SOFT COMPUTING METHODS FOR
“ON LINE” DIAGNOSTICS OF
NUCLEAR COMPONENTS AND SYSTEMS

Tesi di Dottorato di:
Piero BARALDI

Relatore:  
Prof. Enrico ZIO

Tutor:  
Prof. Marzio MARSEGUERRA

Coordinatore del Corso di Dottorato:  
Prof. Marzio MARSEGUERRA

Correlatore:  
Dott. Davide ROVERSO

XVIII CICLO
Acknowledgments \ Ringraziamenti

First of all, if it has been possible to conceive and realize this PhD research activity, I have to thank Paolo Fantoni who created from scratch the collaboration between the Halden Reactor Project and the LASAR group of the Polytechnic of Milan. Grazie Paolo!

I would like to express my deepest gratitude to Fridtjov Øwre, Øivind Berg and Jan Erik Farbrot for the financial support provided by the Halden Reactor project. I also want to thank all members of the IFE for providing an excellent and inspiring working atmosphere.

I am deeply indebted to Davide Roverso for guiding this work during my 13 Halden’s months and for the many fruitful discussions we had. Grazie Davide!

Vorrei poi ringraziare è il relatore di questa tesi, Enrico Zio. Folle da avere creduto in me all’inizio di questo dottorato, folle da avere creduto in me durante questo dottorato, folle da credere in me per il dopo di questo dottorato. Enrico, la tua triplice follia mi trasmette grande fiducia!
Grazie per avermi comunicato la capacità di stupirmi e di entusiasmarmi facendo ricerca.
Grazie per tutto quello che mi hai insegnato e per come me l’hai insegnato!

Sono poi molto grato al Prof. Marseguerra, ho molto apprezzato l’esperienza, l’allegria e il rigore con cui affronta ogni cosa.

Grazie a Davide e Nicola, lavorare con voi è stato bellissimo. Grazie a Giulio e Max, ad Andrea, Giovanni, Irina, e Matteo, indimenticabili compagni d’ufficio a Milano. Grazie a Janne e Kalin, grandi amici trovati in Norvegia, they do not speak Italian but they will understand these words!

Grazie a Edo e il Cado, i migliori compagni di viaggio possibili, siano i viaggi lunghi tre anni tra le sabbie della ricerca italiana o 9 giorni tra le sabbie africane.

Un grazie lungo 30 anni a mamma e papà.

Grazie a mia moglie, Beatrice, per le cose meravigliose che abbiamo vissuto insieme in questi tre anni. Grazie in particolare per avermi sopportato in Norvegia per 13 mesi e per avere reso in tutti i modi possibili meno dura la nostra lontananza.
Contents

Part I

Nomenclature .................................................................................................................... 9
Acronyms ....................................................................................................................... 10
1 Introduction ............................................................................................................ 11
  1.1 Introduction to the feature selection problem .................................................. 13
    1.1.1 A brief overview on Genetic algorithms .............................................. 14
    1.1.2 Genetic algorithm feature selection .................................................... 15
  1.2 Introduction to the transient classification problem ......................................... 17
    1.2.1 A brief overview on clustering techniques ........................................... 18
    1.2.2 A novel possibilistic classifier ........................................................... 19
  1.3 Structure of the thesis ...................................................................................... 19
2 Feature selection ..................................................................................................... 24
  2.1 An overview on feature selection techniques ................................................. 25
  2.2 GA-based feature selection for transient classification .................................. 27
3 Transient identification ........................................................................................... 30
  3.1 Fuzzy and possibilistic clustering .................................................................. 30
  3.2 The supervised evolutionary possibilistic clustering algorithm for
classification ............................................................................................................. 32
4 Classification of transients in the feedwater system of a boiling water reactor .... 36
  4.1 Feature selection ............................................................................................. 39
  4.2 Transient classification ............................................................................... 41
5 Conclusions ............................................................................................................ 45
References .................................................................................................................. 47
Part II

Report A\textsuperscript{a}: Enrico Zio, Piero Baraldi and Davide Roverso, \textit{Feature selection for nuclear transient diagnostics}, LASAR\textsuperscript{b} and Halden Reactor Project Work Report, January 2006.

Report B\textsuperscript{a}: Enrico Zio and Piero Baraldi, \textit{Clustering techniques and their application to the classification of transients in nuclear power plant}, LASAR\textsuperscript{b} and Halden Reactor Project Work Report, March 2006.

\textsuperscript{a} These reports are being submitted as official Halden Work Report (HWR)

\textsuperscript{b} LASAR (Laboratorio di Analisi di Segnale ed Analisi di Rischio): Laboratory of Signal Analysis and Risk Analysis of the Department of Nuclear Engineering of the Polytechnic of Milan
Part III


PART I
Nomenclature

\[ X = \{ \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N \} \] Set of available patterns \( \tilde{x}_k, k = 1, \ldots, N \)

\[ i = 1, \ldots, c \] index of the class or cluster to which the generic pattern belongs to

\[ c \] number of classes/clusters

\[ \Phi(\cdot) \] classification mapping function

\[ S_O \] complete feature set (formed by all the \( n \) available features)

\[ S_l \] generic feature subset (formed by \( m \) features)

\[ S_f \] optimal feature subset (formed by \( h \) features)

\[ n \] number of all the available features

\[ m \] number of features contained in the generic feature subset \( S_l \)

\[ h \] number of features contained in the optimal feature subset \( S_f \)

\[ l \] index associated to the generic chromosome

\[ F = \{ f(\cdot) \} \] set of \( n_f \) real objective functions to be maximized/minimized by means of genetic algorithms

\[ U = \{ u \} \] decision variables from which the set of objective functions \( F \) depends

\[ V_i \] generic binary transformation vector created by the genetic algorithms

\[ B \] set of \( m \)-dimensional patterns obtained by applying the transformation vector \( V_i \) to \( X \)

\[ X' \] Set of \( n \)-dimensional patterns used for the result validation

\[ B' \] Set of \( m \)-dimensional patterns obtained by applying the transformation vector \( V_i \) to \( X' \)

\( \mu_{ik} \) fuzzy or possibilistic membership of the \( k \)-th pattern to the \( i \)-th cluster

\[ M = \{ \mu_{ik} \} \] set of all the values of membership of all data \( \tilde{x}_k, k = 1, \ldots, N \) to all clusters \( i, i = 1, \ldots, c \)

\( \bar{v}_i \) cluster prototype

\[ V = \{ \bar{v}_1, \bar{v}_2, \ldots, \bar{v}_c \} \] set of all the cluster prototypes
Identity matrix

Mahalanobis matrix relative to cluster $i$

distance between pattern $\bar{x}_k$ and cluster center $\bar{v}_i$

optimal partition of the patterns in $c$ clusters

a priori known partition of the patterns in $c$ classes

a priori known membership of the $k$-th pattern to the $i$-th class

iteration of the evolutionary supervised algorithm

threshold for the classification of a pattern to a class

distance between the a priori known partition $\Gamma'$ and the found partition $\Gamma^*$

distance between the a priori known class $\Gamma'_i$ and the found cluster $\Gamma^*_i$

**Acronyms**

NPP  Nuclear Power Plant

O&M  Operation and Maintenance

GAs  Genetic Algorithms

MOGA  Multi-Objective Genetic Algorithm

NPGA  Niched Pareto Genetic Algorithm

KNN  K-Nearest Neighbour

FKNN  Fuzzy K-Nearest Neighbour

rr  recognition rate

WOLP  Wavelet On Line Pre-processing
1 Introduction

Many efforts have been devoted in recent years to the development of diagnostic techniques in a joint intent to rationally improve the safety and economic performance of existing and future power plants [1, 2]. Indeed, in the United States and most of Europe, the deregulation of the electric power industry begun in the Nineties has put pressure on the nuclear power industry to become more competitive economically, while maintaining the same high standards of safety. In this respect, Figure 1-1 shows the cost breakdown of electricity generation from coal, gas and nuclear fission in the US [3]. The fuel accounts for the major portion of the cost of electricity production from coal and gas: since coal and gas prices are not expected to decline significantly in the future, there is not much room to reduce the cost of electricity production from coal and gas. On the contrary, the fuel accounts for only one third of the total cost of electricity generation from nuclear energy so that significant economic advantages can be gained from more efficient operation and maintenance (O&M) activities. This is exactly what the nuclear industry is aiming at to meet the challenges of a deregulated marketplace.

In particular, in the last 15 years, the availability of nuclear power plants in United States and most European country has increased dramatically, from around 60% to 90%. Some of these improvement has been due to longer fuel cycles, but the major gains have been due to on-line maintenance and improved operation and planning by the utilities, accomplished primarily by training of plant operators and support personnel. However, there are emerging indications that the benefits of training are asymptotically approaching the limit beyond which it cannot improve availability. Indeed further improvement appears to be dependent upon the effective introduction and use of plant operation support systems [2].

In this respect, assisting operators in tasks such as distinguishing normal from abnormal conditions (fault detection) and identifying the causes of process trends (fault diagnosis), has the potential to increase plant availability, reliability and safety by avoiding errors that lead to trips or that endanger the safety of the plant. This is of paramount importance in nuclear power plants where the large number of process parameters and system interactions poses difficulties for the human operators of the

References from [1] to [40] are literature references whereas references from [41] to [49] have been produced within this PhD work.
control room, especially during abnormal operations or emergencies. Moreover, during such situations, individuals can also be affected by stress or emotions which may influence their performance to varying degrees [4].

In this context, the goal of the PhD thesis here presented has been to contribute to the development of efficient diagnostic techniques for the ‘on line’ classification of faults and malfunctioning in nuclear power systems and components. The approach to the classification of the causes responsible of a pre-defined set of transients is performed through a sequence of three steps which include (Figure 1-2):

- the pre-processing of the dynamical data for extraction of the features most characteristic of the system transient behaviour;
- the selection of the features relevant for the classification (feature selection phase);
- the classification of the dynamic patterns into different transient types (transient classification phase).

With respect to the first step, a technique developed in [5] for the extraction of compact wavelet features from multivariate transient data that accounts for the intrinsically dynamic nature of the transients, has been employed.

Steps 2 (feature selection) and 3 (transient classification) have instead been addressed directly in this PhD work, through the introduction of innovative methods and technologies based on soft computing methodologies.

The second step of feature selection is particularly important since irrelevant or noisy features unnecessarily increase the complexity of the diagnostic problem and can degrade modeling performance [6]. Moreover, in modern industrial plants hundreds of parameter are monitored for operation and safety reasons so that expert judgment alone cannot effectively drive the feature selection. The technique proposed in this PhD work to carry out this task combines a genetic algorithm search [7] with a Fuzzy K-Nearest Neighbors (FKNN) classification algorithm [8]. The performance achieved by the latter is used as criterion for comparing the different feature subsets searched by the former (Figure 1.3).

The third task of transient classification is framed as a dynamic pattern classification problem and tackled within a possibilistic clustering approach [9]. Given the complexity and variety of cluster shapes and dimensions which can be expected in the transient classification, an approach based on different Mahalanobis metrics for each cluster has been here embraced. The a priori known information regarding the true
classes which the available patterns belong to is exploited in the classification construction phase to select, by means of a supervised evolutionary algorithm, the optimal Mahalanobis metrics (Figure 1-3).

The work has been performed within a cooperation between the Laboratorio di Analisi di Segnale ed Analisi di Rischio (LASAR, Laboratory of Signal Analysis and Risk Analysis) of the Department of Nuclear Engineering of the Polytechnic of Milan (http://lasar.cesnef.polimi.it/) and the division of Computerised Operation Support Systems (COSS) of the Institute For Energy Technologies (IFE), OECD Halden Reactor Project in Halden, Norway (http://www.ife.no).

1.1 Introduction to the feature selection problem

Feature selection regards the problem of discerning out of the several obtained features those to be used for efficient, early transient classification. The feature selection task is important since irrelevant or noisy features unnecessarily increase the complexity of the diagnostic problem and can degrade modelling performance [6]. Further, for the particular case of nuclear power plants the problem is also critical with respect to the very large number of parameters monitored for operation and safety reasons [41].

Feature selection aims at finding the optimal feature subset that achieves the highest accuracy in the pattern classification task with the minimum number of features. In this respect, given \( n \) features, the problem of selecting a subset of \( m \) relevant ones can be formulated as an optimization problem. Each feature subset is encoded by a binary sequence: for \( n \) features, there are \( n \) bits in the sequence, where each bit indicates whether a feature is present (1) or absent (0) in the subset. Hence, the size of the search space is \( 2^n \), so that the search for the optimal feature subset involves a search among a large number of potential alternatives. Thus, the problem becomes impossible to tackle with an exhaustive search, unless \( n \) is very small [10]. On the other hand, a heuristic approach, based on greedy search strategies such as forward selection or backward elimination, which iteratively add or subtract features, can fail to select features which do poorly alone but offer valuable information together [11]. For these reasons, a genetic algorithm approach which evolves a population of solutions for efficiently searching the feature space has been embraced.
1.1.1 A brief overview on Genetic algorithms

Genetic Algorithms (GAs) are optimization methods aiming at finding the global optimum of a set of real objective functions, \( F = \{ f(t) \} \), of one or more variables, \( U = \{ u \} \), possibly subject to various linear or non-linear constraints. The main characteristics are that the search is conducted \( i) \) using a population of multiple solution points or candidates, \( ii) \) using operations inspired by the evolution of species, such as breeding and genetic mutation, \( iii) \) using probabilistic operations, \( iv) \) using only information on the objectives or search functions and not on their derivatives.

GAs owe their name to their operational similarities with the biological and behavioural phenomena of living beings. After the pioneering theoretical work by John Holland [12], in the last decade a flourishing literature has been devoted to their application to real problems. The basics of the method may be found in Goldberg [7]; some applications in various context are included in Chambers [13].

The terminology adopted in GAs contains many terms borrowed from biology, suitably redefined to fit the algorithmic context. Thus, GAs operate on a set of (artificial) chromosomes, which are strings of numbers, generally sequences of binary digits 0 and 1. If the objective function of the optimization has many arguments (typically called control factors or decision variables), each string is partitioned in as many substrings of assigned lengths, one for each argument and, correspondingly, we say that each chromosome is partitioned in (artificial) genes. The genes constitute the so-called genotype of the chromosome and the substrings, when decoded in real numbers, constitute its phenotype. When the objective function is evaluated in correspondence of a set of values of the control factors of a chromosome, its value is called the fitness of that chromosome. Thus, each chromosome gives rise to a trial solution to the problem at hand in terms of a set of values of its control factors.

The GA search is performed by constructing a sequence of populations of chromosomes, the individuals of each population being the children of those of the previous population and the parents of those of the successive population. The initial population is generated by randomly sampling the bits of all the strings. At each step, the new population is then obtained by manipulating the strings of the old population in order to arrive at a new population hopefully characterized by an increased mean fitness. This sequence continues until a termination criterion is reached. As for the natural selection, the string manipulation consists in selecting and mating pairs of chromosomes in order to groom chromosomes of the next population. This is done by
repeatedly performing on the strings the four fundamental operations of reproduction, crossover, replacement and mutation, all based on random sampling: the parents’ selection step determines the individuals which participate in the reproduction phase; reproduction itself allows the exchange of already existing genes whereas mutation introduces new genetic material; the substitution defines the individuals for the next population. This way of proceeding enables to efficiently arrive at optimal or near-optimal solutions.

With regards to their performance, it is acknowledged that GAs take a more global view of the search space than many other optimization methods. The main advantages are i) fast convergence to near global optimum, ii) superior global searching capability in complicated search spaces, iii) applicability even when gradient information is not readily achievable.

1.1.2 Genetic algorithm feature selection

In the current PhD research work, the Genetic Algorithm search is combined with a Fuzzy K-Nearest Neighbors (FKNN) classification algorithms. The feature subsets selected by the genetic algorithm are compared using as objective functions the recognition rate achieved by the Fuzzy K-Nearest Neighbors classifier [8] and the number, \( m \), of features included in the input patterns. The second objective has been considered in order to drive the GA towards more parsimonious solutions in terms of number of features retained. Thus, the obtained solutions are more easily interpretable from a physical point of view.

In a multi-objective optimization problem, as the one here considered, several possibly conflicting objective functions \( f_i(\cdot), \ i=1,2,...,n_f \), must be evaluated in correspondence of each decision variable vector \( U \) in the search space. The goal is to identify the solution vector \( U^* \) which gives rise to the best compromise among the various objective functions. The comparison of solutions is achieved in terms of the concepts of Pareto optimality and dominance [7, 14]: with reference to a maximization problem solution \( U_a \) is said to dominate \( U_b \) if:

\[
\forall i \in \{1,2,...,n_f\}, f_i(U_a) \geq f_i(U_b)
\]

and

\[
\exists j \in \{1,2,...,n_f\}, f_j(U_a) > f_j(U_b).
\]

The decision variable vectors which are not dominated by any other of a given set are called nondominated with respect to this set; the decision variable vectors that
are nondominated within the entire search space are said to be Pareto optimal and constitute the so called Pareto optimal set or Pareto optimal front.

A comparison of different genetic algorithm techniques in multi-objective optimization problem, carried on within this PhD work [42], has shown that the multi-objective Pareto-based approach is most effective in feature selection problems.

In this approach, reported first in [15], once a population of chromosomes has been created, these are ranked according to the Pareto dominance criterion by looking at the \( n_f \)-dimensional space of the fitnesses \( f_i(U), \ i = 1, 2, \ldots, n_f \). Firstly, all nondominated individuals are identified and rank 1 is assigned to them. Then, these solutions are virtually removed from the population and the next set of nondominated individuals are identified and assigned rank 2; this process continues until every solution in the population has been ranked. Every solution belonging to the same rank class has to be considered equivalent to any other of the same class in the sense that it has the same probability of the others to be selected as a parent for the mating.

During the optimization search, an archive of solution vectors \( U \), each one constituted by a nondominated chromosome and by the corresponding \( n_f \) fitnesses, representing the dynamic Pareto optimality front is recorded and updated [16, 17, 18]; this procedure allows implementation of elitism in the genetic algorithm: every individual in the archive (or a pre-established fraction) is chosen once as a parent in each generation guaranteeing a better propagation of the genetic code of nondominated solutions, and thus a more efficient evolution of the population towards Pareto optimality.

At the end of the search procedure the result of the optimization is constituted by the archive itself which gives the Pareto optimality front.

The efficiency of the search depends on the ability to maintain genetic diversity through the generations so as to arrive at a population of individuals which uniformly represent the real nondominated solutions of the Pareto set [7]. In this respect, during the present PhD work, a “niching” technique has been investigated [43, 44]. A “controlled niched pressure” is applied to spread out the population in the search space so that convergence is shared on different niches of the Pareto front which is thus evenly covered.
1.2 Introduction to the transient classification problem

A fundamental issue for any diagnostic systems is to recognize transients in the first instants of their evolution so as to enable the operator to schedule the proper corrective and mitigation actions to protect from the adverse consequences of the malfunctioning [19]. Other two important requirements for the practical implementation of model-based fault diagnostic systems in nuclear power plants (NPPs) regard the possibility of defining and controlling the boundaries of their utilization and their capability to diagnose a fault independently from the plant operational state before its occurrence [19, 46].

The basis for the identification of a change in the functioning of a system is that different system faults and anomalies lead to different patterns of evolution of the process features [20, 21, 47].

Transient identification is here framed as a pattern classification problem. The task of pattern classification may be viewed as a problem of partitioning of objects (the measured data patterns) into classes (the faults). From a mathematical point of view, a classifier is a mapping function \( \Phi(\cdot) \) which assigns a pattern \( \tilde{x} \) in an \( h \)-dimensional domain \( S_f \subset \mathbb{R}^{h} \) to a given class \( i \). Due to the complexity of transient classification problems in nuclear power plants, it is not possible to determine the analytical expression of the function \( \Phi \), i.e. to individuate the exact boundaries of the classes in the \( S_f \) space.

Hence, one resorts to empirical classification techniques in which the classifier is built through a process of learning based on a set of classification examples. In other words, in the learning phase, labelled patterns belonging to the different classes are utilized to estimate the values of the parameters of the empirical classification function.

Among the empirical classification techniques, over the last 25 years, a large number of artificial intelligence based techniques have been proposed for pattern classification. In particular, artificial neural network and fuzzy logic based systems have been successfully applied. Artificial neural networks are data-processing systems consisting of a number of simple processing elements interconnected in an architecture inspired by the cerebral cortex of the brain. When used as diagnostic tools, properly trained artificial neural networks offer the capability of providing real-time responses and of recognizing faults even when fed with noisy or partially incomplete features [21].
Further, their empirical nature allows modelling complex systems and processes when only input and output data are available [22]. Fuzzy logic systems are also widely proposed in modern diagnostic technologies. Based on the principles of Zadeh’s fuzzy set theory, fuzzy logic provides a formal mathematical framework for dealing with the vagueness of everyday reasoning [23]. As opposed to binary reasoning based on ordinary set theory, fuzzy logic allows for classification into multiple classes with different degrees of membership [24]. Further, measurement uncertainty and estimation imprecision can be properly accommodated within the fuzzy logic framework.

In the present PhD research work, an alternative approach to pattern classification based on supervised clustering techniques has been developed.

\[ \text{1.2.1 A brief overview on clustering techniques} \]

Cluster analysis is one of the main techniques in pattern classification. Given a set of \( N \) multivariate patterns \( X = \{ \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N \} \), cluster analysis aims at identifying any relationship existing among them by organizing them into \( c < N \) clusters (hereafter also called groups) such that the patterns belonging to a given cluster are more similar to each other than are those belonging to different clusters.

The conventional (hard) clustering methods constrain each pattern to belong to one cluster only. In practice, however, the clusters may not be completely disjoint and patterns could be classified as belonging to one cluster almost as well as to another. In these cases, one must assign to each pattern a set of membership values, one for each class identified. The implication of this is that the class boundaries are not ‘hard’ but rather ‘fuzzy’ and the representation of the data structure can be more accurately handled by fuzzy or possibilistic clustering methods [24-27].

In this PhD work, the possibilistic clustering algorithm proposed in [26] has been exploited. This algorithm is based on an iterative procedure of minimization of an objective function that attains its minimum for a partition which maximally separates the clusters. In this approach, the only constraint that the objective function must satisfy is that the membership values do represent degrees of compatibility, or possibility, i.e. they must lie in \([0,1]\). Compatibility (typicality) is measured with respect to the cluster prototypical members [9, 24] found by the algorithm itself. In this view, the memberships of representative patterns are high, while unrepresentative (atypical) points bear low membership to all clusters.
1.2.2 A novel possibilistic classifier

When cluster analysis is to be used within the framework of classification of data belonging to physically different classes an obvious desiderata is that the clusters eventually identified be indeed representative of the underlying physical differences.

In this respect, in the present PhD research work, the feasibility of building a classifier by means of a supervised evolutionary procedure [28, 48, 49] applied to the possibilistic clustering algorithm has been discussed [46]. To tackle the classification problem with a supervised technique we assume to have available a partition \( \Gamma' \) of \( N \) “training” labelled data, \( \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N\} \) in \( c \) subsets. The evolutionary algorithm searches for the optimal Mahalanobis metrics to be used by the clustering algorithm so as to achieve clusters as close as possible to the real physical classes that permit, during operation, an accurate classification of the transients. Moreover, the possibilistic classifier is capable to avoid misclassification of unknown transient patterns given that it can assign a pattern with low membership values to all the clusters.

1.3 Structure of the thesis

The thesis comprises three parts. Tables 1-1, 1-2 and 1-3 summarize the thesis structure. Part I, subdivided in five Sections, introduces the transient diagnostic problem in further details, illustrates the methodological approaches developed in this PhD work and proposes a complete application to a transient classification problem. Part II contains two extensive reports on feature selection (Report A) and on transient classification (Report B). Part III is a collection of six papers published as a result of the work and which the reader is referred for further details to.

Section 2 of Part I regards feature selection. In particular, Section 2.1 gives an overview on classical feature selection techniques making extensive reference to part II (Report A, Section 2). The feature selection technique introduced within the PhD work is depicted in Section 2.2 of Part I and discussed in more details in Part II (Report A, Section 3) and in Part III (Papers I-III).

Section 3 of Part I tackles the transient classification problem. In particular, Section 3.1 introduces the basic concept behind fuzzy and possibilistic clustering and Section 3.2 proposes a novel algorithm for the transient classification. Report B in Part II describes more accurately clustering algorithms (Section 2) and the developed classification algorithm (Section 3).
Section 4 of Part I considers an application of the developed techniques to a problem of transient classification in the feedwater system of a boiling water reactor. More details on the feature selection application are given in Part II (Report A, Section 4,5,6) and in Part III (papers I,II,III) whereas other diagnostic applications of the developed classification algorithm are presented in Part II (Report B, Section 4) and Part III (papers IV, V,VI).
Table 1-1. Structure of the thesis

<table>
<thead>
<tr>
<th>Topic</th>
<th>Part I</th>
<th>Part II</th>
<th>Part III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature selection</td>
<td>2</td>
<td>A</td>
<td>I-III</td>
</tr>
<tr>
<td>Transient classification</td>
<td>3</td>
<td>B</td>
<td>IV-VI</td>
</tr>
</tbody>
</table>

Table 1-2. Structure of the thesis for what concern the feature selection topic

<table>
<thead>
<tr>
<th>Topic</th>
<th>Part I</th>
<th>Part II</th>
<th>Part III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview on classical techniques</td>
<td>2.1</td>
<td>Section 2</td>
<td></td>
</tr>
<tr>
<td>Proposed feature selection technique</td>
<td>2.2</td>
<td>Section 3</td>
<td>I-III</td>
</tr>
<tr>
<td>Case study</td>
<td>4.1</td>
<td>Sections 4,5,6</td>
<td>I-III</td>
</tr>
</tbody>
</table>

Table 1-3. Structure of the thesis for what concern the transient classification topic

<table>
<thead>
<tr>
<th>Topic</th>
<th>Part I</th>
<th>Part II</th>
<th>Part III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview on classical clustering techniques</td>
<td>3.1</td>
<td>Section 2</td>
<td></td>
</tr>
<tr>
<td>Proposed classifier</td>
<td>3.2</td>
<td>Section 3</td>
<td>IV-VI</td>
</tr>
<tr>
<td>Case studies</td>
<td>4.2</td>
<td>Section 4</td>
<td>IV-VI</td>
</tr>
</tbody>
</table>
Figure 1-1: Cost Components of U.S. Electricity Production [1]

Figure 1-2. Components of a diagnostic system
Figure 1-3. Pictorial view of the issues considered in the current PhD research work and of the developed techniques
2 Feature selection

The diagnostic task may be viewed as a problem of partitioning of objects (the measured data patterns) into classes (the faults). From a mathematical point of view, a classifier is a mapping function $\Phi(\cdot)$ which assigns an object $\tilde{x}$ in an $h$-dimensional domain $\Omega \subset \mathbb{R}^h$ to a given class $i$. If one knew the exact expression of $\Phi(\cdot)$, the question of which features of $\tilde{x}$ to use would not be of interest. In fact, in such situation adding features does not decrease the accuracy of the classifier, and hence restricting to a subset of features is never advised. However, as it is often the case in engineering, it is not possible to determine the exact analytical expression of the function $\Phi(\cdot)$ due to the complexity of the systems under analysis. Hence, one resorts to empirical classification techniques in which the classifier is built through a process based on a set of classification example pairs $\{(\tilde{x}, i)\}$, each one constituted by a pattern $\tilde{x}$ in the feature space labelled by the corresponding class $i$.

In practice, the number of measured features is quite large. At least various reasons call for a reduction of this number for use in the classification model. First of all, irrelevant, non informative features result in a classification model which is not robust [29, 30]. Second, when the model handles many features, a large number of observation data are required to properly span the high-dimensional feature space for accurate multivariable interpolation [11, 29]. Third, by eliminating unimportant features the cost and time of collecting the data and developing the classifier can be reduced [11, 29]. Finally, reducing the number of selected features permits to achieve more transparent and easily interpretable models [6].

Given a labelled dataset, the objective of feature selection is that of finding a subset of the features such that the classifier built on these features classifies the available data with the highest possible accuracy [6].

Notice that different classification algorithms may require different feature subsets to achieve highest accuracy, so that a universal optimal feature subset does not exist.

Finally, other requirements are often added to the feature selection objective of maximum classification accuracy, e.g. the reduction of the number of features for reducing the computational and data storage burdens.
2.1 An overview on feature selection techniques

Feature selection involves conducting a search for an optimal feature subset in the space of possible features. The inclusion or not of a feature in the subset can be encoded in terms of a binary variable which takes value 1 or 0, respectively. For \( n \) features, the size of the binary vector search space is \( 2^n \). Thus, an exhaustive search is impractical unless \( n \) is small.

Each feature subset selected during the search must be evaluated with respect to the given objective functions, e.g. classification performance and number of features.

Several methods of feature selection have been proposed. They are usually classified into two categories: filter and wrapper methods [6].

In filter methods, the feature selector algorithm is independent of the specific algorithm used in the classification and it is used as a filter to discard irrelevant features, a priori of the construction of the classifier. A numerical evaluation function is used to compare the feature subsets proposed by the search algorithm. The subset with the highest value of the evaluation function is the final feature set which feeds the algorithm for the classification.

The evaluation functions are usually of two types:

- those which apply distance metrics in the feature space to measure the separability between classes, e.g. the classifiability evaluation function firstly proposed by [9] and then extended to the transient classification case within this PhD work [41].
- those based on the rationale that a good feature subset contains features highly correlated with the class label but uncorrelated with the other features of the subset, e.g. those based on the concept of mutual information [31].

Contrary to filter methods, in wrapper methods the feature selector behaves as a "wrapper" around the specific algorithm used to construct the classifier whose performance is used to compare the different feature subsets [6].

The filter approach is generally computationally more efficient than the wrapper one because for each feature subset of trial, the computation of an evaluation function is less time consuming than the development of a complete classification model. Indeed, a high number of feature subsets are tested during the search for the optimal and the time consumption of a wrapper approach depends mainly from the time necessary for the
development of the classifier and the subsequent classification of the patterns to test its performance. Hence, for many practical applications the wrapper approach is feasible only if the classifier is a fast-computing algorithm, e.g. the K-Nearest Neighbour (KNN) [32] algorithm or its fuzzy extension (FKNN) [8]. On the other hand, wrapper approaches are more performing than the filter ones since the former ensure the selection of the features more suitable for the specific classification algorithm used, whereas the latter totally ignore the effects of the selected feature subspace on the performance of the classifier that will actually be used.

With respect to the search algorithms, three approaches are commonly adopted: complete, heuristic and probabilistic [6].

In the complete approach, the properties of a pre-defined evaluation function are used to prune the feature space to a manageable size, thus avoiding that the complete search is also exhaustive [33]. Only some evaluation functions give rise to a search that guarantees the optimum feature subset selection without being exhaustive.

The heuristic approach does not guarantee that the best feature subset is achieved, but is less time consuming than the complete one and may be employed in combination with any evaluation function [41]. At present, the most employed heuristic methods are greedy search strategies such as the sequential forward selection (SFS) or the sequential backward elimination (SBE) “hill climbing” methods, which iteratively add or subtract features and at each iteration the evaluation function is evaluated. The forward selection refers to a search that begins with no features and at each step a feature is added to the subspace; on the contrary, the backward elimination refers to a search that begins with the n-dimensional feature set and at each step a feature is removed. At each step, the choice of which feature to add or remove is driven by its effect on the classifiability function in the direction of climbing towards its maximum value. The hill-climbing search is usually stopped when adding or removing new features does not increase the value of the classifiability function or when the number of features has reached a predefined threshold.

The hill climbing methods suffer from the so called “nesting effect”: if the features added cannot be removed, a local minimum of the evaluation function may be found. To reduce this effect, it is possible to use the so called plus-\(l\)-take-away-\(r\) method (PTA) [6]. In this method, after \(l\) steps of the forward selection, \(r\) steps of the backward elimination are applied so as to allow escaping from local minima. Still, there is no guarantee of obtaining the absolute optimum.
The probabilistic approach is based on population-based metaheuristics guided by fittest solutions, such as the genetic algorithms, presented in this PhD work, or on methods like simulated annealing and tabu search algorithms [34]. In the next Section, a genetic algorithms-based method is introduced.

2.2 GA-based feature selection for transient classification

In the present PhD work, the wrapper scheme is adopted and a Multi-Objective Genetic Algorithm (MOGA) for searching the optimal feature subset upon which to perform the diagnostics of nuclear transients is presented.

The total number of $n$-dimensional pre-labelled available data are partitioned into a set (hereafter denoted by $X$) to be used for the feature selection task and a separate set (hereafter denoted by $X'$) to be used for validating the performance of the classifier resting upon the optimal feature subset selected.

The structure of the chromosome is straightforward [42]. For $n$ features, the size of the chromosome is $n$ bits and each bit of the chromosome is associated with a feature (Figure 2-1): if the $r$-th bit equals 1, then the $r$-th feature is included in the subset and vice versa if the bit is 0. Thus, the number $m$ of features in the subset is the total number of 1’s in the chromosome.

In this view, the $l$-th chromosome represents a binary transformation vector $V_l$ of dimension $n$, which operates on the $n$-th dimensional patterns of set $X$ producing a modified set of $m$-dimensional patterns $B=V_l(X)$ (Figure 2-2). Note that contrary to other GA applications, in this case of feature selection the binary chromosome does not encode real-valued control factors: the information regarding the features presence or absence in the optimal subset for classification is included in the bits themselves so that no decoding is necessary.

![n-dimensional binary chromosome](image)

Figure 2-1. $n$-dimensional binary chromosome
Two objective functions (fitness) are used for evaluating and comparing the feature subsets during the search: the fraction of patterns correctly classified (recognition rate) by a Fuzzy K-Nearest Neighbors classifier (to be maximized) and the number \( m \) of features forming the subsets (to be minimized).

With respect to these two objective functions, the feature subsets are compared in terms of dominance and the optimization leads to the identification of the so-called Pareto optimal set or front of non-dominated feature subsets [7, 14].

![Diagram of feature selection process](image)

**Figure 2-2.** GA-based feature selection using classification accuracy and number \( m \) of selected features as optimization criteria. Each binary chromosome from the GA population transforms the original patterns to a reduced feature space and passes them to the classifier. The objective function values of the chromosome are the classification accuracy attained on the transformed patterns and their dimension \( m \).

The efficiency of the search depends on the ability to maintain genetic diversity through the generations so as to arrive at a population of individuals which uniformly represent the real nondominated solutions of the Pareto set [7]. This can be achieved by resorting to niching techniques, e.g. sharing [7, 35], which apply a “controlled niched pressure” to spread out the population in the search space so that convergence is shared.
on different niches of the Pareto front which is thus evenly covered. The interested reader is referred to [43, 44] for a discussion of the Niched Pareto-based Genetic Algorithm developed within this PhD work.

For the FKNN evaluation of the classification performance fitness function associated to a given feature subset $S_l$ with corresponding binary transformation vector $V_l$, the labelled patterns constituting the transformed dataset $B=V_l(X)$ are randomly subdivided into a set $B_1$ containing 75% of the data which are used for the classifier construction and a tuning set $B_2$ of 25% of the data, which are used to compute the performance of the classifier in terms of its accuracy. By trial and error, a number $K=5$ of neighbors has been found to produce good classification results. The obtained fuzzy partition of the tuning dataset $B_2 \{ \mu_i(\bar{x}_k) \}$, where $0 \leq \mu_i(\bar{x}_k) \leq 1$ is the membership function of pattern $\bar{x}_k$ to class $i$, is converted into a hard partition assigning each pattern to the class with highest membership value.

The subdivision of the available patterns in training and tuning sets ($B_1$ and $B_2$, respectively) is randomly repeated 10 times (10 cross-validation) and for each tuning set $B_2$ the accuracy of the FKNN classifier operating on the proposed subset of features $S_l$ is evaluated in terms of the recognition rate (the fraction of tuning patterns of $B_2$ correctly classified) [6]: then, the mean recognition rate is calculated and sent back to the GA as the fitness value of the transformation chromosome $V_l$ used to produce the transformed set of patterns $B$.

At convergence, a family of non-dominated chromosomes (feature subsets) with different trade-offs of classification performance (FKNN recognition rate) and complexity (number of features) are obtained. Based on these results, an informed choice can be made on the features to be actually monitored for the diagnostic task considering also practical issues of costs, ease of data acquisition and data interpretability. Eventually, the analyst must select the preferred solution according to some subjective preference values.

Finally, the pre-constructed validation set $X'$, separate from the training and tuning sets ($B_1$ and $B_2$) used for the feature selection task, is processed through the optimal transformation vector $V^*$ and the corresponding classifier to verify the classification accuracy on a set $B'$ of new patterns (never used during the feature selection process) specified in terms of the optimal selected feature subset encoded by the transformation vector $V^*$ (Figure 2-2). This validation procedure is of paramount
importance for safety applications in critical technologies such as the nuclear one considered in the present PhD work.

3 Transient identification

In the previous Section, a wrapper scheme for selecting the features relevant for the classification task has been devised by combining a MOGA search with a FKNN classification. The former effectively searches the large feature space whereas the latter offers the needed fast computing characteristics.

For increased accuracy and reliability in the classification, more refined soft computing techniques may be adopted [19]. In the following, an approach to transient identification based on pattern classification by possibilistic clustering [9] is presented as an extension of fuzzy clustering.

3.1 Fuzzy and possibilistic clustering

Fuzzy clustering algorithms have been widely studied and applied in various substantive areas such as taxonomy, medicine, geology, business, engineering, image processing and others. A general classification of these algorithms is offered in [25] in terms of three categories: fuzzy clustering based on fuzzy relations, fuzzy clustering based on the minimization of an objective function and the class of nonparametric classifiers based on the generalized fuzzy K-nearest neighbors rule. The interested reader is referred to [25] for a detailed discussion of the three categories and an extensive literature review of works in the field.

The approach illustrated in the following paragraphs falls under the second of the above categories. The fuzzy clustering is performed by minimizing an objective function under the following constraints on the membership functions \( \mu_{ik} \) of pattern \( \vec{x}_k \) in cluster \( i \) [25, 36]:

\[
0 \leq \mu_{ik} \leq 1 \quad i=1,2,\ldots,c, \quad k=1,2,\ldots,N \tag{1}
\]

\[
\sum_{i=1}^{c} \mu_{ik} = 1 \quad k=1,2,\ldots,N \tag{2}
\]

In particular, the ‘probabilistic’ constraint (2), that the memberships of a given pattern must sum up to 1, is a generalization of the condition which ensures that in a

\[\text{The interested reader is referred to Part II, Report B, Section 2 for a detailed discussion of fuzzy and possibilistic clustering algorithms.}\]
‘hard’ (crisp) partition a pattern is a member of one class only and avoids the trivial solution of all memberships equal to 0. As a result of this constraint, the membership of a pattern to a cluster depends on the memberships to all other clusters; geometrically speaking, it depends on where the pattern is located with respect to not only that cluster but also to the others. Hence, in the framework of fuzzy clustering the membership functions take the meaning of degrees of sharing, i.e. they measure how much a pattern belongs to a cluster relatively to the others.

Under these conditions, two major drawbacks arise [9]:

1. The constrained memberships cannot distinguish between ‘equal evidence’ and ‘ignorance’ or, in other words, between ‘equally likely’ and ‘unknown’ membership to a cluster.
2. Since most distance functions used in fuzzy clustering are geometric in nature, noisy patterns, which typically lie far from the clusters, can drastically influence the estimates of the clusters prototypes and, hence, the final partition and the resulting classification.

In this situation, an ‘unknown’, atypical pattern not belonging to any cluster would still belong more to one cluster than to the others, relatively speaking, and thus it may receive high membership values to some clusters even if it lies far from all clusters in the feature space.

On the contrary, in the diagnostic practice it is required that unknown, atypical patterns be recognized as such, i.e. bear low membership to all clusters. In this respect, thus, the ‘conservation of total membership’ constraint (2) is too restrictive since it gives rise to relative membership values, dependent on the number of clusters.

