Fuzzy Decision Trees as Intelligent Decision Support Systems for Fault Diagnosis

Enrico Zio, Piero Baraldi, and Irina Crenguta Popescu

Energy Department, Polytechnic of Milan, Via Ponzio 34/3, 20133 Milan, Italy
enrico.zio@polimi.it, piero.baraldi@polimi.it, irina.popescu@polimi.it

Summary. In the present work, an intelligent decision support system is proposed to assist the operators in fault diagnosis tasks. The underlying approach relies on a systematic procedure to manipulate measured data of the monitored variables for constructing transparent fuzzy if-then rules associating different patterns of evolution to different faults and anomalies. The resulting fuzzy classification model can then be represented in the form of a Fuzzy Decision Tree. A case study regarding the classification of simulated faults in the feedwater system of a Boiling Water Reactor is presented.

1 Introduction

A fundamental task of fault diagnosis consists in the identification of the occurred fault on the basis of monitored signals representative of the system behavior. Control Room operators are alerted by any meaningful departure of the monitored signals from their steady state and then required to identify the associated fault causes, based on the different patterns of evolution thereby developing. Assisting the operators in this complex diagnostic task has the potential to significantly increase the availability, reliability and safety of the systems and plants, by avoidance of errors that lead to trips or that endanger safety. This is of paramount importance in major hazard plants, such as the nuclear power plants, where the large number of process parameters and the complexity of the system interactions pose great difficulties to the human operators of the control room, especially during abnormal operation and emergency when stress and emotional states come into play [1].

In recent years, many efforts have been devoted to the development of automatic diagnostic techniques for the support of the control room operators in the diagnostic tasks. In particular, techniques based on statistical or geometric methods, neural networks, expert systems, fuzzy and neuro-fuzzy approaches have proven very effective, although often remain "black boxes" as to the interpretation of the physical relationships underpinning the fault classification [2–5].

E. Zio et al.: *Fuzzy Decision Trees as Intelligent Decision Support Systems for Fault Diagnosis*, Studies in Computational Intelligence (SCI) **117**, 187–210 (2008)
www.springerlink.com

In this work, a systematic approach to fault classification is propounded. To obtain the classification model, available pre-classified, labeled data are first fuzzy-clustered using the algorithm proposed in [6]; then, the procedure presented by the authors in [7] is applied to the fuzzy clusters found in order to derive the Fuzzy Sets (FSs) and the Fuzzy Rule Base (FRB) underpinning the classification model.

Once the classification model has been built and its fuzzy rules explicited, every FS in the Universes of Discourse (UODs) of the monitored signals is associated to a symptom of the fault and the FRB of the model is translated into a Symptom Table in which the relationships between faults and symptoms are explicitly laid out [8]. During operation, when some symptoms are detected it is usually difficult to attribute them to a given fault type, given that one fault may cause several symptoms and thus a symptom may describe more than one possible fault. To solve this problem, the relationships between faults and symptoms contained in the FRB are systematically represented in a Fuzzy Decision Tree (FDT). The occurrence of a symptom is measured by the degree of Membership Function (MF) of the associated FS: the degrees of activation of the symptoms are propagated through the FDT to obtain the fuzzy classification of the transient patterns in the different fault classes [8].

The design of the FDT entails the successive consideration of the symptoms. These can be considered in different orders, leading to different structures of the FDT and thus different classification performances. Hence, a combinatorial optimization problem arises with regards to the FDT design: a single-objective genetic algorithm search is devised to find the sequence of symptoms leading to the optimal configuration of the FDT, i.e. that which achieves the maximum classification performance.

The main advantages of the proposed approach are the transparency of the resulting classification model and its visualization to the operator in the form of a Decision Tree (DT).

The Chapter is organized as follows. In Sect. 2, the basic concepts underpinning the fuzzy reasoning are introduced for completeness. Section 3 sketches the steps of the procedure for obtaining a transparent FRB. In Sect. 4, the methodology for constructing the FDT is presented. Section 5 presents a genetic algorithm-based method for optimizing the FDT design. Section 6 reports the case study regarding the classification of faults in a section of the feedwater system of a nuclear Boiling Water Reactor (BWR) [9].

2 Fuzzy Rules for Classification

The classifier proposed in this work is based on a set of fuzzy if-then rules inferred from available data. In this Section, a short description of fuzzy reasoning is provided [10,11]: the content is limited to the general basic concepts, terminologies and notation necessary for completeness and self-consistency of the paper.

The two key elements of fuzzy reasoning are the FRB and the fuzzy inference engine. The former consists of a set of R if-then rules. The generic jth fuzzy rule, $j = 1, \ldots, R$, is made up of a number of *antecedent* and *consequent* linguistic statements, suitably related by fuzzy *connections*:

$$\begin{aligned} &R_j : \ if \ \ (x_1 \ is \ X_{1j}) \ and \ldots \ and \ (x_n \ is \ X_{nj}) \\ &then \ \ (y_1 \ is \ Y_{1j}) \ and \ldots \ and \ (y_m \ is \ Y_{mj}) \end{aligned} \tag{1}$$

The linguistic variables $x_p, \ p = 1, \ldots, n$, are the antecedents, represented in terms of the FSs X_{pj} of the UOD U_{x_p}, with MFs $\mu_{X_{pj}}(x_p)$. The linguistic variables $y_q, \ q = 1, \ldots, m$, are the consequents, represented by the FSs Y_{qj} of the UOD U_{y_q}, with MFs $\mu_{Y_{qj}}(y_q)$. The connective operator *and* links two fuzzy concepts and it is generally implemented by means of a *t-norm*, typically the minimum operator or the algebraic product. The rules of the FRB are joined by the connective *else* and are generally implemented by means of an *s-norm*, typically the maximum operator [10].

The fuzzy inference engine receives the (linguistic) variables which constitute the Fact, viz.,

$$Fact : x_1 \ is \ X'_1 \ and \ \ldots \ and \ x_n \ is \ X'_n \tag{2}$$

where X'_p is a FS on the UOD U_{x_p} of the pth linguistic input variable x_p, and compares it with the antecedents of the rules in the FRB to arrive at the Conclusion, viz.,

$$Conclusion : \ y_1 \ is \ Y'_1 \ and \ \ldots \ and \ y_m \ is \ Y'_m \tag{3}$$

where Y'_q is a FS on the UOD U_{y_q} of the qth output variable y_q.

In the case of fault classification, the antecedents are the monitored variables. A discrete output variable $y_q, \ q = 1, \ldots, c$, is assigned to each fault class to be distinguished [12, 13]. Each output variable is described by two linguistic labels $\{YES, NO\}$, with corresponding singletons FSs Y_q^{NO} and Y_q^{YES} (Fig. 1). In the consequent part of a fuzzy rule representing the jth class, all the output variables $y_q, \ q \neq j$, appear labeled with the FS Y_q^{NO}, except the jth output variable y_j, representing the jth class, which is labeled with Y_q^{YES}:

$$\begin{aligned} &if \ \ (x_1 \ is \ X_{1j}) \ and \ \ldots \ and \ (x_n \ is \ X_{nj}) \ then \\ &\left(y_1 \ is \ Y_1^{NO}\right) \ and \ \ldots \ \left(y_j \ is \ Y_j^{YES}\right) \ \ldots \ and \ \left(y_c \ is \ Y_c^{NO}\right) \end{aligned} \tag{4}$$

This form of the consequents has been chosen because it allows an easier handling of multiple faults [12].

