DOI: 10.18605/2175-7275/cereus.v11n1p195-209

<<Recebido em 11/01/2019. Aceito em 25/02/2019>>

ARTIGO ORIGINAL

REVISTA CEREUS ISSN: 2175-7275

Optimal Kernel Parameter Setting for Faults Detection with Stochastic Methods and Data Preprocessing

José Manuel Bernal de Lázaro[1], Adrián Rodríguez Ramos[2], Carlos Cruz Corona[3], Antônio José da Silva Neto[4], Orestes Llanes-Santiago[5]

ABSTRACT

In this paper, an indirect optimization criterion for parameter setting the kernel-based fault detection process is applied. The procedure analyzed involves the data preprocessing through the Kernel Independent Component Analysis (KICA) method, and the fault detection by using a classifier based on the Kernel Fuzzy C-means (KFCM) algorithm to reduce the classification errors. The main objective of the paper is the adjustment of the kernel parameters to obtain the best possible performance in the fault detection. To achieve this, two different metaheuristic algorithms are used: Differential Evolution and Particle Swarm Optimization. The proposed approach was evaluated by using the Tennessee Eastman (TE) process benchmark.

Keywords: Fault diagnosis, computational intelligence, kernel tools, optimal parameters.

[1] Automation Engineer. Universidad Tecnológica de La Habana José Antonio Echeverría, CUJAE, La Habana, Cuba

E-mail:

jbernal@automatica.cujae.edu.cu

[2] Automation Engineer. Universidad Tecnológica de La Habana José Antonio Echeverría, CUJAE, La Habana, Cuba.

[3] Electronics Engineer. Universidad de Granada, Granada, Espanha.

[4] Mechanical Engineer. IPRJ-UERJ, Nova Friburgo, RJ, Brasil.

[5] Electrical engineer. Universidad Tecnológica de La Habana José Antonio Echeverría, CUJAE, La Habana, Cuba.

1. INTRODUCTION

During the last two decades, the kernel methods have been established as a valuable alternative tool for numerous areas of research (Motai, 2015). In fact, they have played a significant role in reducing dimensionality, removing noise and, extracting features from the databases, including the historical data sets obtained from the complex industrial processes (Bernal de Lázaro et. al., 2016). Many current publications incorporate kernel methods in the fault diagnosis tasks because they allow the mapping of input data into a feature space where it is possible the use of linear algorithms, by avoiding the non-linearity in the original data. However, the aforementioned operation and the structure underlying the data are totally determined by the kernel function selected. This means that the inappropriate parameter setting for these kernel methods may result in non-satisfactory diagnosis results. Both, the choosing of an appropriate kernel, and the proper setting of its parameters, are open problems in the current fault diagnosis applications.

The present paper addresses the use of kernel methods to detect incipient faults with small effects on the monitored systems, which can be hidden by the disturbances in the systems. Besides, it is investigated how the indirect kernel optimization criteria can improve the performance of the kernel classification algorithms used in this stage.

The procedures evaluated in this analysis involve the Kernel Independent Component Analysis (KICA) algorithm for data preprocessing, and the Kernel Fuzzy C-means (KFCM) algorithm as the classifier. In addition, the stochastic Differential Evolution (DE) and Particle Swarm Optimization (PSO) algorithms are used to adjust the kernel parameters in the KFCM algorithm. The study is evaluated using the Tennessee Eastman (TE) process benchmark.

The paper is organized as follows. In Section 2 the techniques used in the fault detection tasks, and the optimization algorithms employed for the adjustment of the KFCM classifier, are described. In Section 3, the study case is presented, and the performance of the proposed fault detection scheme is discussed. Finally, based on the analysis of the results, some conclusions, and future potential research lines are presented.

2. GENERAL CHARACTERISTICS OF THE PREPROCESSING, CLASSIFICATION AND OPTIMIZATION TOOLS

In this section, the general characteristics of the tools used in the preprocessing and classification of the data collected by the fault diagnosis system, and the optimization algorithms used to obtain the parameters of the classification tool are presented.

2.1 Preprocessing by Using Kernel Independent Component Analysis (KICA)

Several papers have shown that when a fault diagnosis system incorporates a stage of data preprocessing, the results in the classification process are improved (Bernal de Lázaro et. al., 2015, 2016).