To overcome the above limitations, the clustering problem can be recast into the framework of possibility theory [9, 27, 37]. In this interpretation, the membership function \( \mu_{ik} \) represents the degree of compatibility of the pattern \( \bar{x}_i \) with the prototypical member \( \bar{v}_j \) of cluster \( i \). If the classes represented by the clusters are thought of as a set of fuzzy sets defined over the Universe of Discourse (UOD), then there should be no constraint on the sum of the memberships. The only constraint is that the membership values do represent degrees of compatibility, or possibility, i.e. they must lie in \([0,1]\). This is achieved by substituting the fuzzy clustering constraints (1-2) with the following [26]:

\[
0 \leq \mu_{ik} \leq 1 \quad i=1,2,\ldots,c, \quad k=1,2,\ldots,N \tag{3}
\]

\[
\max_i \mu_{ik} > 0 \quad k=1,2,\ldots,N \tag{4}
\]
where constraint (4) simply ensures that the set of fuzzy clusters covers the entire UOD. A possibilistic partition derived under these constraints defines a set of distinct, uncoupled possibilistic distributions (and the corresponding fuzzy subsets) over the UOD [26].

3.2 The supervised evolutionary possibilistic clustering algorithm for classification

In this Section, a supervised evolutionary possibilistic clustering algorithm developed within the present PhD work to perform the diagnostic identification of transients is discussed.

The traditional, unsupervised possibilistic algorithm based on a Euclidean metric to measure compatibility leads to spherical clusters that rarely are adequate to represent the data partition in practice [28]. A significant improvement in classification performance is achieved by considering a different Mahalanobis metric for each cluster, thus obtaining different ellipsoidal shapes and orientations of the clusters that more adequately fit the a priori known data partition [28, 49].

The information on the membership of the available patterns \( \tilde{x}_k, k=1,...,N \), to the \( c \) a priori known classes, can be used to supervise the algorithm for finding the optimal Mahalanobis metrics such as to achieve geometric clusters as close as possible to the a priori known physical classes. Correspondingly, the possibilistic clustering algorithm is said to be constructed through an iterative procedure of ‘training’ based on a set of available patterns, pre-labeled with their possibilistic memberships to the a priori classes. The procedure for the optimization of the metrics is carried out via an evolutionary procedure, presented in the literature within a supervised fuzzy clustering scheme [28] and further extended to diagnostic applications within this PhD work (See Part III, Papers IV and V for more details on the extension). Here, the procedure is employed within the possibilistic clustering scheme [46].

To this purpose, the distance between the set \( \Gamma_i' (t = \text{true}) \) of memberships of the \( N \) available patterns to the a priori known class \( i \) and the corresponding set \( \Gamma_i \) of the possibilistic memberships to cluster \( i=1,...,c \), is computed by:

\[
D(\Gamma_i', \Gamma_i) = \sum_{k=1}^{N} \frac{|\mu_{ik}' - \mu_{ik}|}{N} \quad (5)
\]
where $0 \leq \mu_{ik}^t \leq 1$ is the a priori known (possibilistic) membership of the $k$-th pattern to the $i$-th physical class and $0 \leq \mu_k \leq 1$ is the possibilistic membership to the corresponding geometric cluster in the feature space.

The target of the supervised optimization is the minimization of the distance $D(\Gamma^t, \Gamma^*)$ between the a priori known physical class partition $\Gamma^t (\Gamma_1^t, \Gamma_2^t, \ldots, \Gamma_N^t)$ and the obtained geometric possibilistic cluster partition $\Gamma^* = (\Gamma_1^*, \Gamma_2^*, \ldots, \Gamma_N^*)$:

$$D(\Gamma^t, \Gamma^*) = \sum_{i=1}^{c} \sum_{k=1}^{N} \frac{D(\Gamma^t_i, \Gamma^*_i)}{c}$$

The optimal membership function $\mu_{ik}^*$, $i=1,2,\ldots,c$, $k=1,2,\ldots,N$ can be computed accordingly to the possibilistic clustering algorithm [26] in which the distance $s_{ik} = s_i(\vec{x}_k, \vec{v}_i^*)$ between the pattern $\vec{x}_k$ and the optimal cluster center $\vec{v}_i^*$ is computed by:

$$s_i(\vec{x}_k, \vec{v}_i^*) = (\vec{x}_k - \vec{v}_i^*)^T M_i (\vec{x}_k - \vec{v}_i^*)$$

$M_i$ being the metric for the cluster $i$ proposed by the evolutionary supervised procedure and $T$ denoting the transpose operator.

The overall iterative training scheme can be summarized as follows:

1. At the first iteration ($\tau = 1$), initialize the metrics of all the $c$ clusters to the Euclidean metrics, i.e. $M_i(l) = I$, $i=1,2,\ldots,c$, where $I$ is the identity matrix.

2. At the generic iteration step $\tau$, run the possibilistic clustering algorithm (Part II, Report B, Section 2.5) to partition the $N$ training data into $c$ clusters, $\Gamma^* (\tau) = (\Gamma_1^*(\tau), \Gamma_2^*(\tau), \ldots, \Gamma_N^*(\tau))$, based on the current metrics $M_i(\tau)$.

3. Compute the distance $D(\Gamma^t, \Gamma^*(\tau))$ between the a priori known physical classes and the geometric possibilistic clusters obtained in step 2. At the first iteration ($\tau = 1$) initialize the best distance $D^*$ to $D(\Gamma^t, \Gamma^*(1))$, $D_i^*$ to $D(\Gamma^t_i, \Gamma^*_i(1))$ and the best metrics $M_i^*$ to $I$ and go to step 5.

4. If $\Gamma^*(\tau)$ is close to $\Gamma^t$, i.e. $D(\Gamma^t, \Gamma^*(\tau))$ is smaller than a predefined threshold $\varepsilon$, or if the number of iterations $\tau$ is greater than the predefined
maximum allowed number of iterations $\tau_{\text{max}}$, stop: $\Gamma^*(\tau)$ is the optimal cluster partition $\Gamma^*$; otherwise, if $D(\Gamma^*,\Gamma^*(\tau))$ is less than $D^+$ upgrade $D^+$ to $D(\Gamma^*,\Gamma^*(\tau))$, $M^+_i$ to $M_i(\tau)$ and $D^+_i$ to $D(\Gamma^*_i,\Gamma^*_i(\tau))$.

5. Increment $\tau$ by 1. Update each matrix $M^+_i$ by exploiting its unique decomposition into Cholesky factors [28], $M^+_i = G^+_i \cdot G^+_i$, where $G^+_i$ is a lower triangular matrix with positive entries on the main diagonal. More precisely, at iteration $\tau$, the entries $g_{i,l_2}^j(\tau)$ of the Cholesky factor $G_i(\tau)$ are updated as follows:

$$g_{i,l_2}^j(\tau) = g_{i,l_2}^j + N_{i,l_2}^j(0,\delta_i^+) \quad \text{if } l_1 < l_2$$

$$g_{i,l_2}^j(\tau) = \max\{10^{-5}, g_{i,l_2}^j + N_{i,l_2}^j(0,\delta_i^+)\} \quad \text{if } l_1 = l_2$$

where $\delta_i^+ = \alpha D_i^+$, $\alpha$ is a parameter that controls the size of the random step of modification of the Cholesky factor entries $g_{i,l_2}^j, N_{i,l_2}^j(0,\delta_i^+)$ denotes a Gaussian noise with mean 0 and standard deviation $\delta_i$ and eq.(9) ensures that all entries in the main diagonal of the matrices $G_i(\tau)$ are positive numbers and so $M_i(\tau)$ are definite positive distance matrices. Notice that the elements of the $i$-th Mahalanobis matrix are updated proportionally to the distance $D_i^+$ between the $i$-th a priori known class and the $i$-th found cluster. In this way, only the matrices of those clusters which are not satisfactory for the classification purpose are modified.

6. Return to step 2.

The overall structure of the algorithm is depicted in Figure 3-1.
The closed external loop iterates until an acceptable clustering of the training data is found, i.e. until the obtained possibilistic partition \( \Gamma^*(\tau) \) is “close” to the a priori known partition \( \Gamma^j \), where “close” means that the distance \( D(\Gamma^j, \Gamma^*(\tau)) \) is smaller than a defined threshold \( \varepsilon \).

At convergence, the supervised evolutionary possibilistic clustering algorithm provides the \( c \) optimal metrics \( M^+_i \) with respect to the classification task, the \( c \) possibilistic cluster centers \( \bar{v}^+_i \) and the possibilistic membership values \( \mu^+_ik \) of the patterns \( \bar{x}_k, k=1,\ldots,N \) to the clusters \( i=1,2,\ldots,c \).

When fed with a new pattern \( \bar{x} \), the classification algorithm provides the values of the membership functions \( \mu^+_i(\bar{x}), i=1,2,\ldots,c \) to the possibilistic clusters. These values give the degree of compatibility or “typicality” of \( \bar{x} \) to the \( c \) clusters. In practice, three situations may arise (Figure 3-2):

1. \( \bar{x} \) does not belong to any cluster with enough membership, i.e. all the membership values \( \mu^+_i(\bar{x}) \) are below a given threshold \( \varepsilon_c \) (degree of confidence): this means that \( \bar{x} \) is an atypical pattern with respect to the labelled data set used for the construction of the classification algorithm;
2. at least two membership values are above the threshold \( \varepsilon_c \): \( \bar{x} \) is thus ambiguous. In this case, the ambiguity must be regarded as equal “evidence”, i.e. the pattern is typical of more than one class and thus
cannot be assigned to a class with enough confidence. This situation occurs if \( \tilde{x} \) is at the boundary between two classes;

3. \( \tilde{x} \) belongs only to a cluster with a membership value greater than the threshold \( \varepsilon \); in this case, it is assigned to the corresponding class.

Figure 3-2. Classification of pattern \( \tilde{x} \)

4 Classification of transients in the feedwater system of a boiling water reactor

As an example of application, the identification of a predefined set of faults in a Boiling Water Reactor (BWR) is considered. Transients corresponding to the faults have been simulated by the HAMBO simulator of the Forsmark 3 BWR plant in Sweden [38]. Figure 4-1 shows a sketch of the system [38].
The considered faults occur in the section of the feedwater system 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 [39]. Two power levels, 50% and 80% of full power, have been considered.

The faults may be divided into three categories:

- F1-F5,F7 regard the line 1 of the feedwater system.
- F11-F15,F17 regard the line 2 of the feedwater system.
- F6,F8,F9,F10,F16,F18 regard both lines.

Data coming from the simulations of five transients were available for each of the 18 faults, for varying degrees of leakage and valve closures and with step and ramp changes at different leak sizes. 363 plant parameters had been recorded with a sampling frequency of 1 Hz. All transients start after 60 seconds of steady state operation. From the analysis developed in [41], it is clear that faults 6, 10, 16 have no significant consequence on the plant measured parameters because the size of the leakage considered is too small. Hence, these faults were not considered further. Moreover, given that the ramp changes cause variations of the parameters later than the step changes, only the three step changes for each fault were considered.
For the fault classification purpose of this work, the number of parameters was reduced to 123 by combining redundant measurements of the same physical quantity and by eliminating those parameters linearly correlated or carrying basically the same time evolution in all the different faults.

Compact wavelet features were then extracted from the 123 measured signals by Haar wavelet decomposition from a sliding window on the actual signal time-series [40]. The selected wavelet features are: the mean residual signal taken at the highest, i.e. coarsest, scale and the minimum and maximum wavelet coefficients over all the scales [20]. The rationale behind this choice is that the first wavelet feature captures the general trend of the signal across the windows in a compact way, being very much related to the average signal value within the analysis window, while the minimum and maximum wavelet coefficients capture important variations in the signal within a single window which would otherwise be severely smoothed out by the compression process. In particular, the maximum coefficient reflects negative trends, step changes and the negative component of spikes, whereas the minimum coefficient reflects the positive trends, step changes and positive components of spikes. The window size is selected so as to correspond to wavelet dyadic decomposition values (i.e. powers of 2) and consecutive windows are chosen with a slight overlap to avoid missing features that might be present at the window borders. Because of its ability of continuously applying the wavelet transform on a sliding window, and since the transform is used as a pre-processing step for the final transient classification, this technique has been named Wavelet On-Line Pre-processing (WOLP) [20].

In a compromise between a high level of transient compression and an acceptable resolution, time windows which are 16 patterns long, with an overlap of 6 seconds, have been chosen. Consequently, the evolution of anyone of the wavelet features in a given transient from t= 58s to t=133s is summarized in 7 points each one representing the segment of dynamics in one time-window. Thus, the application of the WOLP pre-processing on the 123 original plant measured signals generates 369 wavelet coefficients, increasing from \(2^{123}\) to \(2^{369}\) the dimension of the search space from which the optimal subset of features relevant for the fault classification task is to be selected. The interested reader can refer to Part II, Report B, Section 5 for more details on the WOLP feature extraction techniques and on its application to the case study here considered.
The transients at 50% of full power were used for the feature selection task, whereas the transients at 80% of full power were left out for validation of the resulting classifier.

For comparison purposes, Table 4-1 reports the mean recognition rate and the standard deviation obtained by the classifier based on all the 123 available features and on the feature subset selected by the experts of the plant process. Notice that there is a significant improvement in the classification performance with the reduced feature subset selected by the experts compared with the feature subset formed by all the available 123 features, both considering the test set and the validation set. This confirms the advantage of using a feature selection technique to support the classification task, otherwise based on the large number of available features. Finally, notice that the recognition rates (rr) achieved on the validation set obtained from transients at 80% power level are in general higher than those achieved on the tuning set made of patterns from transients at 50% power. This is due to the fact that signal variations are more relevant when the reactor is working at a higher power level so that the classification task becomes somewhat simpler.

Table 4-1. Classification performances achieved by the FKNN algorithm

<table>
<thead>
<tr>
<th>Feature subset</th>
<th>$m$</th>
<th>rr on test set (50% power level)</th>
<th>rr on validation set (80% power level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>123</td>
<td>0.534 ±0.035</td>
<td>0.647 ±0.055</td>
</tr>
<tr>
<td>Expert-selected</td>
<td>17</td>
<td>0.679 ±0.034</td>
<td>0.789 ±0.029</td>
</tr>
</tbody>
</table>

4.1 Feature selection

The feature selection technique described in Section 2.2 has been applied. Given the large number of possible solutions ($2^{369}$), the task of maintaining genetic diversity in the population in order to explore more accurately the search space is sought by using a Niched Pareto-based Genetic Algorithm (NPGA) [35, 43] with a large population size ($n_p=200$) and a high probability of mutation ($p_m=0.008$). In a single run, the NPGA identifies a family of non-dominated solutions with different classification performance (FKNN mean recognition rate) vs. complexity (number of features) trade-offs.

Figure 4-2 shows the Pareto front and the final population found by the NPGA at convergence. The “nicthing pressure” applied by the equivalence class sharing method
succeeds in spreading the population out along the Pareto optimal front: indeed, the NPGA Pareto solutions cover from $m=0$ to $m=22$ with only individuals with $m=8,9,15$ not present.

![Pareto set and last population](image)

Figure 4-2. Pareto front and final population found by the NPGA at convergence

The analyst has to take the final decision on the features subset to be actually used for the classification, depending on his/her favouring high classification accuracy or low number of features. Thus, the closure of the problem must still rely on techniques of decision analysis such as utility theory, multi-attribute value theory or fuzzy decision making, to name a few.

In the present case, the optimization results for the non-dominated subset with 7 features show a good compromise between high classification accuracy and low number of features (pointing arrow in Figure 4-2).

A set of patterns taken from transients at 80% power level, never employed during the feature selection phase, has then been used to cross-validate the performance of the FKNN classifier resting upon the selected 7-features subset. The resulting recognition rate of $0.9190 \pm 0.0301$ is significantly better than that obtained by using all the 123 features ($0.647 \pm 0.055$) and by using the feature selected by the plant expert ($0.789 \pm 0.029$)(Table 4-1).
4.2 Transient classification

The possibilistic clustering technique illustrated in Section 3 has been applied to the problem of classifying the transients generated by the six faults F1, F2, F3, F4, F5 and F7 that regard line 1 of the feedwater system of Figure 4-1.

Five features are used for the transient classification, extracted from the 7 features identified as optimal compromise in the previous Section 4.1. In particular, the 3 signals that regard line 1 of the feedwater system, one signal that regards line 2 and one signal that regards the common part of the system have been chosen for the classification of the transients generated by the identified faults. The remaining 2 signals not considered regard line 2 of the feedwater system and are basically constant in the considered faults. Table 4-2 reports the five signals selected and Figure 4-3 their behaviour for the 6 faults at 80% of full power.

Table 4-2. Input signals of the classification model

<table>
<thead>
<tr>
<th>Feature number as reported in [43]</th>
<th>Signal name</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>77</td>
<td>drain temperature before valve VB3 in line 1</td>
<td>°C</td>
</tr>
<tr>
<td>160</td>
<td>Water level in tank TD1</td>
<td>m</td>
</tr>
<tr>
<td>195</td>
<td>feedwater temperature after preheater EA2 in line 1</td>
<td>°C</td>
</tr>
<tr>
<td>241</td>
<td>feedwater temperature after preheater EB2 in line 2</td>
<td>°C</td>
</tr>
<tr>
<td>320</td>
<td>position level of the control valve for preheater EA1 in line 1</td>
<td>%</td>
</tr>
</tbody>
</table>
The possibilistic classifier is built on patterns taken every 6 seconds from t=80s to t=200s from simulated transients of each fault type, with the plant at 80% of full power.

After the evolutionary supervised training of the possibilistic classifier, its performance has been tested using patterns taken every second from t=0s to t=300s from both the training transients and from an unknown transient caused by F13. Figure 4-4 shows the obtained transient classification as time progresses. Considering a degree of confidence $\varepsilon=0.7$, the results are quite satisfactory, even though at the beginning of the transients the possibilistic classifier assigns the steady state patterns of the first 60 seconds to the class of fault F2 albeit with low membership. This is explained by the fact that for transients of class F2 there are no significant effects on the relevant signals of Table 4-2 so that understandably the steady state may be confused with a fault of class F2 (Figure 4-3).
Figure 4-4. Time profiles of the pattern assignment to the different classes: (+) membership to class F1, (o) membership to class F2, (*) membership to class F3, (.) membership to class F4, (x) membership to class F5 and (◊) membership to class F7. The horizontal solid line represents the degree of confidence $\varepsilon_c$ here set to 0.7.

Table 4-3 reports the time necessary for the possibilistic classifier to assign the transients to the right class with a membership value greater than $\varepsilon_c = 0.7$. Notice that class F3 is recognized only 6 seconds after the beginning of the transient at $t=60s$ while for the recognition of class F7 it is necessary to wait 34 seconds. This is due to the fact that the two most sensible signals for F7, 320 and 195, start departing from their steady state at $t=82s$ and $t=78s$, respectively, reaching significant variations after 30s from the beginning of the transient at $t = 60s$ (Figure 4-3).
Table 4-3. Time at which the transient is assigned to the right class (the transient begins at t=60s)

<table>
<thead>
<tr>
<th>Class of the test transient</th>
<th>Time (s) at which there is the first assignment with membership value ( \varepsilon_c = 0.7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>69</td>
</tr>
<tr>
<td>F2</td>
<td>75</td>
</tr>
<tr>
<td>F3</td>
<td>66</td>
</tr>
<tr>
<td>F4</td>
<td>87</td>
</tr>
<tr>
<td>F5</td>
<td>68</td>
</tr>
<tr>
<td>F7</td>
<td>94</td>
</tr>
</tbody>
</table>

Also, note that the possibilistic classifier is able to assign to the right class the foreseen transients at times well beyond the temporal domain of 200s considered in the construction phase of the algorithm, due to the increased significance of the signals as the transients continue evolving away from their initial steady state.

Finally, the algorithm is very efficient in filtering out the patterns of the unknown fault F13 as atypical by assigning them very low membership values to all classes (Figure 4-4, bottom). This is particularly important in diagnostic problems, given the impossibility to identify and explicitly enumerate a priori all transient scenarios possibly occurring in a system.

The identification as unknown of those transient conditions that have not been foreseen when building and tuning the diagnostic system, possibly allows to avoid taking wrong protection and emergency actions as a consequence of an incorrect diagnosis [19]. This may be important since an incorrect inference by the diagnostic system during an emergency condition risks to confuse and mislead the operator, with the potential of producing catastrophic consequences upon taking incorrect actions. Hence, the importance of classifying as unknown the transients that are not within the utilization domain of the classification algorithm.
5 Conclusions

The goal of the present PhD work has been to contribute to the development of efficient diagnostic techniques for the ‘on line’ classification of faults and malfunctionings in nuclear power systems and components.

Two key issues to be addressed when developing an on line fault diagnostic technique are:

1) the selection of the features relevant for the classification;
2) the building of a robust classification model capable of discerning the different transient types from the measured dynamic patterns.

A wrapper approach which combines a multi-objective genetic algorithm search with a Fuzzy K-Nearest Neighbors (FKNN) classification algorithm has been embraced to tackle the first issue of feature selection. The performance of the feature subsets examined during the search is evaluated in terms of two optimization objectives: the classification accuracy of a Fuzzy K-Nearest Neighbors classifier and the number of features in the subsets.

The general advantages of the proposed methodology are that the ‘wrapping’ search measures explicitly the goodness of the considered feature subsets on the performance of the ‘wrapped’ classification algorithm (in our case the FKNN) and that the GA search is immune from the “nesting effect” characteristic of “hill climbing” methods, so that the probability of getting stuck in local optima is significantly lower.

A modified Niched Pareto Genetic Algorithm has been developed to spread out the population in the search space so that convergence is shared on different niches of the Pareto front. This leads to the identification of a representative and evenly distributed set of alternative, equivalent feature sets offering different trade-offs in terms of diagnostic power and complexity, from which an informed choice of the preferred feature subset can be taken.

Regarding the second issue of building an efficient and robust classification model, a supervised, evolutionary possibilistic clustering algorithm has been developed. A Mahalanobis metric for each possibilistic cluster is found such that an optimal classification of an available set of labelled patterns is achieved. The possibilistic approach viewpoint on the other hand allows controlling the boundaries of effective
application of the classification model by recognizing as unknown those transient conditions that have not been foreseen when building and tuning the diagnostic system. In practice, this avoids running the risk of taking wrong protection and emergency actions as a consequence of an incorrect diagnosis.

The methods developed have been applied to a case study concerning the classification of simulated nuclear transients in the feedwater system of a Boiling Water Reactor. Using five parameters identified as relevant by the feature selection algorithm, the possibilistic classifier correctly classifies the foreseen plant transients, while filtering out the patterns of an unknown fault type by assigning them very low membership values to all classes.

Regarding future research directions along the lines of work of this thesis, it must be noted that the multi-objective genetic algorithm used for the feature selection task does not solve the decision problem: the preferred solutions should be actually chosen from the Pareto set. The closure of the problem then still relies on techniques of decision analysis such as utility theory, multi-attribute value theory or fuzzy decision making, to name a few, which could be worth investigating. In this respect, it is important to transform into quantitative values (‘preferences’) the qualitative linguistic categories that the decision maker uses to express his/her preferences. Then, a technique to efficiently drive the search of the genetic algorithm towards a specific region of the Pareto optimal front, accordingly to the expressed preferences should be developed.

Furthermore, recent research findings have proved the benefits of using ensembles of classification models for they allow to achieve higher accuracy than single models. For an ensemble to be effective, it should consist of base classifiers that give diverse predictions. One method for constructing an ensemble is to have the base classifiers work on different feature sets. Feature selection techniques like the one developed within this PhD work can indeed be exploited for the construction of the ensemble.

Finally, given the importance, from the diagnostic point of view, of recognizing transients in the first instants of their evolution so as to enable the operator to schedule the proper corrective and mitigation actions, it seems worth to investigate the possibility of developing diagnostic models that attempt also to minimize the delay time in the diagnosis, i.e. the time necessary in the definite assignment of a transient to a class of faults.
References


[27] Fantoni, P.F., *Experiences and applications of PEANO for online monitoring in power plants* Progress in Nuclear Energy, 46, pp 206-225, 2005


[48] Zio, E. and Baraldi, P., Evolutionary fuzzy clustering for the classification of transients in nuclear components, Progress of Nuclear Energy, Vol. 46, No. 3-4, pp. 282-296, 2005

PART II
Report A†

Feature selection for nuclear transient diagnostics

Enrico Zio, Piero Baraldi and Davide Roverso

LASAR‡ and Halden Reactor Project Work Report, January 2006

† This report is being submitted as official Halden Work Report (HWR)
‡ LASAR (Laboratorio di Analisi di Segnale ed Analisi di Rischio): Laboratory of Signal Analysis and Risk Analysis of the Department of Nuclear Engineering of the Polytechnic of Milan
# TABLE OF CONTENTS

1. INTRODUCTION TO FEATURE SELECTION FOR NUCLEAR TRANSIENT CLASSIFICATION ... 2

2. CLASSICAL TECHNIQUES OF FEATURE SELECTION .......................................................... 2
   2.1 Reasons for performing the feature selection .......................................................... 3
   2.2 Objective of feature selection .............................................................................. 3
   2.3 Feature selection techniques .................................................................................. 3
   2.3.1 Filter methods .................................................................................................. 3
   2.3.2 Wrapper methods ............................................................................................ 5
   2.4 The search engine .................................................................................................. 6
   2.4.1 Complete search ............................................................................................. 6
   2.4.2 Heuristic search .............................................................................................. 6
   2.4.3 The probabilistic search ................................................................................... 7
   2.5 Comparison between different feature selection techniques ................................ 8

3. FEATURE SELECTION FOR TRANSIENT DIAGNOSTICS: TWO NOVEL APPROACHES .......... 9
   3.1 A novel filter approach based on the extended classifiability evaluation function ....... 9
   3.2 A genetic algorithm-based wrapper approach ....................................................... 12
   3.2.1 Multi-Objective Genetic Algorithm (MOGA) .................................................... 14

4. CASE STUDY: CLASSIFICATION OF TRANSIENTS FOR FAULT DIAGNOSIS IN THE FEEDWATER SYSTEM OF A BOILING WATER REACTOR ................................................... 15
   4.1 Filter approach results .......................................................................................... 17
   4.2 Wrapper approach results .................................................................................... 18
   4.3 Comparison of the results obtained by the two techniques ..................................... 23

5. DATA PRE-PROCESSING FOR FEATURE SELECTION .................................................... 25
   5.1 Wavelet on line pre-processing ............................................................................ 25
   5.2 The WOLP-GA-based wrapper feature selector applied to the classification of nuclear transients .............................................................................................................. 28

6. A GENETIC ALGORITHM FOR THE MINIMIZATION OF THE RECOGNITION DELAY TIME .................................................................................................................. 34
   6.1.1 Definition of the Recognition Delay Time and its fuzzy extension ..................... 34
   6.2 Definition of fuzzy performance ........................................................................... 35
   6.3 A Genetic algorithm-based feature selector for ‘early’ fault classification ............... 36
   6.4 Results ................................................................................................................... 38

7. CONCLUSIONS ............................................................................................................. 41

APPENDIX A .................................................................................................................. 43

APPENDIX B .................................................................................................................. 43
NOTATION

Φ(·)  classification function

\( \tilde{x} \)  generic pattern to be classified

N  number of patterns used for the feature selection

\( k = 1, \ldots, N \)  index associated to the generic pattern

\( X = \{ \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N \} \)  set of \( n \)-dimensional patterns used for the feature selection

\( c \)  number of classes

\( i = 1, \ldots, c \)  class of the generic object

\( w^i(\tilde{x}_k) \)  a priori known Boolean assignment of pattern \( \tilde{x}_k \) to the \( i \)-th class

\( w^i(\tilde{x}_k) \)  Boolean assignment of pattern \( \tilde{x}_k \) to the \( i \)-th class (performed by the classifier)

\( S_o \)  complete feature set (formed by all the \( n \) available features)

\( S_f \)  generic feature subset (formed by \( m \) features)

\( S_f^* \)  optimal feature subset (formed by \( h \) features)

\( n \)  number of all the available features

\( m \)  number of features contained in the generic feature subset

\( h \)  number of features contained in the optimal feature subset

\( r \)  number of measured signals

\( d \)  neighboured distance used for the classifiability computation

\( L(S; d) \)  classifiability evaluation function of the feature subset \( S_f \)

\( L^*(S; d) \)  weighed transient classifiability evaluation function of the feature subset \( S_f^* \)

\( K \)  number of nearest neighbors considered by the FKNN classifier

\( \mu^i(\tilde{x}_k) \)  fuzzy membership of the \( k \)-th pattern to the \( i \)-th class

\( l \)  index associated to the generic chromosome

\( V_l \)  generic binary transformation vector created by the Genetic Algorithm

\( V_f \)  optimal binary transformation vector found by the Genetic Algorithm

\( F = \{ f \} \)  set of real objective functions to be maximized/minimized by means of Genetic Algorithm

\( U = \{ u \} \)  decision variables from which the set of objective functions \( F \) depends

\( B \)  set of \( m \)-dimensional patterns obtained by applying the transformation vector \( V_f \) to \( A \).

\( X^* \)  Set of patterns used for the result validation

\( \tilde{x}_k(t_r) \)  pattern taken at time \( t_r \) from the \( k \)-th transient

\( N_t \)  number of available transients

\( t_o \)  first time step analyzed for the diagnosis of the transient

\( t_f \)  generic time step of the transient

\( t_r \)  last time step analyzed for the diagnosis of the transient

\( \Delta t \)  time interval at which the time step are taken

ACRONYMS

KNN  K-Nearest Neighbour algorithm

FKNN  Fuzzy K Nearest Neighbours algorithm

SFS  Sequential Forward Selection method

SBE  Sequential Backward Elimination method

RR  recognition rate index

fp  fuzzy performance index

GAs  Genetic Algorithms

MOGA  Multi-Objective Genetic Algorithm

RDT  Recognition Delay Time

FRDT  Fuzzy Recognition Delay Time
1. INTRODUCTION TO FEATURE SELECTION FOR NUCLEAR TRANSIENT CLASSIFICATION

The early identification of the causes for the onset of a meaningful departure from steady state behaviour is an essential step for the operation, control and accident management of nuclear power plants. Given the safety and economical importance of the problem, several approaches for transient identification have been investigated and many efforts are continuously devoted to the improvement of the results so far obtained.

The approach to the classification of the causes responsible of a pre-defined set of transients is typically performed through a sequence of three steps which include:

1. the pre-processing of the dynamical data for extraction of the features most characteristic of the system transient behaviour;
2. the selection of the features relevant for the classification;
3. the classification of the dynamic patterns into different transient types.

With respect to the first step, in HWR-640 [1] a technique for the extraction of compact wavelet features from multivariate transient data that accounts for the intrinsically dynamic nature of the transient classification problem is described. For the third step, that of the actual classification, advanced techniques based on modern concepts of computational intelligence, like neural networks and fuzzy logic systems, have been recently propounded [1,2,3,4].

In this report, step 2 is addressed, i.e. the problem of discerning out of the several obtained features those to be used for efficient, early transient classification. The feature selection task is important since irrelevant or noisy features unnecessarily increase the complexity of the diagnostic problem and can degrade modelling performance. Further, for the particular case of nuclear power plants the problem is also critical with respect to the very large number of parameters monitored for operation and safety reasons.

Feature selection aims at finding the optimal feature subset that achieves the highest accuracy in the pattern classification task with the minimum number of features. In this respect, transient diagnostics differs from traditional pattern classification problems because the objects to be classified are trajectories traced by the evolution of the plant parameters in time, rather than data points in the feature space.

Specific techniques are here developed to tackle the feature selection task in transient classification problems. In particular, an extension of a filter approach proposed in literature [5] and a wrapper approach based on a multi-objective genetic-based search engine, are developed and compared on a nuclear transient classification problem. The benefits deriving from the integration of the feature selection scheme with the pre-processing of the dynamical data (step 1, above) are also investigated. The requirement of an ‘early’ classification of the transients is tackled by defining a transient recognition delay time. This aspect is fundamental to enable the operator to optimize the scheduling of corrective and mitigating actions on the adverse consequences of the malfunctioning.

2. CLASSICAL TECHNIQUES OF FEATURE SELECTION

In this Section, the feature selection problem is reviewed and the filter and wrapper approaches for its solution are introduced.
2.1 Reasons for performing the feature selection

The diagnostic task may be viewed as a problem of classification. From a mathematical point of view, a classifier is a mapping function $\Phi(\cdot)$ which assigns an object $\bar{x}$ to a given class $i$. If one knew the exact expression of $\Phi(\cdot)$, the question of which features of $\bar{x}$ to use would not be of interest. In fact, in such situation adding features does not decrease the accuracy of the classifier, and hence restricting $\Phi(\cdot)$ to a subset of features is never advised. However, as it is often the case in engineering, it is not possible to determine the exact analytical expression of the function $\Phi(\cdot)$ due to the complexity of the systems under analysis. Hence, one resorts to empirical classification techniques in which the classifier is built through a process based on a set of classification examples $(\bar{x},i)$ constituted by patterns $\bar{x}$ in the feature space labelled by the corresponding class $i$. In this case, the features relevant for the classification are often unknown a priori. Therefore many candidate features are measured and collected to describe the object $\bar{x}$ to be classified. In practice, unfortunately the number of features that are collected is quite large, many of which may be irrelevant for the classification task, i.e. they do not affect the classification in any way, or redundant, i.e. they do not add information to that already carried by the other features.

In practical scenarios, at least five reasons call for a reduction in the number of features used in the classification model. First of all, irrelevant, non informative features result in a classification model which is not robust [6, 7]. Second, studies have shown that it is necessary to remove highly correlated features [7]. Third, when the model handles many features, a large number of observation data are required to properly span the high-dimensional feature space for accurate multivariable interpolation. Fourth, by eliminating unimportant features the cost and time of collecting the data and developing the classifier can be reduced. Finally, reducing the number of selected features permits to achieve more transparent and easily interpretable models.

2.2 Objective of feature selection

In general, the objective of feature selection is that of finding a subset of the original features of a dataset such that an induction algorithm run on data represented only by these features generates a classifier with the highest possible classification accuracy [8]. Notice that, accordingly, different induction algorithms may require different feature subsets and so an optimal universal feature subset does not exist.

Other requirements are often added to the feature selection objective, such as the reduction of the number of features for reducing the computational and data storage burdens.

2.3 Feature selection techniques

Several methods have been proposed to tackle the feature selection problem. They are usually classified into two categories: filter and wrapper methods [8].

2.3.1 Filter methods

In filter methods, the feature selector algorithm is independent of the specific learning algorithm used in the classification and it is used as a filter to discard irrelevant and/or redundant features, a priori of the construction of the classification algorithm (Figure 1). A numerical evaluation function is used to compare the feature subsets $S_f$ proposed by a search algorithm [9]. The subset $S_f$ with the highest value of the evaluation function is the final feature set which feeds the induction algorithm for the classification.
Figure 1. Filter method for feature selection; the features are filtered independently of the induction algorithm.

In the next Section, the characteristics of an evaluation function of literature [5] are discussed as an example of filter methods.

2.3.1.1 The classifiability evaluation function for feature selection by filtering

The definition of the classifiability evaluation function can be motivated on the basis of a visualization of an $m$-dimensional pattern classification problem in $m+1$ dimensions, using the class label as the $(m+1)$-th dimension. Figure 2 shows an example of a binary classification problem on a two dimensional feature subset with the corresponding visualization in three-dimensions. The introduction of the class label defines a surface which is "rough" when patterns of different classes are close in the underlying feature set (Figure 2, top) and "smooth" when patterns of the same class are adjacent (Figure 2, bottom).

Naturally, classification is considerably more complicated when the "class label surface" is rough. Consequently, if the smoothness (or roughness) of the class label surface can be quantified, then a natural measure of classifiability is obtained which depends on the feature subspace. This intuitive notion is the basis of the
algorithm proposed in [5], theoretically founded on the second-order, conditional joint density function \( f(i, j|d) \) of shifting in the feature space from a pattern of class \( i \) to a pattern of class \( j \) within a distance \( d \) from the first pattern. Here, we only give the basic steps of the operative procedure for computing the value of the classifiability function from a set of \( N \) labelled data patterns \((\hat{x}_k, i_k), k = 1, 2, ..., N\), where \( \hat{x}_k = (x_{k1}, x_{k2}, ..., x_{km}) \) is the feature vector represented by a point in the selected feature subspace \( S_l \) of dimension \( \mathbb{R}^m \subseteq \mathbb{R}^n \), and \( i_k \) is the corresponding class label:

1. Initialize to 0 two counters \( s \) and \( o \)
2. For \( k = 1 \) to \( N \)
   - add to the counter \( s \) the number of points of class \( i_k \) inside the hyper-sphere of radius \( d \) with center \( \hat{x}_k \).
   - add to the counter \( o \) the number of points not of class \( i_k \) inside the same hyper-sphere.
3. Evaluate the classifiability of the feature subspace \( S_l \) by:
   \[
   L(S_l; d) = \frac{s - o}{s + o}
   \]  

By definition, the classifiability evaluation function \( L \) takes values in \([-1, 1]\), higher values implying greater classifiability.

Besides the feature subspace \( S_l \), the classifiability depends also on the value of the neighbourhood distance \( d \). Typically, \( d \) should be large enough that at least a few points are present within the neighborhood of each point but small enough that classifiability is evaluated locally. In [5], it is empirically suggested that a value of \( d \) between 2 and 3 times the Root Mean Square distance (RMS) of each pattern from its nearest neighbor gives good results.

### 2.3.2 Wrapper methods

Contrary to filter methods, in wrapper methods the feature selector behaves as a “wrapper” around the specific induction algorithm used to construct the classifier. The feature subsets are compared using as criterion the performance achieved by the classification algorithm itself [8] (Figure 3).

![Figure 3. Wrapper method for feature selection](image)

To measure the performance achieved by a classification algorithm, a commonly used index is the recognition rate (rr) [8], i.e. the fraction of patterns correctly classified. Introducing of a set \( W \) of \( c \) Boolean assignment functions \( w_i(\hat{x}_k), i = 1, ..., c, k = 1, ..., N \) where \( w_i(\hat{x}_k) = 1 \) means that the pattern \( \hat{x}_k \) is of class \( i \) and \( w_i(\hat{x}_k) = 0 \) means that it is not of class \( i \), the recognition rate can be written as:

\[
rr = \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{c} w_i(\hat{x}_k)
\]
\[ r = 1 - \sum_{k=1}^{N} \sum_{i=1}^{c} \frac{w_i'(\bar{x}_k) - w_i'(\bar{x}_i)}{2N} \]  \hspace{1cm} (2)

where \( w_i'(\bar{x}_k) \) is the a priori known Boolean assignment of the \( k \)-th pattern \( \bar{x}_k \), \( k = 1, \ldots, N \) to the \( i \)-th class, \( i = 1, \ldots, c \). Note that this measure is devised such that the misclassification of a pattern leads to two unitary contributions in the sum on the \( c \) classes, one from the actual class of the pattern and one from the erroneously assigned class: hence the division by 2 to get the fraction of misclassified patterns.

The filter approach is generally computationally more efficient than the wrapper approach because for each feature subset of trial, the computation of an index from the available data is less time consuming than the development of a complete classification model. Indeed, given that during the search for the optimal feature subsets an high number of feature subsets are tested, the time consumption of a wrapper approach depends mainly from the time necessary for the development of the classifier and the subsequent classification of the patterns to test its performance. Hence, a wrapper approach is a suitable choice if the classifier is a fast-computing algorithm, e.g. the K-Nearest Neighbour (KNN) algorithm or its fuzzy extension.

On the other hand, wrapper approaches are more performing since they ensure the selection of the features more suitable for the specific classification algorithm used, whereas the filter approach totally ignores the effects of the selected feature subspace on the performance of the induction algorithm that will be used for the classification.

2.4 The search engine

Both the filter and the wrapper approaches conduct a search for an optimal feature subset in the space of possible features. The inclusion or not of a feature in the subset is encoded in terms of a binary variable which takes value 1 or 0, respectively. For \( n \) features, the size of the binary vector search space is \( 2^n \). An exhaustive search is impractical unless \( n \) is small.

To overcome this problem, three approaches are commonly adopted: complete, heuristic and probabilistic [8].

2.4.1 Complete search

In the complete approach, the properties of a pre-defined evaluation function are used to prune the feature space to a manageable size, thus avoiding that the complete search is also exhaustive [10]. Only some evaluation functions give rise to a search that guarantees the optimum feature subset selection without being exhaustive [11].