The assignment of an incoming pattern of signals to a class is realized as follows: the fuzzy inference engine (1) receives as Fact the n values of the monitored variables, possibly fuzzyfied to account for measurement imprecision, (2) computes the 'strength' with which each of the R rules in the FRB

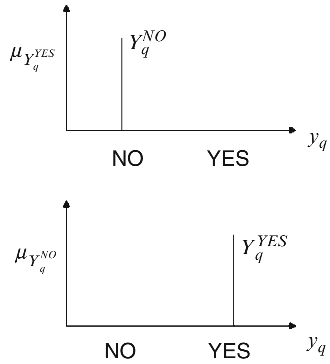


Fig. 1. The two singletons FSs Y_q^{NO} and Y_q^{YES} associated to the qth output variable

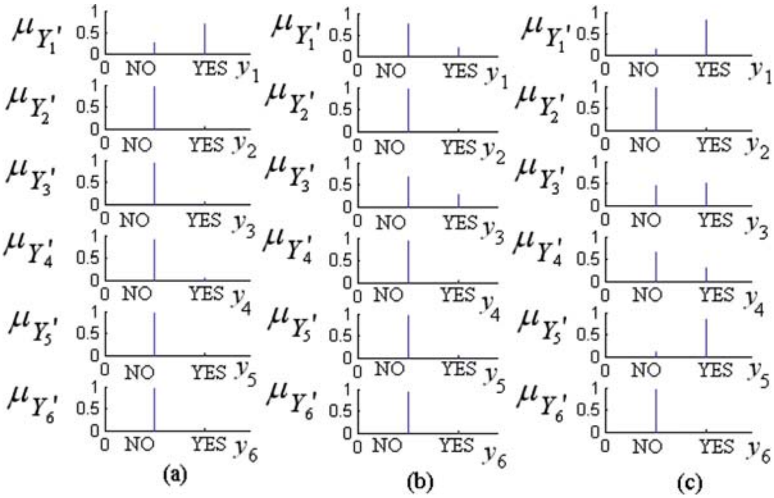


Fig. 2. Example of a classification of a pattern to class 1 (**a**), as atypical (**b**), as ambiguous (**c**)

is activated by the incoming input Fact and (3) properly combines the consequents of the rules, weighed by their respective strengths, to determine the output memberships to the different fault classes [10, 11].

The final assignment of an incoming pattern of signals to a class is conservatively realized as follows: the pattern is assigned to all the classes whose corresponding output y_q, $q = 1, \ldots, c$, has the FS Y_q' with membership value to the linguistic label $\{YES\}$ larger than a threshold γ (chosen equal to 0.6 in the applications which follow). If none of the membership grades to the label $\{YES\}$ is larger than γ, then the pattern is labeled 'atypical'. If more than one membership grade is larger than γ, then the pattern is labeled 'ambiguous'.

Figure 2 shows an example of classification of a pattern to class 1 (a), as atypical (b), and as ambiguous (c).

3 Building a Transparent Fuzzy Rule Base for the Classifier

For ease of presentation, let us consider the four-dimensional artificial classification problem of Fig. 3. The relative data comprise six classes of patterns obtained by random sampling from six different Gaussian distributions and can be assumed to represent the system response signals resulting from six different types of faults to be classified.

For the development of the classification model, a set of N, n-dimensional patterns $\vec{x}_k,\ k = 1, \ldots, N$, pre-classified to c a priori known classes, is assumed available. This information is used to find c geometric clusters as close as possible to the a priori known physical classes, accordingly to the fuzzy clustering algorithm detailed in [13]. The c identified clusters are FSs in the n-dimensional space of the monitored variables, each FS being associated to a different class.

Then, a transparent FRB is constructed from these multidimensional FSs according to the following 3 steps:

1. P*rojection of the n-dimensional fuzzy clusters into n mono-dimensional FSs.* According to the clustering classification algorithm presented in [14], the n-dimensional training patterns $\vec{x}_k,\ k = 1, \ldots, N$ are classified into the c classes with given memberships $\mu_i(\vec{x}_k),\ i = 1, \ldots, c$. This produces c clusters represented by an equal number of n-dimensional FSs, each of which can be projected onto the input variables as follows [15]:

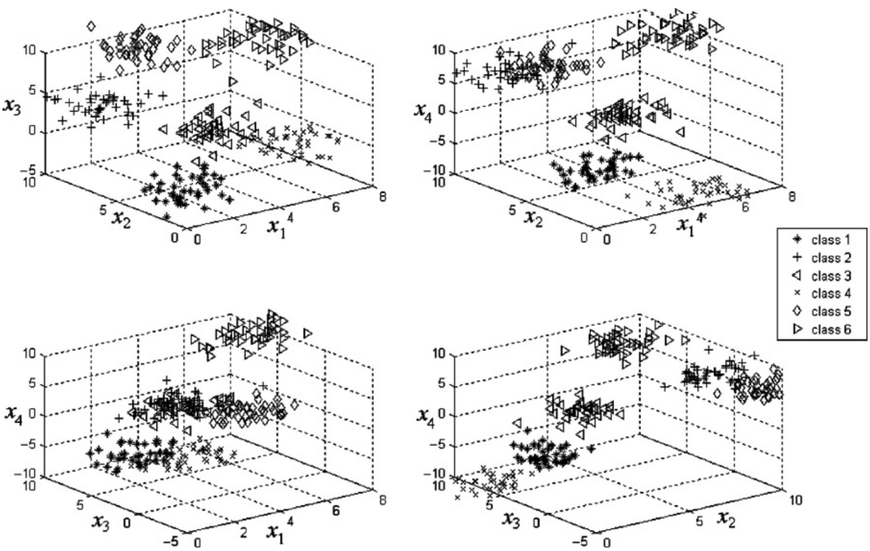


Fig. 3. Four dimensional data set comprised of six classes (for visual clarity, only 240 data out of the 2,400 available have been plotted)

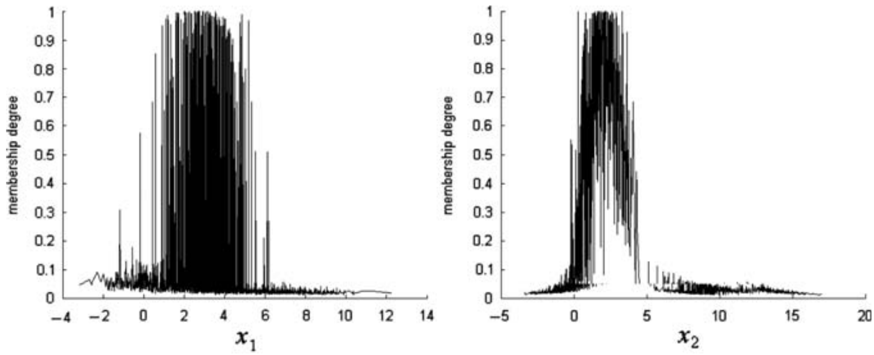


Fig. 4. Projections of the cluster corresponding to class 3 of Fig. 3, onto the UODs of the antecedents x_1 and x_2 (abscissa: antecedent value; ordinate: membership value of the generic kth pattern $\vec{x}_k$ to the cluster projection, $k = 1, \ldots, N$)

