Applied Soft Computing 49 (2016) 676–686

Contents lists available at ScienceDirect

Applied Soft Computing

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

A new fault classification approach applied to Tennessee Eastman benchmark process

Marcos F.S.V. D'Angelo [a,*], Reinaldo M. Palhares [b], Murilo C.O. Camargos Filho [a], Renato D. Maia [a], João B. Mendes [a], Petr Ya. Ekel [c]

[a] Department of Computer Science – UNIMONTES, Av. Rui Braga, sn, Vila Mauricéia, 39401-089 Montes Claros, Brazil
[b] Department of Electronics Engineering, Federal University of Minas Gerais, Av. Antônio Carlos, 6627, 31270-901 Belo Horizonte, Brazil
[c] Graduate Program in Electrical Engineering, Pontifical Catholic University of Minas Gerais, Av. Dom Jose Gaspar, 500, 30535-610 Belo Horizonte, Brazil

ARTICLE INFO

Article history:
Received 4 December 2015
Received in revised form 22 August 2016
Accepted 23 August 2016
Available online 13 September 2016

Keywords:
Fault detection and isolation
Fuzzy/Bayesian approach
Immune/neural formulation
Tennessee Eastman benchmark process

ABSTRACT

This study presents a data-based methodology for fault detection and isolation in dynamic systems based on fuzzy/Bayesian approach for change point detection associated with a hybrid immune/neural formulation for pattern classification applied to the Tennessee Eastman benchmark process. The fault is detected when a change occurs in the signals from the sensors and classified into one of the classes by the immune/neural formulation. The change point detection system is based on fuzzy set theory associated with the Metropolis–Hastings algorithm and the classification system, the main contribution of this paper is based on a representation which combines the ClonALG algorithm with the Kohonen neural network.

1. Introduction

Currently, in the industry, it is common the application of fault detection and isolation (FDI) techniques in order to keep productivity standards, ensuring safety and allowing a cost effective maintenance policy. As a matter of fact, it is possible to design corrective actions, system redundancies and safety policies by using fault analysis techniques to mitigate the effects of a fault [1]. Typically, a fault diagnosis procedure can be divided into three tasks: (i) the fault detection indicating the occurrence of some fault in a monitored system; (ii) the fault isolation establishing the type and/or location of the fault and; (iii) the fault identification determining the magnitude of the fault. In some applications, after detecting and diagnosing the fault, is required a self-correction process, usually done via controller reconfiguration. This self-correction process is referred to as fault accommodation.

1.1. Literature review

Several FDI approaches have been presented in the literature and they can be categorized in quantitative models [2] and qualitative models [3,4]. Usually, quantitative model-based approaches demand the knowledge of mathematical models for the process whereas qualitative approaches are based on some pattern analysis of historic process data.

Over the last few decades, an important methodological and analytically focused approach based on observers/filters to the FDI problem [5–14], emerged in the context of generating signals to reflect inconsistency between normal and faulty operation in the system. Practical applications of the techniqued can be found in [7–9,15]. Sometimes it is important to consider the presence of time delay in fault detection problems in dynamic systems, this can be approximated by techniques based on observers/filters dedicated [13]. In the literature, approaches involving strategies based on observers are often built under the context of an observer with unknown inputs, since they have good results in some characteristics of uncertainties and nonlinearities that can be grouped in the unknown input [9,12,13]. Overall, there are more difficulties in addressing the problem of fault detection for systems that present parametric uncertainties in the model, considering that the residual generation for fault detection does not take into account these uncertainties. Other important techniques in the context of analytical approaches are parity relations [9,16] and Kalman or robust

* Corresponding author.
E-mail addresses: marcos.dangelo@unimontes.br (M.F.S.V. D'Angelo), rpalhares@ufmg.br (R.M. Palhares), murilo.camargosf@gmail.com (M.C.O. Camargos Filho), renato.dourado@unimontes.br (R.D. Maia), joao.mendes@unimontes.br (J.B. Mendes), ekel@pucminas.br (P.Ya. Ekel).

http://dx.doi.org/10.1016/j.asoc.2016.08.040
1568-4946/

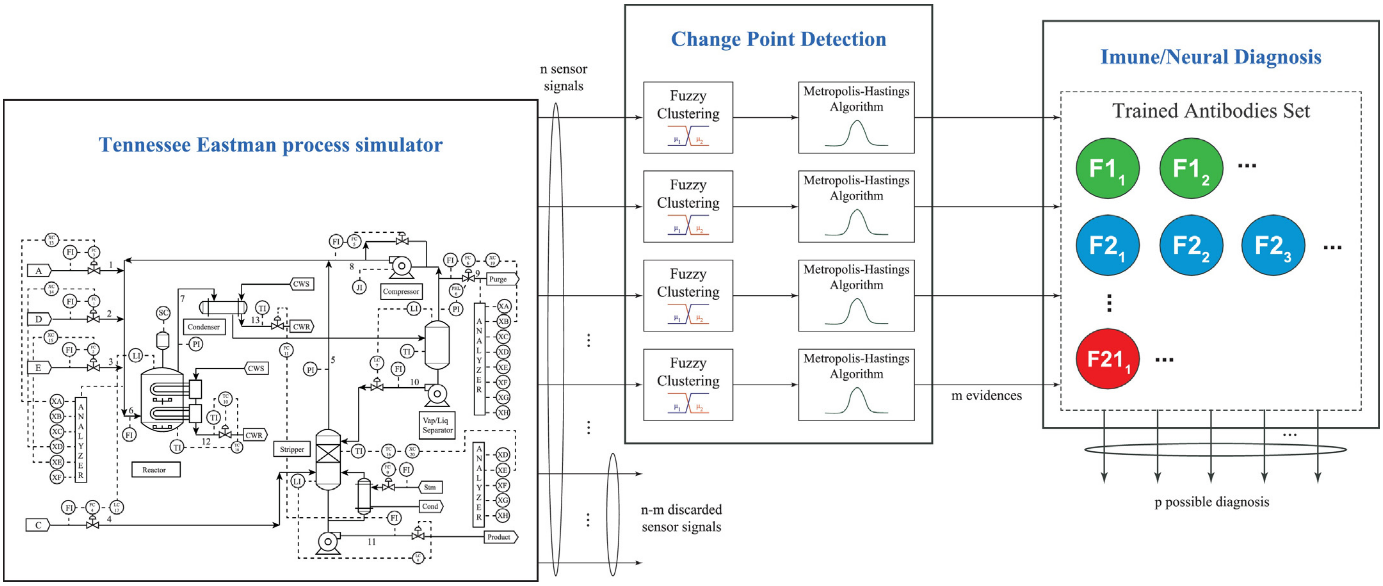

Fig. 1. FDI framework.

filters [17,18]. The requirement of a mathematical model for the process can lead to several difficulties in the implementation of these approaches due to factors such as system complexity, high dimensionality, nonlinearities, and parametric uncertainties.

Another approach to the FDI problem in dynamic systems is the use of neural networks [19] and fuzzy logic [20]. These techniques may be more attractive because they do not require explicit mathematical models and can be used both in quantitative models (such as observers) and qualitative models (such as classifiers). In cases where mathematical models are required and dynamic system is non-linear, the use of Takagi-Sugeno fuzzy models has been a good instrument [19]. Practical applications of the techniques can be found in [22–27].

Other qualitative model approaches, which are based on some pattern analysis of the historic process data, are signed directed graph [28], fault tree [29], qualitative trend analysis [30,31], mutual information [32], artificial immune systems [33–35], Bayesian networks [36,37], and others exploring hybrid strategies [38–40]. An important discussion on qualitative model approaches for FDI is multivariate statistical process monitoring, and techniques like principal component analysis (PCA) [41,42] and partial least squares (PLS) [43] have been widely used for fault detection and diagnosis in industrial practice. These kinds of methods first project the multivariate and collinear data onto a lower dimensional subspace; then a test statistics like T^2 and SPE is developed to monitor the multivariate data.