2.4.2 Heuristic search

The heuristic approach does not guarantee that the best feature subset is achieved, but is less time consuming than the complete one and may be employed in combination with any evaluation function [5]. At present, the most employed heuristic methods are greedy search strategies such as the sequential forward selection method (SFS) or the sequential backward elimination method (SBE), both known as “hill climbing” methods, which iteratively add or subtract features and correspondingly the evaluation function is evaluated. The forward selection refers to a search that begins with no features and at each step a feature is added to the subspace; on the contrary, the backward elimination refers to a search that begins with the \( n \)-dimensional feature set \( S_o \) and at each step a feature is removed. At each step, the choice of which feature to add or remove is driven by its effect on the classifiability function so as to climb toward its maximum value. Since in our applications typically the goal is that of reducing from a large number of features to a subspace of less than 15-20, it is computationally more convenient to start from no features. Figure 4 shows an example of sequential forward selection in a space \( S_o \) with \( n=4 \) features.
Figure 4. Example of sequential forward selection applied in a space of 4 features. Each node is connected to nodes that have one feature added.

The SFS hill-climbing search is usually stopped when adding new features does not increase the value of the classifiability function or when the number of features has reached a predefined threshold.

As a remark, note that the hill climbing methods suffer from the so called “nesting effect”: if the features added cannot be removed, a local minimum of the evaluation function may be found. To reduce this effect, it is possible to use the so called plus-l-take-away-r method (PTA) \[4,8\]. In this method, after \(l\) steps of the forward selection, \(r\) steps of the backward elimination are applied so as to allow escape from local minima. However, there is no guarantee of obtaining the absolute optimum.

2.4.3 The probabilistic search

The probabilistic approach is based on population-based metaheuristics guided by fittest solutions, such as the genetic algorithms, presented in this paper, or on methods like simulated annealing and tabu search algorithms \[12\]. Here we briefly introduce the genetic algorithm search technique.

2.4.3.1 An example of probabilistic search: the genetic algorithms

Genetic Algorithms (GAs) are optimization methods aiming at finding the global optimum of a set of real objective functions, \(F = \{f(i)\}\), of one or more variables, \(U = \{u\}\), possibly subject to various linear or non-linear constraints. The main characteristics are that the search is conducted i) using a population of multiple solution points or candidates, ii) using operations inspired by the evolution of species, such as breeding and genetic mutation, iii) using probabilistic operations, iv) using only information on the objective or search function and not on its derivatives.

GAs owe their name to their operational similarities with the biological and behavioural phenomena of living beings. After the pioneering theoretical work by John Holland \[13\], in the last decade a flourishing literature has been devoted to their application to real problems. The basics of the method may be found in Goldberg \[14\]; some applications in various context are included in Chambers \[15\].

The terminology adopted in GAs contains many terms borrowed from biology, suitably redefined to fit the algorithmic context. Thus, GAs operate on a set of (artificial) chromosomes, which are strings of numbers, generally sequences of binary digits 0 and 1. If the objective function of the optimization has many arguments (typically called control factors or decision variables), each string is partitioned in as many substrings of assigned lengths, one for each argument and, correspondingly, we say that each chromosome is partitioned in (artificial) genes. The genes constitute the so called genotype of the chromosome and the substrings, when decoded in real numbers, constitute its phenotype. When the objective function is evaluated in correspondence of a set of values of the control factors of a chromosome, its value is called the fitness of that chromosome. Thus,
each chromosome gives rise to a trial solution to the problem at hand in terms of a set of values of its control factors.

The GA search is performed by constructing a sequence of populations of chromosomes, the individuals of each population being the children of those of the previous population and the parents of those of the successive population. The initial population is generated by randomly sampling the bits of all the strings. At each step, the new population is then obtained by manipulating the strings of the old population in order to arrive at a new population hopefully characterized by an increased mean fitness. This sequence continues until a termination criterion is reached. As for the natural selection, the string manipulation consists in selecting and mating pairs of chromosomes in order to groom chromosomes of the next population. This is done by repeatedly performing on the strings the four fundamental operations of reproduction, crossover, replacement and mutation, all based on random sampling: the parents’ selection step determines the individuals which participate in the reproduction phase; reproduction itself allows the exchange of already existing genes whereas mutation introduces new genetic material; the substitution defines the individuals for the next population. This way of proceeding enables to efficiently arrive at optimal or near-optimal solutions.

With regards to their performance, it is acknowledged that GAs take a more global view of the search space than many other optimization methods. The main advantages are i) fast convergence to near global optimum, ii) superior global searching capability in complicated search spaces, iii) applicability even when gradient information is not readily achievable.

2.5 Comparison between different feature selection techniques

In this work, different feature selection techniques are compared with respect to the accuracy of a FKNN classifier fed by the different selected feature subsets. Clearly, the classification accuracies computed on the same data used for the feature selection are optimistically biased (“a subtle means of training the classifier on the test data” [8]). Hence, to compare the different feature selection algorithms, their classification accuracy is measured with respect to a separate set of data from that used for the feature selection task. To this purpose, the total number of initially available \( n \)-dimensional data are partitioned into a set \( X \) used for the feature selection task and a separate set \( X' \) used for validating the performance of the classifier resting upon the optimal feature subset \( S_f \) formed by the selected \( h \) features (Figure 5).

![Figure 5. Subdivision of the available pre-classified data into a set \( X \) for the feature selection and a set \( X' \) for validating the performance of a classifier resting on the optimal feature subset \( S_f \).](image-url)
3. FEATURE SELECTION FOR TRANSIENT DIAGNOSTICS: TWO NOVEL APPROACHES

Transient diagnostics differs from traditional classification problems because the transients to be classified are trajectories traced by the evolution of the measured parameters in time, rather than data points in the feature space. The classification usually regards the early identification of the causes which initiates the transient, for diagnostic and accident management purposes.

In this Section two new methods of feature selection for transient classification are presented, one based on a filter approach and the other on a wrapper approach.

The first method is based on the extension to the transient case of the classifiability evaluation function introduced in Section 2.3.1.1. The modification takes into account the particular distribution in the feature space of the points representative of the transients. The proposed evaluation function is combined with a sequential forward search within a filter framework for the feature selection task.

The second method consists in a genetic algorithm-based wrapper feature selector.

Table 1 summarizes the basic characteristics of the two proposed methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Evaluation function</th>
<th>Search engine</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Extended classifiability</td>
<td>Sequential Forward Selection (SFS)</td>
<td>3.1</td>
</tr>
<tr>
<td>Wrapper</td>
<td>Classification performance of a Fuzzy K Nearest Neighbour classifier</td>
<td>Multi-Objective Genetic Algorithm (MOGA)</td>
<td>3.2</td>
</tr>
</tbody>
</table>

3.1 A novel filter approach based on the extended classifiability evaluation function

In Section 2.3.1.1 the filter method for feature selection in classification problems, based on the classifiability evaluation function, has been described.

When dealing with transient classification, the corresponding representative trajectories are transformed into collections of points in the feature space, by taking the values of all the features along the trajectories over a fixed time window. Each point is labelled correspondingly to the class to which the transient belongs. The particular distribution in the feature space of the points obtained from the transients leads us to modify the definition of the classifiability evolution function. Indeed, in the first time instants after the transient initiation from steady state (which is actually the period of interest for early diagnostics) the features are very close to their nominal values, independently of the initiating cause. From a geometric point of view, the small zone of the feature space close to the point that represents the steady state contains a great number of “quasi-steady” or “early-transient” points representing indistinguishable transients from all the different classes. For example, Figure 6 shows the distribution of 385 transient points in the two-dimensional feature space of the case study illustrated in the successive Section 4. These points represent the values of the 2 features at the beginning of 55 transients of 5 different classes. As expected, at the early times the majority of points are concentrated near the steady state point (0,0.22) whereas only few are located far. By definition, the value of the classifiability evaluation function (eq.(1) in Section 2.3.1.1) is influenced more by a point with a lot of neighbors inside its sphere of radius \(d\) than by a point with few neighbors. Thus, the counter \(o\) is very large for the points close to the
steady state, which have several neighbors of different classes, and the value of the classifiability evaluation function is small even if the class surface becomes smooth far from the steady state, i.e. at longer times. To overcome this problem, inherent in the transient classification problem, the same importance in the evaluation of the classifiability is given to the \( N \) points representative of the transients, independently of the number of neighbors. This is done by defining for each point \( \bar{x}_k \), \( k=1,...,N \), the pointwise classifiability \( L_k(S;d) \):

\[
L_k(S;d) = \frac{s_k - o_k}{s_k + o_k}
\]

(3)

where \( s_k \) and \( o_k \) are the counters of the number of points in the sphere centered at \( \bar{x}_k \) of the same class and of different class of \( \bar{x}_k \), respectively. The transient classifiability function \( L(S;d) \) is then defined as the average of the pointwise classifiabilities:

\[
L(S;d) = \frac{\sum_{k=1}^{N} L_k(S;d)}{N}
\]

(4)

By so doing, the points close to the steady state, with a lot of neighbors of different classes, contribute to the value of \( L(S;d) \) in the same way as the points far away from the steady state.

Hence, for two feature subspaces S1 and S2 equally “rough” near the steady state region but with S1 more “smooth” than S2 elsewhere, \( L(S_1;d) \) is larger than \( L(S_2;d) \). In principle, then, the new classifiability \( L(S;d) \) is effective in evaluating the capability of a feature subspace of distinguishing transients away from the steady state, after some time instants from the beginning of the transient.

Nevertheless, a crucial problem in transient classification remains the early diagnosis of the malfunction. The ‘early’ requirement leads to the need of classifying small deviations from the steady state. The classifiability \( L(S;d) \) is not effective in this task because the value of the pointwise classifiability \( L_k(S;d) \) for points close to the steady state is typically very small since it contains many points of different classes. Actually, for a point close to the steady state, we are interested in its nearest neighbors. In fact, even with a lot of points inside the sphere of radius \( d \), a situation in which the nearest neighbors are of the same class is preferable, from a classification point of view, to a situation in which the nearest neighbors are of several classes. Obviously, to
give importance to the nearest neighbors one could reduce the sphere radius \(d\), but this would cause the points far from the steady state and more dispersed, to have empty spheres. A more effective way to proceed amounts to assigning a weight to the points inside the sphere according to their distance from the center. In this way a weighed pointwise classifiability \(L'_k(S_i;d)\) can be computed, which gives a local information on the roughness of the class surface independently of the number of points inside the sphere. The new algorithm for evaluating the classifiability function \(L''(S_i;d)\) for a given feature subspace \(S_i\) and distance \(d\) is the following:

1. For each point \(\tilde{x}_k\), \(k=1,\ldots,N\):
   - Initialize to 0 the counters \(sw_k\), \(ow_k\).
   - Sort the \(N_i\) points inside the hyper-sphere of radius \(d\) with center at point \(\tilde{x}_k\) of class \(i_k\) according to their distance from \(\tilde{x}_k\).
   - Assign weight \(w_p = \exp[(p-1)\ln(0.5)/9]\) to the \(p\)-th nearest point, \(p=1,2,\ldots,N_i\). The weight function is obviously arbitrary. In our case, it was chosen such that the nearest point has weight \(w_1 = 1\), the 10-th nearest has weight \(w_{10} = 0.5\) and the 25th-nearest has weight \(w_{25} = 0.15\). By so doing, only the first 20-25 neighbors give significant contributions.
   - For \(p=1\) to \(N_i\), add \(w_p\) to \(sw_k\) if the \(p\)-th nearest point in the hyper-sphere of \(\tilde{x}_k\) is of class \(i\); otherwise add \(w_p\) to \(ow_k\).
   - Compute the “smoothness” of the surface around \(\tilde{x}_k\) by the pointwise classifiability:
     \[
     L'_k(S_i;d) = \sum_{j=1}^{N_i} \frac{w_j}{N_i - sw_k - ow_k}
     \] (5)

2. Compute the weighed transient classifiability of the feature subset \(S_i\) by:
   \[
   L''(S_i;d) = \frac{\sum_{k=1}^{N} L'_k(S_i;d)}{N}
   \] (6)

To evaluate the classifiability evaluation function \(L''\) of a given feature subspace \(S_i\) one has to identify, for each of the \(N\) points \(\tilde{x}_k\), the points which are within the neighborhood of radius \(d\). This implies computing the \((N-1)\) distances \(\Delta k j\) between \(\tilde{x}_k\) and all the other points \(\tilde{x}_j\), \(j=1,2,\ldots,N, j \neq k\), with an overall computation complexity of \(O(N^2)\). Given that during the feature selection process it is necessary to compute the classifiability of several different feature subspaces, a reduction of the computational complexity in the evaluation of the classifiability of the single feature subspace is very important.

To achieve such reduction, we propose the following procedure which avoids computing all \(N(N-1)\) distances \(\Delta k j\), \(k = 1,2,\ldots,N, k \neq j\). Let us fix arbitrarily a point \(\tilde{x}_o\) in the \(R^n\) subspace and compute the \(N\) distances \(\Delta o j\) between \(\tilde{x}_o\) and the \(N\) points \(\tilde{x}_j\), \(j = 1,2,\ldots,N\) expressed in units of \(d\). Then, each of the \(N\) points is labeled by the first majoring integer of the computed distance (Figure 7). For example a point with distance 2.3 \(d\) from \(\tilde{x}_o\) has label 3. By so doing, all the points in the sphere of radius \(d\) and center \(\tilde{x}_k\) of label \(r_k\) have integer labels in \([r_k-1,r_k+1]\) (Figure 7). Thus, to find the points that fall inside a sphere of radius \(d\) centered at \(\tilde{x}_k\) it is not necessary to compute the distances from \(\tilde{x}_k\) of all the other \(N-1\) points, but only of those with labels \(r_k-1\), \(r_k\), \(r_k+1\), thus achieving a significant saving of computation time.
Figure 7. Labelling of the points based on their distances in $d$ units from an arbitrary point $\tilde{x}_o$ (large dot). Only the points with label $r_k-1$, $r_k$ or $r_k+1$ fall inside the sphere of radius $d$ centered at a point $\tilde{x}_k$ (asterisk) with label $r_k$.

In the proposed feature selection approach, the transient classifiability evaluation function has been combined with the classical sequential forward search engine described in Section 2.4.2.

3.2 A genetic algorithm-based wrapper approach

In the approach proposed in the last section two main criticalities may be identified. First of all, the classifiability evaluation function, as any filter evaluation function, attempts to assess the merit of the features directly from data geometric properties, ignoring the merit of the selected features with respect to the actual transient classification algorithm used. For this reason, a wrapper approach is employed in this Section. Secondly, the sequential forward selection search suffers from the “nesting effect”: the features added cannot be removed, so that a local minimum of the evaluation function is likely to be reached. For this reason, genetic algorithms (GAs), that are immune from this effect, are here considered as search engine for optimization.

Indeed, the problem of feature selection illustrated in Section 2 can be formulated as an optimization problem. In this view, given a set $X$ of $n$-dimensional input patterns, a GA can be devised to find an optimal binary transformation vector $V'$, of dimension $n$, which maximizes/minimizes a set of optimization criteria. Let $m$ be the number of 1’s in $V'$ and $n - m$ the number of 0’s. Then, a modified set of patterns $B = V'(X)$ is obtained in an $m$-dimensional space ($m < n$). Figure 8 shows the structure of a multi-objective GA-feature extractor that uses classification accuracy (to be maximized) and dimension $m$ of the transformed patterns (to be minimized) as optimization criteria. The GA creates a population of competing transformation vectors $V'_i, i = 1, 2, \ldots$, which are evaluated by the following steps [16]:

- The vector $V'_i$ is applied to each pattern of set $X$, giving a modified pattern which is then sent in input to the classifier.
- The set $B$ of modified patterns thereby obtained is divided into a training set, used to train the classifier, and a testing set, used to evaluate the classification accuracy on new patterns.
- The classification accuracy obtained and the number of selected features, $m$, are used by the GA as a measure of the quality of the transformation vector $V'_i$ used to obtain the set of transformed patterns.
On the basis of this feedback, the GA conducts its probabilistic search for a vector or a set of vectors which give rise to the best compromise among classification accuracy and parsimony in the selection of features.

For the GA feature selector of Figure 8, the organization of the chromosome is quite straightforward [17]: each bit of the chromosome is associated with a parameter (Figure 9) and interpreted such that if the \( j \)-th bit equals 1, then the \( j \)-th parameter is included as feature in the pattern for classification or vice versa if the bit is 0. Note that contrary to other GA applications, in this case of feature selection the binary chromosome does not encode real-valued control factors: the information regarding the features presence or absence in the optimal set for classification is included in the bits themselves so that no decoding is necessary.

Concerning the classification accuracy fitness function, in this work each subset of features encoded in a chromosome is evaluated on a set of testing data using a nearest neighbor classifier. More specifically, in the applications which follow in Section 4, the total number of pre-labelled patterns available is randomly subdivided into training and test sets consisting of 75% and 25% of the data, respectively. The Fuzzy K-Nearest Neighbor algorithm (FKNN) [18], with \( K=5 \), has been applied to classify the test data on the basis of the location of the labelled training data. The obtained fuzzy partition of the test data \( \{ \mu_i(\vec{x}_k) \} \), where \( 0 \leq \mu_i(\vec{x}_k) \leq 1 \) is the membership function of pattern \( \vec{x}_k \) to class \( i \), is converted into a hard partition assigning each pattern to the class with highest membership value. The random subdivision of the available patterns in training and test sets is repeated 10 times (10 cross-validation): the mean recognition rate, i.e. the mean fraction of correct classifications [19], is calculated and sent back to the GA as the fitness value of the transformation chromosome used to produce the transformed set of patterns \( B \).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure8.png}
\caption{GA-based feature selection using classification accuracy and number of selected features as optimization criteria. Each binary chromosome from the GA population is used to transform the original patterns, which are then passed to a classifier. The objective function values of the chromosome are the classification accuracy attained on the transformed patterns \( B \) and its dimension \( m \).}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure9.png}
\caption{\( n \)-dimensional binary chromosome.}
\end{figure}

In [20], a single-objective GA method, based on the FKNN mean recognition rate as fitness function, has been applied to select the relevant features from a large number of measured parameters in a nuclear transient
classification problem. The algorithm finds an optimal subset of features with a significantly better recognition rate than the case with all the features. The results remain, however, only “black-box”-appreciable because the feature subset is still too large to be interpreted from a physical point of view. In order to drive the GA search towards more parsimonious solutions in terms of number of features retained in the optimal subset, three strategies have been explored:

1. hierarchical 2-stage search procedure which progressively screens unimportant features in successive single-objective GA searches;
2. two objective search in which the maximization of the classification accuracy and minimization of the number of selected features are simultaneously sought by aggregating the corresponding objective functions in a weighted sum which is then subject to the optimization in order to drive the search towards effective and parsimonious solution;
3. Pareto-based multi-objective genetic algorithm (MOGA) search in which the two objectives of procedure 2. above are kept separate so that the population is allowed to evolve towards alternative, Pareto equivalent feature subsets offering different trade-offs in terms of Pareto accuracy and dimension of the feature space.

With respect to the case study tested in [20], the Pareto based multi-objective GA seems the most interesting, since it finds a variety of solutions with different trade-offs between classification performance and dimensionality reduction. In the next subsection the main concepts of this method are introduced.

### 3.2.1 Multi-Objective Genetic Algorithm (MOGA)

In a multi-objective optimization problem, several possibly conflicting objective functions \( f_i(), \quad i=1,2,...,n_f \), must be evaluated in correspondence of each decision variable vector \( U \) in the search space. The goal is to identify the solution vector \( U^* \) which gives rise to the best compromise among the various objective functions. The comparison of solutions is achieved in terms of the concepts of Pareto optimality and dominance [14, 16]: with reference to a maximization problem solution \( U_a \) is said to dominate \( U_b \) if

\[
\forall i \in \{1,2,...,n_f\}, f_i(U_a) \geq f_i(U_b) \\
\text{and} \\
\exists j \in \{1,2,...,n_f\}, f_j(U_a) > f_j(U_b).
\]

The decision variable vectors which are not dominated by any other of a given set are called nondominated with respect to this set; the decision variable vectors that are nondominated within the entire search space are said to be Pareto optimal and constitute the so called Pareto optimal set or Pareto optimal front.

In Pareto-based GA methods, reported first in [14], once a population of chromosomes has been created, these are ranked according to the Pareto dominance criterion by looking at the \( n_f \)-dimensional space of the fitnesses \( f_j(U) , j=1,2,...,n_f \). Firstly, all nondominated individuals are identified and rank 1 is assigned to them. Then, these solutions are virtually removed from the population and the next set of nondominated individuals are identified and assigned rank 2; this process continues until every solution in the population has been ranked. Every solution belonging to the same rank class has to be considered equivalent to any other of the same class in the sense that it has the same probability of the others to be selected as a parent for the mating.

During the optimization search, an archive of solution vectors \( U \), each one constituted by a nondominated chromosome and by the corresponding \( n_f \) fitnesses, representing the dynamic Pareto optimality front is recorded and updated [21,22,23]; this procedure allows implementation of elitism in MOGAs: every individual in the archive (or a pre-established fraction) is chosen once as a parent in each generation guaranteeing a better propagation of the genetic code of nondominated solutions, and thus a more efficient evolution of the population towards Pareto optimality.
At the end of the search procedure the result of the optimization is constituted by the archive itself which gives the Pareto optimality front.

4. CASE STUDY: CLASSIFICATION OF TRANSIENTS FOR FAULT DIAGNOSIS IN THE FEEDWATER SYSTEM OF A BOILING WATER REACTOR

In this Section, the two novel techniques for feature selection in transient classification problems proposed in Section 3 are applied to a case regarding nuclear transients. The diagnostic problem considered regards the early identification of a predefined set of faults in a Boiling Water Reactor (BWR). The corresponding transients have been simulated by the HAMBO simulator of the Forsmark 3 BWR plant in Sweden [24].

Figure 10. Sketch of the feedwater system.

The considered faults occur in the section of the feedwater system 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. Figure 10 shows a sketch of the system. Process experts have identified a set of 18 faults [25,26] that are generally hard to detect for an operator and that produce efficiency losses if undetected. Two power levels, 50% and 80% of full power, have been considered in this work.

The faults may be divided in three categories:
1. F1-F5, F7 regard the line 1 of the feedwater system.
2. F11-F15, F17 regard the line 2 of the feedwater system.
3. F6,F8,F9,F10,F16,F18 regard both lines.

At both power levels, five transients were simulated for each of the 18 faults, considering different degrees of leakage and valve closures and with step and ramp changes. All transients start after 60 seconds of steady state operation. From the analysis developed in [27,28], it is clear that faults F6,F10,F16 have no significant consequence on the plant measured parameters because the size of the leakage considered is too small. Hence these faults were not considered further. The data relative to \( n=363 \) features were recorded with a sampling frequency of 1Hz.

For the classification purposes of the present work, this number was reduced to 123 by combining the redundant measurements of the same physical quantity and by eliminating those features linearly correlated or carrying
basically the same time evolution in all the different faults. This pre-processing of the data was done in a controlled way so as to ensure that information redundancy were still kept, with the aim of making the model robust with respect to sensors degradation and failure.

Since the goal is early diagnosis, only the data of the first 65 seconds after the beginning of the transient (i.e. 66 data points from 60s to 125s) have been considered. Moreover, given that the ramp changes cause variations of the features later than the step changes, only the three step changes for each fault are considered.

Given the complexity of the evaluation function computation, the number of data points has been reduced to 7 for each transient by taking values every 10 seconds. Thus, the total number of points employed for the feature selection is 315 (7 points for each of the 45 transients).

The transients at 50% of full power were used for the feature selection task, whereas the transients at 80% of full power were left out for validation of the resulting classifier. Although all at 50% power level, the transients for the feature selection begin from slightly different initial parameter values due to the fact that some faults, like for example those involving boron dilution, are associated to processes with very long characteristic times [25]. To avoid that for this reason the feature selection and classification algorithms are applicable only from a specific steady state of the reactor and to allow the validation on the transients at 80% power, the difference between the feature values at the generic sampling time during the transients and their steady-state values before the beginning of the transients are used.

Validation of the performance of the classifiers based on the optimal feature subsets identified by the GA has been carried out with respect to the transients at 80% of full power, none of which has been used during the feature selection phase. Treating these transients analogously to those at 50% power, a total of 315 patterns are obtained for the validation set. The different feature selection techniques are then compared with respect to the mean recognition rate resulting from a 25-fold cross validation performed on such validation set. The 315 data points of the validation set were randomly subdivided into training and test sets consisting of 75% and 25% of the data, respectively. The Fuzzy K-Nearest Neighbor algorithm (FKNN) [18], with $K = 5$ was applied to classify the test data on the basis of the location of the labelled training data. The obtained fuzzy partition of the test data \{$\mu_i(x_k)$\}, where $0 \leq \mu_i(x_k) \leq 1$ is the membership function of data point $x_k$ to class $i$, has been converted in a hard partition assigning each data point to the class with highest membership value. The mean recognition rate, i.e. the fraction of correct classifications [19], obtained by repeating 25 times the random subdivision of the available data in train and test sets, has been employed for the comparison of the different feature selection techniques.

For comparison purposes, Table 2 reports the mean recognition rate and the standard deviation obtained by the classifier based on all the 123 available features and on the feature subset selected by experts of the plant process (Table 8 in appendix A). Notice that there is a significant improvement in the classification performance with the feature subset selected by the expert compared with the feature subset formed by all the available 123 features both considering the test set and the validation set. This confirms the advantage of using a feature selection technique to support the classification task, otherwise based on a large number of available features. Finally, notice that the recognition rates achieved on the validation set obtained from transients at 80% power level are in general higher than those achieved on the tuning set made of patterns from transients at 50% power. This is due to the fact that signal variations are more relevant when the reactor is working at a higher power level so that the classification task becomes somewhat simpler.
Table 2. Classification performances achieved by the FKNN algorithm.

<table>
<thead>
<tr>
<th>Feature subset</th>
<th>Number of features in the optimal feature subspace</th>
<th>FKNN recognition rate on the data used for the feature selection (50% power level)</th>
<th>FKNN recognition rate on the validation set (80% power level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All available features</td>
<td>123</td>
<td>0.534 ± 0.035</td>
<td>0.647 ± 0.055</td>
</tr>
<tr>
<td>Selected by the expert</td>
<td>17</td>
<td>0.679 ± 0.034</td>
<td>0.789 ± 0.029</td>
</tr>
</tbody>
</table>

4.1 Filter approach results

In this section, the filter feature selection technique described in Section 3.1 is applied. The original classifiability evolution function, $L$, and its extensions $L^*$, $L^{**}$ are considered combined with a sequential forward selection search engine.

Figure 11 shows the evolution of the classifiability functions $L$, $L^*$ and $L^{**}$ as features are added. At the first step, the algorithm for the computation of $L^{**}$, for example, evaluates 123 values of $L^{**}$, one for each feature, and finds that a maximum value of -0.50 is achieved by the one-dimensional feature subspace that contains parameter 320 (position level control valve for EA1). At the next step, the $L^{**}$ values of the 122 possible two-dimensional feature subspaces that contain feature 320 and another one among the remaining 122 are computed, and the maximum value of -0.12 is obtained by the feature subspace {320,321}. Starting from this point, the procedure then moves to considering three-dimensional feature subspaces and so on.

Note that the $L$ function starts decreasing beyond the three dimensional optimal feature subspace, $S_3^{opt}$. This confirms that the classical definition of classifiability cannot identify features salient for transient classification because of the high importance that it gives to the points very close to the steady state (see Section 3.2). On the contrary, $L^*$ identifies an optimal subspace of 12 features while the weighed classifiability $L^{**}$ finds the best, optimal space with 16 features.

![Figure 11: Evolution of the classifiability functions $L$, $L^*$ and $L^{**}$ with the number of features in the subspace. The line with diamonds refers to $L$ (eq.1 in Section 2.3.1.1), the line with circles to $L^*$ (eq.4 in Section 3.1) and the line with squares to $L^{**}$ (eq.6 in Section 3.1).](image-url)
The features selected by using the different definitions of classifiability and the selection of 17 features declared relevant by expert judgment are reported in Table 13 in Appendix A. It turns out that 1 out of 3 features in the optimal subspace identified by using $L$ appears also in the expert list, 7 out of 12 in the case of $L^*$ and 11 out of 16 for $L^{**}$. Given the latter result, it seems justified to say that there is a fair agreement between the feature selection by the weighed algorithm proposed, based on $L^{**}$, and the expert selection.

Next, we examine the performances of the selected feature subspaces when applied to the FKNN algorithm. Table 3 reports a comparison between the feature subset selected by the different techniques. The feature subset obtained by using the evaluation function $L$ leads to a completely unsatisfactory classification. On the contrary, those obtained by $L^*$ and $L^{**}$ allow achieving a higher classification accuracy, and with less features, than the feature subspace proposed by the expert. This confirms the advantage of using a feature selection technique to support or replace the expert judgement in selecting the features relevant for transient classification.

The best recognition rate is achieved by the subspace identified using the weighed classifiability evaluation function $L^{**}$ considering both the data at 50% power level used for the feature selection and the validation data at 80% power level. The features thereby selected turn out to distinctly regard either only line 1, only line 2 or both lines [27].

Table 3. Classification performances achieved by FKNN algorithm.

<table>
<thead>
<tr>
<th>Feature selection technique</th>
<th>Number of features in the optimal feature subspace</th>
<th>FKNN recognition rate on the data used for the feature selection (50% power level)</th>
<th>FKNN recognition rate on the validation set (80% power level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All available features</td>
<td>123</td>
<td>0.534 ± 0.035</td>
<td>0.647 ± 0.055</td>
</tr>
<tr>
<td>Selected by the expert</td>
<td>17</td>
<td>0.679 ± 0.034</td>
<td>0.789 ± 0.029</td>
</tr>
<tr>
<td>$L$</td>
<td>3</td>
<td>0.253 ± 0.040</td>
<td>0.256 ± 0.051</td>
</tr>
<tr>
<td>$L^*$</td>
<td>12</td>
<td>0.748 ± 0.042</td>
<td>0.883 ± 0.034</td>
</tr>
<tr>
<td>$L^{**}$</td>
<td>16</td>
<td>0.813 ± 0.041</td>
<td>0.899 ± 0.039</td>
</tr>
</tbody>
</table>

4.2 Wrapper approach results

In this Section, the GA-based wrapper feature selector described in Section 3.2 is applied.

In particular, the potentialities of a Pareto-based approach (Section 3.2.1) are investigated with respect to the goal of identifying equivalently optimal feature subsets characterized by different performances in terms of classification accuracy (FKNN mean recognition rate) and number of features.

The performance of a Pareto-based MOGA depends largely on its ability to maintain genetic diversity while attempting to produce a population of individuals uniformly distributed in the vicinity of the various members of the Pareto front itself, so as to achieve a front widely representative of the real nondominated solutions [14, 29]. Thus, the effects on the genetic diversity of different combinations of selection (Sel) and replacement (Repl) procedures, probabilities of mutation ($P_m$), population sizes ($Pop\_size$) and number of generations ($N\_gen$), have been studied by performing the following MOGA computations:

- A1: {Sel: random; Repl: fittest; $P_m = 0.001$; $Pop\_size = 100; N\_gen = 500$};
- A2: {Sel: random; Repl: fittest; $P_m = 0.008$; $Pop\_size = 100; N\_gen = 500$};
- A3: {Sel: random; Repl: fittest; $P_m = 0.001$; $Pop\_size = 200; N\_gen = 1000$};
- B1: {Sel: fit-fit; Repl: children-parents; $P_m = 0.001$; $Pop\_size = 100; N\_gen = 500$};
- B2: {Sel: fit-fit; Repl: children-parents; $P_m = 0.008$; $Pop\_size = 100; N\_gen = 500$};
- C1: {Sel: fit-fit; Repl: fittest; $P_m = 0.001$; $Pop\_size = 100; N\_gen = 500$};
Most of these MOGA strategies and parameter values are widely adopted in literature [30]: in particular, \( P_m = 0.001 \) is recommended in [31], whereas \( P_m = 0.008 \) is the inverse of the number of bits constituting a chromosome, as suggested in [32, 33 and 34]. The fraction of the population chosen through elitist selection [15] in each generation is set to 0.25 in all cases.

Table 4 contains the objective function values (number of selected features and mean recognition rates) of the sets of nondominated solutions for the different MOGA settings. With respect to the dimension \( m \) of the Pareto solutions, note that it can range from 1 to 13 but only case C2 identifies a complete set from \( m = 1 \) to \( m = 13 \). Also, note that two different MOGA computations may reach a solution with an equal number \( m \) of different features, to which correspond different mean recognition rates (e.g. cases A1, A2, B1, B2 for \( m = 3 \)). Figure 12 and Figure 13 show the corresponding Pareto frontiers, the final populations and the performance achieved by the reference FKNN classifier which uses all the 123 signals (for graphical reasons, this latter is reported out of scale with respect to the number of features, at \( m = 19 \) instead of \( m =123 \)).

From the results obtained, general considerations can be drawn about the effect of different genetic operators on the structure of the Pareto optimal set and on the convergence of both population and archive towards the Pareto front:

- \( P_m \): low probabilities of mutation (0.001, as in MOGA configurations A1, A3, B1, C1) allow the population to uniformly converge towards the optimal frontier; higher values of \( P_m \) act negatively when combined with fitness-ignoring replacement procedures (children-parents, as in MOGA configuration B2), reducing the extension of the front and producing a scattered final population.
### Table 4. Objective function values of the nondominated solutions found by the Pareto-based MOGA for different combinations of genetic operators.

<table>
<thead>
<tr>
<th>Number of features in the transformed patterns, $m$</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>B1</th>
<th>B2</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3431</td>
<td>0.3431</td>
<td>0.3431</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>0.3431</td>
<td>0.3431</td>
</tr>
<tr>
<td>2</td>
<td>0.6662</td>
<td>0.6662</td>
<td>0.6662</td>
<td>/</td>
<td>/</td>
<td>0.6662</td>
<td>0.6662</td>
<td>0.6662</td>
</tr>
<tr>
<td>3</td>
<td>0.7960</td>
<td>0.8159</td>
<td>0.8159</td>
<td>0.6776</td>
<td>0.7721</td>
<td>0.7960</td>
<td>0.8159</td>
<td>0.8159</td>
</tr>
<tr>
<td>4</td>
<td>0.8310</td>
<td>0.8313</td>
<td>0.8313</td>
<td>0.7537</td>
<td>0.8022</td>
<td>0.8310</td>
<td>0.8313</td>
<td>0.8313</td>
</tr>
<tr>
<td>5</td>
<td>0.8374</td>
<td>0.8376</td>
<td>0.8376</td>
<td>0.7949</td>
<td>0.8310</td>
<td>0.8374</td>
<td>0.8393</td>
<td>0.8393</td>
</tr>
<tr>
<td>6</td>
<td>0.8395</td>
<td>0.8401</td>
<td>0.8404</td>
<td>0.8290</td>
<td>0.8404</td>
<td>0.8388</td>
<td>0.8455</td>
<td>0.8455</td>
</tr>
<tr>
<td>7</td>
<td>0.8444</td>
<td>0.8444</td>
<td>0.8444</td>
<td>0.8444</td>
<td>/</td>
<td>0.8421</td>
<td>0.8488</td>
<td>0.8488</td>
</tr>
<tr>
<td>8</td>
<td>0.8547</td>
<td>0.8547</td>
<td>0.8547</td>
<td>0.8547</td>
<td>/</td>
<td>0.8534</td>
<td>0.8547</td>
<td>0.8547</td>
</tr>
<tr>
<td>9</td>
<td>0.8584</td>
<td>0.8584</td>
<td>0.8584</td>
<td>0.8584</td>
<td>0.8444</td>
<td>0.8572</td>
<td>0.8584</td>
<td>0.8584</td>
</tr>
<tr>
<td>10</td>
<td>0.8598</td>
<td>0.8598</td>
<td>0.8598</td>
<td>0.8598</td>
<td>0.8508</td>
<td>0.8598</td>
<td>0.8598</td>
<td>0.8598</td>
</tr>
<tr>
<td>11</td>
<td>0.8610</td>
<td>0.8610</td>
<td>0.8610</td>
<td>0.8610</td>
<td>0.8606</td>
<td>/</td>
<td>0.8622</td>
<td>0.8610</td>
</tr>
<tr>
<td>12</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>0.8624</td>
<td>/</td>
<td>0.8647</td>
<td>/</td>
</tr>
<tr>
<td>13</td>
<td>0.8613</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>0.8648</td>
<td>/</td>
<td>0.8658</td>
<td>/</td>
</tr>
</tbody>
</table>

On the other hand, in strongly fitness-oriented searches (e.g. fittest replacement procedures), large values of $P_m$ positively improve genetic diversity, favouring a deep search of the feature space and producing a wide Pareto optimal front (MOGA configurations A2, C2, C3);

- **Repl**: fitness-guided replacement procedures (e.g. fittest) efficiently move the population towards solutions on the optimal frontier; on the other hand, when they are combined with fitness-guided selection methods (e.g. fit-fit), genetic diversity must be ensured by means of a relatively high mutation probability to avoid entrapment in local optima and to obtain an extended front (MOGA configuration C1 vs C2);

- **Sel**: selection seems less determining than replacement; however, it is interesting to observe that the combination of a uniform sampling of the population (random selection) with a fitness-guided replacement procedure (e.g. fittest) helps improving genetic diversity and finding a wide optimal frontier (MOGA configuration A1 vs C1).

MOGA configuration C2 presents the best Pareto set in terms of completeness, homogeneity and classification accuracy of its members: its characteristics are summarized in Table 5.
Table 5. Detailed description of the best Pareto optimal frontier corresponding to MOGA configuration C2.

<table>
<thead>
<tr>
<th>Number of features in the transformed patterns, $m$</th>
<th>Mean recognition rate</th>
<th>Included features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3431 ± 0.0395</td>
<td>$S_1^{opt,Pareto}$</td>
</tr>
<tr>
<td>2</td>
<td>0.6662 ± 0.0380</td>
<td>$S_2^{opt,Pareto}$</td>
</tr>
<tr>
<td>3</td>
<td>0.8159 ± 0.0325</td>
<td>$S_3^{opt,Pareto}$</td>
</tr>
<tr>
<td>4</td>
<td>0.8313 ± 0.0442</td>
<td>$S_4^{opt,Pareto}$</td>
</tr>
<tr>
<td>5</td>
<td>0.8393 ± 0.0379</td>
<td>$S_5^{opt,Pareto}$</td>
</tr>
<tr>
<td>6</td>
<td>0.8455 ± 0.0425</td>
<td>$S_6^{opt,Pareto}$</td>
</tr>
<tr>
<td>7</td>
<td>0.8488 ± 0.0368</td>
<td>$S_7^{opt,Pareto}$</td>
</tr>
<tr>
<td>8</td>
<td>0.8547 ± 0.0344</td>
<td>$S_8^{opt,Pareto}$</td>
</tr>
<tr>
<td>9</td>
<td>0.8584 ± 0.0363</td>
<td>$S_9^{opt,Pareto}$</td>
</tr>
<tr>
<td>10</td>
<td>0.8598 ± 0.0384</td>
<td>$S_{10}^{opt,Pareto}$</td>
</tr>
<tr>
<td>11</td>
<td>0.8622 ± 0.0367</td>
<td>$S_{11}^{opt,Pareto}$</td>
</tr>
<tr>
<td>12</td>
<td>0.8647 ± 0.0345</td>
<td>$S_{12}^{opt,Pareto}$</td>
</tr>
<tr>
<td>13</td>
<td>0.8658 ± 0.0339</td>
<td>$S_{13}^{opt,Pareto}$</td>
</tr>
</tbody>
</table>

The features (Appendix B) included in these nondominated solutions are:

- $S_1^{opt,Pareto} = \{321\};$
- $S_2^{opt,Pareto} = S_1^{opt} + \{320\};$
- $S_3^{opt,Pareto} = S_2^{opt} + \{357\};$
- $S_4^{opt,Pareto} = S_3^{opt} + \{193\};$
- $S_5^{opt,Pareto} = \{59, 192, 194, 320, 321\};$
- $S_6^{opt,Pareto} = S_5^{opt} + \{193\};$
- $S_7^{opt,Pareto} = S_6^{opt} + \{240, 241\};$
- $S_8^{opt,Pareto} = \{139, 160, 195, 213, 241, 242, 320, 321\};$
- $S_9^{opt,Pareto} = S_8^{opt} + \{359\};$
- $S_{10}^{opt,Pareto} = S_9^{opt} + \{194\};$
- $S_{11}^{opt,Pareto} = \{60, 77, 140, 195, 241, 242, 320, 321, 324, 325, 330\};$
- $S_{12}^{opt,Pareto} = S_{11}^{opt} + \{359\};$
- $S_{13}^{opt,Pareto} = S_{12}^{opt} + \{143\}.$
In a single run, the Pareto-based MOGA identifies a range of nondominated solutions with different complexity/performance trade-offs: once such a subset is available, an informed choice can be made of the features to be actually monitored for the diagnostic task, by considering diagnostic classification accuracy and possibly also practical issues related to costs and ease of data acquisition.