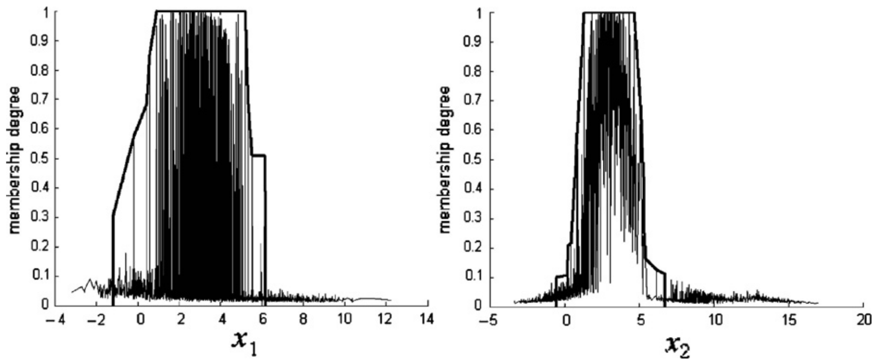


Fig. 5. Approximation of the cluster projections of Fig. 4 into convex FSs

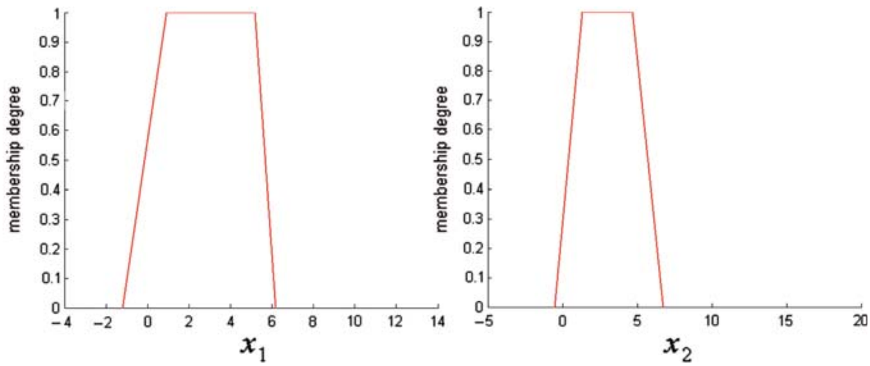


Fig. 6. The trapezoidal FSs corresponding to the cluster projections of Fig. 4

(a) The mono-dimensional MFs of the antecedents FSs are generated by pointwise projection of the membership value $\mu_i(\vec{x}_k)$ onto the antecedent variables UODs [1, 12, 15, 16] (Fig. 4).
(b) The resulting non-convex MFs are transformed into convex MFs (Fig. 5). To do this, starting from the smallest value of the antecedent x_n, only the membership of those values that have a higher membership than the previous one are kept, until the maximum membership value is reached [17]. Then, the same procedure is applied starting from the highest value of the antecedent, until the maximum MF is reached.
(c) The convex FSs are approximated by linear interpolation to MFs of trapezoidal shape (Fig. 6). Before performing the linear interpolation, all membership values under a threshold (chosen to be 0.1 in the present work) are rounded off to 0 and analogously all membership values above an upper threshold (chosen to be 0.9 in the present work) are rounded off to 1.

By so doing, the n-dimensional FS X_i representing the ith cluster is transformed into a fuzzy proposition of the kind:

$$if\ \ (x_1\ is\ X_{1i})\ and\ \ldots\ and\ (x_n\ is\ X_{ni}) \tag{5}$$

where X_{pi} is the projection of cluster i onto the pth input variable, $i = 1, \ldots, c$, $p = 1, \ldots, n$. Obviously, the method is approximate and some information on the cluster is inevitably lost in the projection, due to the decomposition error arising from projecting the multi-dimensional FS into its mono-dimensional constituents [15]. On the other hand, it enables expressing the FRB in a form with a clear and interpretable semantic meaning.

2. *Enforcement of appropriate semantic constraints on the obtained FSs.* To achieve the physical interpretability of the model, semantic constraints are imposed to the FSs obtained in the previous step in an attempt to obtain an "optimal" interface [18]. This is sought through the procedure described below in which each of the FSs modifications required is actually carried out only if the classification performance on the training data is not significantly decreased.
   (a) *Pruning of FSs covering a large portion of the UOD.* Some FSs projections can turn out to be covering great portions of the variables UODs, adding little specific information to the model and over-shadowing more focused FSs. An hypothetical example of FS pruning is shown in Fig. 7. Such sets can be removed from the antecedents of the rules [19]. The criterion for elimination of the FSs widely covering the UOD U_{x_p} is [20]:

$$\beta_o l_{X_{pi}} \geq U_{x_p}; p = 1, \ldots, n;\ i = 1, \ldots, c \tag{6}$$

where $l_{X_{pi}}$ is the width at half-height of the ith FS X_{pi} of variable x_p and $\beta_o \geq 1$ is the so-called overlap parameter. The larger is the value of β_o, the more severe is the pruning criterion.

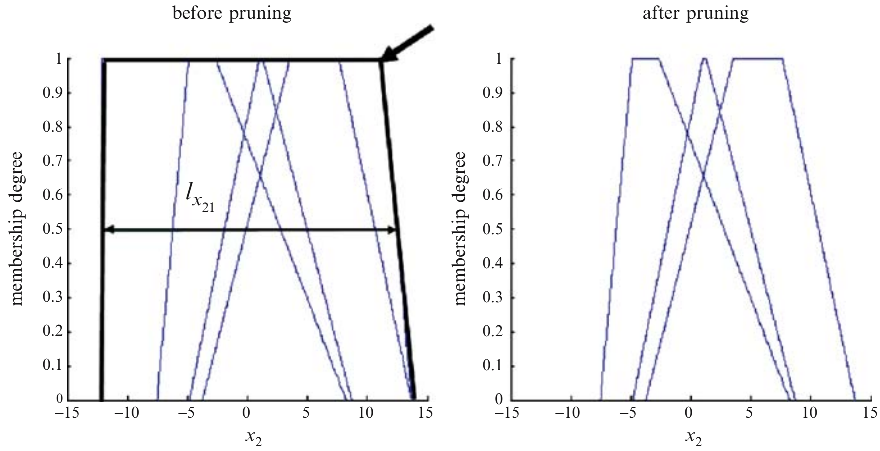


Fig. 7. Overlapping MFs obtained from the clusters projection. The thick solid line in the left Figure denotes the FS to be pruned

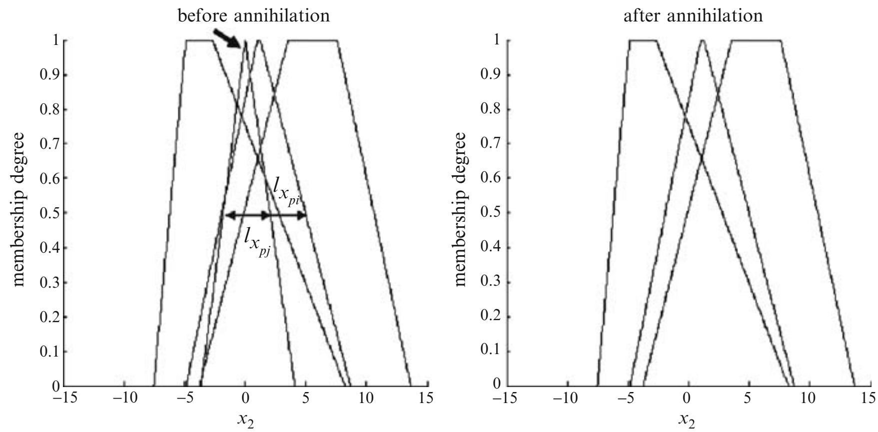


Fig. 8. Annihilation of a narrow FS (*arrow*)

The pruning of a FS modifies only the rules in which the FS appears as antecedent. The modification amounts to canceling from the antecedents the one corresponding to the eliminated FS.

(b) *Addition of FS "nearly zero"*. If the training data do not contain realizations from the class of no faults (stationary state), there is no cluster representing such situation and correspondingly no antecedents FSs and no rules. In this case, a new triangular FS called "nearly zero" is forced in the partition of the UOD U_{x_p} of each variable x_p. The new FS is centered in 0 and the zero-membership vertices are arbitrarily chosen equal to ± 0.1 of the minimum and maximum of the

UOD U_{x_p} of the antecedent variable x_p, respectively. A rule tailored to stationary conditions can then be added to the FRB.

(c) *Annihilation of narrow FSs.* In order to avoid the overlapping among pairs of linguistic terms and the possible consequent semantic inconsistencies, it is necessary to have sufficiently distinct FSs. If a FS X_{pj} were too narrow, for example as in Fig. 8, its contribution is too specific and model transparency is somehow lost. Annihilation of FS X_{pj} is performed if there is a FS X_{pi} for which the following criterion is satisfied [19]:

$$l_{X_{pi}} \mu_{X_{pi}} \left(\frac{z_1 + z_2 + z_3 + z_4}{4} \right) \geq \beta_a l_{X_{pj}};$$
$$i = 1, \ldots, c; j = 1, \ldots, c; i \neq j \tag{7}$$

where $l_{X_{pi}}$ and $l_{X_{pj}}$ are the half-height widths of the FSs X_{pi} and X_{pj} of the same input variable x_p, $\beta_a \geq 1$ is the annihilation parameter and z_s, $s = 1, 2, 3, 4$, stands for the input variable values corresponding to the four vertices of a trapezoidal MF [21–23]. The larger is the value of β_a, the more severe is the annihilation criterion [20, 21].

The FRB is appropriately modified by replacing the canceled FS X_{pj} with the FS X_{pi}.

(d) *Fusion of similar FSs.* If two FSs describing the same variable are sufficiently overlapped, then they should be fused into a single FS because similar [20, 21]. Appropriate measures can be used in order to asses the pairwise similarity of the FSs in the FRB.