1.2. Contributions

In this paper, a data-based FDI methodology based on two steps is presented. In the first step, the possible fault event is associated with a change point detection problem in time series. To do this, a fuzzy clustering is performed such that a given time series is transformed into a time series with a beta distribution and the Metropolis–Hastings algorithm [44] is used to detect the change point, i.e., a possible fault event evidence. The Metropolis–Hastings algorithm indicates a change point probability and this information can be used for classification purpose. The objective of the fuzzy/Bayesian FDI approach is to compute a change point probability vector in its variables. This vector is used by the classification approach in the second step increasing correctness classification and reducing the total patterns used in training step of classification methodology. The idea of the second step, the main contribution of this paper, is to deal with the fault classification in a new way using an immune/neural representation which combines the ClonALG [45] with the Kohonen neural network [46]. The stopping criteria and the antibody selection of the proposed new approach had to be modified in order to accommodate a non-fixed number of antibodies. One of the contributions of this paper is to propose this new FDI scheme in which there is no requirement of a mathematical or statistical models for the plant or signals, neither any threshold specification. Other important contribution is to propose an approach that has low implementation complexity with high performance in fault detection and isolation, which is a crucial point for on-line real-world applications.

The FDI qualitative approach presented in this paper is applied to the Tennessee Eastman benchmark process [47]. The benchmark primarily received attention in the conventional process control community, and, recently, applications have been reported in soft computing community, focused on case studies for fault detection and isolation [48]. The main methods used in literature, for fault detection and isolation in Tennessee Eastman benchmark process, are based on principal component analysis (PCA) [41,42] and partial least squares [43] (other methods as neural networks [49], fuzzy systems [50], neuro fuzzy system [51], support vector machine (SVM) [42] and two or more associations methodologies [52,53] were also used). The effectiveness of these conventional methods (PCA and PLS) requires that the process data approximately follow multivariate Gaussian distributions for the derivation of control limits. However, industrial data often obeys non-Gaussian distributions in a way that PCA/PLS based monitoring techniques become ill-suited, justifying the use of pattern recognition techniques to identify similarities between multivariate time series data sets.

An overview of the FDI framework proposed in this paper is illustrated in Fig. 1. The first module reads the sensors available in the Tennessee Eastman benchmark process, extracting windows of the measurements. Each extracted window consists of a time series that is used in fuzzy/Bayesian approach to indicate the probability of change point occurrence along the time series. The second module consists of the immune/neural system in which the probability of change point occurrence is used to classify the fault.

The remainder of this paper is organized as follows. Section 2 presents a synthesis of change point detection formulation, based on fuzzy set theory associated with Metropolis–Hastings algorithm,

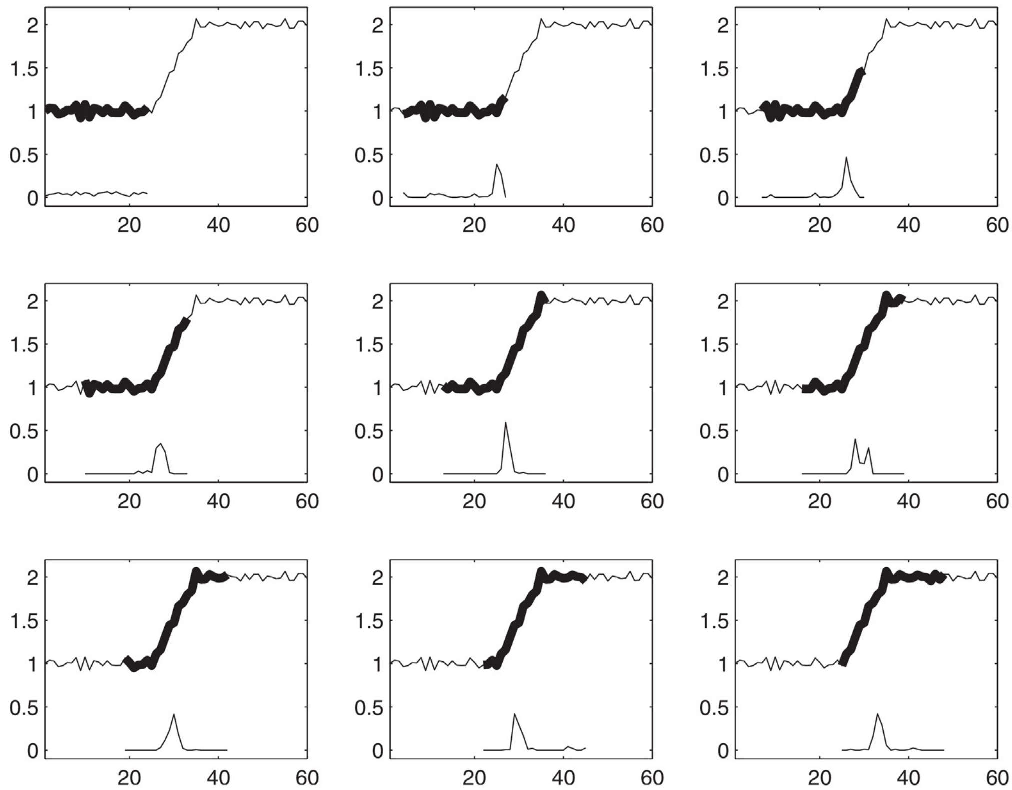

Fig. 2. Each subfigure illustrates the time series $y(t)$ in the top with the time window of samples indicated by the thick line, and the obtained results in the bottom.

used for fault detection, see details in [44]. Section 3 presents a new immune/neural approach, for fault classification, based on a representation which combines the ClonALG algorithm with the Kohonen neural network (the main contribution of this work). Section 4 shows the results for fault detection and isolation considering the Tennessee Eastman process simulator and compares them with other methods. Finally, Section 5 presents the concluding remarks.

2. Fuzzy/Bayesian approach used on fault detection

This strategy is based on steps as proposed in [44], in which a given time series is transformed into a beta distribution via fuzzy clustering. After this transformation, the Metropolis–Hastings algorithm is used to detect the change point. The steps of the methodology are given by Algorithm 1.

Algorithm 1. Change point detection algorithm

Data: Time series window, $y(t)$
Result: Change point in time series window, m
begin
Found the k centers of time series;
Compute a new time series as the fuzzy membership of each point of time series, $y(t)$, for each center, C_i, by:

$$\mu_i(t) \triangleq \left[\sum_{j=1}^{k} \frac{|y(t)-C_i|^2}{|y(t)-C_j|^2}\right]^{-1} \quad (1)$$

Compute the change point, m, by Metropolis–Hastings algorithm using the time series transformed, $\mu_i(t)$
return *Change point in time series window, m*
end

To illustrate the methodology, the following time series is used:

$$y(t) = \begin{cases} p_1 + 0.1 * \epsilon(t) - 0.1 * \epsilon(t-1), & \text{if } t \leqslant m_1, \\ (t - m_1)\dfrac{p_2 - p_1}{m_2 - m_1} + p_1 + 0.1 * \epsilon(t) - 0.1 * \epsilon(t-1), & \text{if } t > m_1 \text{ and } t \leqslant m_2, \\ p_2 + 0.1 * \epsilon(t) - 0.1 * \epsilon(t-1), & \text{if } t > m_2 \end{cases} \quad (2)$$

where p_1 is the first operation point, p_2 is the second operation point, $\epsilon(t)$ is a noise signal with $\pi(\cdot)$ distribution and $[m_1, m_2]$ is the interval where the change point occurs in (2).

Fig. 2 shows several time windows, with fixed size (24 samples), from a time series $y(t)$ (given by (2)) of 60 samples (x-axis), where $p_1 = 1$, $p_2 = 2$, $m_1 = 25$, $m_2 = 35$ and $\pi(\cdot) \sim U(0, 1)$ (uniform distribution with the support defined between 0 and 1). Notice that in each subfigure of Fig. 2, the time series $y(t)$ is depicted in the top with the time window illustrated by the thick line, and the obtained results are depicted in the bottom. It is clear to see that the probability of change point is in the neighborhood of 30.

The objective of the fuzzy/Bayesian FDI approach in Tennessee Eastman process is to compute a change point probability vector in its variables. This vector is used by the classification hybrid formulation and it increases correctness classification and reduces the total patterns used in training step of classification methodology.

3. Immune/neural approach used on fault isolation

The pattern recognition algorithm is based on the immune-inspired algorithm ClonALG [45], aided by the Kohonen neural network [46] training algorithm in order to increase its performance.

Inspired in the biological immune system, the artificial system must be able to react to any foreign substance that affects its integrity. These substances are called antigens and the system elements that react to them are called antibodies [54]. The ClonALG approach explicitly takes into account the antibodies affinity where only the fittest are selected to proliferate through a process called maturation (or mutation). In this paper, the affinity criterion is determined by the Euclidian distance.

