database.

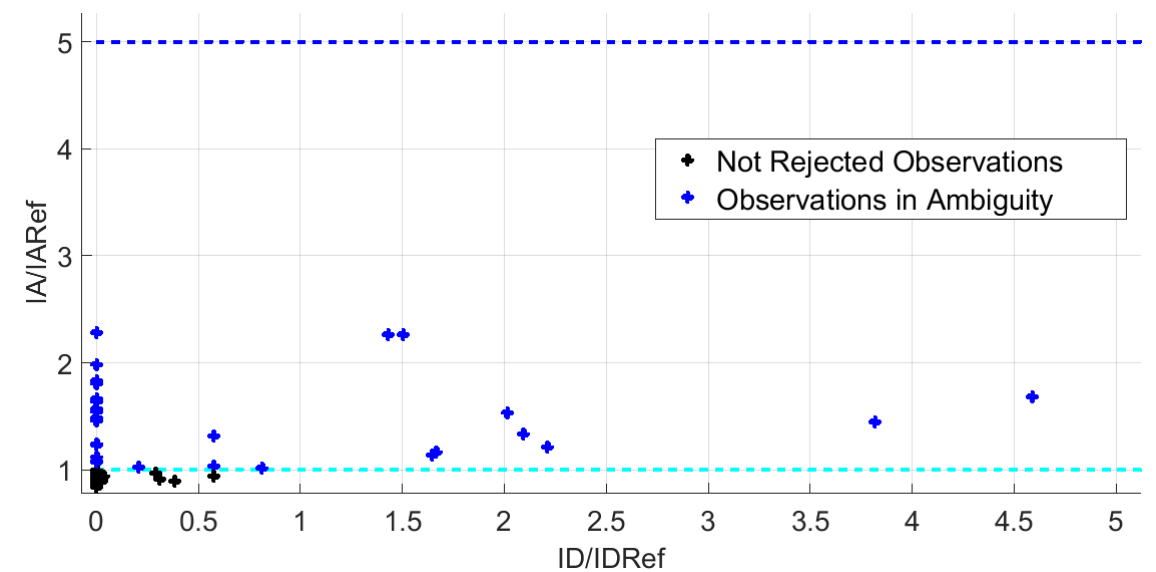


Figure 16. Decision map for the second database (zoom).

9. Comparison with other classifiers

We compared the proposed classifier with some classical ones. The comparison was held on the database obtained on the above multi-fault induction machine. Techniques like Fuzzy Support Vector Machine (SVM-FA), Decision Tree (ADD), K Nearest Neighbours (KNN) and Naive Bayesian Network (BN) were compared. The comparison results are given in Tables 17 to 19 in terms of the rate of well-classified samples, the amount of memory that is necessary to realize the algorithm and the computing time to classify the whole test database.

Table 17. Precision.

| Precision P (%) | SVM – FA | ADD | KNN | BN |
|-------------------|----------|-------|-------|-------|
| Maximum | 100 | 75 | 75 | 75 |
| Mean | 97.5 | 73.34 | 71.67 | 74.17 |
| Minimum | 91.67 | 70.84 | 70.84 | 70.84 |

Table 18. Memory.

| Memory (kB) | SVM – FA | ADD | KNN | BN |
|-------------|----------|------|------|------|
| Amount (kB) | 14.5 | 28.2 | 18.5 | 17.6 |

Table 19. Computation time.

| Test time (ms) | SVM – FA | ADD | KNN | BN |
|----------------|----------|------|------|------|
| Maximum | 102 | 6.67 | 14.5 | 83.9 |
| Mean | 47.4 | 2.84 | 5.04 | 38.2 |
| Minimum | 24.3 | 0.40 | 1.30 | 17.8 |

It can be noticed that SVM-FA presents the best performance in terms of precision and memory while the highest computation time remains still fully acceptable for electrified transportation systems applications.

10. Conclusion

A new algorithm for classification of faults with detection of unknown classes is proposed in this paper. This algorithm is based on support vector machine, fuzzy membership functions and information fusion through triangular norms. It produces distance and ambiguity indicators. These indicators allow detecting observations belonging to unknown classes. These classes can correspond to all new fault modes or to combined fault modes.