The similarity measure Ω of the two FSs X_{pi} and X_{pj} here adopted is given by the ratio between the intersection and the union of their two areas [24]:

$$\Omega(X_{pi}, X_{pj}) = \frac{|X_{pi} \cap X_{pj}|}{|X_{pi} \cup X_{pj}|} = \frac{|X_{pi} \cap X_{pj}|}{|X_{pi}| + |X_{pj}| - |X_{pi} \cap X_{pj}|} \tag{8}$$

If the value of Ω is higher than a pre-established threshold, the two FSs are deemed similar and they are fused (Fig. 9). The four parameters of the new, fused trapezoidal MF will be:

$$z_{fus,s} = \frac{z_i l_{X_{pi}} + z_j l_{X_{pj}}}{l_{X_{pi}} + l_{X_{pj}}}; s = 1, 2, 3, 4 \tag{9}$$

where $z_{fus,s}$ stands for the input variable values corresponding to the four vertices of the trapezoidal MF [20–23] resulting from the fusion and $l_{X_{pi}}$, $l_{X_{pj}}$ are the half-height widths of the FSs X_{pi} and X_{pj}, respectively.

3. *Generation of the fuzzy rules.* The implementation of the previous steps 1 and 2 leads to the generation of a FRB formed by c rules, one for each physical class, of the kind (4).

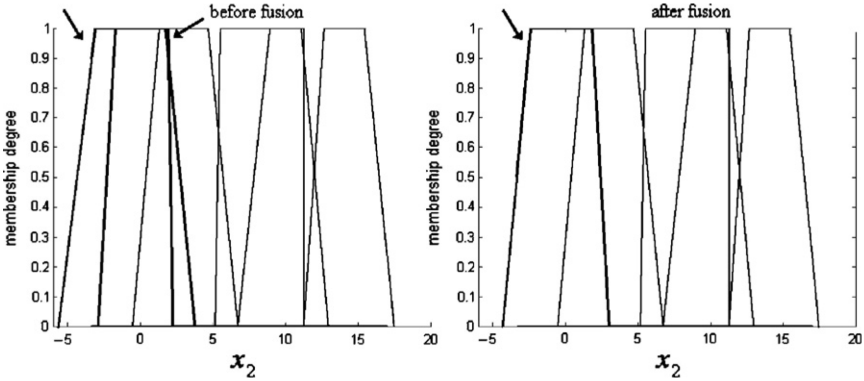


Fig. 9. Fusion of two similar FSs (arrows) corresponding to the projection of class 1 and 2, represented in Fig. 3 with * and +, respectively, onto the second signal UOD

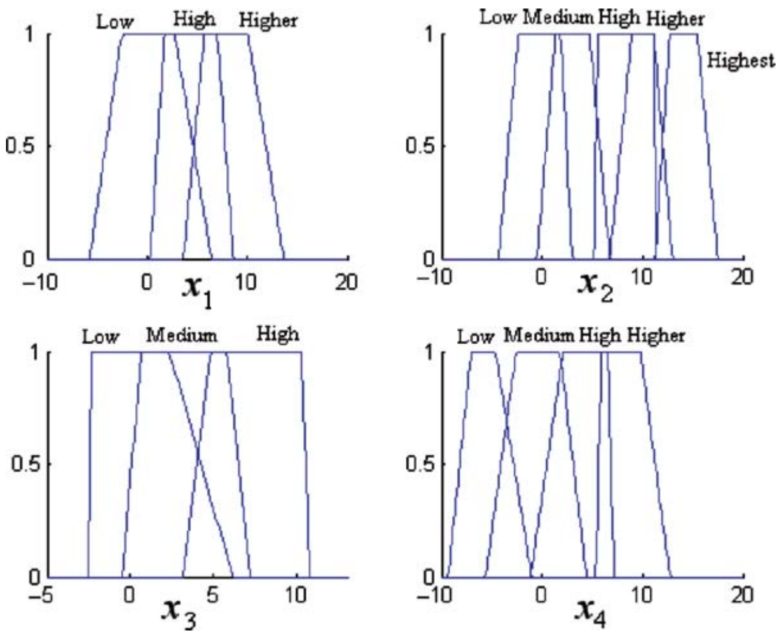


Fig. 10. Final FSs obtained after the projection of the clusters corresponding to the artificial case study

3.1 Application to the Artificial Case Study

Six fuzzy clusters have been identified by applying the algorithm described in [13] to the 2,400 data of Fig. 3.

The application of the procedure just illustrated in Sect. 3 leads to the projection of the six clusters into the FSs of Fig. 10 and to the generation of a corresponding FRB composed of six rules, one for each class (Table 1).

Table 1. The rules of the FRB

Rule		x_1	x_2	x_3	x_4		y_1	y_2	y_3	y_4	y_5	y_6
1	IF	Low S_1	Low S_4	Low S_9	Medium S_{12}	THEN	Yes	No	No	No	No	No
2		High S_2	Medium S_5	Medium S_{10}	Medium S_{13}		No	Yes	No	No	No	No
3		High S_2	High S_6	Medium S_{10}	High S_{13}		No	No	Yes	No	No	No
4		High S_2	Low S_4	Medium S_{10}	Low S_{14}		No	No	No	Yes	No	No
5		High S_2	Higher S_7	Medium S_{10}	Medium S_{12}		No	No	No	No	Yes	No
6		Higher S_3	Highest S_8	High S_{11}	Higher S_{15}		No	No	No	No	No	Yes

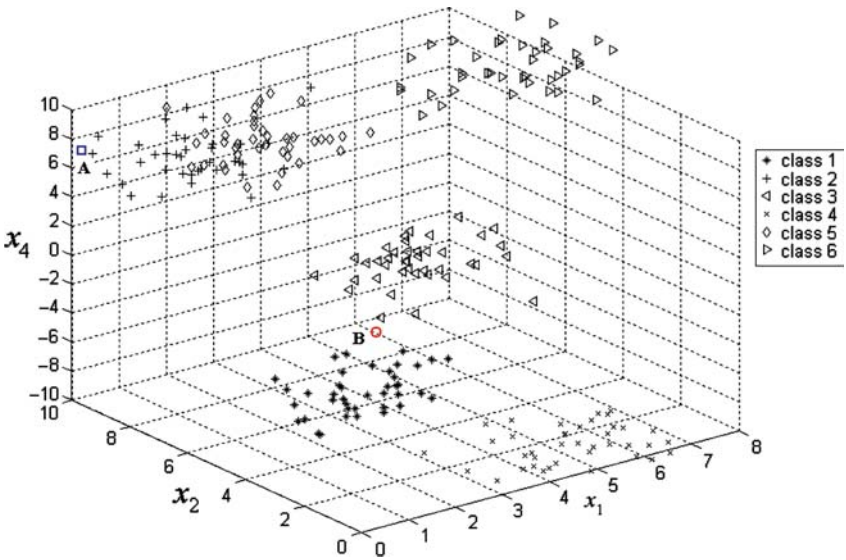


Fig. 11. Example of atypical (A, *square*) and ambiguous (B, *circle*) patterns

Adopting a class membership threshold $\gamma = 0.6$, the classification results for 600 data newly sampled from the underlying six Gaussian distributions are: 85.33% patterns correctly assigned, 7% atypical, 7.67% ambiguous and no pattern assigned to a wrong class.

To picture atypical and ambiguous patterns, consider the patterns A and B represented in Fig. 11 (square and circle, respectively) in the subspace of signals x_1, x_2, x_4. Pattern A belongs to class 2 but is located somewhat far away from the cluster of the other patterns of class 2; for this reason, it is weakly assigned to all six classes with membership values lower than the pre-established classification threshold of 0.6 (Fig. 12a) and, thus, classified as atypical. Pattern B belongs to class 1 but is located at the boundary between classes 1 and 3; for this reason, it is assigned to both classes with membership values above 0.6 (Fig. 12b) and, thus, classified as ambiguous.