![Figure 12. Representation, in the objective functions space, of Pareto optimal frontiers, last-generation populations and an out-of-scale indication of the classification accuracy obtained by a FKNN classifier using all the 123 features for configurations A1, A2, B1, B2 of the MOGA](image1)

![Figure 13. Representation, in the objective functions space, of Pareto optimal frontiers, last-generation populations and an out-of-scale indication of the classification accuracy obtained by a FKNN classifier using all the 123 features for configurations A3, C1, C2, C3 of the MOGA](image2)

It is significant that $S_{3 \text{opt,Pareto}}$ achieves a classification accuracy comparable to those of other nondominated feature subsets characterized by higher numbers of features. The time evolutions of the features contained in $S_{3 \text{opt,Pareto}}$ show that feature 357, which evolves in a characteristic way in case of faults F8, F9, F12 and F18,
completes the information borne by features 320 and 321 giving rise to the efficient recognition of almost all the faults.

Further, it is notable that the core of features in $S_{11}^{opt.\text{Pareto}}$ is also maintained in $S_{12}^{opt.\text{Pareto}}$ and $S_{13}^{opt.\text{Pareto}}$, the latter feature set achieving slightly higher classification accuracy thanks to the additions of features 359 and 143.

Validation of the classifier applied to the transients at 80% power level has confirmed the above findings.

4.3 Comparison of the results obtained by the two techniques.

The solutions found in Section 4.2 by the GA-based wrapper feature selector are more satisfactory than the solutions found in Section 4.1 by the filter approach, both from the point of view of the final classification performance on the test set and of the number of selected features. In fact, Figure 14 shows that all the solutions on the Pareto front with a number of features greater than 2 have higher classification accuracy than the optimal feature subset selected by the filter method. For example, considering the classification performance obtained on the validation set, the 11 features in $S_{11}^{opt.\text{Pareto}}$ allow achieving a recognition rate of 0.899 ± 0.033 which is of the same order of that obtained by the filter approach considering $L^*$, but with 16 features.

![Figure 14. Comparison of the feature subsets selected by the different techniques and out of scale indication of the recognition rate obtained by a FKNN classifier using all the 123 features.](image)

Note that contrary to the sequential forward method, which adds one input at a time, the GA approach may find Pareto-front solutions by an increase in the number of selected features which not necessarily implies simple addition of one feature. Instead, it may be the case that some features are removed and others are added: for example, when moving from $m=4$ to $m=5$, features 193 and 357 are removed whereas 59, 192 and 194 are added (Table 6).

Also, as explained earlier, the wrapper approach is more performing since it ensures the selection of the features more suitable specifically for the FKNN classification algorithm, whereas the filter approach totally ignores the effects of the selected feature subspace on the classification algorithm.

On the other hand, the computation time required by the filter approach has been approximately 10 times shorter than the computation time required by the wrapper approach (Table 6). Notice, however, that the feature
selection is performed only once, off line, before the developing of the classification model. Thus, a long computational time is not a strong limitation to the use of genetic algorithm-based wrapper feature selectors.

Table 6. Computational time on a 2.66GHz personal computer.

<table>
<thead>
<tr>
<th>Feature Selector</th>
<th>Computational time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter approach</td>
<td>74 minutes</td>
</tr>
<tr>
<td>GA-based feature selector</td>
<td>731 minutes</td>
</tr>
</tbody>
</table>
5. DATA PRE-PROCESSING FOR FEATURE SELECTION

Given the intrinsically dynamic nature of the transient classification problem, techniques for the representation of the transients in a more compact form have been largely studied [35,36,37,38]. In particular, in [1,39] it has been shown that a feature extraction phase, based on a wavelet decomposition of the transient data, increases significantly the classification accuracy. In this Section, feature selection by the previously illustrated GA-based wrapper approach is investigated on the features obtained from a pre-processing feature extraction of the transient data (Figure 15).

![Feature Extraction Diagram](image)

Figure 15. Data pre-processing and feature selection scheme

For the extraction of the features, the wavelet on line pre-processing tool (WOLP) has been considered. Wavelet transform (see [40] for a general reference on wavelets) is a signal decomposition technique closely related to the classical Fourier transform. Just as the Fourier transform is based on the notion of expressing a time series as a linear combination of sinusoids, the idea underlying the wavelet transform is to express the time series as a linear combination of wavelets. In Fourier analysis, each sinusoid is associated with a particular frequency, so that one can deduce what frequencies are most important in a particular signal by observing the magnitudes of the coefficients of the various sinusoids in the linear combination. In contrast to sinusoids, wavelets are functions that are nonzero only inside a specific interval of time where they spend an almost equal amount of time below and above zero, therefore appearing as "small waves". This property implies that the description of an event is strictly related to the time during which it occurs. Instead of transforming a pure “time description” into a pure “frequency description” (as is the case with the Fourier transform), the wavelet transform provides a good compromise by generating a time-frequency description. By decomposing the original signal using wavelets at different scales (i.e. of different duration), one can learn how a time series varies on particular scales across time. These properties make the wavelet transform a much more adequate tool for transient analysis, where an accurate localization in time of signal features is often required for a correct classification of the courses generating the transients.

In the next section, the Wavelet on line pre-processing module, taken from the ALLADIN transient classifier [1,39], is illustrated.

5.1 Wavelet on line pre-processing

The solution here proposed for the feature extraction from multivariate transient data is based on an on-line procedure for the extraction of compact wavelet features. The features are extracted by Haar wavelet
decomposition [40] from a sliding window on the actual signal time-series. The selected wavelet features are the following:

- The mean residual signal taken at the highest, i.e. coarsest, scale.
- The maximum wavelet coefficient over all the scales.
- The minimum wavelet coefficient over all the scales.

The rationale behind this choice is that (a) captures the general trend of the signal across the windows in a compact way, while (b) and (c) capture important variations in the signal within a single window (e.g. downward or upward trends, step changes, spikes, etc.) which would otherwise be severely smoothed out by the compression process. The window size is selected so as to correspond to wavelet dyadic decomposition values (i.e. powers of 2), and consecutive windows are chosen with a slight overlap to avoid missing features that might be present at the window borders. Since we are using the Haar wavelet on a dyadic window, there is no need to be concerned with edge effects of the wavelet transform.

Because of the ability of continuously applying the wavelet transform on a sliding window, and since the transform is used as a pre-processing step for the final transient classification, the technique is named Wavelet On-Line Pre-processing (WOLP).

For clarification on how the WOLP pre-processing actually works in practice, Figure 16, Figure 17 and Figure 18 show three examples of application to actual process signals. The upper graphs in these Figures show 300 points of the original signal data stream. Also shown in the upper graphs are the overlapping windows from which the wavelet features are dynamically extracted. In these examples, the windows are 32 patterns long and the overlap is 5 patterns. This means that a new wavelet transform is applied each time 27 new patterns are acquired from the process.

The lower graph in the Figures shows the output of the WOLP pre-processing. WOLP generates three data streams for each original signal, at a rate equal to the original sampling rate divided by the window slide step size, which in this case is 27. In the example of Figure 16, an original data stream of 275 samples generates three data streams of 10 samples each.

**Figure 16. Example 1 of application of WOLP to a process signal**
Looking at the correspondence between the original signal in Figure 16 and the WOLP output, one sees that the WOLP transformation produces a very compact, and yet very significant, description of the original signal. The mean residual feature reflects the general trend of the signal just as the average signal value within the analysis window would. Actually, there are no technical reasons for choosing one representation over the other. The choice of using the wavelet mean residual is mainly for consistency with the other WOLP outputs.

The value of the maximum coefficient reflects negative trends, step changes, and the negative components of spikes within the analysis window. In a dual way, the minimum coefficient reflects the positive trends, step changes, and positive components of spikes. For example, in the third window, a big increase in the signal value corresponds to a big value of the minimum coefficient and a small decrease in the signal value (in correspondence with the two small peaks) is matched by a small value of the maximum coefficient. When compared with the data corresponding to the second window, one also appreciates how the WOLP transformation maintains the correct proportions of the registered changes: the signal decrease in the second window is roughly double the signal increase in the third window and, correspondingly, the maximum coefficient in the second window is roughly double the minimum coefficient in the third window.

The choice of window size in WOLP can be used to strike a balance between a high level of transient compression and a resolution still sufficient to discriminate among the event classes. In the example of Figure 16, one can see that the two small peaks in the third window contribute to a single set of wavelet features. A similar transient, containing only one peak, would generate a practically identical WOLP output. If this transient were to correspond to a different event class one would not be able to discriminate the two. In this case, a smaller window size (i.e. smaller than the distance between the two peaks) would be necessary in order to make the WOLP outputs distinguishable.

Figure 17. Example 2 of application of WOLP to a process signal

The example in Figure 17 shows the WOLP outputs obtained on another process signal. Again the mean residual closely follows the global behaviour of the signal, while the two wavelet coefficients accurately capture the detailed behaviour within each analysis window.
The example shown in Figure 18 is also interesting because it clearly highlights the ability of WOLP of capturing a wide range of behaviours. The big drop of the signal in window one is clearly recorded in a correspondingly very big maximum wavelet coefficient component of that window. The wavelet coefficients, not only capture the size of the change, but also its steepness. This can be seen by comparing this signal with the signal of Figure 16. In both cases the observed drop in value is close to 1.6, however, the drop is gradual in one case, and abrupt in the other. This is reflected in a much larger value of the wavelet coefficient in the second case (about 2.6, compared with 1.2 in the first case). At the same time, in the example of Figure 18, WOLP is shown capable of capturing also proportionally much smaller features, such as the small step changes in windows three, five, eight, nine, and ten.

5.2 The WOLP-GA-based wrapper feature selector applied to the classification of nuclear transients.

In this Section the complete scheme of Figure 15, implemented by the WOLP feature extractor followed by the GA-based wrapper feature selector is applied to the nuclear transient classification problem of Section 4.

To balance between a high level of transient compression and an acceptable resolution, time windows which are 16 patterns long, with an overlap of 6 seconds, have been chosen. Then, the evolution of a wavelet coefficient in a given transient from t=58s to t=133s is summarized in 7 points, each one corresponding to a time-window.

Figure 19 shows the seven time windows considered and an example of application of WOLP to the evolution of signal 320 in a transient of class 3. It is interesting to observe the behavior of this signal in the first time window. The big value of the maximum wavelet coefficient catches the decrease of the signal, whereas the small negative value of the minimum wavelet coefficient catches the slight rise. If, as done in the previous Sections, the feature selection were applied directly to the plant measured parameters without WOLP pre-processing, only the values of the signal at the time steps t=65,75,85,...,125s would be considered (Figure 20) and no information on the behavior of the signal between the time steps is provided to the classifier.
Figure 19. Time evolution of Signal 320 in a transient of class 3 (top) and the 3 features obtained from it by the wavelet transformation.

Figure 20. Feature obtained from Signal 320 directly considering the signal value at the time steps t=65, 75, 85, ..., 125s.

The GA-based wrapper approach to feature selection illustrated in Section 3.2 has been applied to the 369 wavelet coefficients extracted by the WOLP pre-processing from the 123 original plant signals. Note that the feature space is three-fold increased, and, correspondingly, the number of candidate solutions goes from $2^{123}$ to $2^{369}$.

For what concerns the GA parameters and search strategy, the selection procedure ($Sel$) and the probability of mutation ($P_m$) have been fixed to the optimal settings found in the previous Section ($Sel$=fit-fit and $P_m$=0.008), whereas three different combinations of replacement procedures ($Repl$), population sizes ($Pop_size$) and number of generation ($N_gen$) have been studied. Thus, the following MOGA computations have been performed:

A: \{Repl: fittest; Pop_size = 100; N_gen = 500\}

B: \{ Repl: fittest; Pop_size = 200; N_gen = 1000\}

C: \{ Repl: 50% of the cases fittest, 50% weakest; Pop_size = 200; N_gen = 1000\};

Figure 21 shows the last populations and the Pareto optimal frontiers obtained by these three procedures. Procedure A, although characterized by the same GA parameters and search strategy of the best configuration found in Section 4.2, is less satisfactory in terms of classification accuracy, completeness and homogeneity of the members of its Pareto set. This is due to a much larger search space ($2^{369}$ possible feature subsets) than in the previous case ($2^{123}$). To explore more effectively the search space, procedures B and C are devised, characterized
by larger populations. In both cases, convergence of the last population to the Pareto frontier is not achieved, although the number of generations has been increased to 1000, and the final population turns out to be very scattered. With procedure C, 50% of the times the weakest individual replacement procedure is applied: this results in no individuals in the Pareto front with a number of features greater than 30, because this replacement procedure shortens the permanence of the worst individuals in the population. Moreover, for what concerns the completeness of the Pareto front, solutions with 4 and 5 individuals are found only using procedure C. However, no solutions with less than 4 features have been obtained. In this respect, work is in progress in the area of new GA selection procedures based on a concept of niche evolution which allows for greater genetic diversity [41].

Figure 21. Representation, in the objective functions space, of the Pareto optimal front and the last generation population for GA procedures A, B and C. The last Figure compares the three obtained Pareto optimal fronts.

Table 7 summarizes the characteristics of the nondominated solutions of procedure C. A good solution with respect to both objectives is $S_{14}^{opt, Pareto+wolp}$, composed of 11 mean residuals, 2 minimum and 1 maximum wavelet coefficients (Table 8).

Validation of the classifier applied to the transients at 80% power level gives a recognition rate of 0.910 ± 0.030, confirming the satisfactory results.
Table 7. Pareto optimal front corresponding to MOGA procedure C.

<table>
<thead>
<tr>
<th>Number of features in the transformed patterns, m</th>
<th>Mean recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.7506 ± 0.0438</td>
</tr>
<tr>
<td>5</td>
<td>0.7805 ± 0.0381</td>
</tr>
<tr>
<td>6</td>
<td>0.8494 ± 0.0432</td>
</tr>
<tr>
<td>7</td>
<td>0.8818 ± 0.0357</td>
</tr>
<tr>
<td>8</td>
<td>0.8896 ± 0.0334</td>
</tr>
<tr>
<td>9</td>
<td>0.8933 ± 0.0355</td>
</tr>
<tr>
<td>10</td>
<td>0.8999 ± 0.0392</td>
</tr>
<tr>
<td>11</td>
<td>0.9068 ± 0.0358</td>
</tr>
<tr>
<td>12</td>
<td>0.9134 ± 0.0339</td>
</tr>
<tr>
<td>13</td>
<td>0.9176 ± 0.0331</td>
</tr>
<tr>
<td>14</td>
<td>0.9253 ± 0.0341</td>
</tr>
<tr>
<td>16</td>
<td>0.9264 ± 0.0338</td>
</tr>
<tr>
<td>17</td>
<td>0.9302 ± 0.0319</td>
</tr>
<tr>
<td>20</td>
<td>0.9302 ± 0.0321</td>
</tr>
<tr>
<td>21</td>
<td>0.9328 ± 0.0358</td>
</tr>
<tr>
<td>25</td>
<td>0.9329 ± 0.0361</td>
</tr>
</tbody>
</table>

Table 8. Features in Pareto optimal subset \( S_{14, \text{opt,Pareto+posp}} \).

<table>
<thead>
<tr>
<th>Type of wavelet coefficient</th>
<th>Signals from which the wavelet coefficient is extracted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean residual</td>
<td>100,141,151,195,242,314,320,321,324,325</td>
</tr>
<tr>
<td>Minimum wavelet coefficient</td>
<td>74,194</td>
</tr>
<tr>
<td>Maximum wavelet coefficient</td>
<td>146</td>
</tr>
</tbody>
</table>

To investigate the benefits of using the wavelet transformation for transient feature extraction, let us consider the minimum wavelet coefficient of signal 194, which turns out to be very efficient for the classification of transients caused by fault F4 (Figure 22). If simply the values of the signal at the time steps \( t=65,75,85,\ldots,125 \)s were used without any WOLP pre-processing for feature extraction, as done in Sections 3 and 4, it would be
impossible to identify this fault before the second time step (t=75s); on the contrary, the minimum wavelet coefficient shows a relatively significant variation already in the first time window allowing transients caused by fault F4 to be promptly discerned from transients caused by other faults in which this feature is constant.

Figure 22. Evolution of signal 194 in a transient of class 4 (top) and its wavelet transformed features.

In Figure 23, the Pareto fronts obtained with the feature selection on the original 123 plant signals (Section 4.2) are compared to those obtained from the 369 wavelet coefficients obtained by WOLP preprocessing. All solutions obtained from the wavelet coefficients with a number of used features $m>7$ are more satisfactory than the corresponding ones obtained directly from the 123 measured signals. This is due to the additional information carried by the wavelet coefficients that capture the entire evolution of the signal in the time window and not only some point values at given time steps.

On the other hand, for what concerns the part of the Pareto front with $m \leq 7$, the solutions obtained from the 123 measured signals dominate those obtained from the 369 wavelet coefficients. This is due to the difficulties in the convergence of the GA search on the significantly larger feature space. In fact, it was verified that if the feature selection is applied to only the 123 mean residuals wavelet coefficients, i.e. to a subset of the 369 wavelet coefficients obtained by WOLP, the GA is capable of finding solutions that dominate those obtained from the 123 measured signals not pre-processed by WOLP (Figure 23). These more satisfactory solutions are obviously included also in the bigger search space of all the 369 wavelet coefficients, but the GA is not capable of finding them, unless the number of generations is significantly increased.
Figure 23. Comparison of the Pareto fronts obtained by the MOGA-based feature selection applied on the 369 wavelet features (diamond) and on the 123 measured signals (*+, Section 4.2).

Figure 24. Comparison of the Pareto fronts obtained by the MOGA-based feature selection applied on the 123 measured signals (+) and on the 123 mean residual wavelet coefficients (*).

In conclusion, the combination of the WOLP data pre-processing phase with the GA-feature selection procedure has been found efficient in identifying the Pareto optimal feature subsets for transient diagnosis.
6. A GENETIC ALGORITHM FOR THE MINIMIZATION OF THE RECOGNITION DELAY TIME

Early diagnostic systems aim at recognizing transients in the first instants of their evolution so as to enable the operator to schedule the proper corrective and mitigation actions to protect from the adverse consequences of the malfunctioning [3]. Correspondingly, the feature selection phase has to be oriented towards the selection of features that allow the earliest classification of the transients. As a simple illustrative example, let us assume that the 2 features reported in Figure 25 are the ones sensible to a given malfunctioning: feature 1 would be more efficient for the early identification of the transient since it deviates from its stationary value before feature 2 does ($t_{10} < t_{20}$); yet feature 2 remains sensible to the transient-initiating cause for a longer time ($t_{2f} - t_{20} > t_{1f} - t_{10}$), thus possibly allowing for a more reliable classification. It appears, then, that the need of an early identification may be contradictory with the requirement of a high classification performance. To account for this in the feature selection phase, the concept of Recognition Delay Time (RDT) is introduced in the next Section as an additional fitness measure and a fuzzy formalization is provided to account for the uncertainty in the transient classification.

6.1.1 Definition of the Recognition Delay Time and its fuzzy extension

Consider an online diagnostic system. The delay time in the diagnosis, i.e. in the definite assignment of a transient to a class of faults, is here taken as the time elapsed between the beginning of the transient and its correct identification by the diagnostic system.

In this work, this concept is captured by introducing the Recognition Delay Time (RDT) defined as the time elapsed between the beginning of the transient and its first assignment to the correct class of fault. This quantity is easy to compute and independent from the specific rule that the system uses to give the diagnosis (e.g. a given number of confirmatory consecutive assignments of the transient to the same class of faults). By this definition, the RDT does not account for the negative fact that the diagnostic system may give a false alarm (assignment of the transient to a wrong class) before the first correct assignment. However, in the multi-objective framework of feature selection here adopted, this limitation in the definition of the RDT is overtaken by considering that the minimization of the number of incorrect classification is implicitly done by considering the minimization of the recognition rate as one of the objectives of the feature selection.

The mathematical definition of the RDT in crisp terms stands on the introduction of a set $W$ of $c$ Boolean assignment functions $w_i(\tilde{x}_k(t_i))$, $i=1,...,c$, $k=1,...,N$, where $w_i(\tilde{x}_k(t_i))=1$ means that the pattern $\tilde{x}_k(t_i)$ taken
from the \(k\)-th transient at time \(t\) is of class \(i\), \(w_i(\hat{x}_k(t))=0\) means that it is not of class \(i\) and \(N_t\) is the total number of available transients. Then, the number of time steps necessary for the first assignment of transient \(k\) to its true class \(i\) is measured by:

\[
RDT(k) = \sum_{t_{j=0}}^{t_f} \left(1 - \max\{w_i(\hat{x}_k(t))\}\right)
\]

(8)

where \(t_0\) is the first time step analyzed for the diagnosis and \(t_f\) the last.

The extension of the above concept in fuzzy terms is readily obtained by substituting the Boolean assignment function \(w_i(\hat{x}_k(t))\) in (eq.8) with the fuzzy degree of membership \(\mu_i(\hat{x}_k(t))\), i.e. the membership to the \(i\)-th class of pattern \(\hat{x}_k(t)\) taken from the \(k\)-th transient at time \(t\); the Fuzzy Recognition Delay Time (FRDT) is then defined as:

\[
FRDT(k) = \sum_{t_{j=0}}^{t_f} \left(1 - \max\{\mu_i(\hat{x}_k(t))\}\right)
\]

(9)

For a global measure of the diagnostic system delay time, a mean fuzzy delay time in the recognition of all \(N_t\) transients may be computed as:

\[
FRDT = \frac{\sum_{k=1}^{N_t} RDT(k)}{N_t}
\]

(10)

### 6.2 Definition of fuzzy performance

The recognition rate introduced in Section 3.2 for evaluating the performance in the classification of a set of \(N\) patterns can be computed with respect to only the patterns of a generic \(k\)-th transient:

\[
rr(k) = 1 - \frac{\sum_{t_{j=0}}^{t_f} \sum_{i=1}^{N} \left| w_i(\hat{x}_k(t)) - w'_i(k) \right|}{2 \left( \frac{t_f - t_0}{\Delta t} + 1 \right)}
\]

(11)

where \(w'_i(k)\) is the a priori known Boolean assignment of the \(k\)-th transient to the \(i\)-th class and \(\left( \frac{t_f - t_0}{\Delta t} + 1 \right)\) is the total number of data patterns sampled at time intervals \(\Delta t\) from the transient. Then, the mean recognition rate may be computed by:

\[
rr = 1 - \frac{\sum_{k=1}^{N_t} \sum_{t_{j=0}}^{t_f} \sum_{i=1}^{N} \left| w_i(\hat{x}_k(t)) - w'_i(k) \right|}{2N}
\]

(12)

where \(N = N_t \cdot \left( \frac{t_f - t_0}{\Delta t} + 1 \right)\) is the total number of data patterns sampled.

In this Section, the possibility to extend the recognition rate to the case of fuzzy classification is investigated. Hence the fuzzy performance index (fp) is proposed:

\[
fp = 1 - \frac{\sum_{k=1}^{N_t} \sum_{t_{j=0}}^{t_f} \sum_{i=1}^{N} \left| \mu_i(\hat{x}_k(t)) - w'_i(k) \right|}{2N}
\]

(13)

To appreciate the difference in the information content carried by the recognition rate (eq.12) and the fuzzy performance index (eq.13) consider an artificial example in which a pattern \(P\) of class 1 has the same four
nearest neighbors (patterns \(A, B, C, D\)) in two different feature sets \(S_X\) and \(S_Y\), but with different distances (Table 9).

**Table 9. Distances of point \(P\) to its 4 nearest neighbours \(A, B, C, D\) in the feature sets \(S_X\) and \(S_Y\).**

<table>
<thead>
<tr>
<th>nearest neighbor</th>
<th>class</th>
<th>Distance from (P) in (S_X)</th>
<th>Distance from (P) in (S_Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>1</td>
<td>0.010</td>
<td>0.010</td>
</tr>
<tr>
<td>(B)</td>
<td>3</td>
<td>0.011</td>
<td>0.500</td>
</tr>
<tr>
<td>(C)</td>
<td>2</td>
<td>0.012</td>
<td>0.700</td>
</tr>
<tr>
<td>(D)</td>
<td>1</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

The classification of pattern \(P\) using the FKNN with \(K=4\) leads to the membership function reported in Table 10.

**Table 10. Fuzzy classification of pattern \(P\) in the two feature sets \(S_X\) and \(S_Y\).**

<table>
<thead>
<tr>
<th>Feature set</th>
<th>(\mu_1(P))</th>
<th>(\mu_2(P))</th>
<th>(\mu_3(P))</th>
<th>crisp assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_X)</td>
<td>0.3967</td>
<td>0.2755</td>
<td>0.3278</td>
<td>class 1</td>
</tr>
<tr>
<td>(S_Y)</td>
<td>0.9994</td>
<td>0.0002</td>
<td>0.0004</td>
<td>class 1</td>
</tr>
</tbody>
</table>

In the crisp definition of the recognition rate (eq.11), the contribution of pattern \(P\) to the total recognition rate would be equal to 1 using both feature sets, because pattern \(P\) is crisply assigned in both cases to class 1, the class with the highest membership value. The feature selector would thus be unable to distinguish that \(S_Y\) is a more efficient feature set for the classification of \(P\) than \(S_X\) due to its superior clustering capabilities as seen from the distance values of Table 9. On the contrary, the fuzzy performance (eq.12) recognizes \(S_Y\) as more efficient than \(S_X\), taking values \(fp(S_X)=0.3967\) and \(fp(S_Y)=0.9994\).

6.3 A Genetic algorithm-based feature selector for ‘early’ fault classification

In this section a multi-objective Genetic Algorithm based feature selector is presented for the identification of optimal feature subsets for early fault classification. Three objectives are considered:

- the minimization of the fuzzy recognition delay time (FRDT) (eq.10)
- the maximization of the fuzzy performance (fp) (eq.13);
- the minimization of the number of selected signals \((m)\).

The search has been applied to the 123 mean residual features extracted by the WOLP pre-processing from the 123 original plant signals. In the methodological investigation carried out, only these 123 features have been considered for the three-objective optimization instead of all the 369 wavelet features used for the two-objective optimization, in order to reduce the dimension of the search space \(2^{123}\) instead of \(2^{369}\) and simplify the GA convergence. Following a trial-and-error procedure, the optimal parameter settings and GA procedures reported in Table 11 have been employed. Notice the very large population size and the high number of generations which need to be employed for the three-objective search.
Table 11. GA parameters and search strategies employed.

<table>
<thead>
<tr>
<th>Selection procedure</th>
<th>fit-fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replacement procedure</td>
<td>66% of the cases fittest, 34% weakest</td>
</tr>
<tr>
<td>Probability of mutation</td>
<td>0.008</td>
</tr>
<tr>
<td>Number of generations</td>
<td>4000</td>
</tr>
<tr>
<td>Population size</td>
<td>300</td>
</tr>
</tbody>
</table>

With respect to the computation of the FRDT objective (eq.10), it is required to compute FRDT(k) for all the transients \( k=1,...,N_t \). For the computation of FRDT(k) (eq.9) it is necessary that all the patterns of the \( k \)-th transients are in the test set. Thus, for the computation of the 3 fitness functions of a given feature set, it is not possible to use a 10-experiment cross validation approach in which the patterns are randomly subdivided into training and test sets without considering from which transient they have been obtained, as it was done in the previous Sections. The transients should, then, be subdivided into training transients, from which the training patterns are taken, and test transients, from which the test patterns are obtained. This procedure is not here applied because for each class of fault only three transients are available.

For this reason, a new strategy has been adopted for the computation of the fitness values corresponding to a given feature subset \( S_l \) (Figure 26). The computation of FRDT(k) and fp(k) is done subdividing all the available patterns (data set \( B \)) in a test set formed by the patterns taken from the \( k \)-th transient \( \{\tilde{x}_k(t_0), \tilde{x}_k(t_1),...,\tilde{x}_k(t_f)\} \) and in a training set formed by the remaining patterns \( \{\tilde{x}_1(t_0), \tilde{x}_1(t_1),...,\tilde{x}_N(t_f),\tilde{x}_{k-1}(t_0),\tilde{x}_{k-1}(t_1),...,\tilde{x}_{k-1}(t_f),\tilde{x}_{k+1}(t_0),\tilde{x}_{k+1}(t_1),...,\tilde{x}_N(t_f),\tilde{x}_N(t_0),\tilde{x}_N(t_1),...,\tilde{x}_N(t_f)\} \).

Notice, indeed, that the partitioning of the data in training and test sets and the following classification of the test patterns is performed \( N_t \) times, each time with different training and test sets. Finally, the mean fuzzy recognition delay time and the mean fuzzy performance of the feature set \( S_l \) are computed using respectively (eq.12) and (eq.10).

This technique differs from the one used in the previous Sections because each pattern is tested only once (no cross validation) and because in the training set there are no patterns taken from the transient of the tested patterns. This characteristic is important because in practical transient classification problems the training transients are never the same of the test transients.
6.4 Results

Figure 27 reports the nondominated Pareto solutions obtained for fixed values of the number of features, $m$. As expected, increasing $m$ the solutions are more satisfactory with respect to the fuzzy recognition delay time (FRDT) and the fuzzy performance (fp). Note that the minimization of FRDT and the maximization of fp are objectives mildly contradictory in this case given that for a fixed $m$, the maximum variation of the fuzzy delay time is of about 0.2 time steps, corresponding to 2 seconds. This fact can be explained considering that a feature set that allows achieving a high fuzzy performance typically allows achieving also the early classification of the transients. For example, in the ideal case of a perfect classifier with fuzzy performance equal to 1 (all the patterns are correctly classified), the fuzzy delay time would be equal to 0 (the transients are correctly classified at the first time step). Yet, in more realistic cases of non perfect classifiers, a fixed number of features solutions can be satisfactory for what concerns the fuzzy performance but less satisfactory for what concerns the delay time or vice versa.
Figure 27. Pareto nondominated solutions (fuzzy performance on the $x$-axes, FRDT on the $y$-axes) for various values of $m$ (number of selected features). Notice that different axis scales are used in the first row and that the delay time is not measured in seconds, but in time steps of 10 seconds.

Table 12. Characteristics of the two optimal solutions found with 3 features.

<table>
<thead>
<tr>
<th>Solution</th>
<th>fuzzy performance ($fp$)</th>
<th>Fuzzy Recognition Delay Time (FRDT)</th>
<th>selected features (Appendix B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>0.556</td>
<td>1.91</td>
<td>320-321-160</td>
</tr>
<tr>
<td>$B$</td>
<td>0.545</td>
<td>1.77</td>
<td>320-321-338</td>
</tr>
</tbody>
</table>

Table 3 reports the two optimal solutions identified with $m = 3$. The difference in the fuzzy recognition delay time is mainly due to the classification of class 9 transients for which the FRDT is 4.0 time steps using solution $A$ and 0.5 time steps using solution $B$. On the other side, solution $A$ is better for what concerns the classification of patterns of the classes 5 and 15 (the mean fuzzy performance in the classification of patterns of these 2 classes is 0.97 whereas with solution $B$ it is 0.64). In particular the features in solution $B$ are not able to discern between the transients of classes 3 and 5 and between the transients of classes 13 and 15, all caused by small intensity steps. On the contrary, features 320, 321, 160 selected by solution $A$ allow discerning between these classes because of the sensible variations of signal 160 in response to class 3 and 13 transients (Figure 28).
Figure 28. Evolutions of the features selected by solution A (mean residuals of features 320, 321, 160 with respect to their standard values) and by solution B (mean residuals of features 320, 321, 138 with respect to their stationary values) for faults of classes 3, 5, 9, 13, 15.

Figure 29. Comparison of the Pareto optimal solutions identified by the three-objective search (o) and by the two objective search (x). Fuzzy performance on the x-axes, FRDT on the y-axes are reported for different values of m (number of selected features). Notice that different axis scales are used in the first row and that the delay time is not measured in seconds but in time steps of 10 seconds.

Finally, the solutions found by the three-objective GA search are compared with the solutions found by a two-objective GA search with the goal of minimizing only the number of features and maximizing the fuzzy performance, i.e. ignoring the FRDT. Figure 29 compares the nondominated solutions found in the two cases. For \( m > 4 \) the three-objective search finds the same solution of the two-objective search plus other solutions characterized by different trade offs between the FRDT and the fuzzy performance. In particular, solutions characterized by slightly shorter FRDT but significantly lower fuzzy performance are identified by the three-
objective search. It is interesting to notice that for a given number of features $m > 4$, all the solutions found by the two-objective search have the longest FRDT and the highest fuzzy performance. This fact confirms the benefit of considering the three-objective search to find solutions with shorter FRDT that are not identified by the two-objective search. For what concerns the solution found with a number of features $m \leq 4$, the three-objective search encounters difficulties in the convergence due to the increased complexity of the search space.

7. CONCLUSIONS

In this work, the problem of discerning, among the several measured plant parameters, those features to be used for early transient diagnosis has been tackled. This is a crucial issue to be resolved for the application of advanced monitoring and diagnostic techniques to complex plants, like the nuclear power plants, where hundreds of parameters are measured.

Traditionally, feature selection algorithms developed for classical data classification problems have proven not efficient in selecting features salient for transient classification because transients are trajectories traced in the feature space rather than data points.

To overcome this problem, specific techniques have here been developed and tested by application to the classification of simulated transients in the feedwater system of a Boiling Water Reactor.

First of all, a standard classifiability evaluation function, recently introduced in the literature, has been extended to account for the transient evolution in time. The corresponding proposed feature selection filter approach identifies 16 features, from an initial number of 123 potentially relevant plant parameters, which are shown capable of achieving a satisfactory accuracy in a Fuzzy K-Nearest Neighbor classification, significantly better than with the features selected by a plant expert and by the algorithm based on the original, standard feature selection evaluation function.

The employed filter approach, as any filter approach, assesses the merit of the features directly from data, ignoring the merit of the selected features in the actual transient classification algorithm used.

To overcome this limitation of filter approaches, a wrapper approach based on a multi-objective search engine has been developed. A Pareto-based GA technique has been embraced to arrive at a set of nondominated feature subsets characterized by different classification performances and dimensionalities (in terms of number of features). The performance deriving from such approach has been analyzed in details through a parameter analysis regarding the main characteristics of the GA search procedure. The approach is demonstrated to be both more accurate, in comparison to the filter technique, and parsimonious in the number of features required to obtain that accuracy.

From the general point of view of the methodology, a key advantage of the GA feature selection wrapper technique with respect to a filter approach is that the search measures explicitly the goodness of the considered feature subsets on the performance of the classification algorithm (in our case the FKNN) that is used for the diagnostic task. Additionally, the analysis of the features that prove to be relevant for the diagnostic classification task can lead to a deeper understanding of the data and underlying physical relationships, allowing feature selection to be used in exploratory data analysis and data mining. Finally, the GA-FKNN wrapper approach is immune from the “nesting effect” characteristic of “hill climbing” methods, so that the probability of getting stuck in local optima is significantly lower.

An integration of the multi-objective GA feature selection scheme with a pre-processing of dynamical data has also been investigated. More precisely, the features most characteristic of the system transient behaviour have been extracted from the measured signals using a technique based on the wavelet transform. In the case study analyzed, the number of features contained at convergence in the Pareto solutions, obtained by applying the
feature selection to the extracted features, ranges from \( m = 4 \) to \( m = 25 \) and the corresponding recognition rates from 0.7506 to 0.9329. This is to be compared with the previous optimal results obtained considering directly the measured signals and not their wavelet extracted features: the number of signals ranges from \( m = 1 \) to \( m=13 \), with recognition rate ranging from 0.341 to 0.8658. Thus, the feature selection integrated with the pre-processing phase is superior in producing solutions with more satisfactory performances.

Finally a novel definition of recognition delay time has been introduced to emphasize the requirement of early transient identification. A corresponding objective of minimization of the recognition delay time can be added to the minimization of the number of features and the maximization of the recognition rate in the multi-objective search. The three-dimensional Pareto optimal front obtained at convergence provides the analyst with the possibility to choose a compromise solution between these three relevant diagnostic objectives. In particular, for a given number of features (complexity of the classification algorithm), different trade-offs can be taken between the early classification of the transients and the classification performance.
APPENDIX A

Table 13. Features selected by the expert (first column), by the feature selection algorithm based on the classical definition of classifiability $L$ (second column), on the transient classifiability $L'$ (third column) and on the weighed transient classifiability $L''$ (fourth column). For $L$, $L'$,$L''$ the features are ordered in their order of selection and the bold features are those which appear also in the expert selection.

<table>
<thead>
<tr>
<th>Expert</th>
<th>$L$</th>
<th>$L'$</th>
<th>$L''$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>319</td>
<td>321</td>
<td>320</td>
</tr>
<tr>
<td>73</td>
<td>45</td>
<td>320</td>
<td>321</td>
</tr>
<tr>
<td>74</td>
<td>223</td>
<td>342</td>
<td>77</td>
</tr>
<tr>
<td>75</td>
<td>357</td>
<td>157</td>
<td>160</td>
</tr>
<tr>
<td>76</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>175</td>
<td>223</td>
<td>342</td>
<td>77</td>
</tr>
<tr>
<td>194</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>195</td>
<td>223</td>
<td>342</td>
<td>77</td>
</tr>
<tr>
<td>240</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>241</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>242</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>318</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>319</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>320</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>321</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>324</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
<tr>
<td>325</td>
<td>157</td>
<td>325</td>
<td>195</td>
</tr>
</tbody>
</table>

APPENDIX B

Table 14. Feature number, name, and unit of measurement.