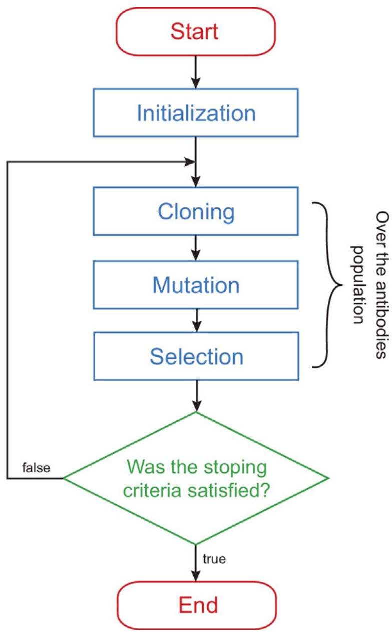

Fig. 3. Algorithm flow chart, where a part of the mutation mechanism is carried out by the Kohonen neural network.

3.1. *Training process*

Every antigen used in the training step was tagged with its type so that the algorithm could train each antibody to recognize a specific type of antigen. The metric used to determine if an antibody recognized an antigen was the Euclidian distance, in which the antigen will be recognized by the nearest antibody.

The training process begins with one antibody created in a central position (related to the antigens' positions in space). Then, all antibodies are submitted to three steps that iterate until the stopping criteria are satisfied: cloning, mutation and selection (Fig. 3). It is important to know that this approach is based on a non-fixed size population of antibodies.

3.1.1. *Selection mechanism*

As shown in Fig. 4, the antibody selection happens when all combinations of two elements out of the antibodies set are analyzed. When the set of antigens recognized by two of them are of the same type and they are close with each other, a new antibody is created between them and those two must be deleted. Also, the antibodies which did not recognize any type of antigen are deleted. The threshold used as proximity criterion between two antibodies is calculated at every iteration; it is set to 25% of the average distance between every two antibodies.

3.1.2. *Cloning mechanism*

In the cloning mechanism (Fig. 5), each antibody of the system is cloned twice and each one of these clones suffers a random mutation using a uniform distribution in its space position. This guarantees diversity in the antibody population.

3.1.3. *Mutation mechanism*

The mutation began at the cloning mechanism and will be improved by a Kohonen neural network (Fig. 6). Each antigen linearly reduces the distance between him and the closest antibody. This algorithm changes the antibody's position as if it was weights of a neuron.

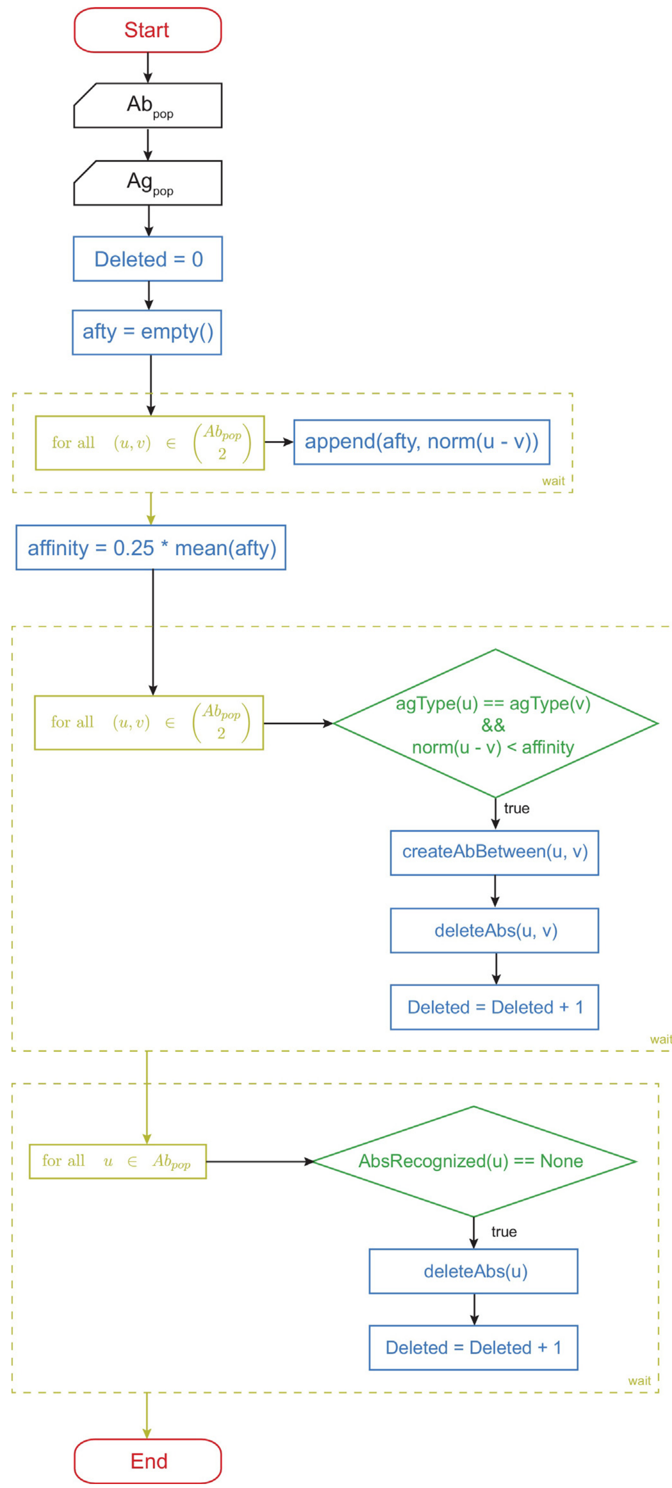

Fig. 4. Selection mechanism flow chart, where $\binom{Ab_{pop}}{2}$ represents every two element combination out of the antibody population set.

3.1.4. *Stopping criteria*

Two stopping criteria were defined to limit the amount of repetitions of the processes of cloning, mutation and selection. The first criterion checks the algorithm's convergence; the proposed methodology has two steps, cloning and selection, that creates and deletes antibodies, respectively. It was a good approximation to confirm its convergence by checking if it is creating and deleting antibodies by the same amount. Nevertheless, there is some

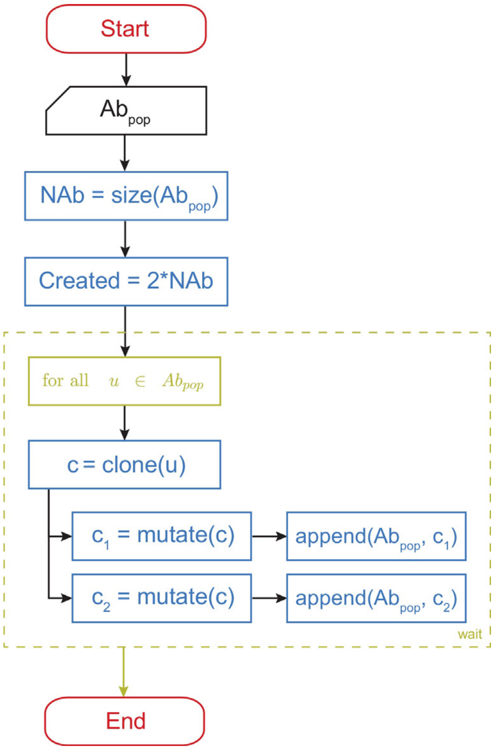

Fig. 5. Cloning mechanism flow chart.

situations where these numbers (created and deleted) are never equal, so the use of another criterion is necessary. The second criterion limits the maximum number of iterations by an integer K.

To show the advantage of the proposed methodology over some types of problems, an experiment with two different scenarios was carried out. The first scenario (Fig. 7) describes a problem with three patterns spread in space but not too mixed with each other. The proposed methodology was put against the ClonALG and Kohonen algorithms individually.

It is important to know that ClonALG and Kohonen algorithms use a fixed number of cluster prototypes (antibodies for ClonALG and neurons for Kohonen). In this example, they were configured to work with 3 and 7 cluster prototypes; the number 7 was chosen to offer a fair competition given the average number of antibodies used by the proposed methodology (Table 2). The maximum number of iterations was set to 50 and 100 simulations were performed for each algorithm.