Kernel ICA (KICA) is an advanced version of the Independent Component Analysis (ICA) algorithm. The aforementioned technique is mainly used for non-gaussian processes in order to transform multivariate data into statistically independent components (Manabu, 2003). The basic idea of kernel ICA is to perform a non-linear mapping of the data into a hyper-dimensional feature space $\mathcal{H}$, and then to extract the useful information by using the ICA algorithm (Jiang et. al., 2013).

2.s Kernel Fuzzy C-means (KFCM)

The kernel clustering methods have been widely used in several fields (Ding & Fu, 2016). Nevertheless, their applications are still an innovative topic in the field of fault diagnosis (Cao et. al., 2012). Specifically, KFCM is a kernelized version of the Fuzzy C-means algorithm with a high potential for fault detection tasks. The KFCM algorithm can be formalized as follows, by first defining:

$$J_{\mathrm{KFCM}} = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m \left\| \Phi(\mathbf{x_k}) - \Phi(\mathbf{v_i}) \right\|^2 \quad (1)$$

where $\|\varPhi(X_k) - \varPhi(V_i)\|^2$ is the square of the distance between the mapping data $\varPhi(X_k)$ and $\varPhi(V_i)$, V_i is the center of each cluster, *c* is the total number of clusters, µ indicates the member-ship of each data point to each cluster, and m is a control parameter, that may be adjusted. The distance in the feature space is calculated through the kernel in the input space as follows:

$$\left\| \Phi(\mathbf{x_k}) - \Phi(\mathbf{v_i}) \right\|^2 = \mathbf{K}(\mathbf{x_k}, \mathbf{x_k}) - \mathbf{2K}(\mathbf{x_k}, \mathbf{v_i}) + \mathbf{K}(\mathbf{v_i}, \mathbf{v_i}) \quad (2)$$

Using a Radial Basis Function (RBF) as the kernel function, results

$$\left\|\Phi(\mathbf{x_k}) - \Phi(\mathbf{v_i})\right\|^2 = \mathbf{2}\left(\mathbf{1} - \mathbf{K}(\mathbf{x_k}, \mathbf{v_i})\right) \tag{3}$$

where

$$\mathbf{K}(\mathbf{x_k}, \mathbf{v_i}) = e^{-\|\mathbf{x_k} - \mathbf{v_i}\|^2/\sigma^2} \tag{4}$$

As a result, Eq. (1) may be rewritten as:

$$J_{\mathrm{KFCM}} = 2\sum_{i=1}^{c}\sum_{k=1}^{N} (\mu_{ik})^m \left\|1 - \mathbf{K}(\mathbf{x_k}, \mathbf{v_i})\right\|^2 \tag{5}$$

Then, minimizing the above expression under the conditions for local extreme allows to determine the center of each cluster and the pertinence of each data point to each cluster as follows:

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^m \mathbf{K}(\mathbf{x_k}, \mathbf{v_i})\mathbf{x_k}\right)}{\sum_{k=1}^{N} \mu_{ik}^m \mathbf{K}(\mathbf{x_k}, \mathbf{v_i})} \tag{6}$$

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{1-\mathbf{K}(\mathbf{x_k},\mathbf{v_i})}{1-\mathbf{K}(\mathbf{x_k},\mathbf{v_j})}\right)^{1/(m-1)}} \tag{7}$$

2.3 Optimization Algorithms and Kernel Functions

In many scientific areas, and in particular in the fault diagnosis field, metaheuristic algorithms have been widely used, with excellent results in the solution of optimization problems (Camps Echevarría et. al., 2010). They can locate efficiently the neighborhood of the global optimum in most of the occasions, with an acceptable computational time.

In this paper, the Differential Evolution (DE) and the Particle Swarm Optimization (PSO) algorithm are employed to adjust the parameter for KFCM-based classifier, with the goal to obtain the best results in the classification task.