<table>
<thead>
<tr>
<th>Feature number</th>
<th>Name</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>total feedwater flow</td>
<td>kg/s</td>
</tr>
<tr>
<td>59</td>
<td>Temperature high-pressure drain</td>
<td>°C</td>
</tr>
<tr>
<td>60</td>
<td>Temperature after EA1 (high pressure preheater A1)</td>
<td>MPa</td>
</tr>
<tr>
<td>73</td>
<td>Pressure drain 6 before VB1</td>
<td>MPa</td>
</tr>
<tr>
<td>74</td>
<td>Pressure drain 5 before VB2</td>
<td>MPa</td>
</tr>
<tr>
<td>75</td>
<td>Temperature drain 6 after VB1</td>
<td>°C</td>
</tr>
<tr>
<td>76</td>
<td>Temperature drain 5 after VB2</td>
<td>°C</td>
</tr>
<tr>
<td>77</td>
<td>Temperature drain 4 before VB3</td>
<td>°C</td>
</tr>
<tr>
<td>100</td>
<td>Pressure before EA1 (preheater A1)</td>
<td>MPa</td>
</tr>
<tr>
<td>107</td>
<td>Temperature after 424EA1 (preheater A1)</td>
<td>°C</td>
</tr>
<tr>
<td>109</td>
<td>Temperature after EA1 train A (preheater A1)</td>
<td>°C</td>
</tr>
<tr>
<td>139</td>
<td>Temperature of condensate after EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>140</td>
<td>Temperature of condensate after EB2 train A</td>
<td>°C</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>143</td>
<td>Temperature of condensate after EB3 train B</td>
<td>°C</td>
</tr>
<tr>
<td>155</td>
<td>Temperature of bearing, drive end PB1.M1</td>
<td>°C</td>
</tr>
<tr>
<td>157</td>
<td>Temperature of bearing, drive end PB2.M1</td>
<td>°C</td>
</tr>
<tr>
<td>160</td>
<td>Water level tank TD1</td>
<td>M</td>
</tr>
<tr>
<td>165</td>
<td>Temperature of axial bearing PC3 (pump 3)</td>
<td>°C</td>
</tr>
<tr>
<td>168</td>
<td>Temperature seal water to PC3.E1</td>
<td>°C</td>
</tr>
<tr>
<td>175</td>
<td>Drain flow to EB4</td>
<td>Kg/s</td>
</tr>
<tr>
<td>191</td>
<td>Water level in EB1 train B (preheater B1)</td>
<td>M</td>
</tr>
<tr>
<td>192</td>
<td>Water level in EA2 train A (preheater A2)</td>
<td>M</td>
</tr>
<tr>
<td>193</td>
<td>Water level in EB2 train B (preheater B2)</td>
<td>M</td>
</tr>
<tr>
<td>194</td>
<td>Temperature feedwater before EA2 train A</td>
<td>°C</td>
</tr>
<tr>
<td>195</td>
<td>Temperature feedwater after EA2 train A</td>
<td>°C</td>
</tr>
<tr>
<td>196</td>
<td>Temperature of radial bearing PA1.P1 (pump A1)</td>
<td>°C</td>
</tr>
<tr>
<td>197</td>
<td>Temperature of radial bearing PA1.P1 (pump A1)</td>
<td>°C</td>
</tr>
<tr>
<td>223</td>
<td>Vibration PA1.P1 outer vertical</td>
<td>mm/s</td>
</tr>
<tr>
<td>229</td>
<td>Pressure after pump PB1</td>
<td>MPa</td>
</tr>
<tr>
<td>240</td>
<td>Temperature in common pipe before preheating</td>
<td>°C</td>
</tr>
<tr>
<td>241</td>
<td>Temperature feedwater before EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>242</td>
<td>Temperature feedwater after EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>243</td>
<td>Temperature feedwater after high-pressure preheating</td>
<td>°C</td>
</tr>
<tr>
<td>281</td>
<td>Temperature of radial bearing PD1.P1 (pump D1)</td>
<td>°C</td>
</tr>
<tr>
<td>318</td>
<td>Active effect</td>
<td>MW</td>
</tr>
<tr>
<td>319</td>
<td>Internal core cooling flow</td>
<td>kg/s</td>
</tr>
<tr>
<td>320</td>
<td>Position level control valve for EA1</td>
<td>%</td>
</tr>
<tr>
<td>321</td>
<td>Position level control valve for EB1</td>
<td>%</td>
</tr>
<tr>
<td>324</td>
<td>Position level control valve before EA2</td>
<td>%</td>
</tr>
<tr>
<td>325</td>
<td>Position emergency drain valve EB2</td>
<td>%</td>
</tr>
<tr>
<td>330</td>
<td>Position valve for level I EB4</td>
<td>%</td>
</tr>
<tr>
<td>332</td>
<td>Position drain valve TD1</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>338</td>
<td>Position drain valve TA1</td>
<td></td>
</tr>
<tr>
<td>342</td>
<td>Position valve in drain for TB1</td>
<td></td>
</tr>
<tr>
<td>344</td>
<td>Position valve in drain for TB2</td>
<td></td>
</tr>
<tr>
<td>357</td>
<td>Position steam stop control valve %</td>
<td></td>
</tr>
<tr>
<td>359</td>
<td>Position steam stop control valve %</td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES


Report B

Clustering techniques and their application to the classification of transients in nuclear power plants

Enrico Zio and Piero Baraldi

LASAR** and Halden Reactor Project Work Report, March 2006

---

§ This report is being submitted as official Halden Work Report (HWR)
** LASAR (Laboratorio di Analisi di Segnale ed Analisi di Rischio): Laboratory of Signal Analysis and Risk Analysis of the Department of Nuclear Engineering of the Polytechnic of Milan
# CONTENTS

- NOMENCLATURE .................................................................................................................................. 1
- ACRONYMS ........................................................................................................................................ 2
- 1. INTRODUCTION .................................................................................................................................. 2
- 2. UNSUPERVISED CLUSTERING TECHNIQUES ......................................................................................... 3
  - 2.1 Hard clustering vs fuzzy clustering ................................................................................................ 4
  - 2.2 Hard C-Means algorithm ............................................................................................................... 5
  - 2.2.1 Operative procedure .................................................................................................................. 6
  - 2.3 Fuzzy C-Means algorithm ............................................................................................................ 6
  - 2.3.1 Determination of optimal membership functions $\mu_d$ ........................................................... 7
  - 2.3.2 Determination of optimal prototype parameters (clusters centroids) $\vec{v}_i^*$ ....................... 9
  - 2.3.3 Operative procedure ................................................................................................................ 10
  - 2.3.4 Generalization to include Fuzzy Covariance ............................................................................ 10
  - 2.3.5 Operative procedure ................................................................................................................ 12
  - 2.4 Clustering validity ........................................................................................................................ 13
  - 2.4.1 Partition coefficient .................................................................................................................. 13
  - 2.4.2 Partition entropy ...................................................................................................................... 14
  - 2.4.3 Compactness and separation function ....................................................................................... 14
  - 2.4.4 A technique for the determination of the optimal number of clusters ........................................ 15
  - 2.5 Possibilistic clustering .................................................................................................................. 15
  - 2.5.1 Limitations of the FCM algorithm ............................................................................................ 16
  - 2.5.2 Possiblistic Clustering algorithm ............................................................................................ 19
  - 2.5.3 Operative procedure ............................................................................................................... 22
  - 2.5.4 Generalization to include the fuzzy covariance in the possibilistic algorithm ......................... 22
- 3. A CLUSTERING TECHNIQUE FOR CLASSIFICATION ....................................................................... 23
  - 3.1 A supervised evolutionary clustering approach to classification ................................................. 23
  - 3.1.1 The evolutionary FCM classifier .............................................................................................. 25
  - 3.1.2 The evolutionary possibilistic classifier ..................................................................................... 26
- 4. APPLICATIONS TO TRANSIENT CLASSIFICATION IN NUCLEAR POWER PLANTS ....................... 27
  - 4.1 Case study 1: classification of transients in a U-tube steam generator ............................................ 28
  - 4.1.1 Results achieved by the evolutionary FCM classifier ................................................................ 29
  - 4.1.2 Results achieved by the evolutionary possibilistic classifier .................................................... 31
  - 4.1.3 Comparison of the two classification models ........................................................................... 32
  - 4.2 Case study 2: classification of transients in the feedwater system of a Boiling Water Reactor .... 34
  - 4.2.1 Filtering out unknown transients ............................................................................................. 35
  - 4.2.2 Classification of transients at different power levels .................................................................. 37
NOMENCLATURE

- $S_f$: feature space
- $h$: dimension of the input space
- $X = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_N\}$: set of available patterns $\hat{x}_k$, $k = 1, ..., N$
- $i = 1, ..., c$: index of the class or cluster to which the generic pattern belongs to
- $c$: number of clusters
- $\tilde{x}_i^{(i)}$: generic pattern of the $i$-th cluster
- $N_i$: number of available patterns of cluster $i$
- $r_m$: fuzziness index
- $w_{ik}$: Boolean assignment function of the $k$-th pattern to the $i$-th cluster
- $\mu_{ik}$: fuzzy or possibilistic membership of the $k$-th pattern to the $i$-th cluster
- $\mu_{ik}$: set of all the values of membership of all data $\hat{x}_k$, $k = 1, ..., N$ to all clusters $i$, $i = 1, ..., c$
- $\hat{v}_i$: cluster prototype
- $V = \{\hat{v}_1, \hat{v}_2, ..., \hat{v}_c\}$: set of all the cluster prototypes
- $\hat{v}_i^*$: optimal cluster prototype
- $I$: Identity matrix
- $M_i$: Mahalanobis matrix relative to cluster $i$
- $P_i$: Fuzzy covariance matrix relative to cluster $i$
- $s_{ik}$: distance between pattern $\hat{x}_k$ and cluster center $\hat{v}_i$
- $\Gamma = (\Gamma_1, \Gamma_2, ..., \Gamma_c)$: fuzzy or possibilistic partition of the patterns in $c$ clusters $\Gamma_i$
- $\mathcal{J}(\cdot)$: objective function of the clustering algorithm
- $\lambda_\xi$, $\omega_\xi$: Lagrange multipliers
- $\alpha$: iteration of the clustering algorithm
- $\epsilon$: convergence threshold used in the clustering algorithm for the maximum change in the membership values from one iteration to the next
- $F$: partition coefficient
- $E$: partition entropy
- $S$: Compactness and separation validity measure
- $\eta_i$: distance at which the membership values of a pattern become 0.5 in a possibilistic approach
- $\Phi(\cdot)$: classification mapping function
- $\varepsilon_c$: threshold for the classification of a pattern to a class
Given a set of $N$ multivariate data points $X = \{x_1, x_2, ..., x_N\}$, unsupervised cluster analysis aims at identifying any relationship existing among them by organizing them into $c < N$ clusters (hereafter also called groups) such that the points belonging to a given cluster are more similar to each other than are those belonging to different clusters.

The conventional (hard) clustering methods constrain each pattern to belong to one cluster only. In practice, however, the clusters may not be completely disjoint and patterns could be classified as belonging to one cluster almost as well as to another. In these cases, one must assign to each pattern a set of membership values, one for each class identified. The implication of this is that the class boundaries are not ‘hard’ but rather ‘fuzzy’ and the representation of the data structure can be more accurately handled by fuzzy clustering methods [1,2].

Fuzzy clustering algorithms have been widely studied and applied in a variety of substantive areas such as taxonomy, medicine, geology, business, engineering, image processing and others. A general classification of these algorithms is offered in [2] in terms of three categories: 1) fuzzy clustering based on fuzzy relations 2) fuzzy clustering based on the minimization of an objective function 3) the class of nonparametric classifiers based on the fuzzy generalized K-Nearest Neighbors rule.

In this work, the first unsupervised clustering technique considered is the Fuzzy C Means (FCM) algorithm, belonging to the second category, originally defined by Dunn [3], then generalized by Bezdek [1], and nowadays available in a variety of variations and generalizations [4], some of which are here discussed to some detail. In this fuzzy clustering approach, the number of clusters has to be a priori fixed and the obtained fuzzy partition must satisfy the ‘probabilistic’ constraint that forces to 1 the sum of the memberships of a given pattern to the different clusters. The algorithm is based an iterative procedure of minimization of an objective function that attains its minimum for a partition which maximally separates the naturally-occurring clusters. As a result of the ‘probabilistic’ constraint, the membership of a pattern to a cluster depends on the memberships to all other clusters, i.e., geometrically speaking, it depends on where the pattern is located with respect to not only that cluster but also the others. Hence, in the framework of fuzzy clustering, the membership functions take the meaning of degrees of sharing, i.e. they measure how much a pattern belongs to a cluster relatively to the others. A conceptual drawback of this approach is that the constrained membership cannot distinguish between ‘equal evidence’ and ‘ignorance’ or, in other words, between ‘equally likely’ and ‘unknown’ membership to a cluster. Moreover, an unknown, atypical pattern not belonging to any cluster would still belong more to one cluster than...
to the others, relatively speaking, even if it lies far from all clusters in the feature space and thus it may receive high membership values to some clusters.

To overcome the above limitations the clustering problem can be recast into the framework of possibility theory [5,6]. In the possibilistic clustering algorithm proposed in [7] and here discussed, the only constraint that the objective function must satisfy is that the membership values do represent degrees of compatibility, or possibility, i.e. they must lie in [0,1]. Compatibility (typicality) is measured with respect to the cluster prototypical members [5,6]. In this view, the memberships of representative patterns are high, while unrepresentative (atypical) points bear low membership to all clusters.

When cluster analysis is to be used within the framework of classification of data belonging to physically different classes an obvious desiderata is that the clusters eventually identified be indeed representative of the underlying physical differences. This leads to supervised clustering. In this respect, the feasibility of building a classifier by means of a supervised evolutionary procedure [8] applied to a clustering algorithm [1] is presented. To tackle the classification problem with a supervised technique we assume to have available a partition Γ, in general fuzzy, of \( N \) “training” labelled data, \( \{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_N\} \) in \( c \) subsets. An evolutionary algorithm searches for the optimal Mahalanobis metrics to be used by the clustering algorithm so as to achieve clusters as close as possible to the real physical classes. The evolutionary approach has been developed both within the FCM algorithm defining an evolutionary FCM classifier [9] and within the possibilistic clustering algorithm defining an evolutionary possibilistic classifier [10].

The developed classification approaches have been applied to the problem of transient classification in nuclear power plants. The early identification of the causes for the onset of a meaningful departure from steady state behaviour is an essential step for the operation, control and accident management of nuclear power plants. The basis for the classification is that different system faults and anomalies lead to different patterns of evolution of the involved process variables. Two case studies of transient classification are here presented [9,10,11].

2. UNSUPERVISED CLUSTERING TECHNIQUES

Given a set of \( N \) multivariate data points (hereafter also called samples, vectors, patterns) \( X = \{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_N\} \), cluster analysis aims at identifying any relationship existing among them by organizing them into \( c < N \) clusters (hereafter also called groups) such that the points belonging to a given cluster are more similar to each other than are those belonging to different clusters. In practice, two different situations may occur: the number \( c \) of clusters in which the set \( X \) of \( N \) data has to be partitioned may be a priori known or not. In the latter case the problem is that of finding both the number of clusters \( c \) and the corresponding mapping function, which performs the partition of the data of set \( X \) into the \( c \) clusters, such that the data points belonging to a given cluster are more similar to each other than are those belonging to different clusters.

Each one of the above situations may be framed within an optimisation problem with respect to a defined measure of similarity among data. Once the clustering is established, i.e. the number of clusters \( c \) and the mapping function are known, it must be validated against appropriately defined validity functions which measure the separation among clusters and the cohesion within clusters. This problem is termed unsupervised given that the partition of the data in classes or clusters is not a priori known.
2.1 Hard clustering vs fuzzy clustering

We consider an $h$-dimensional feature space $S_f \subset \mathbb{R}^h$. In a hard clustering procedure, it is assumed that a generic point $x \in S_f$ can belong only to one cluster: thus, each data sample is assigned to only one cluster and the clusters are regarded as disjoint gatherings of the data set. A hard partition into $c$ clusters is then a set $W$ of $c$ Boolean functions $w_i(x), i=1, 2, \ldots, c$, defined on the feature space $S_f \subset \mathbb{R}^h$ which satisfy:

$$w_i(x) = 0 \text{ or } 1 \quad \forall x \in S_f, i=1,2,\ldots,c$$

$$\sum_{i=1}^{c} w_i(x) = 1 \quad \forall x \in S_f$$

If $I_i$ represents (1) the $i$-th cluster, with $I_i \cap I_j = \emptyset$ $\forall i \neq j$ and $\bigcup_{i=1}^{c} I_i = \Gamma$, then $w_i(x) = 1$ means that $x \in I_i$ and (2) insures that $x$ is a member of precisely one class only.

In practice, however, this ‘all or none’ membership restriction is not a realistic one since many data points may share characteristics common to several clusters. In other words, there are many cases in which the clusters are not completely disjoint and data could be classified as belonging to one cluster almost as well as to another. Such a situation cannot be described by a crisp clustering process. In these cases, it comes more natural to assign to each data point a set of membership values, one for each class identified. The implication of this is that the class boundaries are not ‘hard’ but rather ‘fuzzy’ and the representation of the data structure can be more accurately handled by fuzzy clustering methods. It is possible to pass from the formal definition of a hard partition as given by (1) and (2) to a corresponding fuzzy partition by retaining condition (2) but replacing (1) with the relaxed condition $0 \leq w_i(x) \leq 1, i=1, 2, \ldots, c$. Following the conventional notation of fuzzy sets, the function $w_i(x)$ represents the membership function of $x$ to class $i$ and will be denoted as $\mu_i(x), x \in I_i, i=1, 2, \ldots, c$.

The following advantages have been found to result from the use of fuzzy clustering procedures:

1. The number of clusters derived from a ‘hard’ partitioning algorithm is generally rather large, thus making an exhaustive search extremely complicated and time consuming. Fuzzy clustering, on the contrary, tends to smaller numbers of clusters and indeed more computational tractability [4].

2. The values of membership assigned to a given data point $x$ relative to the various clusters identified provide a measure of the significance of the classification. Hence, for example, if a sample is assigned membership 0.9 to one class and 0.05 to two other classes, we can be reasonably sure that the class of 0.9 membership is the one to which the sample belongs.

3. For what said in item 2. above, outlying data points are more easily recognised by a fuzzy clustering approach than by a hard one, since the degree of membership to clusters is continuous rather than ‘all-or-none’.

In general, any clustering process consists of an iterative optimisation procedure. In the next Sections, three techniques for the unsupervised clustering of a data set in a known number $c$ of clusters will be illustrated:

- the Hard C-Means (HCM) algorithm (Section 2.2);
- the Fuzzy C-Means (FCM) algorithm (Section 2.3)
- the possibilistic clustering algorithm (Section 2.5)
2.2 Hard C-Means algorithm

In what follows, the Hard C-Means (HCM) algorithm is illustrated in such a way as to render intuitive its extension to the Fuzzy C-Means algorithm [12]. In order to ease the illustration of the detailed computations, we shall adopt an objective function which relates to a Euclidean similarity measure.

A set \( X \) of indexed data points in the \( h \)-dimensional feature space \( S_f \subset \mathbb{R}^h \) is considered:

\[
X = \{ \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N \}; \quad \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N \in S_f
\]  

(3)

We wish to group these data points in a known number \( c \) of clusters. We start from an initial, approximate or random partition of the \( N \) \( h \)-dimensional vectors in \( c \) clusters \( \{ \bar{x}_i^{(k)} \}, i=1,2,\ldots,c, k=1,2,\ldots,N_i \), with \( \sum_{i=1}^c N_i = N \).

The Euclidean quadratic distance \( (\bar{x}_i^{(k)} - \bar{v}_i)^T I (\bar{x}_i^{(k)} - \bar{v}_i) \), where \( I \) is the identity matrix, is taken as the similarity measure between the sample \( \bar{x}_i^{(k)} \), \( k=1,2,\ldots,N_i \), and the cluster \( i \), as represented by the characteristic parameter \( \bar{v}_i \), hereafter also called cluster prototype, centroid or center. The prototypes \( \bar{v}_i \), \( i=1,2,\ldots,c \), are to be determined by the clustering algorithm which, correspondingly, proceeds by updating iteratively the partition of the \( N \) samples into the \( c \) clusters in such a way as to minimize the following quadratic objective function:

\[
J(V) = \sum_{i=1}^c \sum_{k=1}^{N_i} s_{ik} (\bar{x}_i^{(k)} - \bar{v}_i)^T I (\bar{x}_i^{(k)} - \bar{v}_i) = \sum_{i=1}^c \sum_{k=1}^{N_i} |\bar{x}_i^{(k)} - \bar{v}_i|^2
\]

(4)

where \( V = \{ \bar{v}_i \} \) is the set of the prototypes of all \( c \) clusters which defines the hard partition.

The minimization of \( J(V) \) of eq. (4) proceeds iteratively in two steps:

\( i. \) Determination of the clusters prototypes \( \bar{v}_i \), \( i=1,2,\ldots,c; \)

\( ii. \) Regrouping of the sample data by reassignment of each \( \bar{x}_k \), \( k=1,\ldots,N \) to the cluster of respective ‘most similar’ or ‘closest’ prototype in the sense of the Euclidean quadratic distance.

As for the first step, by the definition of the objective function (4) the clusters centers \( \bar{v}_i \) which minimize it lead to the maximum ‘euclidean similarity’ among the elements of the same cluster, in the sense that the data in a cluster will be compactly assembled in \( \mathbb{R}^h \) in the vicinity of the corresponding cluster center. Imposing to \( J(V) \) the condition of stationarity with respect to \( \bar{v}_i \):

\[
\frac{\partial J(V)}{\partial \bar{v}_i} = 2 \sum_{k=1}^{N_i} (\bar{x}_i^{(k)} - \bar{v}_i) = 0
\]

(5)

we obtain the optimizing centers \( \bar{v}_i^* \):

\[
\bar{v}_i^* = \frac{\sum_{k=1}^{N_i} \bar{x}_i^{(k)}}{N_i} = \bar{x}_i \quad \bar{v}_i^* = i = 1, 2, \ldots, c
\]

(6)

which are the sample means of the \( N_i \) data \( \{ \bar{x}_i^{(k)} \} \) belonging to the respective clusters \( \Gamma_i \), \( i=1,2,\ldots,c \), and where the asterisk denotes association with optimality.
Now, based on this information regarding the positions of the new clusters centers, we can proceed to the second step of regrouping the available samples $\bar{x}_k, k=1,...,N$, in a way as to minimize the objective function in (4). Obviously, the objective function $J(V)$ achieves its minimum when each sample is associated to its ‘most similar’ or ‘closest’ cluster center: hence, we reassign the generic vector $\bar{x}_k$ to that cluster $i$ such that $s_{ik} = s(\bar{x}_k, \bar{v}_j) = |\bar{x}_k - \bar{v}_j|^2$ is minimum when $j=i, j=1,2,...,c$ and $k=1,...,N$.

Once the samples are regrouped, the cluster centers $\bar{v}_i$ in set $V$ need to be recomputed by means of eq. (6) so as to minimize $J(V)$ (step i.). With the new cluster centers found we can again perform a regrouping of the samples $\bar{x}_k, k=1,...,N$, into the clusters of respective ‘most similar’ or ‘closest’ centers so as to minimize $J(V)$ (step ii.). The process can be repeated iteratively until $J(V)$ cannot be reduced further or, equivalently, until the cluster centers do not change sensibly from one iteration to the next.

2.2.1 Operative procedure
1. Assign the number $c$ of clusters in which you want to partition the data set $X$;
2. Perform an initial partition of the data set into $c$ clusters. If no a priori information is available to guide an approximate partition, the initial partition can be random: this can be done by selecting randomly $c$ samples from the n available data $\bar{x}_k, k=1,...,N$, and taking them as clusters centers $\bar{v}_j, j=1,2,...,c$;
3. Assign each sample point $\bar{x}_k, k=1,...,N$, to the ‘most similar’ class $i$, i.e. that which has the closest center $\bar{v}_j, j=1,2,...,c$, in terms of the Euclidean quadratic distance $s_{ik}$;
4. By using eq. (6), compute the optimal centers $\bar{v}_i^*, i=1,2,...,c$ of the clusters formed in step 3;
5. Repeat steps 3 and 4 until no change in $J(V)$ is observed from one iteration to the next or, equivalently, until the maximum change in the centers $\bar{v}_i^*, i=1,2,...,c$, from one iteration to the next, is below a predefined convergence threshold $\varepsilon$, i.e.
$$\max_i (\bar{v}_i^*)^a - (\bar{v}_i^*)^{a-1} \leq \varepsilon$$
$\alpha$ denoting the iteration number.

2.3 Fuzzy C-Means algorithm
The Fuzzy C-Means algorithm is the best known and most widely used unsupervised fuzzy clustering technique [1,3]. Considering the indexed set of samples $X = \{\bar{x}_1, \bar{x}_2, ..., \bar{x}_N\}$, we denote by $\mu_{ik} = \mu(\bar{x}_k, \bar{v}_i)$ the membership function value of the point $\bar{x}_k, k=1,2,...,N$, with respect to the $i$-th class parameterised by the class prototype $\bar{v}_i$. The Fuzzy C-Means clustering algorithm represents the fuzzy extension of the Hard C-Means algorithm and aims at minimizing the following objective function with respect to both the prototype parameters $\bar{v}_i$ and the membership values $\mu_{ik}, i=1,2,...,c, k=1,...,N$:
$$J_\alpha (M, V) = \sum_i \sum_k (\mu_{ik})^\alpha s_{ik}$$
(7)
where $M = \{\mu_{ik}\}$ is the set of all values of membership of all data in $X$ to all clusters and $r_\alpha>1$ is an exponential weight which controls the degree of fuzziness of the clusters. The sets $M$ and $V$ define the fuzzy partition of the indexed set $X$. 

6
In the commonly adopted case of \( s_{ik} \) being the Euclidean quadratic distance \((\bar{x}_k - \bar{v}_i)^T I(\bar{x}_k - \bar{v}_i)\) between the sample \( \bar{x}_k \) and the center \( \bar{v}_i \) of the \( i \)-th cluster, the objective function reads

\[
J_m(M,V) = \sum_{i=1}^{N} \sum_{k=1}^{c} (\mu_{ik})^r (\bar{x}_k - \bar{v}_i)^T I(\bar{x}_k - \bar{v}_i) = \sum_{i=1}^{N} \sum_{k=1}^{c} (\mu_{ik})^r |\bar{x}_k - \bar{v}_i|^2
\]  

(7a)

As mentioned in Section 2.1, for a fuzzy partition the elements \( \mu_{ik}, i=1,2,...,c, k=1,...,N \) of \( M \) must satisfy the following conditions:

\[
0 \leq \mu_{ik} \leq 1 \quad i=1,2,...,c, k=1,...,N
\]  

(8)

\[
\sum_{i=1}^{c} \mu_{ik} = 1 \quad k=1,...,N
\]  

(9)

This leads also to:

\[
0 \leq \sum_{k=1}^{N} \mu_{ik} \leq N \quad i=1,2,...,c
\]  

(10)

As mentioned above, to determine the fuzzy partition of the data set \( X \) by the Fuzzy C-Means algorithm one must determine both the elements \( \mu_{ik} \) of \( M \) and the prototype parameters \( \bar{v}_i \) of \( V, i=1,2,...,c, k=1,...,N \), which minimize \( J_m(M,V) \).

### 2.3.1 Determination of optimal membership functions \( \mu_{ik} \)

We consider the problem of minimizing \( J_m(M,V) \) with respect to the elements of the set \( M=\{ \mu_{ik} \} \), subject to \( r_m>1 \) and the constraints (8) and (9). We use the method of the Lagrange multipliers. Within this technique, constraint (8) may be accounted for by setting \( \mu_{ik} = z_{ik}^{2} \), \( i=1,2,...,c, k=1,...,N \), with \( z_{ik} \) real. To proceed with the minimization, we adjoin the constraints (8) and (9) to \( J_m(M,V) \) by means of a set of Lagrange multipliers \( \Lambda = \{ \lambda_i \} \) and obtain the extended objective function

\[
L(Z,V,\Lambda) = \sum_{i=1}^{N} \sum_{k=1}^{c} z_{ik}^2 s_{ik} + \sum_{k=1}^{N} \lambda_k \left( \sum_{i=1}^{c} z_{ik}^2 - 1 \right)
\]  

(7b)

where \( Z = \{ z_{ik} \} \).

As usual, the first order necessary conditions for stationarity are found by setting to zero the gradients of \( L(Z,V,\Lambda) \) with respect to \( Z \)

\[
\frac{\partial L(Z,V,\Lambda)}{\partial z_{ik}} = 2r_m \cdot (z_{ik})^{r_m-1} s_{ik} + 2z_{ik} \cdot \lambda_k = 0 \quad i=1,2,...,c, k=1,...,N
\]  

(11)

thus obtaining:

\[
z_{ik} \cdot \left( r_m \cdot (z_{ik})^{r_m-1} s_{ik} + \lambda_k \right) = 0 \quad i=1,2,...,c, k=1,...,N
\]  

(12)

which is to be coupled to constraint (9), with \( \mu_{ik} = z_{ik}^2 \):
\[ \sum_{j=1}^{c} (z_{ik})^2 = 1 \quad k = 1, \ldots, N \]

The pair of equations (12) and (13) comprise a set of \( c \cdot N + N \) equations which can be solved for the \( c \cdot N + N \) unknowns \( Z^* = \{ z_{ik}^* \} \) and \( \Lambda^* = \{ \lambda_{ik}^* \} \) where as before the asterisk denotes association with optimality.

### 2.3.1.1 Case \( r_m > 1 \)

We proceed first by assuming that \( \mu_{ik}^* \neq 0 \) (and, thus \( z_{ik}^* \neq 0 \)) \( \forall i, k \), which is consistent with the assumption \( r_m > 1 \).

From (12) we have:

\[ \mu_{ik}^* = (z_{ik}^*)^2 = \left( \frac{1}{r_m \cdot s_{ik}} \right)^{1/(\alpha_m - 1)} \quad i = 1, 2, \ldots, c, \quad k = 1, \ldots, N \]

(14)

By summing over \( i \) and using (9) we get:

\[ \left( \frac{1}{r_m \cdot s_{ik}} \right)^{1/(\alpha_m - 1)} = k = 1, \ldots, N \]

(15)

Substituting (15) into (14) we obtain the optimal \( \mu_{ik}^* \):

\[ \mu_{ik}^* = \frac{1}{\sum_{j=1}^{c} \left( \frac{1}{r_m \cdot s_{ik}} \right)^{1/(\alpha_m - 1)}} \quad i = 1, 2, \ldots, c; \quad k = 1, 2, \ldots, N \]

(16)

The memberships assigned to sample \( x_i \) are, thus, functions of the inverse of the measures of similarity with the class prototypes \( \bar{v}_i \). \( i = 1, 2, \ldots, c \). The inverse distance serves to weigh a class more if its representative prototype is more similar and less if it is less similar to the sample \( x_i \) under consideration.

Then, from (7) the associated extremum of \( J_{r_m}(M,V) \) is [3]:

\[ J_{r_m}(V) = \min_{M} J_{r_m}(M,V) = \sum_{i=1}^{c} \left[ \sum_{k=1}^{N} \left( \frac{1}{s_{ik}} \right)^{1/(\alpha_m - 1)} \right] 

(17)

In the commonly adopted case of \( s_{ik} \) being the Euclidean quadratic distance \( (x_i - \bar{v}_i)^T I(x_i - \bar{v}_i) \) between the sample \( x_i \) and the centroids \( \bar{v}_i \) of the \( i \)-th cluster, the optimal membership values read

\[ \mu_{ik}^* = \left( \frac{1}{s_{ik}} \right)^{1/(\alpha_m - 1)} \quad i = 1, 2, \ldots, c; \quad k = 1, \ldots, N \]

(16a)
In this case, the memberships assigned to sample \( \tilde{x}_i \) are functions of the inverse of the distances from the class prototypes \( \tilde{v}_i, i = 1, 2, \ldots, c \). The inverse distance serves to weigh a class more if its representative prototype is closer and less if it is farther from the sample \( \tilde{x}_i \) under consideration.

### 2.3.1.2 Case \( r_m \to 1 \)

In the limiting case of \( r_m \to 1 \), from (7),

\[
J_{r_m}(M, V) \to \sum_{i=1}^{c} \sum_{k=1}^{N_c} \mu_{ik} s_{ik}
\]

(18)

the argument given in [3] leads to the fact that \( \forall i, k \) the optimal membership values \( \mu_{ik}^* \) are boolean in nature i.e.

\[
\mu_{ik}^* = \begin{cases} 
1 & \text{if } s_{ik} = \min_j s_{ij}, \text{i.e. } \tilde{x}_i \in \Gamma_i \\
0 & \text{otherwise, i.e. } \tilde{x}_i \not\in \Gamma_i
\end{cases} \quad i = 1, 2, \ldots, c; \quad k = 1, 2, \ldots, N
\]

(19)

Indeed, considering a given data sample \( \tilde{x}_i \) and denoting by \( i \) the index of the cluster ‘closer’ to it (i.e. \( s_{ik} = \min_j s_{ij} \)), it can be seen that when \( r_m \to 1 \) the dominating term in the sum at denominator of (16) is that with \( j=i \) and correspondingly

\[
\mu_{ik}^* = \begin{cases} 
1 & j = i \\
0 & j \neq i
\end{cases}
\]

(20)

In this case, renaming \( \tilde{x}_i \in \Gamma_i \), for which (20) is verified, as \( \tilde{x}_i^{(i)}, k=1,\ldots,N_i \), and taking \( s_{ik} \) as the Euclidean quadratic distance \( (\tilde{x}_i^{(i)} - \tilde{v}_i) \mathbf{T} I(\tilde{x}_i^{(i)} - \tilde{v}_i) \) between the sample \( \tilde{x}_i^{(i)} \) and the center \( \tilde{v}_i \) of the \( i \)-th cluster, the extremum of the objective function (18) becomes

\[
J^*_{r_m}(V) = \sum_{i=1}^{c} \sum_{k=1}^{N_i} |\tilde{x}_i^{(i)} - \tilde{v}_i|^2
\]

(18a)

which is the objective function (6) at the basis of the Hard C-Means algorithm.

### 2.3.2 Determination of optimal prototype parameters (clusters centroids) \( \tilde{v}_i^* \)

We now turn to the problem of finding the optimal prototype set \( V = \{\tilde{v}_1^*, \tilde{v}_2^*, \ldots, \tilde{v}_c^*\} \). As in the case of the Hard C-Means, this can be done by imposing, with respect to \( V \), the first-order necessary conditions for a local minimum of \( J_{r_m}(M^*, V) \) in (7), knowing the optimal membership functions \( M^* \) given by (16):

\[
\frac{\partial J_{r_m}(M^*, V)}{\partial \tilde{v}_i} = \sum_{k=1}^{N_c} (\mu_{ik}^*)^* \frac{\partial s_{ik}}{\partial \tilde{v}_i} = 0 \quad i = 1, 2, \ldots, c
\]

(21)

To proceed further, we need to specify the parameterization of \( s_{ik} = s_{ik}(\tilde{x}_i, \tilde{v}_i) \) in terms of the prototype parameter \( \tilde{v}_i \).

Considering:

\[
s_{ik} = s_{ik}(\tilde{x}_i, \tilde{v}_i) = (\tilde{x}_i - \tilde{v}_i) \mathbf{T} I(\tilde{x}_i - \tilde{v}_i) \quad i = 1, 2, \ldots, c; \quad k = 1, 2, \ldots, N
\]

(22)

then eq. (21) gives:
\[
\sum_{k=1}^{N} (\mu_k^*)^\alpha (\tilde{x}_k - \tilde{v}_i^*) = 0 \quad i = 1, 2, \ldots, c
\]  

(23)

where the optimality asterisk now identifies also the optimal \( \tilde{v}_i \)'s. Then, we can solve for the optimal prototype parameters \( \tilde{v}_i^* \):

\[
\tilde{v}_i^* = \frac{\sum_{k=1}^{N} (\mu_k^*)^\alpha \cdot \tilde{x}_k}{\sum_{k=1}^{N} (\mu_k^*)^\alpha} \equiv m_\beta \quad i = 1, 2, \ldots, c
\]  

(24)

We call \( m_\beta \) the fuzzy mean of cluster \( i \) in recognition of its limiting property that when \( r_m \to 1 \),

\[
\frac{\sum_{k=1}^{N} \tilde{x}_k}{N_i} \equiv \hat{m}_i \quad \text{which is the sample mean of the cluster } \Gamma_i \text{ as obtained in (6) for the case of hard partitioning.}
\]

In this case, by construction all clusters induced by the algorithm will bear a spherical shape.

2.3.3 Operative procedure

1. Assign the number \( c \) of clusters in which you want to partition the data set \( X \);

2. Assign a value to the fuzziness index \( r_m \);

3. Perform an initial fuzzy partition of the data set into \( c \) fuzzy clusters. If no a priori information is available to guide an approximate partition, the initial partition can be random: this can be done by selecting the \( cN \) values of \( \mu_k^*, i=1,2,\ldots,c, k=1,\ldots,N \), from a uniform distribution in \( [0,1) \) in such a way as to respect constraints (8) and (9), or by selecting randomly \( c \) cluster centers \( \tilde{v}_i, i=1,2,\ldots,c \) and then computing the \( cN \) values of \( \mu_k^*, i=1,2,\ldots,c, k=1,\ldots,N \), from (16a);

4. With the given \( M=\{ \mu_k \} \), compute the optimal prototype vectors \( \tilde{v}_i^*, i=1,2,\ldots,c \) from (24);

5. With the optimal \( V^* = \{ \tilde{v}_i^* \} \) compute the optimal membership values \( \mu_k^*, i=1,2,\ldots,c, k=1,\ldots,N \) from (16a).

6. Repeat steps 4 and 5 until no change in \( J_n (M,V) \) is obtained from one iteration to the next or, in other words, until the maximum change in the membership functions, from one iteration to the next, is below a predefined convergence threshold \( \varepsilon \), i.e.

\[
\max_k \left| \mu_k^\alpha - \mu_k^{\alpha-1} \right| \leq \varepsilon
\]

\( \alpha \) denoting the iteration number.

The Fuzzy C-Means algorithm always converges to a strict local minimum of \( J_n (M,V) \) starting from an initial guess of \( \mu_k^*, i=1,2,\ldots,c, k=1,\ldots,N \), and thus different initial choices might lead to different local minima [1].

2.3.4 Generalization to include Fuzzy Covariance

It is possible to generalize the algorithm illustrated in 2.3.1 by replacing (22) with an inner product induced norm metric of the form [4]
\[ s_{ik} = s_{ik}(\bar{x}_k, \bar{v}_i, M_i) = (\bar{x}_k - \bar{v}_i)^T M_i (\bar{x}_k - \bar{v}_i) \quad i = 1, 2, ..., c; \quad k = 1, 2, ..., N \] (25)

with \( M_i \) symmetric and positive definite for any \( i \). In this case, the objective function \( J_{\omega}(M, V, B) \)

\[ J_{\omega}(M, V, B) = \sum_{i=1}^{c} \left( \sum_{k=1}^{N} \mu_{ik} (\bar{x}_k - \bar{v}_i)^T M_i (\bar{x}_k - \bar{v}_i) \right) \] (26)

where \( B = \{ M_i \} \), is linear in \( M_i \), thus leading to a singular problem. The objective function can be made as small as desired by simply rendering \( M_i \) less positive definite. To get a feasible solution, we must constrain \( M_i \). Ideally, we would like the metric to handle different scalings along the directions of the feature space, thus allowing variations in the shape of each induced cluster but at the same time not letting the metric grow without bound in any direction. A way of accomplishing this by means of a single parameter \( \rho > 0 \) is to constrain the determinant, \( \left| M_i \right| \) thus inducing a volume constraint on the clusters. Adjoining the additional constraints:

\[ \left| M_i \right| = \rho_i, \quad \text{with} \quad \rho_i > 0 \quad \text{and} \quad i = 1, ..., c \]

to equation (7b) leads to consider the extended objective function:

\[ L(Z, V, B, \Lambda, \Omega) = \sum_{i=1}^{c} \left( \sum_{k=1}^{N} (z_{ik})^{2\omega} s_{ik} + \sum_{k=1}^{N} \lambda_k \left( \sum_{i=1}^{c} z_{ik}^2 - 1 \right) + \sum_{i=1}^{c} \omega_i \left( \left| M_i \right| - \rho_i \right) \right) \] (27)

where \( \Omega = \{ \alpha_i \} \) is a set of Lagrange multipliers. As usual, the optimal values of \( \bar{v}_i \) and \( z_{ik} \) are obtained by setting to 0 the gradient of \( L(Z, V, B, \Lambda, \Omega) \) with respect to \( \bar{v}_i \) and \( z_{ik} \) respectively:

\[ \frac{\partial L}{\partial \bar{v}_i} = -2 \sum_{k=1}^{N} (z_{ik})^{2\omega} M_i (\bar{x}_k - \bar{v}_i) \quad \text{with} \quad i = 1, 2, ..., c \] (28)

\[ \frac{\partial L}{\partial z_{ik}} = 2r_k \cdot (z_{ik})^{2\omega-1} s_{ik} + 2z_{ik} \cdot \lambda_k \quad \text{with} \quad k = 1, ..., N \quad \text{and} \quad i = 1, 2, ..., c \] (29)

As in the case of the Euclidean distance (eqs. (24) and (16)), \( \bar{v}_i \) and \( \mu_{ik} \) are given by:

\[ \bar{v}_i = \frac{\sum_{k=1}^{N} (z_{ik})^{2\omega} \cdot \bar{x}_k}{\sum_{k=1}^{N} (z_{ik})^{2\omega}} = \frac{\sum_{k=1}^{N} (\mu_{ik})^{2\omega} \cdot \bar{x}_k}{\sum_{k=1}^{N} (\mu_{ik})^{2\omega}} = m_i \] (30)

\[ \mu_{ik}^* = \left( \frac{1}{s_{ik}} \right)^{\frac{1}{2\omega-1}} \] (31)

Thus, using the optimal membership functions \( \mu_{ik} = \mu_{ik}^* \), \( \bar{v}_i \) is still the fuzzy mean of cluster \( i \). For what concerns the distance matrices \( M_i \), by setting to zero the gradient of \( L(Z, V, B, \Lambda, \Omega) \) with respect to \( M_i \), we get:
\[
\frac{\partial L}{\partial \mu_i} = \sum_{i=1}^{N} (z_a) \sum_{v_i} (\bar{x}_i - \bar{v}_i) (\bar{x}_i - \bar{v}_i)^T + \omega_i M_i^{-1}
\] (32)

To get eq.(32) from eq (27), we have used the 2 identities:
\[
\frac{\partial}{\partial A} (\bar{x}^T A \bar{x}) = \bar{x} \bar{x}^T, \quad \frac{\partial}{\partial A} |A| = |A|^{-1}
\]
which hold for a non-singular matrix \( A \) and any compatible vector \( \bar{x} \). Eq. (32) evaluated for \( \bar{v}_i = \bar{v}_i^* \) gives the optimal metrics \( M_i^* \):
\[
M_i^{-1} = \frac{1}{\omega_i} \left( \sum_{v_i} (\mu_i)^* (\bar{x}_i - \bar{v}_i) (\bar{x}_i - \bar{v}_i)^T \right)
\] (33)

Defining the fuzzy covariance matrix relative to cluster \( \Gamma_i \) by:
\[
P_{\mu} = \frac{\sum_{k=1}^{N} (\mu_{ik})^* (\bar{x}_k - \bar{v}_i) (\bar{x}_k - \bar{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^*}
\] (34)

which reduces to the sample covariance matrix when \( r_m \to 0 \). Now, considering for the determination of \( \omega_i \) that \( |M_i| = \rho_i \), it is possible to write eq. (33):
\[
(M_i^*)^{-1} = \left( \frac{1}{\rho_i} P_{\mu} \right)^{1/2} P_{\mu}^{1/2}
\] (35)

The presence of the fuzzy covariance into the optimal matrix \( M_i^* \) of (35) allows obtaining ellipsoidal clusters.