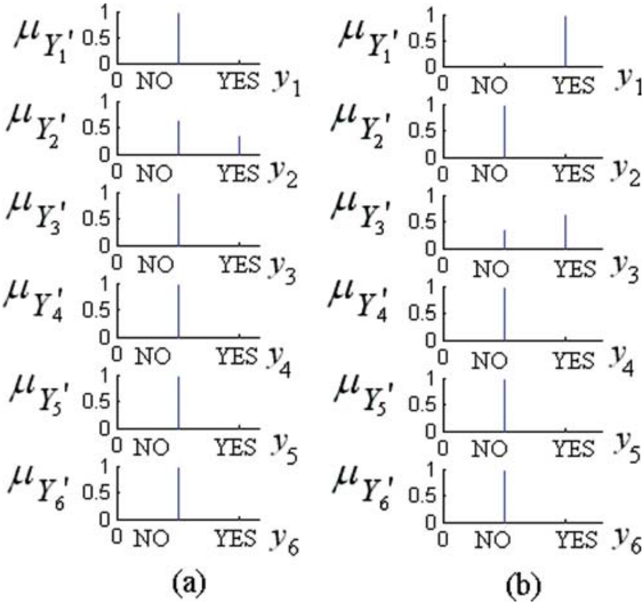


Fig. 12. Classification of an (**a**) atypical pattern and (**b**) ambiguous pattern

4 The Fuzzy Decision Tree

In this Section, the procedure for constructing a FDT starting from the fuzzy rule-based model presented in the previous Section is proposed. In general, DTs are a standard tool used by control room operators for fault classification. Thus, the fact of translating the classifier into a DT bears the great advantage of rendering the diagnostic tool easily received and accepted by the operators.

When a generic fault of class Γ_j, $j = 1, \ldots, c$, occurs, corresponding representative symptoms should be observable by the monitoring system. A symptom associated to the fault of class Γ_j is a deviation, caused by the occurrence of fault Γ_j, of a monitored signal from its reference value. In this work, each one of the FSs obtained in the previous Section represents a deviation and thus a symptom, except those FSs representing steady state conditions of the signals, i.e. the introduced "Nearly Zero" FSs. Correspondingly, the generic FS X_{pj} associated to the pth antecedent in rule j, $p = 1, \ldots, n, j = 1, \ldots, c$, represents a symptom for the class of faults Γ_j.

Notice that the relations between faults and symptoms (signals deviations) are not univocal: one fault may cause several symptoms and in turn one symptom may represent several possible faults. However, if the monitoring system is adequately designed it should be possible to associate to each fault a unique set of symptoms (signals deviations). In our fuzzy classification scheme, these are the FSs representing the signals deviations in the antecedent part

Table 2. Symptom Table: Reference relations between faults and symptoms [25]

Fault class	Symptom type				
	S_1	$\dots$	S_r	$\dots$	S_s
Γ_1	I_{11}	$\dots$	I_{1r}	$\dots$	I_{1s}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
Γ_j	I_{j1}	$\dots$	I_{jr}	$\dots$	I_{js}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
Γ_c	I_{c1}	$\dots$	I_{cr}	$\dots$	I_{cs}

(5) of the corresponding rule. This leads to a Symptom Table such as the one reported in Table 2, where $S_r,\ r = 1, \dots, s$, denotes the generic symptom.

The binary vector $\boldsymbol{\sigma}_{\mathrm{j}} = [I_{j1},\ I_{j2}, \dots, I_{js}]$ represents the reference symptoms vector for fault class $\Gamma_j,\ j = 1, \dots, c$. Each I_{jr} is a binary value that corresponds to the presence or absence of symptom r when fault Γ_j is present, $r = 1, \dots, s,\ j = 1, \dots, c$. For example, $\boldsymbol{\sigma}_1 = (1, 0, 0, 1, 0, 1)$ implies that the occurrence of fault type Γ_1 causes S_1, S_4, and S_6 to appear, among the $s = 6$ possible symptoms.

During operation, an observation vector $\sigma' = (I'_1, I'_2, \dots, I'_s)$ carries the information on the presence or absence of the symptoms, obtained from the measurements of the plant signals. As explained earlier, a symptom is present in the system if its representative measured signal has deviated from its nominal value. For example, a patient has the symptom "fever" if his or her monitored temperature rises to a "high" value, i.e. above 37 °C. However, often in practice the presence or absence of a symptom is affected by uncertainty and ambiguity due to the complexity of the nonlinear signal behaviors associated to the various faults, to the measurement errors of the monitoring sensors and to the imprecise and ambiguous definition of the signal deviation ranges and the associated linguistic labels. In practice then, to a pattern of deviations of the monitored signals measured in correspondence of a given fault, a fuzzy observation vector $\sigma'_f = (\mu'_1, \mu'_2, \dots, \mu'_s)$ can be associated, where $\mu'_r,\ r = 1, \dots,\ s$, is the value of the membership of the FS corresponding to the symptom and gives the degree of presence of symptom S_r in the monitored situation being examined.

Once the fuzzy observation vector σ'_f has been obtained, the problem is to identify which fault type is occurring in the plant. To tackle this problem a systematic procedure for constructing a DT is proposed.

4.1 Decision Tree

The architecture of the tree is obtained by means of a procedure, derived from [25], which applies a hierarchy of Boolean tests to split the sample space into disjoint sections.

Taking into consideration all possible combinations of symptoms, the DT would have 2^s branches given that each of the s symptoms can be either present or absent. On the other hand, only one combination of symptoms corresponds to a given fault: thus, only c of the 2^s tree branches correspond to a class while the remaining $2^s - c$ combinations of symptoms cannot be associated to a class.

For building a smaller, more transparent and easier to interpret DT, two main hypotheses are assumed [25]: (1) if a symptom is indicated as present in the measured observation vector σ', it is certainly present in the system; (2) the presence of a single symptom characteristic of a fault suffices to conclude that the measured pattern of signals belongs to that fault class.

In this context, an "unwanted" symptom is defined as a symptom that, although not present in the system, somehow is present by mistake in the observation vector and a "missing" symptom as a symptom that is not observed although it is present in the system [25]: the first hypothesis can then be called of "impossibility of unwanted symptoms" and the second of "possibility of missing symptoms".

The procedure for building the DT proceeds as follows:

1. A root node is placed at the top of the tree. This node refers to all possible fault classes identified for the system under analysis.
2. A symptom from the Symptom Table is associated to this node.
3. The root node is split into two branches: the left corresponding to the presence of the symptom, the right to the absence of the symptom.
4. The fault classes for which the symptom is present are associated to a node under the left branch. If only one fault class is found to contain the symptom, then the associated node is a terminal leaf of the branch and its identification is guaranteed by the fact that it has been assumed that a symptom that is absent in the system cannot be indicated as present (impossibility of unwanted symptoms hypothesis). The fault class associated to the identified leaf may be also associated to other leaves, at the end of other branches in the tree. This accounts for the possibility that a symptom is not indicated as present by the monitoring system although it actually is (possibility of missing symptoms hypothesis). If more than one fault class are associated to the node characterized by the identified symptom, a new symptom is searched in the Symptom Table and associated to the node in order to differentiate between the identified fault classes. To select the new differentiating symptom, the previous procedure is applied, starting from step 2.
5. The right branch from the root node is further developed by first adding a node associated to all possible fault classes. This node is then treated as a local root node to which the branching procedure is applied starting from step 2.
6. The tree development terminates when all symptoms have been considered and their associated branches developed down to the distinguishing leaves of the individual fault classes.

7. A path through the branches of the tree, from the root node to a leaf, identifies a crisp observation vector σ' of symptoms representative of the fault class associated to the leaf. As pointed out above, different paths may lead to different leaves associated to the same fault class, due to the possibility of missing symptoms.

In operation, the DT gives the correct diagnosis when the measured symptom vector matches completely with the reference symptom vector of a fault class; on the contrary, the diagnosis is conservative in case of a missing symptom, i.e. it is not necessary to have all the symptoms to diagnose the fault.

Finally, in case of unwanted symptoms, the classification is driven by the structure of the tree and the classification will be wrong if the first symptom considered is an unwanted symptom.

From the above it appears that the DT design must be optimized with respect to the order with which the successive symptoms are considered, for optimal classification performance.

4.2 Classification by the FDT

In the realistic case of ambiguity in the presence or absence of a symptom, in correspondence of a given pattern of signal deviations the degree of activation of each symptom S_r, $r = 1, \ldots, s$, is computed from the MF of the corresponding FS. The DT then becomes a FDT and the classification of a given pattern of measured signal deviations is performed by proceeding through all the branches of the tree and computing the MFs to each fault class, at the tree leaves.