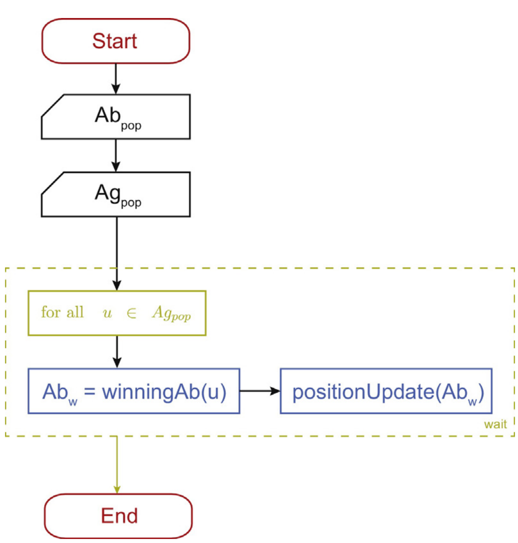

Fig. 6. Mutation mechanism flow chart.

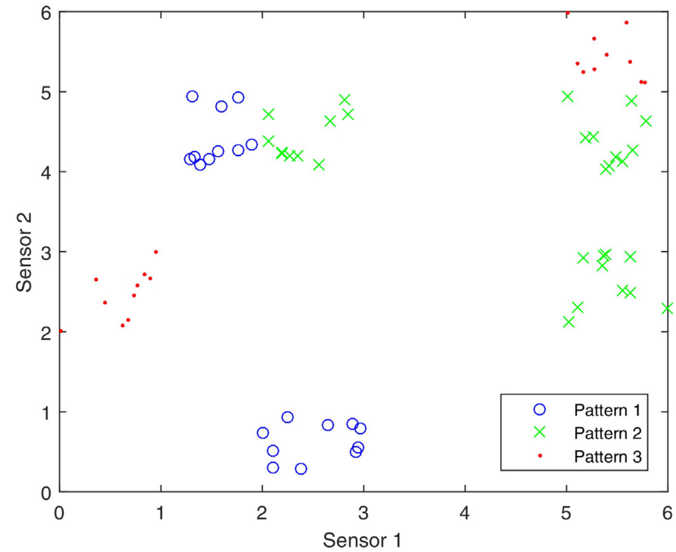

Fig. 7. Scenario 1: spread patterns without mixing.

Table 1
Average performance statistics using the PDBAC's mean in scenario 1.

Algorithm	Average (%)	Best case (%)	Worst case (%)
Hybrid	86.86%	88.93%	77.47%
ClonALG (3 cluster prototypes)	58.22%	73.03%	45.94%
ClonALG (7 cluster prototypes)	67.75%	80.03%	55.15%
Kohonen (3 cluster prototypes)	60.33%	80.25%	50.88%
Kohonen (7 cluster prototypes)	78.68%	88.55%	66.99%

For multiclass problems, using the sample accuracy (correct predictions divided by total of examples) can lead to some erroneous interpretations, because this method does not account for class imbalances in the data set [55]. The alternative was to use a Bayesian approach to estimate the posterior distribution of the balanced accuracy (PDBAC) for a multiclass problem, given in [56]. The results using this metric are presented in Table 1.

It is possible to argue that a simple increase in the cluster prototypes' number would boost these algorithms' performances, but it is important to remember that this is an illustrative problem with low dimensionality and only a few data types. As for the proposed algorithm, some numbers are important to show how it behaves. These numbers are shown in Table 2.

For a simple algorithm convergence analysis of hybrid methodology, the maximum number of iterations was set to 50, the number of performed simulations was 100 and the results are presented in Table 2, illustrating perfectly the second stopping criteria. In some simulations, the algorithm could not stop before the maximum ($K = 50$) number of iterations, but the majority stopped before that by checking the amount of created and deleted antibodies.

Clearly, more tests are needed in order to show the advantage of the proposed methodology. Let us consider another scenario, this time with the patterns strongly mixed. This experiment consists

Table 2
Hybrid methodology behavior.

#	Average	Best case	Worst case
Iterations to convergence	16.65	3	50
Antibodies	6.9	6	8

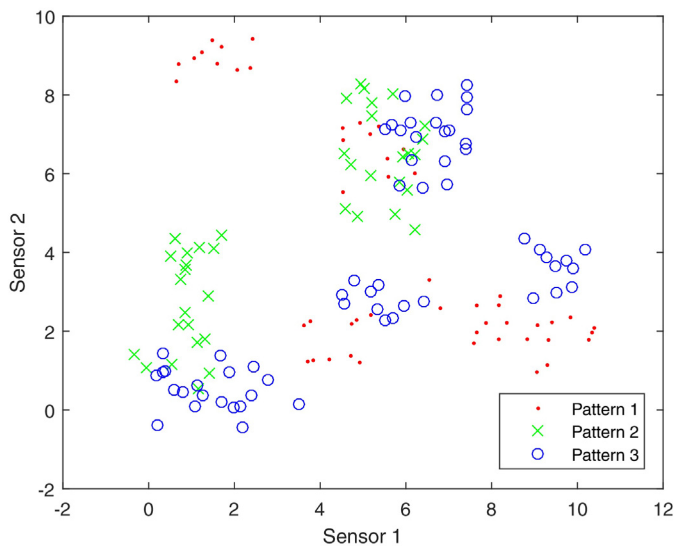

Fig. 8. Scenario 2: spread and heavily mixed patterns.

Table 3
Average performance statistics using the PDBAC's mean in scenario 2.

Algorithm	Average (%)	Best case (%)	Worst case (%)
Hybrid	86.67%	92.28%	80.39%
3-NN	75.56%	87.62%	62.99%
5-NN	74.02%	85.04%	59.24%
Binary tree	69.48%	81.56%	54.51%
Naive Bayes	38.01%	72.89%	23.16%

in testing over a new data set (Fig. 8) other known classification algorithms against the proposed hybrid.

To validate the proposed methodology, a training set containing 70% randomly picked (using a uniform distribution) observations was extracted from the original data set; the other 30% were used to validate the performance of each classifier using PDBAC as the metric, once again. For this experiment, the chosen algorithms were: K nearest neighbors (with K = 3, 5), binary tree and naive Bayes. Table 3 shows that the proposed methodology could beat the simplest forms of some well known classification algorithms.

3.1.5. *Algorithm summary*

The steps of the methodology are given by Algorithm 2.

Algorithm 2. Immune/neural algorithm

```
Data: Dataset
Result: Dataset rate
begin
    while Does not reach the stopping criteria do
        Selection mechanism
        Cloning mechanism
        Mutation mechanism
    end
    return Dataset rate
end
```

3.2. *Generic example*

In this example is considered five types of antigens which has been randomly created. Fig. 9a shows the algorithm initialization, in which the first antibody is created in the space's central position.

Initially, the selection mechanism would take place but, when the number of antibodies is smaller than the number of types of antigens, no antibody will be deleted. Therefore, only the mechanisms of cloning and mutation were depicted. Fig. 9b shows that the first antibody is cloned twice and each clone receives a random mutation using a uniform distribution. The Kohonen neural network is responsible for the second mutation process. In Fig. 9c, the positions of the previously created antibodies were adjusted to a new set of antigens.

Again, the selection mechanism was not depicted because of the lower number of antibodies created so far. Fig. 9d shows the previously explained cloning mechanism over three antibodies. Then, the Kohonen network algorithm was applied in order to increase the affinity of an antibody with a set of antigens (Fig. 9e).

Finally, every antibody that did not recognize at least one antigen must be deleted. The final situation of the training is depicted in Fig. 9f.

4. Case study: faults detection and isolation in the Tennessee Eastman process simulator

The FDI strategy was applied in the Tennessee Eastman (TE) process simulator [47]. The TE process results in two products (G and H) and one (undesired) byproduct F from four reactants (A, C, D and E), according to the following reaction stoichiometry:

$$A(g) + C(g) + D(g) - G(liq) \rightarrow \text{Product 1}$$
$$A(g) + C(g) + E(g) - H(liq) \rightarrow \text{Product 2}$$
$$A(g) + E(g) - F(liq) \rightarrow \text{Byproduct}$$
$$3D(g) - 2F(liq) \rightarrow \text{Byproduct}$$

All reactions are irreversible, exothermic and first order approximately with respect to the reactants concentration. Moreover, production of G requires greater temperature sensitivity to have a higher activation energy. Fig. 10 shows a flowsheet of the Tennessee Eastman (TE) process. There are five unit operations: reactor, condenser, vapor-liquid separator, recycle compressor and stripper. The process summary is given by:

- The gaseous reactants are transformed into liquid products in the reactor;
- The dissolved reactions in liquid phase are catalyzed by a non-volatile catalyst in the gas phase;
- The liquid vapor separator is used to remove the heat from reaction;
- Non condensed components go back to the reactor by a centrifugal compressor;
- Condensed components will for the stripper to remove the remaining reagents;
- Inert and byproduct are discharged from the separator.