Differential Evolution Algorithm. Differential Evolution (DE) is one of the most popular optimization algorithms due to its good convergence and easy implementation (Storn & Price, 1995), (Camps Echevarría et.al., 2014a). This algorithm is based on three operators: Mutation, Crossover and Selection, for which must be defined the population size NP, the number of parameters to be optimized, the crossover constant $\mathrm{C_R}$ and the scale factor F. The crucial idea behind DE is the combination of these operators at each *j-th* iteration using vector operations to obtain a new solution candidate. The configuration of DE can be summarized using the notation $DE/\mathbb{X}^j/\gamma/\lambda^*$ where $\mathbb{X}^j$ denotes the solution to be disturbed in the *j- th* iteration; γ is the number of pair of vectors used for disturbing $\mathbb{X}^j$ and λ^* indicates the distribution function that will be used in the crossover. In this paper, it has been considered the configuration$DE/\mathbb{X}^{j(best)}/1/Bin$, where $\mathbb{X}^{j(best)}$ indicates the best individual of the population, and Bin the Binomial Distribution function. The mutation operator is expressed in the following way:

$$\mathbb{X}^{j+1} = \mathbb{X}^{j(best)} + F_S(\mathbb{X}^{j(\alpha)} - \mathbb{X}^{j(\beta)}) \tag{8}$$

where $\mathbb{X}^{j+1}$, $\mathbb{X}^{j(best)}$, $\mathbb{X}^{j(\alpha)}$, $\mathbb{X}^{j(\beta)} \in \mathbb{R}^n$, $\mathbb{X}^{j(\alpha)}$ and $\mathbb{X}^{j(\beta)}$ are elements of the population *Z*, i.e. one pair of vectors, and *F*F_S is a scaling factor. For complementing the mutation operator, the crossover operator is defined for each component $\mathbb{X}_n$ of the solution vector:

$$\mathbb{X}_n^{j+1} = \begin{cases} \mathbb{X}_n^{j+1}, & \text{if } R < C_R \\ \mathbb{X}_n^{j(best)}, & \text{otherwise} \end{cases} \tag{9}$$

where $0 \leq C_R \leq 1$ is the crossover constant that is another control parameter in DE, and *R* is a random number generated by the distribution λ^*, which in this case is the binomial distribution.

Finally, the selection operator results as follows:

$$\mathbb{X}^{j+1} = \begin{cases} \mathbb{X}^{j+1}, & \text{if } F\left(\mathbb{X}^{j+1}\right) \leq F\left(\mathbb{X}^{j(best)}\right) \\ \mathbb{X}^{j(best)}, & \text{otherwise} \end{cases} \tag{10}$$

where *F* is the objective function such as the one given by Eq. (5).

The pseudo code of DE algorithm is presented in Figure 1.

```
Input: $Z$, $F_S$, $C_R$, $Eval\_max$
Output: $\mathbb{X}^{best}(F(\hat{\theta}) = (m, \sigma))$
Generate initial population of $Z$ solutions,
Select better solution $\mathbb{X}^{best}$,
for $l=1$ to $l=Itr\_max$ do
    Apply Mutation according to Eq. (8)
    Apply Crossover according to Eq. (9)
    Apply Selection according to Eq. (10)
    Update $\mathbb{X}^{best}$
    Verify stopping criterion
end for
```

Figure 1. Pseudo Code of Differential Evolution Algorithm

Particle Swarm Optimization Algorithm. Particle Swarm Optimization (PSO) is an algorithm inspired by the social behavior of different species (Eberhart & Kennedy, 1995). The underlying idea of this algorithm is based on the collaborative work of individual organisms to reduce energy at time of migration or to find food in nature. There are many variants of this algorithm. In the present paper, the conventional PSO version developed by Eberhart & Kennedy (1995) is used, given its simplicity and easy implementation for parameter estimation problems, with kernel methods.

PSO works with a group or population (swarm) of Z agents (particles), which are interested in finding a good approximation to the global minimum or maximum x_0 of the objective function $f : \mathbb{D} \subset \mathbb{R}^n \to \mathbb{R}$. Each agent moves throughout the search space $\mathbb{D}$. The position of the *z - th* particle is identified with a solution for the optimization problem. On each *l –th* iteration, its value is updated and it is represented by a vector $\mathbf{X}_z^l \in \mathbb{R}^n$.