The symptoms degrees of activation are then propagated through the FDT according to the rules of FS theory. In particular, the logic operator of negation of a symptom S_r is implemented by $(1 - \mu_{S_r})$ in the right branch corresponding to the absence of the symptom whereas its complement μ_{S_r} is propagated along the left branch associated to its presence (Fig. 13). The

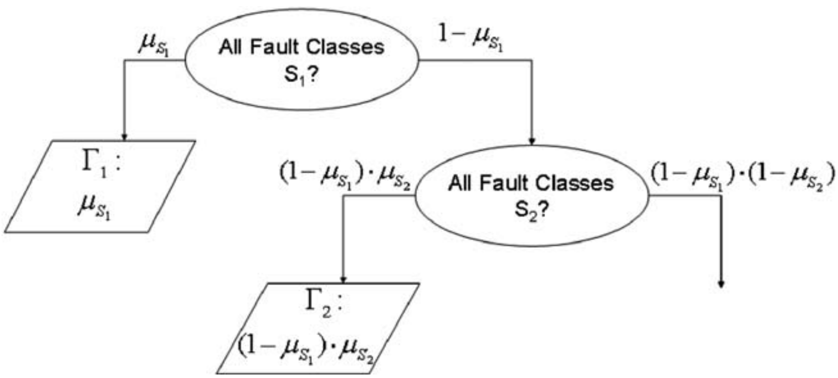


Fig. 13. Propagation of fuzzy information to the DT

connection between two nodes of the tree represents a logic operator of intersection (*and*), here implemented by means of the algebraic product of the membership values.

Finally, since more than one terminal leaf can indicate the same class, the final membership to a given class is computed through the logic operation of union (*or*) of all the leaves associated to that class. The logic operator *or* is here implemented as the MFs sum limited to 1, accordingly to the rules of FS arithmetic.

Differently from the case of crisp symptoms which activate only one terminal leaf, the fuzzy propagation of ambiguous symptoms in the FDT leads to a more realistic classification into different faults with different membership degrees of an ambiguous pattern of deviations, rather than to one definite fault, possibly wrong.

4.3 Application to the Artificial Data

To build the FDT, first each antecedent FS of Fig. 10 is associated to a symptom, resulting in 15 possible symptoms, indicated as S_i, $i = 1, \dots, 15$, in Table 1. This allows the translation of the FRB in the Symptom Table 3.

By applying the steps 1–6 of the procedure for building the DT (Sect. 4.1) on the sequence of symptoms $\Sigma_0 = [S_1; S_2; \dots; S_{15}]$, one obtains the DT reported in Fig. 14.

The quantification of the degree of membership to the different classes is performed as previously described, by propagating through the branches of the tree the degree of activation of each symptom.

The test on the same set of 600 data considered in Sect. 3.1, with membership threshold $\gamma = 0.6$, results in only 40.67% correct classifications to the six fault classes, while 10.5% of the data are considered as atypical, 2.33% as ambiguous and 46.5% are assigned to the wrong class.

The obtained performance is obviously unacceptable and motivates the search for an optimal or near-optimal sequence of symptoms upon which to build the FDT. The objective of the optimization algorithm is to find the sequence of symptoms that leads to the FDT with the best classification performance in terms of percentage of correct classifications. The number of

Table 3. Symptom Table

Class	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
Γ_1	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0
Γ_2	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0
Γ_3	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0
Γ_4	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0
Γ_5	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0
Γ_6	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1

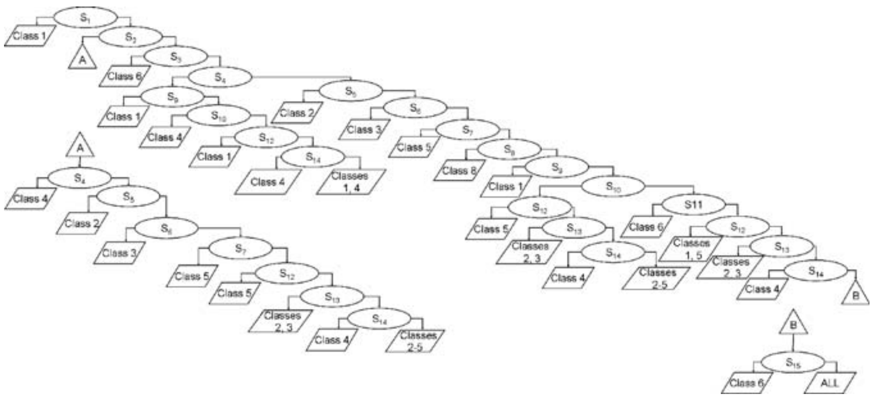


Fig. 14. DT for classification of the artificial data of Sect. 3.1, built with the ordered sequence of symptoms Σ_0

possible sequences of symptoms is 15! ($\sim 10^{11}$). A procedure based on a single-objective genetic algorithm is adopted to solve this combinatorial optimization problem.

5 FDT Optimization by a Genetic Algorithm

In this Section, a procedure based on a single-objective genetic algorithm is carried out for determining the sequence of symptoms to which corresponds the FDT with the maximum classification performance. The genetic algorithm can be seen as performing a wrapper search [26] around the classification algorithm (Fig. 15) in which the symptoms sequence selected during the search is evaluated using as criterion (fitness) the percentage of correct classified data achieved by the FDT itself.

The data and rules of the genetic algorithm search are given in Table 4. These parameters have been established through a systematic procedure of experimentation. The objective (fitness) function to be maximized is the percentage of correct data classifications; the decision variable is the symptoms sequence.

With reference to the artificial case study, each chromosome is made up by 15 genes, one gene for each symptom. The single gene can assume any integer value in $[15, 15]$ that encodes the "swap" position of the symptom along the sequence. An example of a chromosome coding a particular sequence is given in Fig. 16. To decode the chromosome in its corresponding symptom sequence, a 15–steps procedure is performed, one for each gene. At the generic step $i = 1, \ldots, 15$, the ordered sequence Σ_{i-1} and the value k contained in the ith gene are considered: the symptom in the ith position of Σ_{i-1} is then swapped with the symptom in the kth position of the sequence. For example in the first

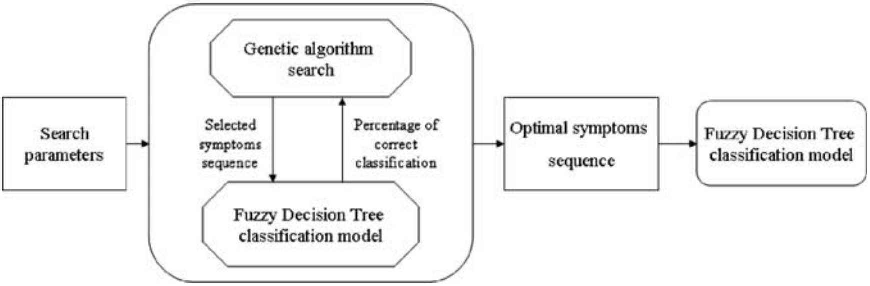
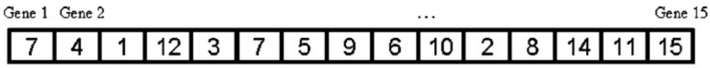


Fig. 15. Single-objective genetic algorithm "wrapper" search

Table 4. GA run parameters

Number of chromosomes in the population	100
Number of generations (termination criterion)	50
Selection	Standard Roulette
Replacement	Children – Parents
Mutation probability	0.01
Crossover probability (one-site)	1



$$\Sigma_0 = [S_1;S_2;S_3;S_4;S_5;S_6;S_7;S_8;S_9;S_{10};S_{11};S_{12};S_{13};S_{14};S_{15}]$$

Step 1 → Gene 1 = 7

$$\Sigma_1 = [S_7;S_2;S_3;S_4;S_5;S_6;S_1;S_8;S_9;S_{10};S_{11};S_{12};S_{13};S_{14};S_{15}]$$

Step 2 → Gene 2 = 4

$$\Sigma_2 = [S_7;S_4;S_3;S_2;S_5;S_6;S_1;S_8;S_9;S_{10};S_{11};S_{12};S_{13};S_{14};S_{15}]$$

...