The benchmark process faults are presented in Table 4 and all the variables in Table 5 (input process variables) and Table 6 (output process variables). The analysis was carried out with a total of n = 52 process variables, in which the agitation speed (XMV(12)) of the reactor's stirrer was not considered [41]. The data set used (available at http://web.mit.edu/braatzgroup) contains 42 data sets for both test and train of each fault along with two files for test and train of the normal operation condition. Each data set was sampled to a three minute interval, in which the introduction of faults happened at the 8th simulation hour for the testing sets and at the 1st simulation hour for the training sets [57]. The number of observations for each of the training and testing data sets are, respectively, 480 (24 h) and 960 (48 h).

The total number of signals available is reduced in two moments. At the first moment, the correlation matrix R is obtained by the

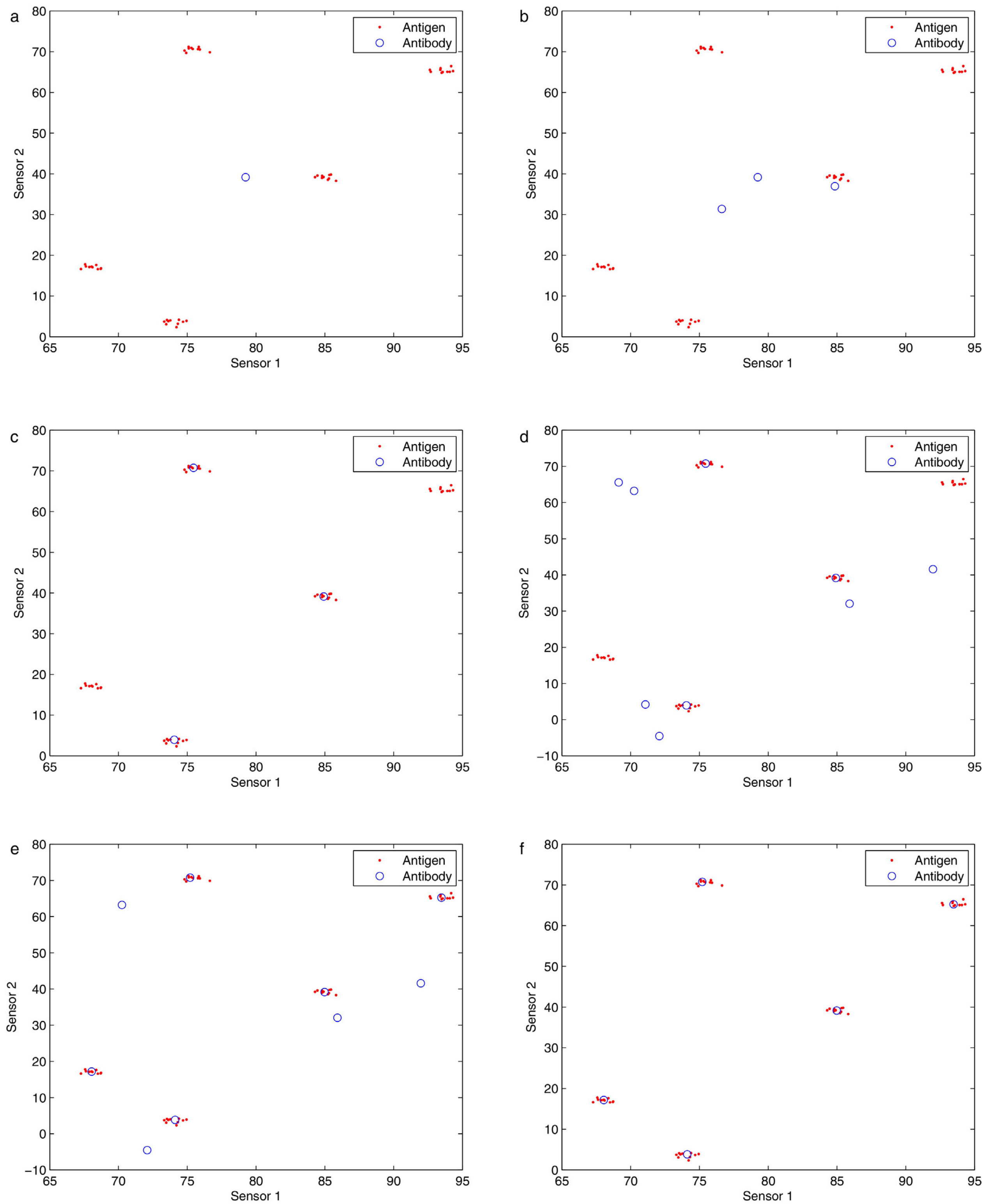

Fig. 9. (a) Initialization with one antibody. (b) Cloning and first mutation of that antibody. (c) Kohonen neural network position adjustment. (d) Cloning and mutating each antibody. (e) Kohonen neural network position adjustment. (f) Removing antibodies who did not recognize any pattern.

set of the 52 measuring variables X. The variables were discarded when considering a matrix of p-values for testing the hypothesis of no correlation. Each p-value is the probability of getting a correlation as large as the observed value by random chance, when the true correlation is zero. If $P(i, j)$ is small, say less than 0.05, then the correlation $R(i, j)$ is significant. This procedure reduced the number of variables used in the FDI system to 37. The removed variables are: XMEAS(13), XMEAS(16), XMEAS(19–22), XMEAS(33), XMV(3), XMV(5–11). At the second moment, seven other variables were discarded due to not presenting any change indication by the fault detection system. These variables are: XMEAS(5–6), XMEAS(12), XMEAS(14–15), XMEAS(17) and XMV(4).

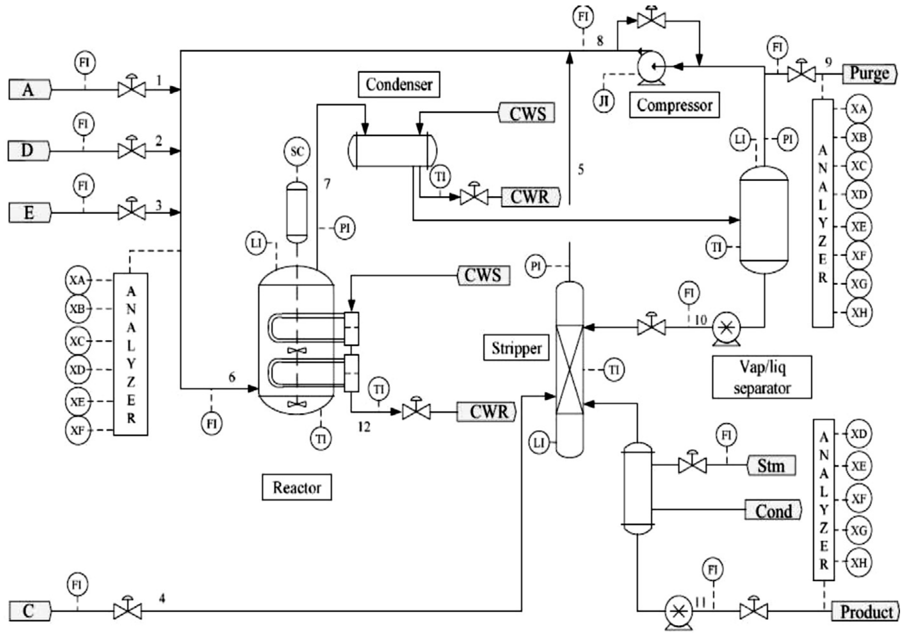

Fig. 10. Flowsheet of the Tennessee Eastman process.

Table 4
Process faults for the Tennessee Eastman process simulator.