Each particle accumulates its historical best position $\mathbf{X}_z^{pbest}$, which represents the best achieved individual experiences. The best position that was achieved along the iterative procedure, among all the agents in the population, i.e. $\mathbf{X}^{gbest}$, represents the collective experience.

The generation of the new position needs the current velocity of the particle $\mathbf{V}_z^l \in \mathbb{R}^n$ and the previous position $\mathbf{X}_z^{l-1}$

$$\mathbb{X}_z^l = \mathbb{X}_z^{l-1} + \mathbb{V}_z^l \tag{11}$$

The vector $\mathbf{V}_z^l$ is updated according to the following expression:

$$\mathbb{V}_z^l = \mathbb{V}_z^{l-1} + c_1\mathbb{R}(\mathbb{X}_z^{pbest} - \mathbb{X}_z^{l-1}) + c_2\mathbb{R}(\mathbb{X}^{gbest} - \mathbb{X}_z^{l-1}) \tag{12}$$

where $\mathbb{V}_z^{l-1}$ is the previous velocity of the *z – th* particle, $\mathbb{R}$ denotes a diagonal matrix with random numbers in the interval [0,1]; and *c₁, c₂* are the parameters that characterize the trend during the velocity updating (Kameyama, 2009), balancing the individual and group experiences. They are called cognitive and social parameters, respectively. They represent how the individual and social experiences influence in the next agent decision. Some studies have been made in order to determine the best values for c_1 and c_2. The values $c_1 = c_2 = 2$, $c_1 = c_2 = 2{:}05$ or $c_1 > c_2$ with $c_1 + c_2 \leq 4.10$ are recommended (Camps Echevarría et.al., 2014b).

Some variants of the algorithm have been developed with the objective of improving some characteristics of PSO, e.g. velocity, stability and convergence.

Equations (11, 12) represent the canonical implementation of PSO. Another well-known variant is the one with inertial weight, which considers either constant inertial weight or inertial weight with reduction. The idea behind this variant is to add an inertial factor ω for balancing the importance of the local and global search (Kaneyama, 2009). This parameter ω affects the updating of each particle velocity by the expression

$$\mathbb{V}_z^l = \omega\mathbb{V}_z^{l-1} + c_1\mathbb{R}(\mathbb{X}_z^{pbest} - \mathbb{X}_z^{l-1}) + c_2\mathbb{R}(\mathbb{X}^{gbest} - \mathbb{X}_z^{l-1}) \tag{13}$$

Nowadays, the most accepted strategy for ω is to establish $\omega \in [\omega_{min}; \omega_{max}]$ and reduce its value according to the number of the current iteration *l* by means of

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{Itr_{max}} l \tag{14}$$

where Itr_{max} is the maximum number of iterations to be reached. The basic PSO is recognized as a particular case for the alternative that considers the inertial weight ω= 1 along all the execution of the algorithm (Kaneyama, 2009).

Figure 2 presents the seudo code of PSO algorithm.

```
Data $c_1$ , $c_2$ , $Z$ , $Itr_{max}$ , $\omega_{max}$ , $\omega_{min}$
Generate randomly $\mathbb{X}_0^z$ and $V_0^z$ for Z particles of the swarm;
Evaluate $F(\mathbb{X}_0^Z)$ for each particle, update $\mathbb{X}_z^{pbest}$ and $X^{gbest}$
for l = 1 to l = $Itr_{max}$ do
    for z = 1 to z = Z do
        Update $V_l^z$ with Eq. (13);
        Update $\mathbb{X}_z^l$ with Eq. (11);
        Compute $F(\mathbb{X}_l^z)$;
        Update $\mathbb{X}_z^{pbest}$;
    end
end
Update $\mathbb{X}^{gbest}$;
Verify stopping criteria; Solution: $\mathbb{X}^{gbest}$
```

Figure 2. Pseudo code of PSO algorithm

Kernel Function. In general, the selection of a kernel depends on the application. However, the Gaussian, or RBF kernel, is one of the most popular (Motai, 2015; Bernal de Lázaro et. al., 2016). This function is a homogeneous kernel, which maps the input space to a higher dimension space. The RBF kernel is mathematically defined as:

$$\mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{15}$$

where σ is called bandwidth, and indicates the degree of smoothness of the function. If σ is overestimated, the exponential tends to show a linear behavior, and its projection in a higher dimensional space loses its ability to separate non-linear data. Meanwhile, if σ is underestimated, the result will be highly sensitive to noise in the training step of the algorithm.