2.3.5 Operative procedure

1. Assign the number \( c \) of clusters in which you want to partition the data set \( X \);
2. Assign a value to the fuzziness index \( r_m \);
3. Perform an initial fuzzy partition of the data set into \( c \) fuzzy clusters. If no a priori information is available to guide an approximate partition, the initial partition can be random: this can be done by selecting the \( c \cdot N \) values of \( \mu_{ik} \), \( i=1,2,...,c, \ k=1,...,N \), from a uniform distribution in [0,1) in such a way as to respect constraints (8) and (9), or by selecting randomly \( c \) cluster centers \( \bar{v}_i \), \( i=1,2,...,c \) and then computing the \( c \cdot N \) values of \( \mu_{ik} \), \( i=1,2,...,c, \ k=1,...,N \), from eq. (31);
4. Compute the the fuzzy covariance matrices \( P_{\mu} \) by eq.(34);
5. With the given \( M=\{ \mu_k \} \), compute the optimal prototype vectors \( \bar{v}_i^* \), \( i=1,2,...,c \) from (30);
6. With the optimal \( V^*=\{ \bar{v}_i^* \} \) compute the optimal membership values \( \mu_{ik}^* \), \( i=1,2,...,c, \ k=1,...,N \) from:
\[
\mu_{ik} = \left( \frac{1}{s_{ik}} \right)^{r_{ik}} \frac{1}{\sum_{j=1}^{c} \left( \frac{1}{s_{jk}} \right)^{r_{jk}}}
\]

where \( s_{ik} = d(\tilde{x}_i, \tilde{v}_i, M^*_i) = (\tilde{x}_i - \tilde{v}_i)^T M^*_i (\tilde{x}_i - \tilde{v}_i) \) in which \( M^*_i \) is given by eq.(35);

7. Repeat steps 4, 5 and 6 until no change in \( J_{\alpha}(M, V, B) \) is obtained from one iteration to the next or, in other words, until the maximum change in the membership functions, from one iteration to the next, is below a predefined convergence threshold \( \varepsilon \), i.e.

\[
\max_{\alpha} \left| \mu_{ik}^\alpha - (\mu_{ik})^{\alpha-1} \right| \leq \varepsilon
\]

\( \alpha \) denoting the iteration number.

### 2.4 Clustering validity

In practical applications, we need a method to verify the validity of the classification structure induced by the applied clustering algorithm. The answer on the quality of the partition obtained is typically sought in terms of quantitative measures of separation among clusters and cohesion within clusters.

In case of unsupervised data partition the quality of the obtained data partition is typically sought in terms of quantitative measures of separation among clusters and cohesion within clusters. In Sections 2.4.1, 2.4.2, 2.4.3 measures of the quality of unsupervised data partitions will be introduced.

The quality of a clustering process depends on many factors such as the method of initialisation, the choice of the number of classes \( c \) and the clustering method itself (particularly the similarity measure \( s_{ik} \) adopted and the associated objective function \( J \) to be optimised). The method of initialisation requires a good a priori estimate of the clusters and is application dependent. Moreover, the quality of the clustering depends on the choice of an optimal number of clusters \( c^* \) in which to partition the dataset if it is not a priori known. In this respect, in Section 2.4.4 a procedure to determine the optimal number of clusters in which the data can be divided is proposed.

#### 2.4.1 Partition coefficient

The partition coefficient \( F(M, V, c) \) measures the amount of ‘overlap’ between clusters and is defined as follows [14]:

\[
F(M, V, c) = \frac{1}{N} \sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{ik}^{\alpha} \leq 1
\]  

(36)

It can be seen that the contribution of small \( \mu_{ik}^{\alpha} \)'s is reduced greatly by the quadratic operator, and significantly more than those \( \mu_{ik}^{\alpha} \)'s close to one. By definition, then, \( F(M, V, c) \) is inversely proportional to the overall average overlap between pairs of fuzzy sets and thus it measures the similarity or closeness of all input samples to the corresponding cluster prototypes. In other words, if each data point \( \tilde{x}_k, k=1,...,N \), is closely associated with only one cluster \( i \), i.e. \( \mu_{ik}^{\alpha} \sim 1 \) and \( \mu_{ij}^{\alpha} \sim 0 \) \( \forall j \neq i \), then \( F(M, V, c) \) is large and the dispersion in the cluster data
is small. In the limit when there is no sharing of membership of any data between any pair of fuzzy clusters, \( F(M, V, c) = 1 \).

The main disadvantages of the partition coefficient as a validity measure are the lack of direct connection to any geometrical property in \( \mathbb{R}^k \) of the data set under examination and its monotonic decreasing tendency with \( c \).

### 2.4.2 Partition entropy

The partition entropy \( E(M, V, c) \) is defined as [14]:

\[
E(M, V, c) = -\frac{1}{N} \sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{ik} \log(\mu_{ik})
\]  

(37)

When all \( \mu_{ik} \)'s have values close to 0.5, which represents a high degree of fuzziness in the clusters, \( E(M, V, c) \) is large, thus indicating a poor quality of the clusters. On the contrary, if all \( \mu_{ik} \)'s have values close to 0 or 1, \( E(M, V, c) \) is small and indicates a good clustering.

This cluster validity measure also does not bear direct connection to any geometrical property of the data set under examination and decreases monotonically with \( c \).

### 2.4.3 Compactness and separation function

The compactness and separation validity function \( S(M, V, c) \) is defined as [14]:

\[
S(M, V, c) = \frac{1}{N} \sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{ik}^2 s(\tilde{x}_k, \tilde{v}_i) = \frac{1}{N} J_z(M, V) \frac{\min_{i,j} s(\tilde{v}_i, \tilde{v}_j)}{\min_{i,j} s(\tilde{v}_i, \tilde{v}_j)}
\]  

(38)

where \( J_z(M, V) \) is given by (7) with \( r_m=2 \). Again, the most common case is that of \( s(\tilde{x}_k, \tilde{v}_i) \) being the Euclidean quadratic distance \( (\tilde{x}_k - \tilde{v}_i)^T I (\tilde{x}_k - \tilde{v}_i) \), so that \( J_z(M, V) \) is given by (6a) with \( r_m=2 \).

The quantity \( \sigma_i = \sum_{k=1}^{N} \mu_{ik}^2 s(\tilde{x}_k, \tilde{v}_i) \) can be taken as representing the (fuzzy) variation or dispersion of the data in cluster \( i \). Then, \( \sigma_i/N_j \) is the average (fuzzy) dispersion in cluster \( i \), where \( N_j = \sum_{k=1}^{N} \mu_{ik} \) is the fuzzy cardinality of class \( i \), i.e. the fuzzy number of samples in \( i \), and \( \sigma_f = \sum_{i=1}^{c} \sigma_i \) is the total (fuzzy) variation or dispersion characterizing the clustering of data set \( X \).

Obviously, both \( \sigma_f \) and \( \sigma_f \) depend on the data set and more importantly on their fuzzy partition in \( c \) clusters, i.e. on \( M \) and \( V \): ‘good’ partitions should give smaller values of \( \sigma_f \). The ratio \( \pi = \sigma_f/N_j \) can be taken as a measure of the compactness of class \( i \): then \( \pi = \sigma_f/N_j \) is the compactness of the fuzzy partition of the data set \( X \). The more compact the classes are, the smaller \( \pi \) is. Note that \( \pi \) depends on the data set \( X \), although it is independent on the cardinality \( N \) of the set, and more importantly on how the data points are partitioned in clusters.

The quantity \( \xi = \min_{i,j} s(\tilde{v}_i, \tilde{v}_j) \) can be taken as a measure of the separation of the fuzzy partition: a large value of \( \xi \) indicates that all clusters are well separated.

Then, the compactness and separation validity function \( S(M, V, c) \) has the intuitive meaning of the ratio between compactness \( \pi \) and separation \( \xi \), i.e.
\[ S(M, V, c) = \frac{\pi}{\xi} \]  

(39)

From a physical point of view, a good clustering procedure should make all input samples as close to their cluster centroids as possible and all cluster centroids as separated as possible. In this respect, the validity measure \( S(M, V, c) \) is the ratio between the average fuzzy similarity of the input data points \( \tilde{x}_k, k=1,2,...,N \), with the cluster centroids \( \tilde{v}_i, i=1,2,...,c \), and the minimum distance between cluster centroids. Then, a smaller \( S(M, V, c) \) indicates a partition in which all the clusters are overall compact and separate from each other.

The measure \( S(M, V, c) \) is directly related to the geometric properties of the data set \( X \) as represented by the similarity measure \( s_{ik} \). In the common case of \( s_{ik} = (\tilde{x}_k - \tilde{v}_i)^T I(\tilde{x}_k - \tilde{v}_i) \) the geometric properties accounted for in \( S(M, V, c) \) are the Euclidean quadratic distance measure and the locations of the cluster centroids \( \tilde{v}_i, i=1,2,...,c \) in \( \mathbb{R}^p \).

However, still the usual drawback applies also to \( S(M, V, c) \) which is monotonically decreasing when \( c \) becomes very large and close to \( N \), though it is true that for most practical applications to obtain satisfactory clustering results there is no need to compute \( S(M, V, c) \) for very large \( c \). A typical value of \( c_{\text{max}} = N/3 \) is a good choice as a stopping criteria in the iterations over \( c \) which would very likely not reach the starting point of decreasing tendency of \( S(M, V, c) \) with increasing \( c \).

### 2.4.4 A technique for the determination of the optimal number of clusters

Several cluster validity measures have been proposed as objective functions to be considered in the optimisation process with respect to the number of clusters [14],[15]. Denoting by \( Q(M, V, c) \) the generic validity measure, the problem of determining the optimal number of clusters \( c^* \) in which to partition the data set with respect to \( Q(M, V, c) \) as objective function can be formulated as

\[
\min_c \left\{ \min_{M, V} \left[ Q(M, V, c) \right] \right\} \quad \text{(40)}
\]

where \( \min \) stands for the \( \min \) or \( \max \) operator depending on the definition of \( Q(M, V, c) \).

Thus, the optimisation of the validity measure \( Q(M, V, c) \) with respect to \( c \) introduces an external loop of iterative optimisation over the possible values of \( c = 1,2,...,N \): at each iteration, in correspondence of the given value of \( c \) a nested optimisation with respect to \( M, V \) is performed through a clustering algorithm such as, for example, the Fuzzy C-Means optimisation procedure illustrated in 2.3.

A common drawback of the measures of the quality of the data partitions introduced in the previous Sections is their monotonic decreasing tendency with \( c \). Thus, the optimization of the validity measure performed with (40) leads to a large, incorrect \( c^* \). To overcome this problem an ad hoc defined penalty function can be introduced in the definitions of \( Q(M, V, c) \) to counter their decreasing tendency.

### 2.5 Possibilistic clustering

The fuzzy logic approaches to clustering generally impose constraints (8) and (9) on the membership functions, i.e.:

\[
0 \leq \mu_{ik} \leq 1 \quad i=1,2,...,c, \quad k=1,2,...,N
\]  

(8)
The ‘probabilistic’ constraint (9), that the memberships of a given pattern must sum up to 1, is a generalization of the condition in a ‘hard’ (crisp) partition which ensures that a pattern is a member of precisely one class only and avoids the trivial solution of all memberships equal to 0.

As a result of this constraint, the membership of a pattern to a cluster depends on the memberships to all other clusters, i.e., geometrically speaking, it depends on where the pattern is located with respect to not only that cluster but also the others. Indirectly, then, the membership of a pattern to a cluster becomes dependent on the number of clusters. Hence, in the framework of fuzzy clustering the membership functions take the meaning of degrees of sharing, i.e. they measure how much a pattern belongs to a cluster relatively to the others.

Under these conditions, two major drawbacks arise:

- The constrained memberships cannot distinguish between ‘equal evidence’ and ‘ignorance’ or, in other words, between ‘equally likely’ and ‘unknown’ membership to a cluster (see example below).

- Since most distance functions used in fuzzy clustering are geometric in nature, noise patterns, which typically lie far from the clusters, can drastically influence the estimates of the clusters prototypes and, hence, the final partition and the resulting classification.

However, the most important potential flaw of fuzzy clustering algorithms lies in the interpretation itself of the membership functions as degrees of sharing. Indeed, it would seem more natural in clustering applications to seek an interpretation of the membership functions as degrees of compatibility, or ‘tipicality’, to a given cluster. Clearly, one would like the memberships of representative (typical) patterns to be as high as possible, while unrepresentative (atypical) patterns should have low membership to all clusters, where representativeness (typicality) can be measured with respect to the clusters prototypical members. For this purpose, the ‘conservation of total membership’ constraint (9) is too restrictive since it gives rise to relative membership values, dependent on the number of clusters.

The following simple examples illustrate the problems associated with the use of constraint (9), in fuzzy clustering methods [7].

Figure 1 shows a case with two clusters. The Fuzzy C-Means algorithm produces different values of membership to cluster 1 for the points A (0.935) and B (0.721), even though they are equally typical of this cluster since they are equidistant to its prototypical member, the cluster center. This situation is precisely due to constraint (9) which forces point B to give up some of its membership to cluster 1 in order to be shared with cluster 2, i.e. to have a non zero membership to such cluster. Thus, the final values of membership of point B to clusters 1 (0.721) and 2 (0.279) reflect its ‘sharing’ between the two clusters. Similarly, point A and point C turn out to have similar values of membership to cluster 1 (0.935 and 0.956, respectively), even though C is more typical than A. The possibilistic clustering approach (without recomputation of the value of \( \eta \)), which will be illustrated
in Section 2.5.2, would give the following memberships to clusters 1 and 2: (0.230, 0.020) for point A; (0.256, 0.091) for point B; (0.616, 0.047) for point C. These relative degrees of within-cluster strength seem to more reasonably quantify the typicality of the points with respect to their ‘similarity’ to the prototypes of the clusters.

Figure 1: Example of a data set with two clusters. By the Fuzzy C-Means algorithm, points A and B turn out to have different values of membership to cluster 1 even though they are equally typical of such cluster; on the contrary, points A and C turn out to have similar membership values with respect to cluster 1 even though they are not equally typical of such cluster. The symbol * gives the position of the cluster centers found by the Fuzzy C-Means whereas the symbol x denotes those found by the possibilistic approach

Figure 2 shows two intersecting circular shell clusters in which point A is a ‘good’ member of both clusters whereas point B is a ‘poor’ member. In this case, constraint (9) would force onto both points A and B a membership of about 0.5 to the two clusters. These membership assignments are coherent with an interpretation of ‘degree of sharing’ of data points among clusters but counterintuitive with respect to the degree of compatibility of a data point with a cluster, as represented by its prototype.
Figure 2: Example of a data set with two intersecting clusters. By the Fuzzy C-Means algorithm, points A and B turn out to have membership values of about 0.5 to both clusters even though point A is a ‘good’ member of both clusters whereas point B is a ‘poor’ member.

Figure 3: Example of a data set with two clusters and two noise points A and B. By the Fuzzy C-Means algorithm, points A and B turn out to have membership values of about 0.5 to both clusters even though point B is much less representative of the characteristics of either cluster than point A.
The possibilistic clustering approach (without recomputation of the value of $\eta$), which will be illustrated in the next Section, assigns the following memberships: (0.081, 0.039) for point A; (0.044, 0.024) for point B. These relative degrees of within-cluster strength seem to more reasonably quantify the tipicality of the points with respect to their ‘similarity’ to the prototypes of the clusters.

### 2.5.2 Possibilistic Clustering algorithm

To overcome the above limitations of the FCM algorithm, the clustering problem can be recast into the framework of possibility theory [5,16,17].

As usual, a data set $X$ in an $h$-dimensional feature space $S \subset \mathbb{R}^h$ is considered. In the framework of possibilistic clustering, the prototype $\tilde{v}_i$, of the generic $i$-th cluster, $i=1,2,\ldots,c$, defines an elastic constraint and the membership function value $\mu_{ik} = \mu(\tilde{x}_k, \tilde{v}_i)$ of the pattern $\tilde{x}_k$, $k=1,\ldots,N$, to the $i$-th cluster parameterised by $\tilde{v}_i$ measures the degree to which such constraint must be stretched to match $\tilde{x}_k$ [18,19].

In this interpretation, then, the membership function $\mu_{ik}$ represents the degree of compatibility of the pattern $\tilde{x}_k$ with the prototypical member $\tilde{v}_i$ of cluster $i$. If the classes represented by the clusters are thought of as a set of fuzzy sets defined over the Universe of Discourse (UOD), then there should be no constraint on the sum of the memberships. The only constraint is that the membership values do represent degrees of compatibility, or possibility, i.e. they must lie in [0,1]. This is achieved by substituting the fuzzy clustering constraints (8)-(10) with the following [7]:

\begin{align}
0 &\leq \mu_{ik} \leq 1 & i = 1, 2, \ldots, c, k = 1, 2, \ldots, N \\
\max_i \mu_{ik} &> 0 & k = 1, 2, \ldots, N \\
0 < \sum_{k=1}^{N} \mu_{ik} < N & & i = 1, 2, \ldots, c
\end{align}

(41) (42) (43)

where constraint (42) simply ensures that the set of fuzzy clusters covers the entire UOD.

A possibilistic partition derived under these constraints defines a set of distinct, uncoupled possibilistic distributions (and the corresponding fuzzy subsets) over the UOD.

To introduce the possibilistic clustering algorithm, we recall that the Fuzzy C-Means clustering algorithm aims at minimizing the following objective function with respect to both the prototype parameters $\tilde{v}_i$ and the membership values $\mu_{ik}$, $i=1,2,\ldots,c$, $k=1,2,\ldots,N$ (Section 2.3):

\begin{equation}
J_{\mu}(M,V) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^r s_{ik}
\end{equation}

(7)

where $M = \{ \mu_{ik} \}$ is the set of all values of membership of all data in $X$ to all clusters, $r_{\text{mu}}>1$ is an exponential weight which controls the degree of fuzziness of the clusters, $s_{ik}$ is a similarity or distance measure between the pattern $\tilde{x}_k$ and the prototype $\tilde{v}_i$ of cluster $i$. The sets $M$ and $V$ define the fuzzy partition of the indexed set $X$. As mentioned before, for a fuzzy partition the elements $\mu_{ik}$, $i=1,2,\ldots,c$, $k=1,2,\ldots,N$, of $M$ must satisfy the conditions (8)-(10).

If we were to simply relax the constraint (9), the solution to the minimization of the objective function (7) would be the trivial solution of all memberships being equal to 0. Instead, to obtain high memberships for patterns
which are good representative of the clusters and low memberships for the unrepresentative data points, the following objective function can be formulated [7]:

\[
J_a(M, V) = \sum_{i=1}^{N} \sum_{k=1}^{K} (\mu_{ik})^\alpha s_{ik} + \sum_{i=1}^{N} \eta_i \sum_{k=1}^{K} (1 - \mu_{ik})^\alpha 
\]

(44)

where \(\eta_i\) are suitable positive numbers. The first term demands that the distances \(s_{ik}\) of the patterns \(x_k\) from the clusters prototypes \(v_i\) be as small as possible, whereas the second term requires that the \(\mu_{ik}\) be as large as possible, thus avoiding the trivial solution.

Since the elements \(\mu_{ik}\) of \(M\) are independent of each other, minimizing \(J_a(M, V)\) with respect to the set \(M\) is equivalent to minimizing the following individual objective function with respect to each of the \(\mu_{ik}\) (provided that the resulting individual solution lies in \([0, 1]\)):

\[
J_{ik}(v_i, \mu_{ik}) = \mu_{ik}^\alpha s_{ik} + \eta_i (1 - \mu_{ik})^\alpha 
\]

(45)

Differentiating with respect to \(\mu_{ik}\) and setting it equal to 0, one obtains:

\[
\mu_{ik}^* = \frac{1}{1 + \left(\frac{s_{ik}}{\eta_i}\right)^{\frac{1}{\alpha - 1}}}
\]

(46)

which obviously satisfies the desired constraints (41)-(43). Hence, in each iteration of the clustering algorithm the updated optimal value of \(\mu_{ik}^*\) depends only on the distance of the pattern \(x_k\) from the clusters prototypes \(v_i\), which is an intuitive result: from the standpoint of compatibility to a class, the membership of a pattern to a cluster should be determined solely by how far it is from the prototype of the class, and should not be coupled with its location with respect to other classes. Equation (46) defines a possibility (membership) distribution function for cluster \(i\) over the universe of discourse consisting of all patterns \(x_k\). The value of \(\alpha\) determines the fuzziness of the final possibilistic partition and the shape of the possibility distribution. When \(\alpha \to 1\), the membership functions are hard and when \(\alpha \to \infty\) they are maximally fuzzy. In passing we note that the form in (46) is similar to the membership function derived from empirical studies in [20] as a good representation of vague classes.

The value of \(\eta_i\) determines the distance at which the membership value of a pattern to the cluster \(i\) becomes 0.5 and thus it relates to the size and shape of the cluster. In this sense, it needs to be chosen on the basis of the desired spread of the possibility (membership) distribution of each cluster which determines the zone of influence of the cluster. On the other hand, \(\eta_i\) determines the importance of the second term in the objective function (45): if both terms have to be weighed equally, then \(\eta_i\) should be of the order of \(s_{ik}\). In practice, \(\eta_i\) is taken proportional to the average fuzzy infrastructure distance of all patterns to cluster \(i\) [7]:

\[
\eta_i = K \frac{\sum_{k=1}^{N} \mu_{ik}^\alpha s_{ik}}{\sum_{k=1}^{N} \mu_{ik}^\alpha} \quad i = 1, 2, ..., c 
\]

(47)

with \(K\) typically chosen unitary. A modification of (47) amounts to taking the average considering only the ‘good’ patterns of cluster \(i\), i.e. those belonging to a given \(\alpha\)-cut which have membership in \(i\) greater than a given
value \( \alpha \). As the iterations proceed, clusters with larger values of \( \eta \) have more mobility because they see more patterns than well-formed compact clusters.

A good procedure seems to be to: i) approximate the values of \( \eta \) by (47) using an initial fuzzy partition; ii) run the clustering algorithm a first time; iii) after the algorithm converges, recompute more accurate values for the \( \eta \) by limiting the average (47) to the patterns of a given \( \alpha \)-cut; iv) run the clustering algorithm a second time. The second run of the algorithm with refined values of \( \eta \) allows the resultant memberships to be nearly ‘noise-free’ and it is necessary only if the actual values of cluster memberships are required. On the contrary, if only the relative degrees of within-cluster strength are needed (e.g. to generate cluster parameters or produce a hard partition), then the second run can be omitted. Any value of \( \alpha \) between 0.1 and 0.4 is reported to yield consistent results [7].

In the case of the generalized algorithm which includes the fuzzy covariance, analog to the extension of the Fuzzy C-Means (Section 2.3.4), a good choice for the value of \( \eta \) would be:

\[
\eta_i = Q^i = \frac{1}{P_{ji}} \quad i = 1, 2, ..., c
\]

where \( Q \) is the expected hypervolume of cluster \( i \), \( P_{ji} \) is the fuzzy covariance matrix relative to cluster \( i \) and \( h \) is the dimensionality of the feature space.

We finally turn to the problem of finding the optimal prototype set \( \mathbf{V}^* = \{ \hat{v}_1^*, \hat{v}_2^*, ..., \hat{v}_c^* \} \). Since the added term in the objective function (44) is independent of \( v_i \) and of \( s_{ik} \), the derivatives of (44) with respect to \( v_i \) are the same as in the Fuzzy C-Means algorithms and so is the updating of the optimal prototypes which depends on the adopted distance measure \( s_{ik} \).

If the distance is a Euclidean metric, as in the case of the basic Fuzzy C-Means algorithm:

\[
s_{ik} = s_{ik} (\hat{x}_k, \bar{v}_i) = (\hat{x}_k - \bar{v}_i)^T \frac{1}{2} (\hat{x}_k - \bar{v}_i) \quad i = 1, 2, ..., c; \quad k = 1, 2, ..., N
\]

then the optimal prototype parameters \( \hat{v}_i^* \) read:

\[
\hat{v}_i^* = \frac{\sum_{k=1}^{N} (\mu_{ik}^*)^\alpha \cdot \hat{x}_k}{\sum_{k=1}^{N} (\mu_{ik}^*)^\alpha} \equiv m_{fi} \quad i = 1, 2, ..., c
\]

where \( m_{fi} \) is called the fuzzy mean of cluster \( i \) in recognition of its limiting property that when \( r_m \to 1 \),

\[
m_{fi} = \frac{\sum_{k=1}^{N} \hat{x}_k}{N_i} = \frac{\sum_{i=1}^{N} \hat{x}_j(i)}{N_i} \equiv \hat{m}_j \quad \text{which is the sample mean of the cluster } \Gamma_i \text{ as obtained in the case of hard partitioning. In this case, we obtain the ‘Possibilistic C-Means algorithm’ which, by construction, generates clusters of spherical shape.}

The results of the possibilistic clustering algorithm depend on the initialization, just as for any clustering technique. However, the situation is somewhat worse because since each pattern sees only one cluster at a time, rather than all the clusters simultaneously, the clusters do not possess great mobility. Therefore, it is important to start from a reasonably good initial membership distribution. This can be obtained from any hard or fuzzy
clustering algorithm. For example, the Fuzzy C-Means algorithm provides a good initialization and the estimates of $\eta$ using (47) are quite adequate, provided that there is not too much noise. In fact, the Fuzzy C-Means algorithm assigns relatively high values of membership to noise patterns. To overcome this problem, a good approach seems to use the $\eta$ estimated using the Fuzzy C-Means memberships in (47) only for a couple of iterations in the possibilistic algorithm and then re-estimate them using only the patterns in a given $\alpha$-cut [7].

2.5.3 Operative procedure
1. Assign the number $c$ of clusters in which you want to partition the data set $X$;
2. Assign a value to the fuzziness index $r_m$;
3. Perform an initial possibilistic partition of the data set into $c$ fuzzy clusters, e.g. by Fuzzy C-Means;
4. Approximate $\eta_i$ using (47);
5. With the given $M=\{\mu_i\}$, compute the optimal prototype vectors $\tilde{v}_i^*, i=1,2,...,c$ from (50);
6. With the optimal $V^*=\{\tilde{v}_i^*\}$ compute the optimal membership values $\mu_i^*, i=1,2,...,c, k=1,2,...,N$ from (46);
7. Repeat steps 5 and 6 until no change in $J_\alpha(M,V)$ is obtained from one iteration $\alpha$-1 to the next $\alpha$ or, in other words, until the maximum change in the membership functions, from one iteration to the next, is below a predefined convergence threshold $\varepsilon$, i.e.
\[
\max_k (\mu_i^*) - (\mu_i^*) \leq \varepsilon
\]
$\alpha$ denoting the iteration number for the clustering cycle;
8. Re-estimate $\eta_i$ considering in (47) only the data patterns in a given $\alpha$-cut;
9. Repeat steps 5, 6 and 7.

2.5.4 Generalization to include the fuzzy covariance in the possibilistic algorithm
As in the case of the FCM algorithm (Section 2.3.4), it is possible to generalize the clustering algorithm by replacing (22) with an inner product induced norm metric of the form
\[
s_a = s_a(\tilde{x}_i, \tilde{v}_j, M_i) = (\tilde{x}_i - \tilde{v}_j)^T M_i (\tilde{x}_i - \tilde{v}_j) \quad i = 1, 2, ..., c; \quad k = 1, 2, ..., N
\]
with $M_i$ symmetric and positive definite $\forall i$. Then, the updating of the optimal prototypes is done using (50), and the optimal $M_i^*$ are still given by eqs.(34) and (35).
3. A CLUSTERING TECHNIQUE FOR CLASSIFICATION

The task of pattern classification may be viewed as a problem of partitioning of objects (hereafter also called data patterns) into classes. From a mathematical point of view, a classifier is a mapping function \( \Phi(\cdot) \) which assigns an object \( \tilde{x} \) in an \( h \)-dimensional domain \( S \subset \mathbb{R}^h \) to a given class \( i \). Often in engineering, due to the complexity of the problems, it is not possible to determine the analytical expression of the function \( \Phi \), i.e. to individuate the exact boundaries of the classes in the \( S_f \) space.

Over the last 25 years, a large number of artificial intelligence based techniques for pattern classification have been proposed. In particular, Artificial Neural Networks [21,22,23] and Fuzzy logic based systems [21,24] have been successfully applied.

In these empirical classification techniques, the classifier is built through a process of learning based on a set of classification examples. In other words, in the learning phase, labelled patterns belonging to the different classes are utilized to estimate the values of the parameters of the empirical classification function. This kind of technique is termed “supervised” in opposition to the “unsupervised” clustering techniques introduced in Section 2, and the available pre-classified data are termed “training data” [4].

In this Section, the classification problem is tackled within a clustering approach properly modified to render it supervised. An evolutionary procedure optimizes a different Mahalanobis metric for each cluster by exploiting a priori known information regarding the true classes to which the patterns belong. The supervised evolutionary scheme is depicted in Section 3.1, its application within a Fuzzy C Means clustering algorithm in Section 3.1.1 and within a possibilistic clustering algorithm in Section 3.1.2.

3.1 A supervised evolutionary clustering approach to classification

Suppose that the physical problem requires building a classifier for assigning patterns to a number \( c \) of physical classes. To tackle the classification problem with a supervised technique we must have available a set \( X \) of \( N \) “training” data grouped into \( c \) subsets representing the classes \( \Gamma_i = \{ \tilde{x}_{ik} \} \) for \( i = 1, \ldots, c \), \( k = 1, \ldots, N_i \) with \( \sum \limits_{i=1}^{c} N_i = N \).

In other words, the number of classes \( c \) into which the patterns are to be classified is known and \( c \) groups of correctly pre-classified data are available to build the classifier, i.e. to find the mapping function \( \Phi(\cdot) \) which assigns the membership of \( \tilde{x}_k \) to class \( i \), i.e. \( \Phi(\tilde{x}_k) = i \).

In this Section, a supervised evolutionary clustering algorithm is developed to perform the pattern classification task.

The traditional, unsupervised clustering algorithms based on a Euclidean metric to measure compatibility (Sections 2.3 and 2.5.2) lead to spherical clusters that rarely are adequate to represent the data partition in practice [6]. A significant improvement is achieved by considering a different Mahalanobis metric for each cluster (Section 2.3.4 and 2.5.4), thus obtaining different ellipsoidal shapes and orientations of the clusters. Notice, however, that the approach to the classification problem offered by the FCM or possibilistic algorithms is unsupervised because the algorithms make no use of the a priori known information on the true physical classes of the training data and so the clustering is based only on the geometric grouping of the data. Then, the geometric clusters obtained by the FCM algorithm do not necessarily yield the actual physical classes.

To overcome this problem, the information on the membership of the available patterns \( \tilde{x}_k \) for \( k = 1, \ldots, N \), to the \( c \) a priori known classes, can be used to supervise the algorithm for finding the optimal Mahalanobis metrics such as...
to achieve geometric clusters as close as possible to the a priori known physical classes. Correspondingly, the clustering algorithm is said to be constructed through an iterative procedure of ‘training’ based on a set of available patterns, pre-labeled with their memberships to the a priori classes. The training procedure for the optimization of the metrics is carried out via an evolutionary procedure presented in the literature within a supervised fuzzy clustering scheme [6] and further extended to diagnostic applications [9,11]. The procedure will be employed within the FCM scheme (see Section 3.1.1) and the possibilistic clustering scheme (see Section 3.1.2).

To this purpose, the distance $D(\Gamma_i^t, \Gamma_i^*)$ between the set $\Gamma_i^t$ ($t = \text{true}$) of memberships of the $N$ available training patterns to the a priori known class $i$ and the corresponding set $\Gamma_i^*$ of the memberships to cluster $i=1,2,...,c$, obtained by using a clustering algorithm, is computed by:

$$D(\Gamma_i^t, \Gamma_i^*) = \sum_{i=1}^{c} \frac{|\mu_{ik}^t - \mu_{ik}^*|}{N}$$  \hspace{1cm} (51)

where $0 \leq \mu_{ik}^t \leq 1$ is the a priori known membership of the $k$-th pattern to the $i$-th physical class and $0 \leq \mu_{ik}^* \leq 1$ is the membership to the corresponding geometric cluster in the feature space found by the algorithm.

The target of the supervised optimization is the minimization of the distance $D(\Gamma^t, \Gamma^*)$ between the a priori known physical class partition $\Gamma^* = (\Gamma_1^*, \Gamma_2^*, ..., \Gamma_c^*)$ and the obtained geometric cluster partition $\Gamma^t = (\Gamma_1^t, \Gamma_2^t, ..., \Gamma_c^t)$:

$$D(\Gamma^t, \Gamma^*) = \sum_{i=1}^{c} D(\Gamma_i^t, \Gamma_i^*) = \sum_{i=1}^{c} \sum_{k=1}^{N} |\mu_{ik}^t - \mu_{ik}^*|$$  \hspace{1cm} (52)

The optimal membership functions $\mu_{ik}^*$, $i=1,2,...,c$, $k=1,...,N$, are obtained by using the clustering algorithm (FCM or possibilistic) introduced in Section 2 in which the distance $s_{ik} = s_j(\bar{x}_k^{\tau}, \bar{v}_i^{\tau})$ between the pattern $\bar{x}_k$ and the optimal cluster center $\bar{v}_i^{\tau}$ is computed by:

$$s_j(\bar{x}_k^{\tau}, \bar{v}_i^{\tau}, M_i^{\tau}) = (\bar{x}_k^{\tau} - \bar{v}_i^{\tau})^T M_i^{\tau} (\bar{x}_k^{\tau} - \bar{v}_i^{\tau})$$  \hspace{1cm} (25)

$M_i^{\tau}$ being the metric for the cluster $i$ proposed by the evolutionary supervised procedure and $T$ denoting the transpose operator.

The overall iterative training scheme can be summarized as follows (Figure 4):

1. At the first iteration ($\tau = 1$), initialize the metrics of all the $c$ clusters to the Euclidean metrics, i.e. $M_i^{\tau}(i) = I$, $i=1,2,...,c$, where $I$ is the identity matrix.

2. At the generic iteration step $\tau$, run the clustering algorithm (FCM or possibilistic) to partition the $N$ training data into $c$ clusters, based on the current metrics $M_i^{\tau}$ and on the “supervising” initial partition $\Gamma^t$ which sets the initial memberships of the $N$ patterns to $c$ clusters equal to the true memberships to the a priori known classes. Then, $\Gamma^{\tau}$ is set equal to the obtained optimal partition $\Gamma^* = (\Gamma_1^*, \Gamma_2^*, ..., \Gamma_c^*)$.

3. Compute the distance $D(\Gamma^t, \Gamma^{\tau})$ between the a priori known physical classes and the geometric clusters by eq.(52). At the first iteration ($\tau = 1$) initialize the best distance $D^+$ to $D(\Gamma^t, \Gamma^{(l)})$, $D^t$ to $D(\Gamma_i^t, \Gamma_i^{(l)})$ and the best metrics $M^{+\tau}$ to $M_i^{(l)}$ and go to step 5.

4. If $\Gamma^{\tau}$ is close to $\Gamma_i^{+\tau}$, i.e. $D(\Gamma^t, \Gamma^{\tau})$ is smaller than a predefined threshold $\varepsilon$, or if the number of iterations $\tau$ is greater than the predefined maximum allowed number of iterations $\tau_{\text{max}}$ stop: $\Gamma^{\tau}$ is the optimal cluster partition $\Gamma^*$; otherwise, if $D(\Gamma^t, \Gamma^{\tau})$ is less than $D^+$ upgrade $D^+$ to $D(\Gamma^t, \Gamma^{(\tau)})$, $M^{+\tau}$ to $M_i^{(\tau)}$ and $D^t$ to $D(\Gamma_i^t, \Gamma_i^{(\tau)})$.

5. Increment $\tau$ by 1. Update each matrix $M_i^{\tau'}$ by exploiting its unique decomposition into Cholesky factors [8], $M_i^{\tau'} = (G_i^{\tau'})^T G_i^{\tau'}$, where $G_i^{\tau'}$ is a lower triangular matrix with positive entries on the main diagonal. More precisely, at iteration $\tau$, the entries $g_{ik}^{\tau\tau'}(\tau)$ of the Cholesky factor $G_i(\tau)$ are updated as follows:
\[ g_{i,j}^i (\tau) = g_{i,j}^i + N_{i,j}^i (0, \delta_i^+) \quad \text{if } l_i < l_2 \]
\[ g_{i,j}^i (\tau) = \max\{10^{-5}, g_{i,j}^i + N_{i,j}^i (0, \delta_i^+)\} \quad \text{if } l_i = l_2 \]

where \( \delta_i^+ = \alpha D_i^+ \), \( \alpha \) is a parameter that controls the size of the random step of modification of the Cholesky factor entries \( g_{i,j}^i \). \( N_{i,j}^i (0, \delta) \) denotes a Gaussian noise with mean 0 and standard deviation \( \delta \), and eq. (54) ensures that all entries in the main diagonal of the matrices \( G_i (\tau) \) are positive numbers and so \( M_i (\tau) \) are definite positive distance matrices. Notice that the elements of the \( i \)-th Mahalanobis matrix are updated proportionally to the distance \( D_i^+ \) between the \( i \)-th a priori known class and the \( i \)-th cluster found. In this way, only the matrices of those clusters which are not satisfactory for the classification purpose are modified.

6. Return to step 2.

The overall structure of the algorithm is depicted in Figure 4. The closed external loop iterates until an acceptable clustering of the training data is found, i.e., until the obtained partition \( \Gamma (\tau) \) is “close” to the a priori known partition \( \Gamma^\tau \), where “close” means that the distance \( D(\Gamma^\tau, \Gamma (\tau)) \) is smaller than a defined threshold \( \varepsilon \).

![Figure 4. Scheme for the training of the supervised evolutionary clustering classifier](image)

### 3.1.1 The evolutionary FCM classifier

The evolutionary FCM classifier is based on the application of the FCM algorithm within the supervised evolutionary scheme described in the previous Section (point 2 of the operative procedure). At each iteration, \( \tau \), of the evolutionary algorithm, the FCM algorithm (Section 2.3) is performed to find the partition of the \( N \) available labeled training data, using the proposed metrics \( M_i^+, i=1,\ldots,c \), to compute the distances \( s_{ik} = s_{ij}(\tilde{x}_k, \tilde{v}_i^+) \) between the pattern \( \tilde{x}_k \) and the optimal cluster center \( \tilde{v}_i^+ \). At convergence, the supervised evolutionary clustering algorithm provides the \( c \) optimal metrics \( M_i^+ \) with respect to the classification task, while the FCM algorithm provides the cluster centers \( \tilde{v}_i^+ \) and the membership values \( \mu_{ik}^+ \) of the training pattern \( \tilde{x}_k, k=1,\ldots,N \), to the cluster \( i=1,\ldots,c \).