Variable	Description	Type
IDV(1)	A/C feed ratio, B composition constant (Stream 4)	Step
IDV(2)	B composition, A/C ratio constant (Stream 4)	Step
IDV(3)	D feed temperature (Stream 2)	Step
IDV(4)	Reactor cooling water inlet temperature	Step
IDV(5)	Condenser cooling water inlet temperature	Step
IDV(6)	A feed loss (Stream 1)	Step
IDV(7)	C header pressure loss – reduced availability (Stream 4)	Step
IDV(8)	A, B, C feed composition (Stream 4)	Random variation
IDV(9)	D feed temperature (Stream 2)	Random variation
IDV(10)	C feed temperature (Stream 4)	Random variation
IDV(11)	Reactor cooling water inlet temperature	Random variation
IDV(12)	Condenser cooling water inlet temperature	Random variation
IDV(13)	Reaction kinetics	Slow drift
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking
IDV(16)		Unknown
IDV(17)		Unknown
IDV(18)		Unknown
IDV(19)		Unknown
IDV(20)		Unknown
IDV(21)	The valve for Stream 4 was fixed at the steady state position	Constant position

The proposed FDI system consists in two main components: the change point detection module (for fault detection) and the immune/neural diagnosis module. The FDI procedure shown in Algorithm 3 is applied to sliding windows and each window is associated with a variable measurement.

Algorithm 3. FDI Algorithm

Data: m sensor measurement time series $y_1(t), y_2(t), \ldots, y_m(t)$ and p antibodies set
Result: Fault classification: F_1 or F_2 . . . or F_p
begin
  for $i = 1, 2, \ldots, m$ **do**
    Obtain the fault evidences for each signal by Algorithm 1
  end
  for $j = 1, 2, \ldots, p$ **do**
    Find the fault F_j by the immune/neural methodology
  end
  return *Fault classification: F_1 or F_2 . . . or F_p*
end

4.1. Results

The data set of the Tennessee Eastman process simulator is divided into two parts: one part to the FDI system training and other to the FDI system test. The training data set was used to generate 730 vectors (or antigens), that were obtained via the fault detection system (change point detection module) considering 10 simulations of each fault, this module reads the sensors available in the Tennessee Eastman benchmark process, extracting windows of the measurements consisting of a set of time series used in fuzzy/Bayesian approach to indicate the probability of change point occurrence along the time series. These results are used in the immune/neural system where the faults were classified using the probability of change point occurrence, resulting in a set of 72 antibodies to classify the 21 faults (immune/neural diagnosis module). The simulations were carried out using MATLAB® Release 2012a.

Table 5
Input process variables for the Tennessee Eastman process simulator.

Description	Variable	Description	Variable
D feed flow (stream 2)	XMV(1)	E feed flow (stream 3)	XMV(2)
A feed flow (stream 1)	XMV(3)	A and C feed flow (stream 4)	XMV(4)
Compressor recycle valve	XMV(5)	Purge valve (stream 9)	XMV(6)
Separator pot liquid flow (stream 10)	XMV(7)	Stripper liquid product flow (stream 11)	XMV(8)
Stripper steam valve	XMV(9)	Reactor cooling water flow	XMV(10)
Condenser cooling water flow	XMV(11)	Agitator speed	XMV(12)

Table 6
Output process variables for the Tennessee Eastman process simulator.

Description	Variable	Description	Variable
A feed (stream 1)	XMEAS(1)	D feed (stream 2)	XMEAS(2)
E feed (stream 3)	XMEAS(3)	A and C feed (stream 4)	XMEAS(4)
Recycle flow (stream 8)	XMEAS(5)	Reactor feed rate (stream 6)	XMEAS(6)
Reactor pressure	XMEAS(7)	Reactor level	XMEAS(8)
Reactor temperature	XMEAS(9)	Purge rate (stream 9)	XMEAS(10)
Product separator temperature	XMEAS(11)	Product separator level	XMEAS(12)
Product separator pressure	XMEAS(13)	Product separator underflow (stream 10)	XMEAS(14)
Stripper level	XMEAS(15)	Stripper pressure	XMEAS(16)
Stripper underflow (stream 11)	XMEAS(17)	Stripper temperature	XMEAS(18)
Stripper steam flow	XMEAS(19)	Compressor work	XMEAS(20)
Reactor cooling water outlet temperature	XMEAS(21)	Separator cooling water outlet temperature	XMEAS(22)
Reactor feed analysis (stream 6) Component A	XMEAS(23)	Reactor feed analysis (stream 6) Component B	XMEAS(24)
Reactor feed analysis (stream 6) Component C	XMEAS(25)	Reactor feed analysis (stream 6) Component D	XMEAS(26)
Reactor feed analysis (stream 6) Component E	XMEAS(27)	Reactor feed analysis (stream 6) Component F	XMEAS(28)
Purge gas analysis (stream 9) Component A	XMEAS(29)	Purge gas analysis (stream 9) Component B	XMEAS(30)
Purge gas analysis (stream 9) Component C	XMEAS(31)	Purge gas analysis (stream 9) Component D	XMEAS(32)
Purge gas analysis (stream 9) Component E	XMEAS(33)	Purge gas analysis (stream 9) Component F	XMEAS(34)
Purge gas analysis (stream 9) Component G	XMEAS(35)	Purge gas analysis (stream 9) Component H	XMEAS(36)
Product analysis (stream 11) Component D	XMEAS(37)	Product analysis (stream 11) Component E	XMEAS(38)
Product analysis (stream 11) Component F	XMEAS(39)	Product analysis (stream 11) Component G	XMEAS(40)
Product analysis (stream 11) Component H	XMEAS(41)		

Table 7
Results of fault isolation.

Description	Correctness (%) Proposed methodology	Correctness (%) PCA	Correctness (%) SVM
Fault 1	100%	87.19%	87.19%
Fault 2	100%	87.5%	85.83%
Fault 3	61%	18.33%	15.21%
Fault 4	68%	72.71%	49.48%
Fault 5	100%	4.06%	50.60%
Fault 6	98%	90.21%	78.85%
Fault 7	97%	89.69%	88.85%
Fault 8	95%	85%	32.19%
Fault 9	36%	20.21%	12.81%
Fault 10	98%	76.15%	22.60%
Fault 11	88%	65.42%	11.88%
Fault 12	100%	85.83%	50.21%
Fault 13	100%	69.06%	21.46%
Fault 14	38%	86.56%	54.90%
Fault 15	100%	23.02%	18.85%
Fault 16	98%	69.48%	12.81%
Fault 17	97%	74.48%	48.02%
Fault 18	85%	59.9%	32.19%
Fault 19	100%	84.06%	46.25%
Fault 20	100%	77.5%	38.96%
Fault 21	100%	85%	8.23%

Table 8
F_1-scores for each fault individually.

	Precision (%)	Recall (%)	F_1-score (%)
Fault 1	97.56%	100%	98.77%
Fault 2	98.04%	100%	99.01%
Fault 3	79.11%	62.5%	69.83%
Fault 4	69.54%	68.5%	69.02%
Fault 5	100%	100%	100%
Fault 6	100%	97.5%	98.73%
Fault 7	100%	96.5%	98.22%
Fault 8	97.42%	94.5%	95.94%
Fault 9	47.43%	43.68%	45.48%
Fault 10	60.18%	99%	74.86%
Fault 11	93.62%	88%	90.72%
Fault 12	100%	99.5%	99.75%
Fault 13	99.5%	100%	99.75%
Fault 14	94.19%	40.5%	56.64%
Fault 15	96.59%	100%	98.26%
Fault 16	93.78%	98%	95.84%
Fault 17	99.48%	96%	97.71%
Fault 18	94.87%	92.96%	93.91%
Fault 19	80.5%	99.49%	88.99%
Fault 20	93.37%	77.89%	84.93%
Fault 21	77.82%	96.5%	86.16%
Average	89.19%	88.14%	87.74%

After the training phase, it was held 100 simulations with the test data set to verify the trained antibodies ability. Previously trained antibodies were able to correctly recognize ~88% of antigens. According to the results, out of 21 faults, 9 have been correctly classified in ~100% of the observations. Also, 19, 17 and 15 faults were correctly classified in over 60%, 80% and 90% of these observations, respectively. The maximum delay for fault detection was two samples. For each fault, the proposed FDI system obtained the results presented in Table 7, with a good correctness performance to deal with fault isolation, when compared with PCA and SVM techniques presented by [42]. The results of the proposed approach were worse than the PCA results in faults 4 and 14. In comparison with SVM, the proposed approach showed worse results in fault 14. Fig. 11 illustrates the histogram results for each fault. Comparing the proposed method with the results showed in [49], we note that the proposed method performs better.