The algorithm was validated on an experimental database via cross validation method. The database contains common faults for an induction motor fed by a voltage-source inverter. The results show the good robustness and the short computation time of the algorithm compared to constraints in aeronautics. Therefore, the proposed algorithm seems suitable for transportation systems like more electric aircrafts.

Nevertheless, more validation tests should be performed, especially for the ambiguity indicator. Database with combined fault modes, for example bearings and capacitor faults, should be tested. More work is planned regarding the indicators and the rejection classifier especially for discovering the novelty. Another future work is on the completion of incomplete databases to get them be closer to aeronautic applications. And finally, this work will be the basis of a study on faults prognosis.

Acknowledgments

We would like to thank SAFRAN Group for funding this research.

Conflict of interest

The authors declare no conflict of interests in this paper.

References

1. Polo G (2011) On maritime transport costs, evolution, and forecast. *Ship Science and Technology*, 5: 19–31.
2. Operations and Maintenance (O&M) Costs Technical Memorandum, 2015. *PARSONS BRINCKERHOFF* –AECOM. Available from: https://www.fra.dot.gov/necfuture/pdfs/feis/volume_2/appendix/app_b09.pdf.
3. Markou C, Cros G, Sng A (2015) Airline Maintenance Cost Executive Commentary. IATA. Available from: <https://www.iata.org/whatwedo/workgroups/Documents/MCTF/AMC-Exec-Comment-FY14.pdf>.
4. Thomson R, Edwards M, Britton E, et al. (2014) Predictive Maintenance : Is the timing right for predictive maintenance in the manufacturing sector ? *Roland Berger Strategy Consultants*.
5. Fumera G, Roli F (2002) Support vector machines with embedded reject option, In: *Pattern Recognition with Support Vector Machines*. Springer, 68–82.
6. Grandvalet Y, Rakotomamonjy A, Keshet J, et al. (2009) Support vector machines with a reject option, In: *Advances in Neural Information Processing Systems*. 537–544.
7. Wegkamp M, Yuan M (2011) Support vector machines with a reject option. *Bernoulli* 17: 1368–1385. Available from: <https://projecteuclid.org/euclid.bj/1320417508>.
8. Abe S, Inoue T (2002) Fuzzy Support Vector Machines for Multiclass Problems. *European Symposium on Artificial Neural Networks*, Bruges (Belgium).
9. Lin CF, Wang SD (2002) Fuzzy support vector machines. *IEEE Transaction on Neural Network* 13: 464–471.
10. Ma H, Xiong Y, Fang H, et al. (2015) Fault diagnosis of bearing based on fuzzy support vector machine, In: *2015 Prognostics and System Health Management Conference (PHM)*, 1–4.
11. Ebrahimi H, Gatabi JR, El-Kishky H (2015) An auxiliary power unit for advanced aircraft electric power systems. *Electr Pow Syst Res* 119: 393–406.
12. Guan Y, Zhu ZQ, Afinowi IAA, et al. (2016) Difference in maximum torque-speed characteristics of induction machine between motor and generator operation modes for electric vehicle application. *Electr Pow Syst Res* 136: 406–414.
13. Wu Y, Jiang B, Lu N, et al. (2016) Multiple incipient sensor faults diagnosis with application to high-speed railway traction devices. *ISA T* 67:183–192.
14. Baraldi P, Cannarile F, Maio FD, et al. (2016) Hierarchical k-nearest neighbours classification and binary differential evolution for fault diagnostics of automotive bearings operating under variable conditions. *Eng Appl Artif Intel* 56: 1–13.