Fitness function. The fitness function used in this paper is the partition coefficient (*PC*) (Li et. al., 2012) which measures the fuzziness degree of the partition *U*. The *PC* is calculated by using Eq. (16), and it is used as a validity index to evaluate quantitatively the result of a clustering method.

$$PC = \frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^2 \qquad (16)$$

The parameters *N, c* and μ_{i_k} are defined in section 2.2.

If the partition *U* is less fuzzy, the clustering process is better. Being analyzed in a different way, it allows to measure the degree of overlapping among the classes. In this case, the optimum comes up when *PC* is maximized, i.e., when each pattern belongs to only one class. Likewise, the minimum comes up when each pattern belongs to all class.

Therefore, the optimization problem is defined as:

$$max\left\{PC\right\} = max\left\{\frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m\right\}$$

subject to:

$$m_{min} < m \leq m_{max} \;\; ; \;\; \sigma_{min} \leq \sigma \leq \sigma_{max}$$

where σ is shown in Eq. (15).

When *PC* is maximized, a better performance is achieved in the classification process, because confusion is reduced in determining to which class an observation belongs.

3. RESULTS AND DISCUSSION

In this section, the techniques described previously are applied in the design of a fault detection system for the Tennessee Eastman (TE) process benchmark. The fault detection system designed shows an improvement in the classification process, because of the best parameters determination to the KFCM algorithm, using optimization algorithms.

3.1 Study Case: Tennessee Eastman Process (TE)

The Tennessee Eastman (TE) process is widely used as a chemical plant benchmark to evaluate the performance of new control and monitoring strategies (Downs & Vogel, 1993). TE contains 21 preprogrammed faults, and one normal operating condition data set. The data sets from the TE are generated during a 48h operation simulation with the inclusion of faults after 8 hours of simulation. Table 1 shows the faults considered in this paper, in order to evaluate the advantages of the presented fault diagnosis proposal. All data sets used to test the procedure hereby proposed were given in Ref. (Downs & Vogel, 1993), and it can be downloaded from http://web.mit.edu/braatzgroup/TE_process.zip.

According to the specialized literature, Faults 3, 5, 9, 10, 11, as well as Fault 15, have small magnitudes, and therefore their detection is very difficult. Fault 3 is generated from one step pulse, in the D feed temperature, but it has a quite close behavior to the normal data in terms of the mean and variance. Beyond that, Fault 5 is due to one step pulse in the condenser cooling water inlet temperature. This variation causes a mean shift on the condenser cooling flow, and a chain reaction in other variables, which produces an out-of-control operation. In this case, the control loops are able to compensate such changes. In consequence of this, the variables return to their set-point, except the condenser cooling water inlet temperature (Chiang et.al., 2001).

As a matter of fact, the fault does not disappear, it is only hidden.
On the other hand, Fault 9 is a result of one random variation in the feed D temperature. It is hard to detect too. Fault 10 appears when the feed C temperature, of stream 4, is randomly changed. It is interesting to observe that as a result of this fault, the temperature and pressure on the stripper also changes. Then, the stripper steam valve is manipulated by the control loops to compensate the changes by means of the stripper steam flow rate, which makes difficult the detection of this fault (Ge et. al., 2009). Fault 15 is a sticking in the condenser cooling water valve. Similarly, to Fault 3, the historical data set of Fault 15 has little difference with respect to the normal data. Therefore, Fault 15 is also hard to detect.

3.2 Experimental Results

To evaluate the proposed diagnostic scheme, two tests were conducted. First, the classifier based on the KFCM algorithm was trained without considering the preprocessing stage, generating the False Alarm Rate (FAR) and the Fault Detection Rate (FDR) indicators. Thereafter, the KICA and the KFCM algorithms were employed together in the fault diagnosis process, generating also the FAR and FDR indicators. In general, a total of 320 observations (samples) for each class (operating stages) was used in the training data set, while, 800 observations for each class were used in the testing data set. As an outstanding aspect, it should be highlighted that the dimension of feature space was significantly reduced $\mathbf{R}^{33} \to \mathbf{R}^{24}$ by the preprocessing with KICA. The number of independent components herein used represents 73% of the information contained in the data set of the TE process.