$$\Sigma_{13} = [S_3;S_{11};S_5;S_{12};S_6;S_8;S_7;S_2;S_1;S_{10};S_4;S_9;S_{14};S_{13};S_{15}]$$

Step 14 → Gene 14 = 11

$$\Sigma_{14} = [S_3;S_{11};S_5;S_{12};S_6;S_8;S_7;S_2;S_1;S_{10};S_{13};S_9;S_{14};S_4;S_{15}]$$

Step 15 → Gene 15 = 15

$$\Sigma_{15} = [S_3;S_{11};S_5;S_{12};S_6;S_8;S_7;S_2;S_1;S_{10};S_{13};S_9;S_{14};S_4;S_{15}]$$

Fig. 16. Example of a chromosome and the corresponding sequence

step of Fig. 16, the value 7 in gene 1 means that the symptom S_1 is placed in position 7 of the sequence and simultaneously the symptom that occupied position 7 is swapped to position 1. This operation is carried out until the 15th gene of the chromosome is worked out, leading to the final sequence:

$$\Sigma_{15} = [S_3; S_{11}; S_5; S_{12}; S_6; S_8; S_7; S_2; S_1; S_{10}; S_{13}; S_9; S_{14}; S_4; S_{15}]$$

Note that this original random design of the chromosome leads to a coherent symptom sequence, i.e. without repetition of symptoms, thus avoiding computationally burdensome chromosome coherence checking a posteriori of its creation.

The optimal sequence found at convergence of the genetic algorithm is:

$$\Sigma_1 = [S_4; S_6; S_7; S_3; S_{12}; S_{10}; S_{13}; S_{15}; S_5; S_1; S_{14}; S_{11}; S_8; S_9; S_2]$$

The FDT built following this sequence increases the fraction of patterns correctly classified from 85.33%, obtained with the FRB classifier, to 91.34%. The percentage of patterns considered atypical is reduced to 5.33% with respect to the previously obtained 7%. Furthermore, the percentage of ambiguous patterns is reduced to 0.33% from the previously obtained 7.6% whereas the percentage of patterns assigned to the wrong class increases from 0 to 3%.

In particular the atypical and ambiguous, patterns A and B of Fig. 11 are now correctly classified. Pattern A is assigned to class 2 with a membership value of 1 due to the symptom S_5 that is characteristic only of this class and that has an activation degree equal to 1 for this pattern. Pattern B is correctly assigned to class 1 due to the degree of activation equal to 1 of the symptom S_9 that is characteristic only for class 1. Thus, the resolution of previously ambiguous and atypical classifications by the FRB is achieved by the FDT thanks to the fact that in the cases considered the activation with high degree of membership of just one characteristic symptom is sufficient for assigning the pattern to the corresponding class. On the other hand, in general the percentage of errors may increase due to the fact that for a given pattern an unwanted symptom activated with a high membership by such pattern, may be placed in the FDT before the representative symptoms for the real class of the pattern.

6 Fault Classification in a Boiling Water Reactor

6.1 Problem Statement

The problem under consideration concerns the identification of a predefined set of faults in a BWR. A set of transients of the monitored signals under different fault conditions have been simulated by the HAMBO simulator of the Forsmark 3 BWR plant in Sweden [9]. Figure 17 shows a sketch of the system [9].

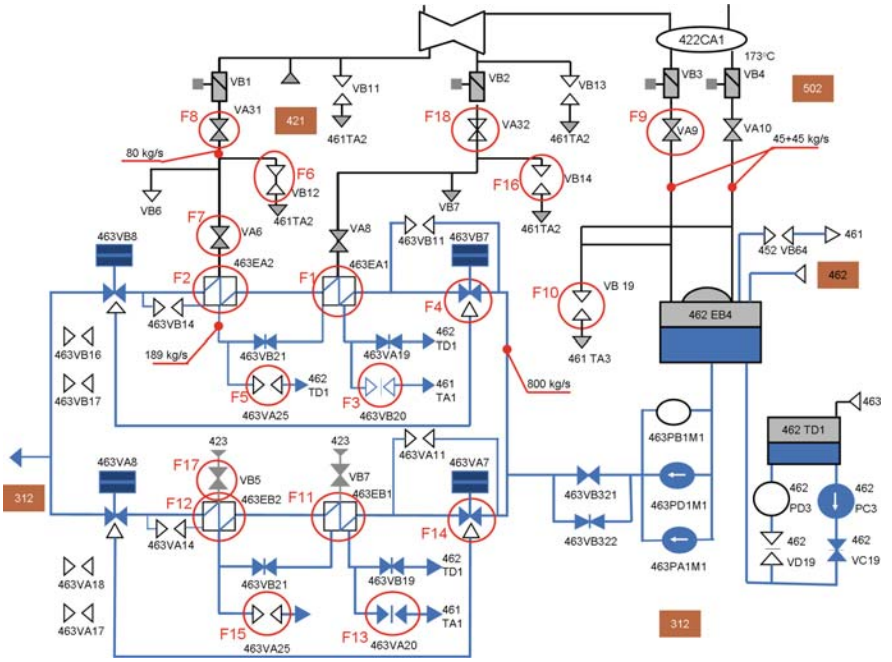


Fig. 17. Sketch of the feedwater system [9]

The considered faults occur in the section of the feedwater system of a BWR where the feedwater is preheated from 169 °C to 214 °C in two parallel lines of high-pressure preheaters while going from the feedwater tank to the reactor. Process experts have identified a set of 18 faults that are generally hard to detect for an operator and that produce efficiency losses if undetected [27]. The $c = 6$ faults regarding line 1 are here considered as the classes to be distinguished by the classification. These are numbered F1–F5 and F7, coherently with the original numbering [9].

For each type of fault, the patterns to be used for building the classification model have been constructed by simulating transients with the plant at 80% of full power, taking values every 6 s from $t_{in} = 80\,s$ to $t_{fin} = 200\,s$.

Among the 363 monitored signals, only $n = 5$ signals have been chosen for the transient classification using the feature selection algorithm proposed in [28]: position level of control valve for preheater EA1 (PLV), temperature of drain 4 before valve VB3 (T1), water level of tank TD1 (WL), feedwater temperature after preheater EA2 (T2) and feedwater temperature after preheater EB2 (T3).

6.2 Application and Results

To test the methodology proposed in this work, the available set of pre-classified patterns is subdivided as follows: 80% have been used for building

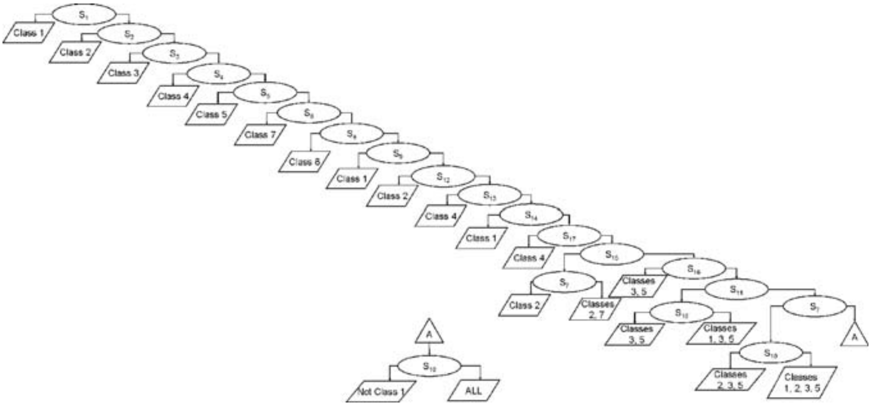


Fig. 18. DT for the classification of BWR feedwater system faults

the diagnostic fuzzy rules and the associated FDT and the remaining 20% have been used for testing the method accuracy.

The application of the fuzzy clustering method presented in this Chapter leads to six clusters, each one corresponding to a different type of fault. Projecting the multi-dimensional clusters onto the UODs of the five input signals and applying the transparency constraints of Sect. 3 for obtaining an optimal partition of the UODs a FRB composed of six rules characterized by five antecedents in the form of (5) is obtained.

As a result, 96% of the test patterns are correctly classified using this more transparent FRB. In particular, all the test patterns are correctly classified except one pattern, which turns out to be characterized by the first input variable x_1 having a value out of the range of the training patterns. This pattern is correctly labelled as atypical by the FRB of the classification model.