When fed with a new pattern \( \tilde{x} \), the classification algorithm provides the values of the membership functions \( \mu_{i}^+(\tilde{x}), i=1,\ldots,c \), to the fuzzy clusters using:
Given the ordered correspondence between classes and clusters, the fuzzy membership information may be further used for the crisp assignment of the patterns to the various classes by setting a classification threshold $\varepsilon_c$: if the largest value of membership of the generic pattern $\tilde{x}$ to any of the $c$ clusters is greater than $\varepsilon_c$ then the pattern is assigned to the class corresponding to the cluster of largest membership; otherwise, if none of the membership of $\mu_r^*(\tilde{x})$ is larger than $\varepsilon_c$, the pattern remains ambiguous, i.e. it is not assigned to any of the $c$ classes. In this sense, the classification threshold $\varepsilon_c$ regulates the confidence in the crisp assignments to classes, along the views illustrated in Section 2.1.

\begin{align}
\mu_r^*(\tilde{x}) = \frac{1}{\sum_{j=1}^{c} \left( \frac{1}{d^2(\tilde{x}, \tilde{v}_j^*)} \right)^{1/\eta_r}}
\end{align}

(55)

3.1.2 The evolutionary possibilistic classifier

This algorithm differs from the previous one only in that the possibilistic clustering algorithm (Section 2.5) is used instead of the FCM to perform the partition of the training data at each iteration of the supervised evolutionary algorithm (point 2 of the operative procedure reported in Section 3.1).

The $c$ optimal metrics $M_r^*$ and cluster centers $\tilde{v}_i^*$, $i=1,...,c$, obtained at convergence of the evolutionary algorithm, are used to compute the membership functions $\mu_r^*(\tilde{x})$, $i=1,...,c$, of a new pattern $\tilde{x}$ to the possibilistic clusters accordingly to:

\begin{align}
\mu_r^*(\tilde{x}) = \frac{1}{1 + \left( \frac{d^2(\tilde{x}, \tilde{v}_i^*)}{\eta_r} \right)^{1/\eta_r}}
\end{align}

(56)

These values give the degree of compatibility or “typicality” of $\tilde{x}$ to the $c$ clusters. In practice, three situations may arise (Figure 6):

- $\tilde{x}$ belongs only to a cluster with a membership value greater than the threshold $\varepsilon_c$: in this case, it is assigned to the corresponding class.
• \( \vec{x} \) does not belong to any cluster with enough membership, i.e. all the membership values \( \mu'_i(\vec{x}) \) are below a given threshold \( \varepsilon_c \) (degree of confidence): this means that \( \vec{x} \) is an atypical pattern with respect to the training patterns.

• at least two membership values are above the threshold \( \varepsilon_c \): \( \vec{x} \) is thus ambiguous. In this case, the ambiguity must be regarded as “equal evidence”, i.e. the pattern is typical of more than one class and thus cannot be assigned to a class with enough confidence. This situation occurs if \( \vec{x} \) is at the boundary between two classes.

Figure 6: Classification of pattern \( \vec{x} \) with the evolutionary possibilistic algorithm

4. APPLICATIONS TO TRANSIENT CLASSIFICATION IN NUCLEAR POWER PLANTS

In this Section, the evolutionary FCM and possibilistic classifiers are applied to problems of transient classification in nuclear power plant components and systems.

Transient classification is an essential step for the operation, control and accident management of complex systems like nuclear power plants. The basis for the identification of a change in the functioning of a system is that different system faults and anomalies lead to different patterns of evolution of the process features.

A fundamental issue for any diagnostic systems is to recognize transients in the first instants of their evolution so as to enable the operator to schedule the proper corrective and mitigation actions to protect from the adverse consequences of the malfunctioning [21]. Other two important requirements for the practical implementation of model-based fault diagnostic systems in Nuclear Power Plants (NPPs) regard the possibility of defining and controlling the boundaries of their utilization and their capability to diagnose a fault independently from the plant operational state before its occurrence [21,25].

The case studies here proposed regard the classification of simulated nuclear transients in a U-tube steam generator of a Pressurized Water Reactor (Section 4.1) and in a feedwater system of a Boiling Water Reactor (Section 4.2).
4.1 Case study 1: classification of transients in a U-tube steam generator

Consider the problem of classifying transients occurring in the steam generator (SG) of a Pressurized Water Reactor (PWR). The steam generator is a component of the Pressurized Light and Heavy Water Reactors and operates as an interface between the nuclear heat generating part and the conventional power conversion part of the plant. Among the existing configurations, we consider here the well-known standard recirculation U-tubes type (Figure 7). A description of this component can be found in [26] whereas a detailed model of its physical functioning is given in [27].

![Steam generator scheme](image)

Figure 7. Steam generator scheme

In our analysis, we have assumed that the pressurizer imposes a constant primary system pressure (P1) of 154.4 bar. Among the possible transient-initiating perturbations (hereafter called forcing functions) the ones here considered are: the feed water mass flow (G_{FW}) and the feed water temperature (T_{FW}) on the secondary side and the inlet water mass flow (G_{1E}) and the inlet water temperature (T_{1E}) on the primary side. These four forcing functions may vary as a consequence of an operator action or because of plant’s anomalies or faults. The transient responses of the SG were obtained with the code UTSG provided by the NEA data bank of Paris. The code has been developed by the "Gesellschaft für Reaktorsicherheit (GRS)" in Garching bei München, and it properly accounts for the feedback due to the heat removal system. It is part of the code ALMOD-2, which simulates the non-linear behavior of a PWR. UTSG computes the solution of the system of equations in two
steps [28]: (1) at first, the time derivatives in the base equations are set to 0 and the initial values are calculated by solving the resulting system of algebraic equations with recursive techniques; (2) then the stationary part of the equations is eliminated and the solution is obtained by an explicit integration procedure [28]. The UTSG code was used to generate 529 transients, each one 50s long, obtained by randomly varying the kind and intensity of the forcing function. More specifically, the generic forcing function $F(t)$ is assumed to vary in 8s according to the following sigmoidal shape:

$$F(t) = F_o + \frac{F_{\text{fin}} - F_o}{1 + e^{-m_{\text{trans}}(t-t_{\text{trans}})}}$$

(57)

where $F_o$ is its initial value at time $t = 0$s, $F_{\text{fin}}$ is its final value for $t = 8$s, $t_{\text{trans}}$ is the delay time and $m_{\text{trans}}$ is the characteristic frequency of the transient, here set to 5. Table 1 reports the intervals of variation of the forcing functions considered in this work for the generation of the transients of interest, as derived from engineering considerations.

Table 1. Forcing functions: reference values and ranges of variability of $F_{\text{fin}}$

<table>
<thead>
<tr>
<th>Forcing Function</th>
<th>Class</th>
<th>Reference Value</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_{\text{FW}}$</td>
<td>1</td>
<td>467.3</td>
<td>420</td>
<td>510</td>
</tr>
<tr>
<td>$T_{\text{FW}}$</td>
<td>2</td>
<td>212</td>
<td>162</td>
<td>262</td>
</tr>
<tr>
<td>$G_{\text{1E}}$</td>
<td>3</td>
<td>5259</td>
<td>4759</td>
<td>6759</td>
</tr>
<tr>
<td>$T_{\text{1E}}$</td>
<td>4</td>
<td>312.8</td>
<td>27</td>
<td>330</td>
</tr>
</tbody>
</table>

(used only in the test phase)

Regarding the monitored signals, they have been selected taking into account their physical measurability. Out of the 15 signals provided by the UTSG code, we considered the following 8: primary outlet water temperature and mass flow ($T_{1A}$ and $G_{1A}$); secondary inlet water temperature and mass flow ($T_{2E}$ and $G_{2E}$); secondary system pressure ($P_2$); secondary water level ($Z_{WL}$); total steam mass flow ($G_{ST}$); generator power ($Q_{GN}$). The generic measurement vector is composed of the values of the 8 measurable signals taken at the same instant. Along the 50s-long transients, we have considered 10 sampling instants, taken every 4s, from 7s to 43s. For improved classification performance, the dimensionality of the measurements vector has been reduced to four features by means of a nonlinear principal component analysis performed by means of an autoassociative artificial neural network [29]. Hence, the classification algorithm aims at establishing, through an appropriate clustering procedure, a mapping between the four-dimensional input space of the transient features and the one dimensional discrete output space of the forcing functions responsible of the transients.

In this analysis, 2100 patterns taken from transients caused by the forcing functions $G_{\text{FW}}, T_{\text{FW}}$ and $G_{\text{1E}}$ (classes 1, 2 and 3) have been used for building the classification systems while 700 patterns of these three forcing function and 192 patterns taken from transients caused by forcing function $T_{\text{1E}}$ (class 4) have been left for testing their performance.

4.1.1 Results achieved by the evolutionary FCM classifier

As explained in Section 3.1.1 the membership values found by the evolutionary FCM classifier have to be converted in crisp class assignments by setting a classification threshold $\varepsilon_c$. In particular, if the largest value of membership of the generic test pattern $\vec{x}$ to any of the $c$ cluster is greater than $\varepsilon_c$ then the pattern is assigned to the class corresponding to the cluster of largest membership; otherwise, if none of the memberships of $\vec{x}$ is larger the pattern remain ambiguous, i.e. it is not assigned to any of the $c$ classes.
Figure 8 shows the performance of the classifier on the 525 test patterns of classes 1, 2 and 3. For example, with a classification threshold $\varepsilon_c=0.85$, the algorithm correctly classifies 97.86% of the test patterns, misclassifies 0.20% and does not assign 1.94%. As expected, decreasing the confidence threshold required for assigning a pattern to a class, both the numbers of correct and incorrect assignments increase and correspondingly the number of pattern not assigned decreases.

The performance of the evolutionary classifier has then been tested with respect to patterns from unforeseen plant conditions. To this aim the 192 patterns of class 4 obtained by varying the intensity of the inlet water temperature $T_{1E}$ have been used. Notice that the transients of this class have been used only in the test phase and not as training patterns for the construction of the evolutionary FCM classifier. Figure 9 shows the performance of the classification algorithm on the 192 patterns of the new class. The classification of a pattern to one of the first three classes is an error, whereas its classification as ‘unknown’ is correct. In this view, the classification results obtained by the evolutionary FCM classifier are not satisfactory: for example, with a classification threshold $\varepsilon_c=0.85$ 59.37% of the patterns are still assigned to a class.
4.1.2 Results achieved by the evolutionary possibilistic classifier

The evolutionary possibilistic classifier has been trained and tested with the same datasets used in the previous Section for the evolutionary FCM classifier. Figure 10 shows a significant worsening of the classification performance obtained by the evolutionary possibilistic classifier compared with the performance of the evolutionary FCM classifier on the test patterns of classes 1, 2 and 3. Considering, for example, a classification threshold $\varepsilon_c=0.85$, the fraction of patterns correctly classified is 79.13% (98.03% with the evolutionary FCM classifier). This decreasing in the fraction of correctly classified patterns comes with an increasing of the patterns not assigned to any class (20.87% for the possibilistic classifier, 1.98 for the FCM classifier), while the fraction of patterns misclassified is 0.00% (0.20% with the FCM classifier). Thus the possibilistic classifier is more “cautious” in the assignment but it makes slightly less errors in the classification than the FCM classifier. In particular, accordingly to the classification scheme of Figure 6, 37.76% of the patterns that are not assigned to any class result atypical (all the membership values are below the classification threshold) and 62.24% ambiguous (more than one membership value is above the classification threshold). This high fraction of ambiguous patterns means that the possibilistic clusters are overlapped, i.e. there is a relatively wide zone of the feature space formed by points that belong to more than one cluster with high membership.

The performance of the evolutionary possibilistic classifier has then been tested with respect to the patterns of class 4. Figure 11 compares the classification results of the possibilistic and FCM classifiers. Notice that on these patterns, the performance of the possibilistic classifier is more satisfactory than the performance of the FCM classifier. With a classification threshold $\varepsilon_c=0.85$, the possibilistic classifier correctly does not assign 90.62% of the patterns of class 4 (40.63% with the FCM classifier). Moreover, as expected, 95.98% of the test patterns of class 4 not assigned are considered by the evolutionary possibilistic classifier as atypical and only 4.02% ambiguous.
4.1.3 Comparison of the two classification models

Tables 2 and 3 compare the classification results obtained by the two classifiers on test patterns of the training classes 1, 2 and 3 and of the unknown test class 4, using a classification threshold of 0.85. The performance of the evolutionary FCM classifier is more satisfactory in terms of the fraction of test patterns correctly classified among those of the classes considered during the training phase but it tends to assign the patterns of the unforeseen class to a wrong class.

Depending on the application, this may be a critical issue as it is important that the classification algorithm be capable of identifying as unknown those transient conditions that have not been foreseen when building and tuning the diagnostic system, so as to avoid taking wrong protection and emergency actions as a consequence of an incorrect diagnosis [19], [25]. In practice, an incorrect inference by the diagnostic system during an emergency condition risks to confuse and mislead the operator, with the potential of producing catastrophic consequences. Hence, the importance of classifying as unknown the transients that are not within the utilization domain of the classification algorithm. For this reason, it has been decided to use for the nuclear diagnostic problem tackled in the next Section the evolutionary possibilistic classifier, in spite of the fact that it is less performing on patterns of classes that have been considered in the training phase.
Table 2. Classification results obtained by the evolutionary FCM classifier and by the evolutionary possibilistic classifier on the test patterns of classes 1, 2, and 3 (classification threshold of 0.85%)

<table>
<thead>
<tr>
<th>Assigned</th>
<th>Evolutionary FCM classifier</th>
<th>Right assigned</th>
<th>Evolutionary FCM classifier</th>
<th>99.80%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Evolutionary possibilistic classifier</td>
<td>100.00%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>98.06%</td>
<td></td>
<td>Evolutionary possibilistic classifier</td>
<td>0.20%</td>
</tr>
<tr>
<td></td>
<td>79.13%</td>
<td>wrong assigned</td>
<td>Evolutionary possibilistic classifier</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>1.94%</td>
<td>atypical</td>
<td>Evolutionary possibilistic classifier</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>20.87%</td>
<td>ambiguous</td>
<td>Evolutionary possibilistic classifier</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 3. Classification results obtained by the evolutionary FCM classifier and by the evolutionary possibilistic classifier on the 175 patterns of the forcing function $T_{1E}$ not considered in the training phase (classification threshold of 0.85%)

<table>
<thead>
<tr>
<th>Assigned</th>
<th>Evolutionary FCM classifier</th>
<th>Right assigned</th>
<th>Evolutionary FCM classifier</th>
<th>0.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Evolutionary possibilistic classifier</td>
<td>0.00%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>59.37%</td>
<td></td>
<td>Evolutionary possibilistic classifier</td>
<td>100.00%</td>
</tr>
<tr>
<td></td>
<td>9.38%</td>
<td>wrong assigned</td>
<td>Evolutionary possibilistic classifier</td>
<td>100.00%</td>
</tr>
<tr>
<td></td>
<td>40.63%</td>
<td>atypical</td>
<td>Evolutionary possibilistic classifier</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>90.62%</td>
<td>ambiguous</td>
<td>Evolutionary possibilistic classifier</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>4.02%</td>
<td></td>
<td>Evolutionary possibilistic classifier</td>
<td>4.02%</td>
</tr>
</tbody>
</table>
4.2 Case study 2: classification of transients in the feedwater system of a Boiling Water Reactor

In this Section, the evolutionary possibilistic classifier is applied to the classification of transients in the feedwater system of a boiling water reactor.

The diagnostic problem regards the early classification of a predefined set of faults in a BWR. The corresponding transients have been simulated by the HAMBO simulator of the Forsmark 3 BWR plant in Sweden [30].

![Figure 12. Sketch of the feedwater system [30]](image)

The considered faults occur in the section of the feedwater system 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. Figure 12 shows a sketch of the system. 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 (see references [31], [32] for their description). Here, the diagnosis considers three power operation levels, i.e. 50%, 80% and 108% of full power.

The faults may be divided into three categories:
1. F1-F5, F7 regard line 1 of the feedwater system.
2. F11-F15, F17 regard line 2 of the feedwater system.
3. F6, F8, F9, F10, F16, F18 regard both lines.

Transient data were made available for each of the fault types, with varying degrees of leakage and valve closures. All transients start after 60 seconds of steady state operation.

Among the 363 measured signals, only the 5 reported in Table 4 have been chosen for the transient classification using the feature selection algorithm proposed in [33]. Figure 13 shows their behavior for the 6 faults at 80% of full power.
### Table 4: Input signals of the classification model

<table>
<thead>
<tr>
<th>Feature number as reported in [33]</th>
<th>Signal name</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>77</td>
<td>Drain temperature before VB3</td>
<td>°C</td>
</tr>
<tr>
<td>160</td>
<td>Water level in tank TD1</td>
<td>m</td>
</tr>
<tr>
<td>195</td>
<td>Feedwater temperature after preheater EA2 in line 1</td>
<td>°C</td>
</tr>
<tr>
<td>241</td>
<td>Feedwater temperature after preheater EB2 in line 2</td>
<td>°C</td>
</tr>
<tr>
<td>320</td>
<td>Position level of the control valve for preheater EA1 in line 1</td>
<td>%</td>
</tr>
</tbody>
</table>

**Figure 13. Behaviour of the features at 80% of full power**

**4.2.1 Filtering out unknown transients**

In this analysis, the patterns used for building the classification system have been taken from the six faults F1, F2, F3, F4, F5 and F7 that regard line 1 of the feedwater system. For each type of fault, the simulated transients with the plant at 80% of full power have been considered, taking patterns every 6 seconds from \( t=80 \)s to \( t=200 \)s.

After the training of the possibilistic classifier, its performance has been tested using patterns taken every second from \( t=0 \)s to \( t=300 \)s from both the training transients and from an unknown transient caused by F13.

Figure 14 shows the obtained transient classification as time progresses. Considering a classification threshold \( \phi_c = 0.7 \), the results are quite satisfactory, even though at the beginning of the transient the possibilistic classifier...
assigns the steady state patterns to the class of fault F2 albeit with low membership. This is explained by the fact that for transients of class F2 there are no significant effects on the relevant signals of Table 4 (Figure 13) so that understandably the steady state may be confused with a fault of class 2.

![Time profiles of the pattern assignment to the different classes: (+) membership to class F1, (o) membership to class F2, (*) membership to class F3, (.) membership to class F4, (x) membership to class F5 and (◊) membership to class F7. The horizontal solid line represents the degree of confidence $\epsilon_c$ here set to 0.7.](image)

Table 5 reports the time necessary for the possibilistic classifier to assign the transients to the right class with a membership value greater than $\epsilon_c = 0.7$. Notice that class F3 is recognized only 6 seconds after the beginning of the transient at $t=60s$ while for the recognition of class F7 it is necessary to wait 34 seconds. This is due to the fact that the two most sensible signals for F7, 320 and 195, start departing from their steady state at $t=82s$ and $t=78s$, respectively, reaching significant variations after 30s from the beginning of the transient at $t = 60s$ (Figure 13).

**Table 5:** time at which the transient is assigned to the right class (the transient begins at $t=60s$)

<table>
<thead>
<tr>
<th>Class of transient</th>
<th>Time (s) at which there is the first assignment with membership value $&gt; \epsilon_c = 0.7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>69</td>
</tr>
<tr>
<td>2</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>66</td>
</tr>
<tr>
<td>4</td>
<td>87</td>
</tr>
<tr>
<td>5</td>
<td>68</td>
</tr>
<tr>
<td>7</td>
<td>94</td>
</tr>
</tbody>
</table>
Also note that, the possibilistic classifier is able to assign to the right class the foreseen transients at times well beyond the temporal domain of training of 200s, due to the increased significance of the signals as the transients continue evolving away from their initial steady state.

Finally, the algorithm is very efficient in filtering out the patterns of the unknown fault F13 as atypical, by assigning them very low membership values (Figure 14, bottom).

4.2.2 Classification of transients at different power levels

In this Section, the capability of the classifier to identify faults that initiate from different plant operational conditions is investigated. This is a very important characteristic of a diagnostic system, because it is not feasible to train the classifier with transients that initiate from all the possible plant conditions. In this respect, the possibilistic classifier is trained using patterns taken from classes F1, F2, F3, F4, F5 and F7 at 50 % and 108 % power whereas in the test phase also patterns taken at 80 % power are considered.

From each of the 12 training transients considered (6 transients for each of the 2 power levels), patterns taken every 6 seconds from 80s to 200s have been used, for a total of 252 patterns.

The performance of the classifier has been tested using the training transients (belonging to the foreseen classes of faults at 50 % and 108 %) and the new transients (belonging to the same classes of faults but at 80 % power), taking a pattern every second from 0s to 300s.

The behavior of the memberships with respect to all the classes at the three power levels are shown in Figure 15, Figure 16 and Figure 17.

Figure 15: Time profiles of the pattern assignment to the different classes for power level 50%: (+) membership to class F1, (o) membership to class F2, (⋆) membership to class F3, (.) membership to class F4, (x) membership to class F5 and (◊) membership to class F7. The horizontal solid line represents the degree of confidence $\varepsilon$ here set to 0.85
Figure 16. Time profiles of the pattern assignment to the different classes for power level 80%: (+) membership to class F1, (o) membership to class F2, (*) membership to class F3, (.) membership to class F4, (x) membership to class F5 and (◊) membership to class F7. The horizontal solid line represents the degree of confidence $c_e$ here set to 0.85.

Figure 17. Time profiles of the pattern assignment to the different classes for power level 108%: (+) membership to class F1, (o) membership to class F2, (*) membership to class F3, (.) membership to class F4, (x) membership to class F5 and (◊) membership to class F7. Upper solid line, $c_e = 0.85$. 

38
Considering a classification threshold $\varepsilon_c = 0.85$, the performance of the classification at the new power level 80 \% is very satisfactory for the first five classes of faults whereas the transients caused by the fault F7 at 80 \% is filtered as atypical at all times. This happens even for the training patterns of class F7 taken at 50 \% of full power.

To further investigate this situation, a sensitivity analysis has been performed, based on the technique reported in [9]. Figure 18 shows the disposition of the patterns in the subspace formed by two of the three signals identified as most important (signal 320 and signal 195).

![Figure 18: Patterns and cluster centers in the subspace formed by features 320 and 195](image)

In this case, the evolutionary algorithm cannot find an optimal metric $M^*_7$ that results in a cluster that contains exclusively all the patterns of class F7 without containing those patterns of the other classes which are close to the patterns of class F7 at 50\% power level.

The reason for this is that two patterns having nearly the same distance from the center of cluster 7 (+), for example pattern A of class F7 and pattern B of class F5 in Figure 18, have nearly the same membership values to class F7, measuring their compatibility with eq.(56). In this situation, it is impossible to have possibilistic clusters with sharp borders and the target of the evolutionary algorithm of minimizing the distance $D(\Gamma^*_7, \Gamma^*_7)$, between the true known physical class memberships $\Gamma^*_7$, and the possibilistic cluster memberships $\Gamma^*_7$, is better satisfied by a small cluster centred on the patterns of class F7 at 108 \% than by a big cluster that contains all patterns of class F7 but also patterns of other classes.

On this premise, two cases may result from the diagnostic point of view:

1. The performance of the possibilistic classifier is regarded suitable, accepting that patterns of class 7 at 50 \% and 80 \% power levels are classified as atypical. In this case, the classifier does not give false alarms for this class of transients.
2. A more effective identification of the patterns of class F7 is required. In this case, it is possible to improve the correct classification of the transients of class F7, by increasing the dimension of the cluster by increasing the value of its characteristic parameter $\eta_7$ (Section 2.5.2). Figure 19 shows the fraction of patterns of class F7 correctly classified as a function of the value of $\eta_7$. 

39
At the nominal value of $\eta_7 = 0.05$ chosen in the present application, the fraction of correct classifications of class F7 patterns is only 0.2458, most of the patterns being classified as atypical (0.7347), whereas the mean fraction of correct classification of patterns of the other classes is 0.8722. For a value $\eta_7$ of 2.5, the fraction of correct classifications of patterns of class F7 reaches 0.8431, with that of the other classes remaining basically the same. When $\eta_7$ increases to 5.0, the fraction of correct classifications of patterns of class F7 increases but the cluster ends up substantially overlapping with the other clusters, resulting in many transients being assigned ambiguous because belonging to multiple classes, thus degrading the overall classification performance.

Figure 19: Fraction of patterns correctly assigned, wrongly assigned, classified as ambiguous and as atypical, as a function of the value of $\eta_7$.

5. CONCLUSIONS

This report summarizes the theory underlying clustering and discusses some potentials for application in transient diagnostics. The general aim of clustering techniques is to partition a collection of data (patterns) into subgroups such that the patterns contained into each cluster (subgroup) have a certain degree of closeness or similarity among themselves.

Firstly, the unsupervised clustering techniques have been discussed. These techniques work on patterns which are ‘unlabelled’, i.e. whose real, physical classes are not a priori known. In this case, clustering indeed aims at finding a partition such that the patterns of each cluster are as similar as possible to each other and, at the same time, as different as possible from those of the other clusters.

Three unsupervised clustering algorithms have been illustrated in this report: the Hard C-Means, the Fuzzy C-Means and the possibilistic clustering algorithm. These techniques are characterized by the attempt to iteratively minimize an objective function that attains its minimum for a partition which maximally separates the naturally-occurring clusters.

The Hard C-Means clustering algorithm constrains each pattern to belong to one cluster only. This ‘all or none’ membership restriction is not a realistic one since in real world problems patterns can show characteristics common to several clusters. In these cases, it comes more natural to assign to each pattern a set of membership values, one for each cluster, thus shifting towards fuzzy clustering techniques.
The Fuzzy C-Means algorithm is the best known and most widely used fuzzy clustering method. The algorithm is based on the ‘probabilistic constraint’ that forces to 1 the sum of the memberships of a given pattern to the different clusters. As a consequence, the membership of a pattern to a cluster depends on where the pattern is located with respect to not only that cluster but also the others. Hence, in the framework of fuzzy clustering, the membership functions take the meaning of degrees of sharing, i.e. they measure how much a pattern belongs to a cluster relatively to the others. A drawback of this approach is that the constrained membership cannot distinguish between ‘equal evidence’ and ‘ignorance’.

To overcome the above limitations, the clustering problem can be recast into the framework of possibility theory. In the probabilistic clustering algorithm considered in this work, the only constraint that the objective function must satisfy is that the membership values do represent degrees of compatibility, or possibility, i.e. they must lie in [0,1]. Compatibility (typicality) is measured with respect to the cluster prototypical members. In this view, the memberships of representative patterns are high, while unrepresentative (atypical) points bear low membership to all clusters.

The quality of the clustering obtained by applying an unsupervised algorithm depends on the choice of the number of clusters in which to partition the patterns. The optimal number of clusters in which the patterns can be divided, if not a priory known, can be found as the number of clusters that permits to achieve the most satisfactory clustering. To guide the optimization, a validity measure of the clustering, which is based on the separation among clusters and cohesion within clusters, has been discussed.

The possibility of using clustering techniques for pattern classification problems has been addressed. When a partition of labelled patterns in different classes is a priori available, the clustering is called ‘supervised’. In these cases, the goal of the clustering is to identify a partition into clusters as close as possible to the real physical classes in the feature space.

A supervised evolutionary algorithm has been propounded in this work. The algorithm attempts to find one Mahalanobis metric for each cluster, with the goal of achieving optimal classification of an available set of labeled patterns. The metrics found are used within a clustering scheme to classify the patterns. Here, the two clustering techniques considered have been the Fuzzy C Means and the possibilistic algorithms.

Two examples of application of the proposed approaches to pattern classification have been given with reference to the classification of simulated transients in the steam generator of a Pressurized Water Reactor and in the feedwater system of a Boiling Water Reactor. From a diagnostic point of view, the most satisfactory performance is achieved by the possibilistic classifier which correctly classifies the foreseen plant transients, while it filters out the patterns of an unknown fault type by assigning them very low membership values to all classes.

REFERENCES


Zio, E.; Baraldi, P. and Mercurio, D. *Identification of transients in nuclear systems by a supervised evolutionary possibilistic clustering approach*. Accepted for the publication on the Proceedings of the 7th International FLINS Conference on Applied Artificial Intelligence, 29-31 August 2006.


[29] Marseguerra, M., Zio, E., Marcucci, F. Continuous Monitoring and Calibration of UTSG Process Sensors by Autoassociative Artificial Neural Network, Accepted for publication on Nuclear Technology, 2006


PART III
An Extended Classifiability Index for Feature Selection in Nuclear Transients

Enrico Zio, Piero Baraldi and Davide Roverso

An extended classifiability index for feature selection in nuclear transients

E. Zio a,*, P. Baraldi a, D. Roverso b

a Department of Nuclear Engineering, Polytechnic of Milan, Via Ponzio 34/3, 20133 Milano, Italy
b Institutt for Energiteknikk, OECD Halden Reactor Project P.O. Box 173, N-1751 Halden, Norway

Received 18 May 2005; accepted 16 June 2005
Available online 15 August 2005

Abstract

A preliminary step in transient classification for diagnostic purposes is the identification of those measured plant parameters (features), which are most sensitive to the faults and malfunctions and thus can be used most effectively for their classification. This is particularly important for nuclear power plants, where hundreds of parameters are monitored for operation and safety reasons so that expert judgment alone cannot effectively drive the feature selection. Moreover, the sensitivity of the particular transient classification technique to the different plant features must be considered for higher classification efficiency.

In this paper, a feature selection algorithm is proposed based on the extension to the transient case of a classifiability index which can be directly computed from the plant measured data and used to filter out irrelevant or redundant features.

© 2005 Elsevier Ltd. All rights reserved.

1. Introduction

To develop a diagnostic system it is of primary importance to select among the measured parameters, only those carrying relevant information related to the faults...
and malfunctions that are to be detected and identified. In nuclear power plants a large number of plant parameters are measured for operation monitoring and safety purposes so that expert judgement may have difficulties in this so called feature selection task. Moreover, the selection should reflect also the sensitivity to the different parameters (features) of the particular diagnostic algorithm adopted.

In this work, we address the problem of selecting the plant features to be used for transient diagnostics. Classically, feature selection techniques aim at finding the optimal feature subset that achieves the highest accuracy in the data classification task with the minimum number of features. In this view, transient diagnostics differs from traditional classification problems because the transients to be classified are trajectories traced by the evolution of the plant parameters in time, rather than data points in the feature space. For this reason, a specific technique is here developed and applied to a nuclear transient classification problem. The proposed technique is based on the extension to the transient case of an evaluation function recently introduced for performing feature selection in standard data classification tasks (Dong and Kothari, 2003).

The work is organized as follows. In Section 2 a brief overview of the classical feature selection techniques is given. In Section 3 the classifiability evaluation function (Dong and Kothari, 2003) is presented as a basis for the extension to feature selection in transient classification problems. In Section 4 an extended classifiability evaluation function is proposed for application to the feature selection in transient classification problems. In Section 5, the developed technique is applied to a nuclear case. Finally, some conclusions are presented in the last section.

2. Classical techniques for feature selection

From a mathematical point of view, a classifier is a mapping function $\Phi(\cdot)$ which assigns an object $\tilde{x}$ to a given class $i$. If we knew the exact expression of $\Phi(\cdot)$, the question of which features of $\tilde{x}$ to use would not be of interest. In fact adding features does not decrease the accuracy of the classifier, and hence restricting $\Phi(\cdot)$ to a subset of features is never advised. However, as it is often the case in engineering, due to the complexity of the systems under analysis it is not possible to determine the exact analytical expression of the function $\Phi$. Hence, one resorts to empirical classification techniques in which the classifier is built through a process of learning based on a set of classification examples.

In practical scenarios, at least four reasons call for a reduction in the number of features (Na, 1997). First of all, irrelevant, non informative features result in a classifier model which is not robust (Verikas and Bacauskiene, 2002). Second, studies have shown that it is necessary to remove highly correlated features (Verikas and Bacauskiene, 2002). Third, when the model handles many features, a large number of observation data are required to properly span the high-dimensional feature space for accurate multivariable interpolation. Finally, by eliminating unimportant features the cost and time of collecting the data and developing the classifier can be reduced.
In general, feature extraction refers to the task of finding a mapping that reduces the \( n \)-dimensional data being classified onto an \( m \)-dimensional space, where \( m < n \) (Dash and Liu, 1997). All \( n \) measurements are used to obtain the \( m \)-dimensional data. Features selection is a special case of feature extraction whereby \((n - m)\) irrelevant features are discarded.

More specifically, the objective of feature selection is that of finding a subset of the original features of a dataset such that an induction algorithm run on data represented only by these features generates a classifier with the highest possible accuracy (Kohavi and John, 1997). Other requirements can be included in the feature selection objective function, such as the computational requirements for data storage and modeling. Notice that different induction algorithms based on different learning techniques, may require different feature subsets and so an optimal universal feature subset does not exist.

Several methods have been proposed to tackle feature selection. They are classified in two categories: filter methods and wrapper methods (Kohavi and John, 1997).

In filter methods, the feature selector algorithm is independent of the specific learning algorithm used in the classification and it is used to discard irrelevant and/or redundant features a priori of the construction of the classification model. A numerical evaluation function, computed directly from the data, is used to score the alternative feature subsets (Dash and Liu, 1997).

On the contrary, in wrapper methods the feature selector behaves as a wrapper around the specific learning algorithm used to construct the classifier. The feature subsets are compared using as criterion the performance achieved by the classification algorithm itself (Kohavi and John, 1997).

The filter approach is generally computationally more efficient than the wrapper approach because for each feature subset, the computation of an index from the available data is less time consuming than the development of a complete classification model through a learning algorithm. On the other hand, wrapper approaches are more performing since they ensure the selection of the features more suitable for the specific classification algorithm used. Both the filter and wrapper approaches conduct a search for an optimal feature subset in the space of possible features. The state space in which the search is performed is such that each state can be encoded by a binary sequence representing a feature subset. For \( n \) features, there are \( 2^n \) bits in each state, where each bit indicates whether a feature is present (1) or absent (0). The size of the search space is \( 2^n \), so that an exhaustive search is impractical unless \( n \) is very small. To overcome this problem, three different search strategies are commonly adopted: complete, heuristic and random (Kohavi and John, 1997). In the complete approach, the properties of a pre-defined evaluation function are used to prune the feature space to a manageable size, thus avoiding that the complete search is also exhaustive (Narendra and Fukunaga, 1977). Only some evaluation functions give rise to a search that guarantees the optimum feature subset selection without being exhaustive (Chen, 2003).

The heuristic approach does not guarantee that the best feature subset is achieved, but is less time consuming than the complete one and may be employed in combination with any evaluation function (Dong and Kothari, 2003). At present, the most
employed heuristic methods are greedy search strategies such as forward selection or backward elimination, which iteratively add or subtract features based on the score given by the evaluation function.

The random approach is based on randomized search algorithms, primarily genetic algorithms, simulated annealing or tabu search algorithms (Zhang and Sun, 2002).

3. The classifiability evaluation function

In this section we discuss the characteristics of an evaluation function, called classifiability (Dong and Kothari, 2003), whose extension is the basis of the filter method here proposed for the feature selection in transient classification problems. The definition of the classifiability evaluation function is motivated by the fact that an $n$-dimensional classification problem may be visualized in $n + 1$ dimensions using the class label as the $(n + 1)$-th dimension. Fig. 1 shows an example of a classification problem in two-dimensions with the corresponding visualization in three-dimensions. The introduction of the class label defines a surface which is “rough” when patterns of different classes are close (Fig. 1, top) and “smooth” when patterns of the same class are adjacent (Fig. 1, bottom). Naturally, classification is considerably more complicated when the “class label surface” is rough.

Fig. 1. Top: Projection of the feature space on features 1, 2 (left) and visualization of features 1 and 2 using as third dimension the class label (right). Bottom: Projection of the feature space on features 3, 4 (left) and visualization of features 3 and 4 using as third dimension the class label (right).
Consequently, if the smoothness (or roughness) of the class label surface can be quantified, then a natural measure of classifiability is obtained which depends on the feature subspace. This intuitive notion is the basis of the algorithm proposed in Dong and Kothari (2003), theoretically founded on the second order joint conditional density function $f(c_i,c_j|d)$ of going from a pattern of class $c_i$ to a pattern of class $c_j$ within a distance $d$. Here, we only give the basic steps of the operative procedure for computing the classifiability index from a set of $N$ labelled data patterns $(\tilde{x}_i, c_i)$, where $\tilde{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ih})$ is the feature vector represented by a point in the selected feature subspace $S_f$ of dimension $R^h \subseteq R^n$, and $c_i$ is the corresponding class label.

(i) Initialize to 0 two counters $s$ and $o$.
(ii) For $i = 1$ to $N$.
   • add to the counter $s$ the number of points of class $c_i$ inside the hyper-sphere of radius $d$ with center $\tilde{x}_i$;
   • add to the counter $o$ the number of points not of class $c_i$ inside the same hyper-sphere;
(iii) Evaluate the classifiability of the feature subspace $S_f$ by:

$$L(S_f; d) = \frac{s - o}{s + o}. \tag{1}$$

By definition, the classifiability evaluation function $L$ takes values in $[-1,1]$, higher values implying greater classifiability.

Besides the feature subspace $S_f$, the classifiability depends also on the value of the neighborhood distance $d$. Typically, $d$ should be large enough that at least a few points are present within the neighborhood of each point but small enough that classifiability is evaluated locally. In Dong and Kothari (2003), it is empirically suggested that a value of $d$ between 2 and 3 times the Root Mean Square distance (RMS) of each pattern from its nearest neighbor gives good results.

4. Feature selection for transient classification

4.1. The extended classifiability evaluation function

In this work, feature selection is framed within a problem of transient classification. The transients to be classified are represented by the time evolutions of the plant measured parameters which trace multi-dimensional trajectories in the feature space.

To perform the classification of the trajectories, these are transformed into collections of points in the feature space, each one labelled with a corresponding class label, by taking the values of all the features along the trajectories over a fixed time window. The particular distribution in the feature space of the points obtained from the transients leads us to modify the definition of classifiability.
Indeed, since all the transients generally begin from a steady state (in our case the full power operation state), in the first time instants after the transients initiation (which is the period of interest for early diagnostics) the features are very close to their nominal values, independently of the initiating cause. From a geometric point of view, the small zone of the feature space close to the point that represents the steady state contains a great number of “quasi-steady” or “early transient” points representing transients of all the different classes. For example, Fig. 2 shows the disposition of 385 transient points in the space of 2 plant features related to the case study of the successive Section 5. These points are obtained considering the values of the 2 features taken every 10 s from 65 to 125 s in 55 transients of 5 different classes. As expected, at these early times the majority of points are concentrated near the steady state point (0,0.22) whereas only few are located far. Now, by definition, the classifiability value (1) is more influenced by a point with a lot of neighbors inside the sphere of radius \(d\) than by a point with few neighbors inside its sphere. Thus, for the points close to the steady state, having several neighbors of different classes, the counter \(o\) is very large and the value of the classifiability evaluation function is small even if the class surface becomes smooth far from the steady state.

To overcome this problem, inherent in the transient classification problem, the \(N\) points are given the same importance in the evaluation of the classifiability, independently of the number of neighbors. This is done by defining for each point \(\bar{x}_i\), \(i = 1, \ldots, N\) the pointwise classifiability \(L_i(S_f; d)\)

![Fig. 2. Feature space of the signals position level control valve for EA1 and position level control valve for EB1.](image-url)
where \( s_i \) and \( o_i \) are counters of the number of points in the sphere centered at \( \bar{x}_i \) of the same class and of different class of \( \bar{x}_i \), respectively. The transient classifiability function \( L^*(S_f; d) \) is then defined as the average of the pointwise classifiabilities

\[
L^*(S_f; d) = \frac{\sum_{i=1}^{N} L_i^*(S_f; d)}{N}.
\]

By so doing, the points close to the stationary state, with a lot of neighbors of different classes, contribute to the value of \( L^*(S_f; d) \) in the same way as the points far away from the steady state. Hence, for two feature subspaces \( S_1 \) and \( S_2 \) equally “rough” near the steady state region but with \( S_1 \) more “smooth” than \( S_2 \) elsewhere, \( L^*(S_1; d) \) is larger than \( L^*(S_2; d) \). In principle, then, the new classifiability \( L^*(S_f; d) \) is effective in evaluating the capability of a feature subspace of distinguishing transients away from the steady state, after some seconds from the beginning of the transient.

Nevertheless, a crucial problem in plant transient classification remains the early diagnosis of the malfunction. The ‘early’ requirement leads to the need of classifying small deviations from the steady state. The classifiability \( L^*(S_f; d) \) is not effective in this task because the pointwise classifiability \( L_i^*(S_f; d) \) of points close to the steady state is typically very small since it contains many points of different classes. Actually, for a point close to the plant steady state, we are interested in its nearest neighbors. In fact, even with a lot of points inside the sphere of radius \( d \), a situation in which the nearest neighbors are of the same class is preferable, from a classification point of view, to a situation in which the nearest neighbors are of other classes. Obviously, to give importance to the nearest neighbors one could reduce the sphere radius \( d \), but this would cause the points far from the steady state, more dispersed, to have empty spheres. A more effective way to proceed amounts to assigning a weight to the points inside the sphere according to their distance from the center. In this way a weighed pointwise classifiability \( L_i^{**}(S_f; d) \) can be computed, which gives a local information on the roughness of the class surface independently of the number of points inside the sphere.