Table 1. Description of faults of the TE process.

Fault	Process Variable	Type	Fault	Process Variable	Type
F1	A/C feed ratio, B composition constant	Step	F9	D feed temperature	Random
F2	B composition, A/C ratio constant	Step	F10	C feed temperature	Random
F3	D feed temperature	Step	F11	Reactor cooling water inlet temperature	Random
F4	Reactor cooling water inlet temperature	Step	F12	Condenser cooling water inlet temperature	Random
F5	Condenser cooling water inlet temperature	Step	F13	Reaction kinetics	Slow Drift
F6	A feed loss	Step	F14	Reactor cooling water valve	Sticking
F7	C header pressure loss-reduced availability	Step	F15	Condenser cooling water valve	Sticking
F8	A, B and C feed composition	Random			

The DE algorithm implemented for the kernel parameter optimization was executed using the following specifications: population size NP = 10, maximum iteration *MaxIter* = 100, difference vector scale factor *F* = 0:1, and crossover criterion C_R = 0:9. Moreover, the following search ranges for the parameters to be estimated were considered $m \in [1,2]$, and $\sigma \in [1,150]$. The PSO algorithm was also configured with such ranges. However, for the PSO algorithm the estimated parameters were searched by using the following specifications: population size = 20, ω_{max} = 0:9, ω_{min} = 0:4, c_1 = 2, c_2 = 2, and Itr_{max} = 100. In this context, were obtained the results for experiment 1 (without data preprocessing) and 2 (with data preprocessing using KICA algorithm).

For the implementation of the DE and PSO algorithms the following stopping criteria were considered:

- Criterion 1: Maximum number of iterations (100).
- Criterion 2: Value of the objective function (0.9999). See Eq. (16).

The value of the *σ* parameter for the KICA algorithm used in experiment 2 was 492:53, and it was taken from (Bernal de Lázaro et. al., 2016).

Table 2 shows the values of the parameters *m* and *σ* estimated for each experiment.

Table 2. Values of the parameters estimated for Experiment 1 (without data preprocessing) and Experiment 2 (with data preprocessing).

Exp. 1 (without data preprocessing)

Parameters	DE	PSO
m	1.7148	1.7150
σ (KFMC)	94.9676	85.9322

Exp. 2 (with data preprocessing)

Parameters	DE	PSO
m	1.3832	1.4284
σ (KFMC)	55.4045	37.7942

Results of the Classification. The information provided by the confusion matrix *C* associated with the classification process was used to evaluate the performance of the fault diagnosis procedures (Fawcett, 2006). In the confusion matrix, the main diagonal represents the number of observations successfully classified. In the first row, outside the main diagonal, the false alarms are represented (i.e.; *i* = 1*, j* = 2, 3, ..., k). The number of missing alarms are shown at the first column (i.e.; *j* = 1, *I* = 2, 3, ..., k). Then, all general information about the fault diagnosis stage is available in the confusion matrix. For example, the detectability of the fault detection schemes can be studied in detail through the False Alarm Rate (FAR), and the Fault Detection Rate (FDR), given by

$$\mathrm{FAR} = \frac{\text{No. of samples } (J > J_{lim} | f = 0)}{\text{total samples } (f = 0)} \cdot 100\% \tag{17}$$

$$\mathrm{FDR} = \frac{\text{No. of samples } (J > J_{lim} | f \neq 0)}{\text{total samples } (f \neq 0)} \cdot 100\% \tag{18}$$

where *J* is the output for the used discriminative algorithms by considering the fault detection stage as a binary classification process, and J_{lim} is the threshold that determines whether one sample is classified as a fault or normal operation.

Table 3 shows the performance of the evaluated procedure in terms of false alarms and missing faults detected.