As explained in Sect. 4 to build the DT, first each antecedent of the FRB is associated to a symptom. This gives rise to the translation of the FRB in the form of a Symptom Table. On the basis of the Symptom Table the DT is developed (Fig. 18) following the guidelines illustrated in Sect. 4.

Propagating the symptoms fuzzy membership information along the DT of Fig. 18, the test pattern classified as atypical using the fuzzy rule-based classifier turns out now to be correctly assigned to fault class F_1. Notice that, in this case S_1 is a missing symptom but the pattern is still correctly classified, thanks to the hypothesis of possibility of missing symptoms underlying the DT construction procedure.

7 Conclusions

Fault classification is often based on ambiguous information which can be effectively handled within a fuzzy logic framework.

In this context, this Chapter has illustrated a fuzzy-logic based intelligent decision support system to assist the operators in the fault diagnosis tasks. Each step of the proposed methodology is presented with respect to a case study regarding the classification of a set of artificial data randomly sampled from six different Gaussian distributions.

The method is based on a FRB made of one fuzzy classification rule for each fault class. The antecedent FSs in each rule represent the characteristic symptoms (signals deviations) for the corresponding fault class.

A DT is then built to logically structure the uncertain information available. Such DT is quantitatively processed by propagating the degrees of presence of the various possible symptoms.

The classification performance by the resulting FDT is dependent on the order in which the symptoms are considered in the building procedure of the DT. This leads to a combinatorial optimization problem with respect to the construction of the tree. As shown in this work, this problem can be effectively tackled by a genetic algorithm search.

The proposed intelligent decision support system has been tested on a case study regarding the classification of simulated faults in a section of the feedwater system of a BWR. The results obtained are very satisfactory in both classification performance and transparency.

Acknowledgements

The authors wish to thank Drs. Paolo Fantoni and Davide Roverso of the IFE, Halden Reactor Project for providing the transient simulation data.

References

1. F. Klawonn, R. Kruse, Derivation of fuzzy classification rules from multidimensional data. In: G.E. Lasker, X. Liu (eds.), *Advances in Intelligent Data Analysis: The International Institute for Advanced Studies in Systems Research and Cybernetics*, pp. 90–94, Windsor, Ontario, 1995
2. R. Isermann, P. Ballé, Trends in the application of model-based fault detection and diagnosis of technical processes, *Control Engineering Practice*, No. 5, pp. 709–719, 1997
3. S. Leonhardt, M. Ayoubi, Methods of fault diagnosis, *Control Engineering Practice*, No. 5, pp. 683–692, 1997
4. J. Du, M.J. Er, Fault diagnosis in air-handling unit system using dynamic fuzzy neural network, *Proceedings of the sixth International FLINS Conference*, pp. 483–488, 2004
5. G. Goddu, B. Li, M.Y. Chow, J.C. Hung, Motor bearing fault diagnosis by a fundamental frequency amplitude based fuzzy decision system, *IEEE Transaction on Systems, Man, Cybernetics*, pp. 1961–1965, 1998

6. E. Zio, P. Baraldi, Evolutionary fuzzy clustering for the classification of transients in nuclear components, *Progress of Nuclear Energy*, Vol. 46, No. 3–4, pp. 282–296, 2005
7. E. Zio, P. Baraldi, I.C. Popescu, *From fuzzy clustering to a rule-based model for fault classification*, submitted for publication in *IEEE Transactions on Fuzzy Systems Journal*
8. E. Zio, P. Baraldi, I.C. Popescu, *A Fuzzy Decision Tree for fault classification*, submitted for publication in Risk Analysis
9. E. Puska, S. Noemann, 3-d core studies for hambo simulator, *Proceedings of Presentation on Man-Machine System Research*, Enlarged Halden Programme Group Meeting, Vol. 2, September 2002
10. G.J. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Application*, Prentice Hall, Englewood Cliffs, NJ, 1995
11. M. Marseguerra, E. Zio, P. Baraldi, I.C. Popescu, P. Ulmeanu, A fuzzy logic-based for the classification of faults in the pump seals of seals of the primary transport system of a CANDU 6 reactor, *Nuclear Science and Engineering*, Vol. 153, No. 2, pp. 157–171, 2006
12. G. Castellano, A.M. Fanelli, C. Mencar, A fuzzy clustering approach for mining diagnostic rules, *IEEE Conference on Systems, Man, Cybernetics*, 2003
13. E. Zio, P. Baraldi, D. Mercurio, Identification of transients in nuclear systems by a supervised evolutionary possibilistic clustering approach. In: D. Ruan, P. D'hondt, P. Fantoni, M.D. Cock, M. Nachtegael, E.E. Kerre (eds.), *Applied Artificial Intelligence: Proceeding of the 7th International FLINS Conference*, pp. 962–969, Genova, Italy, August 2006
14. E. Zio, P. Baraldi, Identification of nuclear transients via optimized fuzzy clustering, *Annals of Nuclear Energy*, Vol. 32, No. 10, pp. 1068–1080, 2005
15. R. Babuska, H.B. Verbuggen, Identification of composite linear models via fuzzy clustering, *Proceedings of 3rd European Control Conference*, pp. 1207–1212, Rome, Italy, September 1995
16. O. Nelles, A. Fink, R. Babuska, M. Setnes, Comparison of two construction algorithms for Takagi–Sugeno fuzzy models, *7th European Congress on Intelligent Techniques & Soft Computing (EUFIT)*, Aachen, Germany, 1999
17. M. Sugeno, T. Yasukawa, A fuzzy-logic-based approach to qualitative modelling, *IEEE Transaction on Fuzzy Systems*, Vol. 1, No. 1, February 1993
18. J.V. De Oliviera, Semantic constrains for membership function optimization, *IEEE Transactions on Systems, Man, Cybernetics-part A: Systems and Humans*, Vol. 29, No. 1, pp. 128–138, 1999
19. R. Babuska, Data-driven fuzzy modeling: transparency and complexity issues, *Proceedings European Symposium on Intelligent Techniques ESIT'99, AB-01*, Crete, Greece, June 1999
20. B.G. Song, R.J. Marks II, S. Oh, P. Arabshahi, T.P. Caudell, J.J. Choi, Adaptive membership function fusion and annihilation in fuzzy if-then rules, *Proceedings of Second IEEE Conference on Fuzzy Systems*, pp. 961–967, San Francisco, CA, USA, 1993
21. J. Botzheim, B. Hàmori, L.T. Kòczy, Applying bacterial algorithm to optimize trapezoidal membership functions in a fuzzy rule base, *MFT Periodika 2001–06*, Hungarian Society of IFSA, Hungary, 2001
22. C.F. Juang, H.W. Nein, C.T. Lin, Techniques and applications of integrated neural-network based fuzzy logic control systems, *Fuzzy Theory Systems: Techniques and Applications*, Vol. 3, pp. 1265–1295, 1999

23. C.T. Lin, C.S.G. Lee, Reinforcement structure/parameter learning for neural-networks-based fuzzy logic control systems, *IEEE Transaction on Fuzzy Systems*, Vol. 2, No. 1, pp. 46–63, 1994
24. M. Salmeri, M. Re, E. Petrongari, G.C. Cardarili, A novel bacterial algorithm to extract the rule from a training set, *Proceedings of the Ninth IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2000*, Vol. 2, pp. 759–761, San Antonio, TX, USA, May 2000
25. K.O. Kim, M.J. Zuo, Two fault classification methods for large systems when available data are limited, *Reliability Engineering and System Safety*, Vol. 92, No. 5, pp. 585–592, 2007
26. J.M. Keller, M.R. Gray, J.A. Givens, A fuzzy k-nearest neighbor algorithm", *IEEE Transaction on Systems, Man, Cybernetics* SMC-15 (4), pp. 580–585, 1985
27. D. Roverso, Fault diagnosis with the aladdin transient classifier, *Proceedings of System Diagnosis and Prognosis: Security and Condition Monitoring Issues III, AeroSense2003, Aerospace and Defense Sensing and Control Technologies Symposium*, Orlando, FL, USA, April 2003
28. E. Zio, P. Baraldi, D. Roverso, An extended classificability index for feature selection in nuclear transients, *Annals of Nuclear Energy*, No. 32, pp. 1632–1649, 2005