To thoroughly assess the quality of the proposed methodology, another metric has been used for 200 new simulations. This selected metric is the F-score which analyzes the classifier in terms of Recall (completeness) and Precision (purity) [58] and it is computed by their harmonic mean [59] as described in (3):

$$F_\beta = (1+\beta^2)\frac{\text{recall} \times \text{precision}}{(\beta^2 \times \text{recall}) + \text{precision}} \tag{3}$$

with

$$\text{recall} = \frac{TP}{TP + FN} \tag{4a}$$

$$\text{precision} = \frac{TP}{TP + FP} \tag{4b}$$

Notice that the recall and precision can be computed as shown in (4) and, in this experiment, there is no special reasons to favor pre-

Table 9
General confusion matrix for a binary classifier.

	Predicted condition	
True cond.	True positive (TP)	False negative (FN)
	False positive (FP)	True negative (TN)

Table 10
Confusion matrix for the predictions made by 72 trained antibodies.

	Faults 1–21: predicted																				
	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	125	11	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	137	0	0	0	0	28	0	0	0	0	0	0	0	0	0	0	1	34
	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	195	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	5	0	0	0	0	0	193	2	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	189	0	0	11	0	0	0	0	0	0	0	0	0	0
	0	0	31	36	0	0	0	0	83	1	0	0	0	0	3	8	0	2	1	8	17
	0	0	2	0	0	0	0	0	0	198	0	0	0	0	0	0	0	0	0	0	0
Faults 1–21: actual	0	0	0	0	0	0	0	0	0	16	176	0	0	5	3	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	1	199	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0	0	0
	0	0	0	4	0	0	0	0	0	114	0	0	0	81	0	0	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	198	0	0	0	0	0	0
	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	196	0	0	0	0	0
	0	4	0	2	0	0	0	0	0	0	0	0	1	0	0	0	192	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	185	13	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	194	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	33	155	4
	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	193

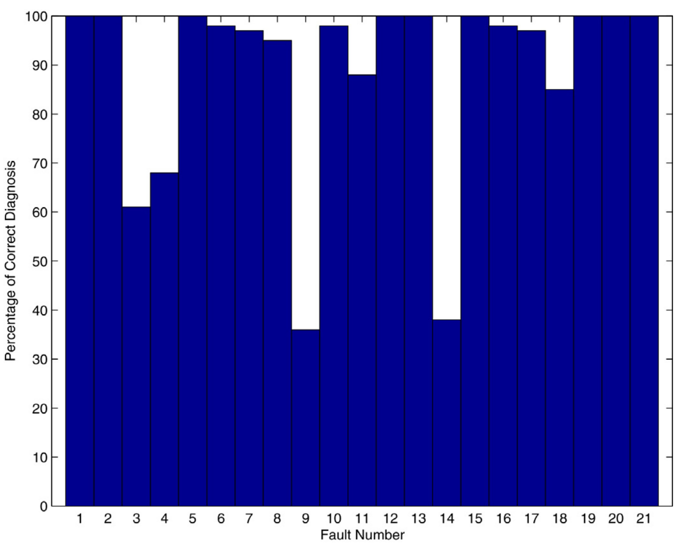

Fig. 11. Percentage of correct diagnosis.

cision over recall thus $\beta = 1$. The F_1-scores for each fault are shown in Table 8.

Further, considering a general confusion matrix for a binary classifier with the form described as in Table 9, for this case study the confusion matrix is given as in Table 10. In order to analyze each fault individually, the problem was made binary in the sense that aside from the subject class, all other classes were reduced to one. This result indicates that the proposed approach has a good performance.

5. Conclusion

In this paper a novel data-based FDI methodology is presented. The proposed FDI system consists in two main modules: (i) a change point indicator based on Metropolis–Hastings algorithm associated with fuzzy set theory that detects changes in time series providing a possible fault evidence (fault detection) and, (ii) a new immune/neural approach which indicates the occurrence of one type of fault (fault classification). This new method for fault classification is based on the association of ClonALG immune systems with the Kohonen neural network. In addition, to combine these two approaches, the stopping criteria and selection mechanism had to be changed to accommodate a non-fixed number of antibodies. For this methodology, no mathematical or statistical models are required decreasing the problem of complexity of implementation.

To show the advantages of this approach, an experiment over two scenarios was carried out. In the first scenario, the proposed methodology performed better than its core algorithms, Kohonen and ClonALG. In another scenario, with heavily mixed data, the approach presented in this paper performed better than the simplest forms of well known classification algorithms. In both scenarios, the metric used to determine the best algorithm was the computation of the mean posterior distribution of the balanced accuracy (PDBAC); this metric was chosen due to the multiclass nature of the problem.

The advantage of the proposed approach over other algorithms in controlled scenarios makes it a good candidate for real-world testing. The chosen environment for this was the well-known Tennessee Eastman benchmark process, which is largely used for tests on fault detection and isolation. The methodology has been successfully applied to this scenario with low implementation complexity and high performance when compared with other methods, e.g. [49,42]. The comparison was made based on the correctness, but to offer more information about the algorithm performance the F-score has also been considered. The F-score, mostly used in binary classification problems, is a more reliable metric as it takes into account the classifier completeness and purity.

Acknowledgements

This work has been supported in part by the Brazilian agencies CNPq, FAPEMIG, and FAPEAM.

References

[1] R. Isermann, P. Balle, Trends in the application of model-based fault detection and diagnosis of technical processes, Control Eng. Pract. 5 (5) (1997) 707–719.
[2] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S.N. Kavuri, A review of process fault detection and diagnosis – part I: quantitative model-based methods, Comput. Chem. Eng. 27 (3) (2003) 293–311.
[3] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, A review of process fault detection and diagnosis – part II: qualitative models and search strategies, Comput. Chem. Eng. 27 (3) (2003) 313–326.
[4] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, K. Yin, A review of process fault detection and diagnosis – part III: process history based methods, Comput. Chem. Eng. 27 (3) (2003) 327–346.