In Experiment 1, the results were obtained without using the data preprocessing with the KICA algorithm. Note that in this case some faults are easily detected (e.g., Faults 1, 2, 4, 6, 7, 8, 12, 13 and 14), with higher values for the FDR measure. Nonetheless, as expected, some faults are difficult to detect (e.g., Faults 3, 5, 9, 10, 11, and 15) due to the fact that they are hidden by the influence of other variables of the process. In general, the performance for these faults is characterized by a high FAR value or a small FDR value. That means a low probability of distinguishing correctly between the normal operating condition (NOC) and the abnormal situations. In general, the experiments performed have shown that PSO gives the best results for the faults 1, 2, 5, 7, 8, 9, 10, 12, 13 and 15. Meanwhile, the parameters obtained with the DE algorithm allowed to achieve a best performance for the faults 1, 3, 4, 6, 11, as well as Fault 15.

In Experiment 2, the results were obtained for the combined work between the KICA and KFCM algorithms. It is interesting to observe that due to preprocessing stage with KICA there is an important reduction of the false alarms. Regarding to the small magnitude faults, the higher results for FDR are obtained with the parameters adjusted by using the DE algorithm. Except for Faults 10 and 11, the fault detection scheme presented a worse performance with respect to the small magnitude results when the PSO algorithm was used. In particular, using KICA and PSO a worse performance for Faults 4 and 9 is obtained.

Table 3. Fault detection performance with preprocessing data through KICA.

	Exp. 1 Without data preprocessing				Exp. 2 With data preprocessing			
	KFCM-DE		KFCM-PSO		KFCM-DE		KFCM-PSO	
Faults	**FAR %**	**FDR %**	**FAR %**	**FDR %**	**FAR %**	**FDR %**	**FAR %**	**FDR %**
F1	9.25	89.13	0.00	95.25	0.00	98.75	0.00	99.13
F2	6.88	96.13	0.00	98.5	0.00	99.25	0.00	92.63
F3	8.25	77.63	6.25	64.88	4.88	88.38	18.75	61.13
F4	6.13	95.75	7.25	89.75	0.00	96.25	0.00	95.38
F5	8.88	76.25	8.13	77.88	8.75	81.75	17.25	64.80
F6	0.00	97.88	6.13	97.25	0.00	95.13	0.00	99.75
F7	9.88	93.13	5.25	93.25	0.00	96.25	0.00	68.38
F8	7.88	91.38	8.75	97.88	0.00	93.25	0.00	66.75
F9	5.25	76.25	0.00	83.13	5.38	92.63	21.88	74.13
F10	0.00	75.38	7.25	76.13	6.25	79.25	9.25	87.88
F11	4.75	82.25	8.88	78.25	7.38	80.75	2.75	81.25
F12	8.5	91.25	8.38	96.75	0.00	92.13	0.00	94.88
F13	8.88	95.13	8.38	96.75	0.00	91.75	0.00	95.63
F14	10.75	93.13	6.88	94.13	0.00	90.13	0.00	99.38
F15	0.00	72.25	10.25	58.88	6.38	63.25	3.25	57.13

In general terms, the performance for Faults 2, 3, 4, 5, 7, 8, 9 and 15 is greater when the classifier uses the DE algorithm to estimate its parameters. By means of the results shown in the table, it is demonstrated that the overall performance of the detection system is better with the data preprocessing based on KICA. However, it should not be forgotten the important role that the configuration of the parameters has for the application of the kernel methods in the fault detection tasks. In fact, it is necessary to emphasize that the detection levels herein achieved, for the small-magnitude faults in this process, are still insufficient for the current industrial standards.

5. CONCLUSIONS AND FUTURE WORK

In this paper was presented a comparative study between two metaheuristic optimization algorithms, Differential Evolution (DE) and Particle Swarm Optimization (PSO). These algorithms were used to estimate the parameters of the Kernel Fuzzy C-means (KFCM) classifier. First, a diagnostic classifier without considering a data preprocessing stage was evaluated. Thereafter, the KICA and the KFCM algorithms were jointly employed in the fault detection process. For the comparative evaluation were established as the comparison criteria the false alarm and fault detection rates. The experiments have shown that the overall performance of the detection scheme is better with the data preprocessing, and higher results are obtained adjusting the kernel parameter by using the DE algorithm.

For future works, it is necessary to analyze the use of KFCM considering the dynamics of the process to improve the detection of incipient and small-magnitude faults. Furthermore, it would be interesting to investigate the use of other optimization techniques for kernel parameters tuning, including the algorithms applied in the preprocessing and classification stages.