15. Bessam B, Menacer A, Boumehraz M, et al. (2016) Detection of broken rotor bar faults in induction motor at low load using neural network. *ISA T* 64: 241–246.
16. Ferracuti F, Giantomassi A, Iarlori S, et al. (2015) Electric motor defects diagnosis based on kernel density estimation and Kullback–Leibler divergence in quality control scenario. *Eng Appl Artif Intel* 44: 25–32.
17. Glowacz A, Glowacz Z (2017) Diagnosis of the three-phase induction motor using thermal imaging. *Infrared Phys Techn* 81: 7–16.
18. Vapnik VN (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.
19. Han G, Meng-Kun L (2018) Induction motor faults diagnosis using support vector machine to the motor current signature. *IEEE Industrial Cyber-Physical Systems (ICPS)*, 417–421.
20. Camarena-Martinez D, Valtierra-Rodriguez M, Amezcua-Sanchez JP, et al. (2016) Shannon Entropy and K-Means Method for Automatic Diagnosis of Broken Rotor Bars in Induction Motors Using Vibration Signals. *Shock Vib* Article ID: 4860309.
21. Burges CJC (1998) A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min Knowl Disc* 2: 121–167.
22. Bottou L, Lin CJ (2007) Support vector machine solvers. *Large scale kernel machines* 3: 301–320.
23. Zadeh LA (1965) Fuzzy sets. *Information and Control* 8: 338–353.
24. Gassert H (2004) Operators on Fuzzy Sets: Zadeh and Einstein. Seminar Paper, Department of Computer Science Information Systems Group, University of Fribourg.
25. Mizumoto M, Tanaka K (1981) Fuzzy sets and their operations. *Information and Control* 48: 30–48.
26. Ondel O, Clerc G, Boutleux E, et al. (2009) Fault Detection and Diagnosis in a Set “Inverter–Induction Machine” Through Multidimensional Membership Function and Pattern Recognition. *IEEE Transaction on Energy Conversion* 24: 431–441.
27. Ondel O (2006) Diagnosis by Pattern Recognition: application on a set inverter – induction machine. Ecole Centrale de Lyon.
28. Bolander N, Qiu H, Eklund N, et al. (2009) Physics-based Remaining Useful Life Prediction for Aircraft Engine Bearing Prognosis. *Annual Conference of the Prognostics and Health Management Society*.
29. Deleroi W (1982) Squirrel cage motor with broken bar in the rotor—Physical phenomena and their experimental assessment. *Proc of Int Conf on Electrical Machines (ICEM)*, Budapest, Hungary, 767–771.
30. Venet P, Lahyani A, Grellet G, et al. (1999) Influence of aging on electrolytic capacitors function in static converters: Fault prediction method. *Eur Phys J-Appl Phys* 5: 71–83.
31. Schoen RR, Habetler TG, Kamran F, et al (1994) Motor bearing damage detection using stator current monitoring, In: *Proceedings of 1994 IEEE Industry Applications Society Annual Meeting* 1: 110–116.
32. Schoen RR, Habetler TG (1993) Effects of time-varying loads on rotor fault detection in induction machines, In: *Conference Record of the 1993 IEEE Industry Applications Conference Twenty-Eighth IAS Annual Meeting* 1: 324–330.

33. Vas P (1992) *Electrical machines and drives: a space-vector theory approach*, Monographs in electrical and electronic engineering. Clarendon Press.
34. Casimir R, Boutleux E, Clerc G, et al. (2003) Broken bars detection in an induction motor by pattern recognition, In: *IEEE Bologna Power Tech Conference Proceedings 2*: 313–319.
35. Kudo M, Sklansky J (2000) Comparison of algorithms that select features for pattern classifiers. *Pattern Recogn* 33: 25–41.
36. Larsen J, Goutte C (1999) On optimal data split for generalization estimation and model selection, In: *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop*, 225–234.
37. Zalila Z, Cuquemelle J, Penet C, et al. (2006) Is your accurate model actually robust? Regulation and validation methods by xtractis. *Sensometrics 2006–Imagine the Senses*, Norway.
38. Plutowski ME (1996) Survey : Cross-validation in theory and practise (Research Report). *David Sarnoff Research Center - Princeton*.
39. Shao J (1993) Linear model selection by cross-validation. *J Am Stat Assoc* 88: 486–494.



AIMS Press

© 2018 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)