[5] M. Fadali, Observer-based robust fault detection of multirate linear system using a lift reformulation, Comput. Electr. Eng. 29 (2003) 235–243.
[6] B. Yan, H. Wang, H. Wang, A novel approach to fault diagnosis for time-delay systems, Comput. Electr. Eng. 40 (2014) 2273–2284.
[7] V. Puig, A. Stancu, T. Escobet, F. Nejjari, J. Quevedo, R. Patton, Passive robust fault detection using interval observers: application to the DAMADICS benchmark problem, Control Eng. Pract. 14 (6) (2006) 621–633.
[8] R.K. Douglas, J.L. Speyer, Robust fault detection filter design, J. Guid. Control Dyn. 19 (1) (1996) 214–218.
[9] J. Chen, R.J. Patton, Robust Model-based Fault Diagnosis for Dynamic Systems, Kluwer Academic Publishers, Dordrecht, 1999.
[10] M. Hou, R.J. Patton, Optimal filtering for systems with unknown inputs, IEEE Trans. Autom. Control 43 (3) (1998) 445–449.
[11] R.H.C. Takahashi, R.M. Palhares, P.L.D. Peres, Discrete-time singular observers: $\mathcal{H}_2/\mathcal{H}_\infty$ optimality and unknown inputs, Int. J. Control 72 (6) (1999) 481–492.
[12] R.H.C. Takahashi, P.L.D. Peres, Unknown input observers for uncertain systems: a unifying approach, Eur. J. Control 5 (2–4) (1999) 261–275.
[13] C. Jiang, D.H. Zhou, Fault detection and identification for uncertain linear time-delay systems, Comput. Chem. Eng. 30 (2005) 228–242.
[14] X. Dai, Z. Gao, T. Breikin, H. Wang, Zero assignment for robust H_2/H_∞ fault detection filter design, IEEE Trans. Signal Process. 57 (4) (2009) 1363–1372.
[15] R.K. Douglas, J.L. Speyer, $\mathcal{H}_\infty$ bounded fault detection filter, J. Guid. Control Dyn. 22 (1) (1999) 129–138.
[16] S. Ploix, O. Adrot, Parity relations for linear uncertain dynamic systems, Automatica 42 (9) (2006) 1553–1562.
[17] A. Zolghadri, An algorithm for real-time failure detection in Kalman filters, IEEE Trans. Autom. Control 41 (10) (1996) 1537–1539.
[18] B.-L. Zhang, A.-M. Feng, J. Li, Observer-based optimal fault-tolerant control for offshore platforms, Comput. Electr. Eng. 40 (2014) 2204–2215.
[19] J.M.F. Calado, J. Korbicz, K. Patan, R.J. Patton, J.M.G.S. da Costa, Soft computing approaches to fault diagnosis for dynamic systems, Eur. J. Control 7 (2–3) (2001) 248–286.
[20] S.M. El-Shal, A.S. Morris, A fuzzy expert system for fault detection in statistical process control of industrial processes, IEEE Trans. Syst. Man Cybern. C 30 (2) (2000) 281–289.
[22] C.H. Lo, E.H.K. Fung, Y.K. Wong, Intelligent automatic fault detection for actuator failures in aircraft, IEEE Trans. Ind. Inform. 5 (1) (2009) 50–55.
[23] M.F.S.V. D'Angelo, P.P. Costa, Detection of shorted turns in the field winding of turbogenerators using the neural network MLP, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Tucson, 2001, pp. 1930–1935.
[24] G. Rigatos, Q. Zhang, Fuzzy model validation using the local statistical approach, Fuzzy Sets Syst. 160 (7) (2009) 882–904.
[25] M. Bartys, R. Patton, M. Syfert, S. de las Heras, J. Quevedo, Introduction to the DAMADICS actuator FDI benchmark study, Control Eng. Pract. 14 (6) (2006) 577–596.
[26] M. Witczak, J. Korbicz, M. Mrugalski, R.J. Patton, A GMDH neural network-based approach to robust fault diagnosis: application to the DAMADICS benchmark problem, Control Eng. Pract. 14 (6) (2006) 671–683.
[27] C.D. Bocaniala, J.S. da Costa, Application of a novel fuzzy classifier to fault detection and isolation of the DAMADICS benchmark problem, Control Eng. Pract. 14 (6) (2006) 653–669.
[28] M.R. Maurya, R. Rengaswamy, V. Venkatasubramanian, A signed directed graph-based systematic framework for steady-state malfunction diagnosis inside control loops, Chem. Eng. Sci. 61 (6) (2006) 1790–1810.
[29] Y. Dutuit, A. Rauzy, Approximate estimation of system reliability via fault trees, Reliab. Eng. Syst. Saf. 87 (2) (2005) 163–172.
[30] M.R. Maurya, R. Rengaswamy, V. Venkatasubramanian, Fault diagnosis using dynamic trend analysis: a review and recent developments, Eng. Appl. Artif. Intell. 20 (2) (2007) 133–146.
[31] M.F.S.V. D'Angelo, R.M. Palhares, R.H.C. Takahashi, R.H. Loschi, Fuzzy/Bayesian change point detection approach to incipient fault detection, IET Control Theory Appl. 5 (4) (2011) 539–551.
[32] S. Verron, T. Tiplica, A. Kobi, Fault detection and identification with a new feature selection based on mutual information, J. Process Control 18 (5) (2008) 479–490.
[33] C.A. Laurentys, R.M. Palhares, W.M. Caminhas, Design of an artificial immune system based on danger model for fault detection, Expert Syst. Appl. 37 (7) (2010) 5145–5152.
[34] C.A. Laurentys, R.M. Palhares, W.M. Caminhas, A novel artificial immune system for fault behavior detection, Expert Syst. Appl. 38 (11) (2011) 6957–6966.
[35] G.C. Silva, R.M. Palhares, W.M. Caminhas, Immune inspired fault detection and diagnosis: a fuzzy-based approach of the negative selection algorithm and participatory clustering, Expert Syst. Appl. 39 (16) (2012) 12474–12486.
[36] B.G. Xu, Intelligent fault inference for rotating flexible rotors using Bayesian belief network, Expert Syst. Appl. 39 (1) (2012) 816–822.
[37] S. Verron, J. Li, T. Tiplica, Fault detection and isolation of faults in a multivariate process with Bayesian network, J. Process Control 20 (8) (2010) 902–911.
[38] C.A. Laurentys, C.H.M. Bomfim, B.R. Menezes, W.M. Caminhas, Design of a pipeline leakage detection using expert system: a novel approach, Appl. Soft Comput. 11 (1) (2011) 1057–1066.
[39] M.F.S.V. D'Angelo, R.M. Palhares, L.B. Cosme, L.A. Aguiar, F.S. Fonseca, W.M. Caminhas, Fault detection in dynamic systems by a fuzzy/Bayesian network formulation, Appl. Soft Comput. 21 (2014) 647–653.
[40] I.V. de Bessa, R.M. Palhares, M.F.S.V. D'Angelo, J.E.C. Filho, Data-driven fault detection and isolation scheme for a wind turbine benchmark, Renew. Energy 87, Part 1 (2016) 634–645, http://dx.doi.org/10.1016/j.renene.2015.10.061.
[41] T.J. Rato, M.S. Reis, Fault detection in the Tennessee Eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (DPCA-DR), Chemom. Intell. Lab. Syst. 125 (2013) 101–108.
[42] C. Jing, J. Hou, SVM and PCA based fault classification approaches for complicated industrial process, Neurocomputing 167 (2015) 636–642.
[43] D. Wilson, G. Irwin, PLS modelling and fault detection on the Tennessee Eastman benchmark, in: Proceedings of the 1999 American Control Conference, 1999, pp. 3975–3979.
[44] M.F.S.V. D'Angelo, R.M. Palhares, R.H.C. Takahashi, R.H. Loschi, L.M.R. Baccarini, W.M. Caminhas, Incipient fault detection in induction machine stator-winding using a fuzzy-Bayesian change point detection approach, Appl. Soft Comput. 11 (1) (2011) 179–192.
[45] L.N. de Castro, F.J.V. Zuben, Learning and optimization using the clonal selection principle, IEEE Trans. Evolut. Comput. 6 (3) (2002) 239–251.
[46] T. Kohonen, Self-organizing maps, in: Springer Series in Information Sciences, Springer, 2001.
[47] J. Downs, E. Vogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (3) (1993) 245–255.
[48] A. Kroll, H. Schulte, Benchmark problems for nonlinear system identification and control using soft computing methods: need and overview, Appl. Soft Comput. 25 (2014) 496–513.
[49] R. Eslamloueyan, Designing a hierarchical neural network based on fuzzy clustering for fault diagnosis of the Tennessee-Eastman process, Appl. Soft Comput. 11 (1) (2011) 1407–1415.
[50] C. Bedoya, C. Uribe, C. Isaza, Unsupervised feature selection based on fuzzy clustering for fault detection of the Tennessee Eastman process, in: J. Pavón, N.D. Duque-Méndez, R. Fuentes-Fernández (Eds.), Advances in Artificial Intelligence – IBERAMIA 2012, Vol. 7637 of Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, 2012, pp. 350–360, http://dx.doi.org/10.1007/978-3-642-34654-5_36.
[51] C. Lau, K. Ghosh, M. Hussain, C.C. Hassan, Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS, Chemom. Intell. Lab. Syst. 120 (2013) 1–14, http://dx.doi.org/10.1016/j.chemolab.2012.10.005.
[52] M.N. Nashalji, M.A. Shoorehdeli, M. Teshnehlab, Fault detection of the Tennessee Eastman process using improved PCA and neural classifier, in: Advances in Intelligent and Soft Computing, Springer Science, 2010, pp. 41–50.
[53] H. Chen, P. Tiňo, X. Yao, Cognitive fault diagnosis in Tennessee Eastman process using learning in the model space, Comput. Chem. Eng. 67 (2014) 33–42.
[54] J. Timmis, A. Hone, T. Stibor, E. Clark, Theoretical advances in artificial immune systems, Theor. Comput. Sci. 403 (1) (2008) 11–32.
[55] K.H. Brodersen, C.S. Ong, K.E. Stephan, J.M. Buhmann, The balanced accuracy and its posterior distribution, in: 2010 20th International Conference on Pattern Recognition, Institute of Electrical & Electronics Engineers (IEEE), 2010, http://dx.doi.org/10.1109/icpr.2010.764.
[56] H. Carrillo, K.H. Brodersen, J.A. Castellanos, Probabilistic performance evaluation for multiclass classification using the posterior balanced accuracy, in: ROBOT2013: First Iberian Robotics Conference, Springer, 2014, pp. 347–361.
[57] E.L. Russell, L.H. Chiang, R.D. Braatz, Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis, Chemom. Intell. Lab. Syst. 51 (1) (2000) 81–93.
[58] D.M. Powers, Evaluation: from precision, recall and *F*-measure to ROC, informedness, markedness and correlation, J. Mach. Learn. Technol. 2 (1) (2011) 37–63.
[59] C. van Rijsbergen, Information Retrieval, Butterworth, 1979.