Acknowledgements

The authors acknowledge the financial support provided by FAPERJ-Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro, CNPq-Conselho Nacional de Desenvolvimento Científico e Tecnológico, and CAPES-Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, research supporting agencies from Brazil.

REFERENCES

BERNAL-DE-LÁZARO, J.M.; PRIETO-MORENO, A.; LLANES-SANTIAGO, O.; SILVA-NETO, A.J., **Optimizing kernel methods to reduce dimensionality in fault diagnosis of industrial systems**. Computers & Industrial Eng, v 87, 2015, p. 140-149.

BERNAL-DE-LÁZARO, J.M.; LLANES-SANTIAGO, O.; PRIETO-MORENO, A.; KNUPP, D.C.; SILVA-NETO, A.J., **Enhanced dynamic approach to improve the detection of small-magnitude faults**. Chemical Engineering Science, v 146, 2016, p. 166-179.

CAMPS ECHEVARRÍA, L.; LLANES SANTIAGO, O.; SILVA NETO, A.J., **An approach for fault diagnosis based on bio-inspired strategies**, Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 2010.

CAMPS ECHEVARRÍA, L.; LLANES-SANTIAGO, O.; SILVA NETO, A.J.; CAMPOS-VELHO, H. F. **An approach to fault diagnosis using metaheuristics: A new variant of the Differential Evolution algorithm**, Computación y Sistemas, v 18, n1, 2014a p. 5–17.

CAMPS ECHEVARRÍA, L.; LLANES SANTIAGO, O.; HERNÁNDEZ FAJARDO, J. A.; SILVA NETO, A. J.; JÍMENEZ SÁNCHEZ, D., **A variant of the particle swarm optimization for the improvement of fault diagnosis in industrial systems via faults estimation**, Engineering Applications of Artificial Intelligence, v 28, 2014b p.36– 51.

CAO, S.; ZUO, X.; TAO, A.; WANG, J.; CHEN, X. **A Bearing Intelligent Fault Diagnosis Method based on Cluster Analysis**, Applied Mechanics and Materials, v152, 2012, p. 1628 –1633.

CHIANG, L.H.; BRAATZ, R.D.; RUSSELL, E.L., **Fault detection and diagnosis in industrial systems**, Springer, 2001.

DING, Y.; FU, X., **Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm**, Neurocomputing, v 188, 2016, p. 233–238.

DOWNS, J.J.; VOGEL, E.F., **A plant-wide industrial process control problem**, Computers & Chemical Eng, v 17 n 3, 1993, p. 245–255.

EBERHART, R, KENNEDY, J., **A new optimizer using particle swarm theory**. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, p. 39-43.

Fawcett, T., **An introduction to ROC analysis**, Pattern Recognition Letters, v 27 n8, 2006, p. 861–874.

GE, Z.; YANG, C.; SONG, Z., **Improved kernel PCA-based monitoring approach for nonlinear processes**, Chemical Engineering Science, V 64 n 9, 2009, p. 2245–2255.

JIANG, L.; ZENG, B.; JORDAN, F.; CHEN, A., **Kernel Function and Parameters Optimization in KICA for Rolling Bearing Fault Diagnosis**. Journal of Networks, v 8, 2013, p. 1913-1919.

KAMEYAMA, K., **Particle swarm optimization – a survey**, IEICE Trans. Inf. Syst. E92-D, 2009, p. 1354-1361.

LI, CH.; ZHOU, J.; KOU, P.; XIAO, J., **A novel chaotic particle swarm optimization based fuzzy clustering algorithm**, Neurocomputing, v 83, 2012, p. 98–109.

MANABU, K., **Monitoring independent components for fault detection**, AIChE Journal, v 49 n 4, 2003, p. 969–976.

MOTAI, Y., **Kernel association for classification and prediction: A survey**, IEEE Transactions on Neural Networks and Learning Systems, v 26 n 2, 2015, p. 208-223.

STORN, R.; PRICE, K., **Differential Evolution-A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces**. Tech. Report, International Computer Science Institute (Berkeley). Available in: http://www1.icsi.berkeley.edu/~storn/TR-95-012.pdf, 1995.