The new algorithm for evaluating the classifiability function \( L^{**}(S_f; d) \) for a given feature subspace \( S_f \) and distance \( d \) is the following:

(i) For each point \( \bar{x}_i, i = 1, \ldots, N \),
- initialize to 0 the counters \( s_{wi}, o_{wi} \);
- sort the \( N_i \) points inside the hyper-sphere of radius \( d \) with center at point \( \bar{x}_i \) of class \( c_i \) according to their distance from \( \bar{x}_i \);
- assign weight \( w_p = \exp[(p - 1)\ln(0.5)/9] \) to the \( p \)th nearest point, \( p = 1, 2, \ldots, N_i \). The weight function is obviously arbitrary. In our case, it was chosen such that the nearest point has weight \( w_1 = 1 \), the 10-th nearest has weight \( w_{10} = 0.5 \) and the 25-nearest has weight \( w_{25} = 0.15 \). By so doing, only the first 20–25 neighbors give significant contributions;
• for $p = 1$ to $N_i$, add $w_p$ to $sw_i$ if the $p$th nearest point in the hyper-sphere of $\bar{x}_i$ is of class $c_i$; otherwise add $w_p$ to $ow_i$;
• compute the “smoothness” of the surface around $\bar{x}_i$ by the pointwise classifiability

$$L_i^*(S_f; d) = \frac{sw_i - ow_i}{\sum_{j=1}^{N_i} w_j}.$$  

(ii) Compute the weighed transient classifiability of the feature subset $S_f$ by

$$L^*(S_f; d) = \frac{\sum_{i=1}^{N} L_i^*(S_f; d)}{N}.$$  \hspace{1cm} (4)

Finally, in the algorithm proposed by Dong and Kothari (2003), to evaluate the classifiability of a given feature subspace $S_f$ one has to identify, for each of the $N$ points $\bar{x}_i$, the points which are within the neighborhood of radius $d$ of $\bar{x}_i$. This implies computing the $(N - 1)$ distances $\Delta ij$ between $\bar{x}_i$ and all the other points $\bar{x}_j$, $j = 1, 2, \ldots, N, j \neq i$, with an overall computation complexity of $O(N^2)$. Given that during the feature selection process it is necessary to compute the classifiability of several different feature subspaces, a reduction of the computational complexity in the evaluation of the classifiability of the single feature subspace is very important.

To achieve such reduction, we propose the following procedure which avoids computing all $N(N - 1)$ distances $\Delta ij$, $i, j = 1, 2, \ldots, N, i \neq j$. Let us fix arbitrarily a point $\bar{x}_i$ in the $R^b$ subspace and compute the $N$ distances $\Delta ij$ between $\bar{x}_i$ and the $N$ points $\bar{x}_j$, $j = 1, 2, \ldots, N$ expressed in units of $d$. Then, each of the $N$ points is labeled by the first majoring integer of the computed distance (Fig. 3). For example a point with distance $2.3d$ from $\bar{x}_i$ has label 3. By so doing, all the points in the sphere of radius $d$ and center $\bar{x}_i$ of label $l_i$ have integer labels in $[l_i - 1, l_i + 1]$ (Fig. 3). Thus, to find the points that fall inside a sphere of radius $d$ centered at $\bar{x}_i$, it is not necessary to compute the distances from $\bar{x}_i$ of all the other $N - 1$ points, but only those with labels $l_i - 1, l_i, l_i + 1$, thus achieving a significant saving of computation time.

4.2. The search engine

The goal of the feature selection search is to find the feature subspace in which the classifiability evaluation function is larger. Since the features are not uncorrelated, the optimal feature subspace is not necessarily made up by the individual best features (Cover and Campenhout, 1977). Hence, theoretically one has to explore the $2^n$ possible combinations of features. Given that in the cases of our interest the number of features $n$ can typically be very large ($n > 100$), an exhaustive search algorithm is too time consuming and so a “greedy search algorithm” has been adopted. Two of the most famous greedy methods are the sequential forward selection (SFS) method and the sequential backward elimination (SBE) method, both known as “hill climbing” methods (Kohavi and John, 1997). The forward selection refers to a search that
begins with no features and at each step a feature is added to the subspace; on the contrary, the backward elimination refers to a search that begins with the n-dimensional feature space and at each step a feature is removed. At each step, the choice of which feature to add or remove is driven by its effect on the classifiability function so as to climb toward its maximum value. Since in our applications typically the goal is that of reducing from a large number of features to a subspace of less than 15–20, it is computationally more convenient to start from no features and thus the SFS method is embraced in this work. The SFS hill-climbing search is stopped when adding new features does not increase the value of the classifiability function or when the number of features has reached a predefined threshold.

As a remark, note that the hill climbing methods suffer from the so called “nesting effect” (Marcelloni, 2003): if the features added cannot be removed, a local minimum of the evaluation function may be found. To reduce this effect, it is possible to use the so called plus-\( l \)-take-away-\( r \) method (PTA) (Kohavi and John, 1997). In this method, after \( l \) steps of the forward selection, \( r \) steps of the backward elimination are applied so as to allow escape from local minima. However, there is no guarantee of obtaining
the absolute optimum and therefore the crude SFS method is here adopted since the optimality of the search algorithm is not the focus of the work.

Finally, we recall that, as mentioned earlier, a disadvantage of the filter approach is that it totally ignores the effects of the selected feature subspace on the performance of the induction algorithm that will be used for the classification.

5. Case study

The diagnostic problem considered is the early identification of a predefined set of faults in a Boiling Water Reactor (BWR). The corresponding transients have been simulated by the HAMBO simulator of the Forsmark 3 BWR plant in Sweden (Puska and Norrman, 2002). Fig. 4 shows a sketch of the system (Puska and Norrman, 2002). The considered faults occur in the section of the feedwater system where the feedwater is preheated from 169 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 (Roverso, 2004, 2003) that are generally hard to detect for an operator and that produce efficiency losses if undetected. It was decided, in first instance, to limit the diagnosis to only one power operation level, i.e., 50% of full power.

The faults may be divided in three categories:

- F1–F5, F7: regard the line 1 of the feedwater system.
- F11–F15, F17: regard the line 2 of the feedwater system.
- F6, F8, F9, F10, F16, F18: regard both lines.

Fig. 4. Sketch of the feedwater system.
The data of five simulations were made available for each of the 18 fault types, with varying degrees of leakage and valve closures, with step and ramp changes at different leak sizes. All transients start after 60 s of steady state operation. The data relative to $n = 363$ features were recorded with a sampling frequency of 1 Hz.

For the classification purpose of the present work, this number was reduced to 128 by combining the redundant measurements of the same physical quantity and by eliminating those features linearly correlated or carrying basically the same time evolution in all the different faults.

Since the goal is early diagnosis, only the data of the first 65 s after the beginning of the transient (i.e., 66 data points from 60 to 125 s) have been considered. Moreover, given that the ramp changes cause variations of the features later than the step changes, only the three step changes for each fault are considered.

Given that the complexity of the classifiability evaluation function computation is $O(N^2)$, with $N$ the number of data points, the number of data points has been reduced to 7 by taking values every 10 s. Thus, the total number of points employed for the feature selection is 385 (7 points for each of the 54 transients and for the steady state).

Although all at 50% power level, the simulated transients begin from slightly different initial feature values due to different boron dilution during the steady state operation (Puska and Norrman, 2002). Hence, to avoid that the feature selection and classification algorithms be applicable only from a specific steady state of the reactor, the differences between the feature values at the generic sampling time during the transient and their steady state values before the beginning of the transient are used.

5.1. Feature selection results

Fig. 5 shows the evolution of the classifiability functions $L$, $L^\ast$ and $L^{**}$ as features are added within the forward selection scheme here adopted. For example, at the first step the algorithm for the computation of $L^{**}$ evaluates 128 values of $L^{**}$, one for each feature, and finds that a maximum value of $-0.50$ is achieved by a one-dimensional feature subspace that contains parameter 320 (position level control valve for EA1). At the next step, the $L^{**}$ values of the 127 possible two-dimensional feature subspaces that contain feature 320 and another one among the remaining 127 are computed, and the maximum value of $-0.12$ is obtained by the feature subspace $\{320, 321\}$. Starting from this point, the procedure then moves to considering three-dimensional subspace and so on.

Note that the $L$ function starts decreasing beyond the three-dimensional optimal feature subspace, $S_3^{opt}$. This confirms that the classical definition of classifiability cannot identify features salient for transient classification because of the high importance that it gives to the points very close to the steady state (see Section 3). On the contrary, $L^\ast$ identifies an optimal subspace of 12 features while the weighed classifiability $L^{**}$ finds an optimal 16 feature space.

Tables 1 and 2 report the features selected by using the different definitions of classifiability and the selection of 17 features declared relevant by an expert. Note
that 1 out of 3 features in the optimal subspace identified by using $L$ appears also in the expert’s list, 7 out of 12 in the case of $L^*$ and 11 out of 16 for $L^{**}$. Given the latter result, it seems justified to say that there is a fair agreement between the feature selection by the weighed algorithm proposed, based on $L^{**}$, and the expert selection.

Next, we examine the performances of the selected feature subspaces when applied to a classification algorithm. To this aim, the 385 data points used for the feature selection were randomly subdivided into training and test sets consisting of 75% and 25% of the data, respectively. The Fuzzy K-Nearest Neighbor algorithm (K-NN) (Keller et al., 1985), with $k = 5$ was applied to classify the test data on the basis of the location of the labeled training data. The obtained fuzzy partition of the test data $\{\mu_i^t(\tilde{x}_k)\}$, where $0 \leq \mu_i^t(\tilde{x}_k) \leq 1$ is the membership function of data point $\tilde{x}_k$ to class $i$, has been converted in a hard partition assigning each data point to the class with highest membership value. The recognition rate, i.e., the fraction of correct classifications (Marcelloni, 2003), has been employed for the comparison. Table 3 reports the mean and the standard deviation of the recognition rate obtained by repeating 25 times the random subdivision of the available data in train and test sets.

The feature subspace obtained by using the evaluation function $L$ leads to a completely unsatisfactory classification. On the contrary, the fuzzy K-NN classifiers that

![Fig. 5. Evolution of the classifiability functions $L$, $L^*$ and $L^{**}$ with the number of features in the subspace. The line with diamonds refers to $L$ (Eq. 1), the line with circles to $L^*$ and the line with squares to $L^{**}$ (Eq. 4).](image)
use the feature subspaces selected by the expert and by means of the $L^*$ and $L^{**}$ evaluation functions outperform significantly the classifier based on all the 128 available features. Further, the feature subspaces obtained by $L^*$ and $L^{**}$ allow achieving a higher classification accuracy, and with less features, than the feature subspace proposed by the expert. This confirms the advantage of using a feature selection technique to support classification task based on a large number of available features.

The best recognition rate is achieved by the subspace identified using the weighed classifiability evaluation function $L^{**}$. The features thereby selected distinctly regard either only line 1, only line 2 or both lines. Figs. 6–8 show their corresponding time evolutions from 60 to 180 s for three transients caused by a step fault. In each graph, the evaluation of the variation of the feature from its steady state value at the starting of the fault is reported. The identification number of the feature (Table 2) and the fault number are indicated at the top of the columns and on the left of the rows of graphs, respectively.

Fig. 6 refers to faults and features related to line 1. Feature 320 evolves clearly in a characteristic way in case of faults 1, 3, 5, although the evolutions for the latter two are quite similar. Feature 195 is sensitive to faults 2, 7 and somewhat to 4. Feature 100, the pressure before preheater A1, has been selected because in the time windows considered for the diagnosis it behaves as a two value signal and thus, in principle, increases significantly the distances between points in the feature space generated by different faults. However, its sensitivity to the transients considered lasts only few instances, so that it might not be so significant for the diagnostic task. Feature 324

<table>
<thead>
<tr>
<th>Expert</th>
<th>$L$ (Eq. 1)</th>
<th>$L^*$ (Eq. 3)</th>
<th>$L^{**}$ (Eq. 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>319</td>
<td>321</td>
<td>320</td>
</tr>
<tr>
<td>73</td>
<td>45</td>
<td>320</td>
<td>321</td>
</tr>
<tr>
<td>74</td>
<td>223</td>
<td>342</td>
<td>77</td>
</tr>
<tr>
<td>75</td>
<td>357</td>
<td>157</td>
<td>160</td>
</tr>
<tr>
<td>76</td>
<td>157</td>
<td>243</td>
<td>195</td>
</tr>
<tr>
<td>175</td>
<td>325</td>
<td>242</td>
<td>242</td>
</tr>
<tr>
<td>194</td>
<td>324</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>195</td>
<td>243</td>
<td>73</td>
<td>73</td>
</tr>
<tr>
<td>240</td>
<td>195</td>
<td>76</td>
<td>76</td>
</tr>
<tr>
<td>241</td>
<td>76</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>242</td>
<td>242</td>
<td>324</td>
<td></td>
</tr>
<tr>
<td>318</td>
<td>100</td>
<td>325</td>
<td></td>
</tr>
<tr>
<td>319</td>
<td></td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>320</td>
<td></td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>321</td>
<td></td>
<td>241</td>
<td></td>
</tr>
<tr>
<td>324</td>
<td></td>
<td>168</td>
<td></td>
</tr>
</tbody>
</table>

For $L$, $L^*$ and $L^{**}$ the features are ordered in their order of selection and the shaded features are those which appear also in the expert selection.
helps feature 320 in discerning between faults 3 and 5. Globally, the selected features seem capable of distinguishing among the faults concerning line 1.

The features selected with respect to line 2 (Fig. 7) are symmetric of those identified for line 1 except that no features analogous to feature 100 is identified. Overall, the selected features seem capable of distinguishing the faults relevant to this line.

For what concerns the faults related with the part of the feedwater system common to the two lines, the feature selection algorithm identifies 9 features (Fig. 8).
Some features have very similar evolutions and seem redundant. For example, feature 77 is almost identical to 59. If we consider, for example, the two $L^\text{opt}$ subspaces, $S_{\text{opt}}^{13}$ (which contains feature 77 among the 13 features) and $S_{\text{opt}}^{14} = S_{\text{opt}}^{13} \cup \{\text{feature59}\}$, we see that the effect of adding the redundant feature 59 is that of increasing the distance between the points associated to transients generated by faults 8, 9, 18 from points associated to other faults. Thus, the classifiability value increases (Fig. 5) although the redundant feature 59 bears basically the same information of feature 77. The selection of redundant features is a typical drawback of the filter approach to the feature selection task (Kohavi and John, 1997).

From Fig. 8 it is also clear that no signals relevant to faults 6, 10, 16 are selected by the algorithm. These faults have no significant consequence on the plant measured features because the size of the leakages considered is too small. In an attempt to identify features relevant to these faults the feature selection algorithm was applied only to the data of the steady state and of transients generated by these faults. The features selected by the algorithm (here not reported for brevity) are almost steady state and have characteristics similar to noise, so that they are not useful for transient classification. This occurs also for large leakages of type 6, 10, 16, as verified by simulation and confirmed by experts.
6. Conclusions

In this paper, the problem of discerning those features, among the several measured plant parameters, to be used for early transient diagnosis has been tackled. This is a crucial issue to be resolved for the application of advanced monitoring and diagnostic techniques to complex plants, like the nuclear power plants, where hundreds of parameters are measured. Traditionally, feature selection algorithms developed for classical data classification problems have proven not efficient in selecting features salient for transient classification because transients are trajectories traced in the feature space rather than data points.

To overcome this problem, a standard classifiability evaluation function, recently introduced in the literature, has been extended to account for the transient evolution in time. The corresponding proposed feature selection filter approach has been tested on a diagnostic problem regarding the classification of simulated transients in the feedwater system of a Boiling Water Reactor. From an initial number of 128 potentially relevant plant parameters, the feature selection algorithm identifies 16 features which are shown capable of achieving a satisfactory accuracy in a fuzzy K-Nearest Neighbor classification, significantly better than with the features selected by a plant
expert and by the algorithm based on the original, standard feature selection evaluation function.

The filter approach here employed, as any filter approach, attempts to assess the merit of the features directly from data, ignoring the merit of the selected features in the actual transient classification algorithm used. In this respect, work is in progress within our research group in the area of wrapper methods for feature selection.

References


Fig. 8. Time evolution of the variations from their steady state values of the features selected by the $L^{**}$ algorithm with respect to both lines 1 and 2 of the feedwater system.
Paper II

Selecting Features for Nuclear Transients Classification by means of Genetic Algorithms

Enrico Zio, Piero Baraldi and Nicola Pedroni

Accepted for publication on IEEE Transactions on Nuclear Science, 2006
Abstract

The issue of feature selection is particularly critical for the application of monitoring and “on condition” diagnostic techniques to complex plants, like the nuclear power plants, where hundreds of parameters are measured. Indeed, irrelevant and noisy features unnecessarily increase the complexity of the problem and can degrade the diagnostic performance.

In this paper, the problem of choosing among the several measured plant parameters those to be used for efficient, early transient diagnosis is tackled by means of genetic algorithms. Three different schemes for simultaneously performing the feature selection and the training of an empirical diagnostic classifier are presented. The first approach employs a single objective genetic algorithm to search the vector of features optimal with respect to the classification performance of a Fuzzy K-Nearest Neighbors classifier. With reference to the same classifier, the second and third approaches embrace a multi-objective perspective to find feature sets that achieve high classification performances with low numbers of features. In all cases, validation of the performance of the classifiers based on the optimal feature subsets identified by the genetic algorithm is successively carried out with respect to transients never used during the feature selection phase. The effectiveness of the proposed approaches is tested on a diagnostic problem regarding the classification of simulated transients in the feedwater system of a Boiling Water Reactor.

1. INTRODUCTION

The approach to the development of model-based techniques for the diagnosis of faults in nuclear power plants is typically performed through a sequence of steps which include the selection of the parameters (hereafter also called features) relevant to the diagnosis, the validation and reconciliation (when needed) of the measured signals, the reduction of the dimensionality of the feature space by means of a procedure of features extraction, the detection of anomalies in the process and the subsequent classification of the type of fault occurred.

The present work focuses on the first step of selection of features relevant to the classification of nuclear transients. In general, the selection of features can have a considerable impact on the effectiveness of the resulting classification algorithm [1, 2, 3]. Indeed, it has been shown experimentally that irrelevant and noisy features unnecessarily increase the complexity of the diagnostic problem and can degrade modeling performance [4, 5, 6, 7]. As a result, feature selection methods have become important techniques for automated pattern recognition [1, 2, 8, 9], exploratory data analysis [10] and data mining [11]. Further, for the particular case of nuclear power plants the problem is also critical with respect to the very large number of parameters monitored for operation and safety reasons.

In this paper, the problem of searching for an optimal feature subset is performed by a wrapper approach [12] based on genetic algorithm (GA) optimization. Each feature subset is encoded by a binary sequence: for $n$ features, there are $n$ bits in the sequence, where each bit indicates whether a feature is present (1) or absent.
(0) in the subset. Thus, the size of the search space is \(2^n\), so that an exhaustive search is impracticable unless \(n\) is very small [13]. On the other hand, a heuristic approach, based on greedy search strategies such as forward selection or backward elimination, which iteratively add or subtract features, can fail to select features which do poorly alone but offer valuable information together [4]. For these reasons, we embrace a genetic algorithm approach which evolves a population of solutions for efficiently searching the feature space.

The genetic algorithm search is combined with a Fuzzy K-Nearest Neighbors (FKNN) classifier, within a wrapper framework for the feature selection task. The feature subsets selected by the genetic algorithm are compared using as objective functions the recognition rate achieved by the Fuzzy K-Nearest Neighbors classifier [14] and the number of features included in the input patterns.

Three GA-searching schemes are explored: in the first, a two-step, hierarchical, single-objective GA progressively screens unimportant features through successive searches; in the second, two objectives, i.e. maximization of the classification accuracy and minimization of the selected features, are aggregated in a weighted sum which is then subject to the optimization by a single-objective GA in order to drive the search towards effective and parsimonious solutions; finally, a Pareto-based Multi-Objective GA (MOGA) is performed on the two previous objectives kept separate to evolve the population towards alternative, equivalent feature sets offering different trade-offs in terms of diagnostic power and complexity.

The proposed feature selection schemes are verified through an application to the classification of simulated transients in the feedwater system of a Boiling Water Reactor [15, 16]. Two power levels, 50% and 80% of full power, are considered: the transients at 50% of full power are used for the feature selection task, whereas the transients at 80% of full power are purposely left out for validation of the resulting classifier. From an initial number of 123 potentially relevant plant parameters, the feature selection GA-based algorithms identify subsets of 11-20 features. The results are very satisfactory considering that the selected feature subsets are capable of achieving a more satisfactory accuracy with less features than the feature subset selected by other filter techniques [17].

The paper is organized as follows. In Section 2, a brief overview of the classical feature selection techniques is given. In Section 3, a general description of the GA’s as search tools is presented. In Section 4, the way different GA approaches can be applied to the feature selection task is explained. In Section 5, the developed techniques are applied to a nuclear case study. Finally, some conclusions are proposed in the last Section.

2. CLASSICAL TECHNIQUES FOR FEATURE SELECTION

From a mathematical point of view, a diagnostic classifier is a mapping function \(\Phi(\cdot)\) which assigns a vector of measured parameters \(\vec{x}\) to a given fault class \(i\). If the exact expression of \(\Phi(\cdot)\) is known, the question of which features of \(\vec{x}\) to use is not of interest. In fact, adding features cannot decrease the diagnostic accuracy of \(\Phi(\cdot)\) and, hence, restricting \(\vec{x}\) to a subset of features is never advised. However, in the engineering practice it is typically not possible to determine the analytical expression of the function \(\Phi(\cdot)\), due to the complexity of the system under analysis. Hence, one resorts to empirical classification techniques in which the diagnostic classifier is built through a process of learning based on a set of classification examples.

In practical scenarios, at least three reasons call for a reduction in the number of features [5]. First of all, irrelevant, non informative features result in a classifier model which is not robust [4, 5, 6, 7]. Second, when the model handles many features, a large number of observation data is required to properly span the high-dimensional feature space for accurate multivariable interpolation. Finally, by eliminating unimportant sensors the cost and time of collecting the data can be reduced.
In general terms, feature extraction refers to the task of finding a mapping that reduces the $n$-dimensional data being classified onto a $m$-dimensional space, where $m < n$ [18]. All $n$ measurements are used for obtaining the $m$-dimensional data. Feature selection is a special case of feature extraction whereby $(n - m)$ irrelevant features are discarded. In other words, feature selection aims at finding a subset of the original features of a dataset such that an induction, or learning, algorithm applied to data represented only by those features generates a classifier with the highest possible accuracy [12].

Several methods have been proposed to tackle feature selection. They can be classified into two categories: filter and wrapper methods [12]. In filter methods, the feature selector algorithm is independent of the specific learning algorithm used in the classification and it is used as a filter to discard irrelevant and/or redundant features a priori of the construction of the classification algorithm. A numerical evaluation function is used to compare the feature subsets with respect to their diagnostic accuracy [18]. On the contrary, in wrapper methods the feature selector behaves as a wrapper around the specific learning algorithm used to construct the classifier. The feature subsets are compared using as criterion the diagnostic performance achieved by the classification algorithm itself [12].

Both the filter and the wrapper approaches conduct a search for an optimal feature subset in the space of possible features. The inclusion or not of a feature in the subset can be encoded in terms of a binary variable which takes value 1 or 0, respectively. For $n$ features, the size of the binary vector search space is $2^n$. An exhaustive search is impractical unless $n$ is small.

To overcome this problem, three search approaches are commonly adopted: complete, heuristic and probabilistic [12].

In the complete approach, the properties of a pre-defined evaluation function are used to prune the feature space to a manageable size, thus avoiding the computational burden of an exhaustive search [19]. Only some evaluation functions give rise to a search that in theory guarantees the optimum feature subset selection without being exhaustive [20].

The heuristic approach does not guarantee that the best feature subset is achieved, but is less time consuming than the complete one and may be employed in combination with any evaluation function [13]. At present, the most popular heuristic methods are greedy search strategies such as the sequential forward selection method (SFS) or the sequential backward elimination method (SBE), both known as “hill climbing” methods, which iteratively add or subtract features. At each iteration the evaluation function is computed to compare the successive subsets thereby obtained. The forward selection refers to a search that begins with no features and at each step a feature is added to the subspace; on the contrary, the backward elimination refers to a search that begins with the entire $n$-dimensional feature space and at each step a feature is removed. At each step, the choice of which feature to add or remove is driven by its effect on the evaluation function so as to climb toward its maximum value. As a remark, note that the hill climbing methods suffer from the so called “nesting effect” [12]: if the features added cannot be removed, a local minimum of the evaluation function may be found.

The probabilistic approach is based on population-based metaheuristics guided by fittest solutions, such as the genetic algorithms, presented in this paper, or on methods like simulated annealing and tabu search algorithms [21].

The filter approach is generally computationally more efficient than the wrapper approach because for each feature subset of trial, the computation of an evaluation function from the available data is less time consuming than the development of a complete classification model through a learning algorithm. On the other hand, wrapper approaches are more performing since they ensure the selection of the features more suitable for the specific classification algorithm used in the diagnostic task. On the contrary, the filter
approach totally ignores the effects of the selected feature subspace on the diagnostic performance of the induction algorithm that will be used for the classification.

3. **A SHORT INTRODUCTION TO GENETIC ALGORITHMS**

Genetic Algorithms (GAs) are optimization methods aiming at finding the global optimum of a set of real objective functions, \( F = \{ f_i \} \), of one or more decision variables, \( U = \{ u_i \} \), possibly subject to various linear or non-linear constraints. Their main properties are that the search is conducted i) using a population of multiple solution points or candidates, ii) using operations inspired by the evolution of species, such as breeding and genetic mutation, iii) using probabilistic operations, iv) using only information on the objective or search function and not on its derivatives.

GAs owe their name to their operational similarities with the biological and behavioural phenomena of living beings. After the pioneering theoretical work by John Holland [22], in the last decade a flourishing literature has been devoted to their application to real problems. The basics of the method may be found in Goldberg [23]; some applications in various contexts are included in Chambers [24].

The terminology adopted in GAs contains many terms borrowed from biology, suitably redefined to fit the algorithmic context. Thus, GAs operate on a set of (artificial) chromosomes, which are strings of numbers, generally sequences of binary digits 0 and 1. If the objective functions of the optimization has many arguments (typically called control factors or decision variables), each string is partitioned in as many substrings of assigned lengths, one for each argument and, correspondingly, we say that each chromosome is partitioned in (artificial) genes. The genes constitute the so called genotype of the chromosome and the substrings, when decoded in real numbers, constitute its phenotype. When the objective functions are evaluated in correspondence of a set of values of the control factors of a chromosome, its values are called the fitness of that chromosome. Thus, each chromosome gives rise to a trial solution to the problem at hand in terms of a set of values of its control factors.

The GA search is performed by constructing a sequence of populations of chromosomes, the individuals of each population being the children of those of the previous population and the parents of those of the successive population. The initial population is generated by randomly sampling the bits of all the strings. At each step, the new population is then obtained by manipulating the strings of the old population in order to arrive at a new population hopefully characterized by increased mean fitness. This sequence continues until a termination criterion is reached. As for the natural selection, the string manipulation consists in selecting and mating pairs of chromosomes in order to groom chromosomes of the next population. This is done by repeatedly performing on the strings the four fundamental operations of reproduction, crossover, replacement and mutation, all based on random sampling: the parents’ selection step determines the individuals which participate in the reproduction phase; reproduction itself allows the exchange of already existing genes whereas mutation introduces new genetic material; the substitution defines the individuals for the next population. This way of proceeding enables to efficiently arrive at optimal or near-optimal solutions.

With regards to their performance, it is acknowledged that GAs take a more global view of the search space than many other optimization methods. The main advantages are i) fast convergence to near global optimum, ii) superior global searching capability in complicated search spaces, iii) applicability even when gradient information is not readily achievable.

### 3.1 Multi-Objective optimization

In a multi-objective optimization problem, several possibly conflicting objective functions \( f_i(\cdot), \ i = 1, 2, \ldots, n_f \), must be evaluated in correspondence of each decision variable vector \( U \) in the search space. The goal is
to identify the solution vector $U^*$ which gives rise to the best compromise among the various objective functions. The comparison of solutions is achieved in terms of the concepts of Pareto optimality and dominance [23, 25]: with reference to a maximization problem solution $U_a$ is said to dominate $U_b$ if

$$\forall i \in \{1, 2, ..., n_f\}, f_i(U_a) \geq f_i(U_b)$$

and

$$\exists j \in \{1, 2, ..., n_f\}, f_j(U_a) > f_j(U_b).$$

The decision variable vectors which are not dominated by any other of a given set are called nondominated with respect to this set; the decision variable vectors that are nondominated within the entire search space are said to be Pareto optimal and constitute the so called Pareto optimal set or Pareto optimal front.

When tackling a multi-objective problem by GAs, the various approaches to fitness definition may be distinguished into three categories [26]:

- Plain aggregation methods combine the objectives into a higher scalar function that is used to evaluate the fitness of a solution $U$; an example of these techniques is represented by the weighted-sum approach [27, 28], in which the fitness of solution $U, f(U)$, is computed by the following weighted sum of the $n_f$ objectives:

$$f(U) = \sum_{i=1}^{n_f} w_i \cdot f_i(U)$$

where the arbitrary constant weights $w_i, i = 1, 2, ..., n_f$, satisfy the following relations:

$$w_i \in [0, 1] and \sum_{i=1}^{n_f} w_i = 1$$

The optimization of a single function, combination of the $n_f$ objectives produces a single compromise solution, requiring no further actions by the decision maker. However, if the solution found is not acceptable as a good compromise, tuning of the aggregating weights may be required, followed by new runs of the optimizer, until a suitable solution is found.

- Population-based non-Pareto approaches are able to evolve multiple nondominated solutions concurrently in a single simulation run: for instance, in [29] sub-populations of the next generation are reproduced from the current population according to each of the objectives, separately; then, the overall population at each generation is formed by merging and shuffling the sub-populations. However, this method produces individuals that perform well for each objective separately, while no consideration of trade-offs is taken.

- In Pareto-based methods, reported first in [23], once a population of chromosomes has been created, these are ranked according to the Pareto dominance criterion by looking at the $n_f$-dimensional space of the fitnesses $f_i(U), i = 1, 2, ..., n_f$. Firstly, all nondominated individuals are identified and rank 1 is assigned to them. Then, these solutions are virtually removed from the population and the next set of nondominated individuals are identified and assigned rank 2; this process continues until every solution in the population has been ranked. Every solution belonging to the same rank class has to be considered equivalent to any other of the same class in the sense that it has the same probability of the others to be selected as a parent for the mating.

During the optimization search, an archive of solution vectors $U$, each one constituted by a nondominated chromosome and by the corresponding $n_f$ fitnesses, representing the dynamic Pareto optimality front is recorded and updated [30, 31, 32]; this procedure allows implementation of elitism in the genetic algorithm: every individual in the archive (or a pre-established fraction) is chosen once as a parent in each generation guaranteeing a better propagation of the genetic code of nondominated solutions, and thus a more efficient evolution of the population towards Pareto optimality.
At the end of the search procedure the result of the optimization is constituted by the archive itself which gives the Pareto optimality front.

In this paper, the first and third approaches are investigated.

4. GENETIC ALGORITHMS FOR FEATURE SELECTION

The problem of feature selection illustrated in Section 2 can be formulated as an optimization problem. In this view, the total number of \( n \)-dimensional pre-labelled data are partitioned into a set (hereafter denoted by \( A \)) used for the feature selection task and a separate set (hereafter denoted by \( A' \)) used for validating the performance of the classifier resting upon the optimal feature subset selected. Then, a GA can be devised to find an optimal binary transformation vector \( V^* \), of dimension \( n \), which operates on \( A \) to maximize/minimize a set of optimization criteria regarding the classifier performance and structure.

Let \( m \) be the number of 1’s in \( V^* \) and \( n - m \) the number of 0’s; then, a modified set of patterns \( B^* = V^*(A) \) is obtained in an \( m \)-dimensional space \((m < n)\). Figure 1 shows the structure of a multi-objective GA-feature extractor that uses classification accuracy (to be maximized) and dimension \( m \) of the transformed patterns (to be minimized) as optimization criteria for the feature selection. The GA creates a population of competing transformation vectors \( V^*_i, i = 1, 2, \ldots \), which are evaluated by the following steps [3, 33]:

i. The vector \( V^*_i \) is applied to each pattern of set \( A \) to obtain a set of modified patterns \( B^*_i \).

ii. The set \( B^*_i \) of modified patterns thereby obtained is divided into a training set and a tuning set: the former is used to build (train) the classifier whereas the latter is used to evaluate its performance in terms of classification accuracy.

iii. The classification accuracy obtained and the number of selected features, \( m \), are the two fitness functions used in the GA search. These quantities measure the quality of the feature subset encoded as the transformation vector \( V^*_i \) applied to the set \( A \) of original patterns of \( n \) features to obtain the set \( B^*_i \) of transformed patterns characterized by a reduced number of features, \( m \).

iv. On the basis of this feedback, the GA conducts its probabilistic search for a vector or a set of vectors which give rise to the best compromise among classification accuracy and parsimony in the selection of features.

For the GA feature selector of Figure 1, the organization of the chromosome is quite straightforward [34]: each bit of the chromosome is associated with a parameter (Figure 2) and interpreted such that if the \( i \)-th bit equals 1, then the \( i \)-th parameter is included as feature in the transformed patterns for classification or viceversa if the bit is 0. Note that contrary to other GA applications, in this case of feature selection the binary chromosome does not encode real-valued control factors: the information regarding the features presence or absence in the optimal set for classification is included in the bits themselves so that no decoding is necessary.

Concerning the classification accuracy fitness function, in this work each subset of features encoded in a chromosome is evaluated on the set of tuning data (step ii. above) using a nearest neighbor classifier. More specifically, in the applications which follow the modified pre-labelled patterns constituting set \( B \) to be used for the GA feature selection are randomly subdivided (step ii.) into a training set containing 75% of the data which are used for the classifier training and a tuning set of 25% of the data, which are used to compute the performance of the classifier in terms of its accuracy. The Fuzzy K-Nearest Neighbor algorithm (FKNN) [14], with \( k = 5 \), has here been applied to classify the tuning data on the basis of the location of the labelled training data. The obtained fuzzy partition of the tuning data \( \{ \mu_l(\tilde{x}_i) \} \), where \( 0 \leq \mu_l(\tilde{x}_i) \leq 1 \) is the membership function of pattern \( \tilde{x}_i \) to class \( l \), is converted into a hard partition assigning each pattern to the class with highest membership value. The random subdivision of the available patterns in training and tuning sets is actually repeated 10 times (10 cross-validation) and for each tuning set the accuracy of the FKNN
classifier operating on the proposed subset of features is evaluated in terms of recognition rate, i.e. the fraction of tuning patterns correctly classified [35]; then, the mean recognition rate is calculated and sent back to the GA as the fitness value of the transformation chromosome $V_i$ used to produce the transformed set of patterns $B$.

After the search has achieved convergence, the validation set $A'$, separate from the training and tuning sets used for the feature selection task, is processed through the optimal transformation vector $V^*$ and the classifier to evaluate the classification accuracy on a set $B'$ of new patterns (never used during the feature selection process) specified in terms of the optimal selected feature subset encoded by the transformation vector $V^*$ (Figure 1). This validation procedure is of especially paramount importance for safety applications in critical technologies such as the nuclear one presented in this work.

![FEATURE SELECTOR](image)

**Figure 1.** GA-based feature selection using classification accuracy and number of selected features as optimization criteria. Each binary chromosome from the GA population is used to transform the original patterns, which are then passed to a classifier. The objective function values of the chromosome are the classification accuracy attained on the transformed patterns and their dimension $m$.

![n-dimensional binary chromosome](image)

**Figure 2.** $n$-dimensional binary chromosome.
5. CASE STUDY: CLASSIFICATION OF TRANSIENTS FOR FAULT DIAGNOSIS IN THE FEEDWATER SYSTEM OF A BOILING WATER REACTOR

The diagnostic problem considered is the early identification of a predefined set of faults in the feedwater system of a Boiling Water Reactor (BWR). The corresponding transients have been simulated by the HAMBO simulator of the Forsmark 3 BWR plant in Sweden [15]. Appendix A shows a sketch of the system [15]. The considered faults occur in the section of the feedwater system 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 [36]. Two power levels, 50% and 80% of full power, have been considered in this work.

The faults may be divided into three categories:
- F1-F5,F7 regard line 1 of the feedwater system.
- F11-F15,F17 regard line 2 of the feedwater system.
- F6,F8,F9,F10,F16,F18 regard both lines.

Five transients were simulated for each of the 18 faults, at the two power levels, considering different degrees of leakage and valve closures and with step and ramp changes at different leak sizes. The data relative to \( n = 363 \) plant parameters were recorded with a sampling frequency of 1 Hz. All transients start after 60 seconds of steady state operation. From the analysis developed in [17], it is clear that faults 6, 10, 16 have no significant consequence on the plant measured parameters because the size of the leakage considered is too small. Hence, these faults were not considered further.

For the classification purpose of this work, the number of parameters was reduced to 123 by combining redundant measurements of the same physical quantity and by eliminating those parameters linearly correlated or carrying basically the same time evolution in all the different faults. This pre-processing of the data was done in a controlled way so as to ensure that information redundancy were still kept, with the aim of making the model robust with respect to sensors degradation and failure. Since the goal is early diagnosis, only the data of the first 65 seconds after the beginning of the transients have been considered. Given that the ramp changes cause variations of the parameters later than the step changes, only the three step changes for each fault were considered.

The transients at 50% of full power were used for the feature selection task (training and tuning), whereas the transients at 80% of full power were left out for validation of the resulting classifier. Although all at 50% power level, the transients for the feature selection begin from slightly different initial parameter values due to the fact that some faults, like for example those involving boron dilution, are associated to processes with very long characteristic times [36]. To avoid that for this reason the feature selection and classification algorithms are applicable only from a specific steady state of the reactor and to allow the validation on the transients at 80% power, the difference between the feature values at the generic sampling time during the transients and their steady-state values before the beginning of the transients are used.

Also, as explained in [17], the number of patterns of a single transient has been reduced to 7 by taking values every 10 seconds, in order to reduce the complexity of the classification. Thus, the number of patterns employed for developing the feature selection and classification algorithms is 315 (7 points for each of the 45 transients considered).

Validation of the performance of the classifiers based on the optimal feature subsets identified by the GA has been carried out with respect to the transients at 80% of full power, none of which has been used during the feature selection phase. Treating these transients analogously to those at 50% power, a total of 315 patterns are obtained for the validation set. The different feature selection techniques are then compared with respect to the mean recognition rate resulting from a 25-fold cross validation performed on such validation set.
5.1 A: Single objective GA search

Following the approach proposed in [37], a single-objective GA using as fitness function the FKNN mean recognition rate has been applied to select the relevant features from the 123 parameters of the previous case study. The GA is then required to search for an optimal solution in a $2^{123}$-dimensional space.

The two-fold aim of the search is represented, as before, by dimensionality reduction with classification accuracy improvement: the target is to improve the base case recognition rate of $0.6472 \pm 0.0551$, obtained on the validation set by a FKNN classifier which uses all the 123 signals and to achieve a recognition rate at least comparable to that value of $0.8839 \pm 0.0342$, obtained by a FKNN classifier operating on 16 of the 123 signals selected by a filter approach of literature based on an extended classifiability evaluation function [17]. Briefly, the definition of the classifiability evaluation function is based on the representation in $m$ dimensions of a classification problem with $m$ features, adding the class label as the $(m + 1)$-th dimension. The resulting class label hyper-surface is “rough” when patterns of different classes are close and “smooth” when patterns of the same class are adjacent. Intuitively, classification is considerably more complicated when the class label hyper-surface is rough. Thus, the smoothness (or roughness) of the class label hyper-surface depends on the feature subspace and can thus be used as a measure of performance within a feature selection filter procedure [13].

The parameters used to run the single objective GA are reported in Table 1:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of chromosomes in the population, $Pop_{size}$</td>
<td>100</td>
</tr>
<tr>
<td>Number of generations (termination criterion), $N_{gen}$</td>
<td>400</td>
</tr>
<tr>
<td>Selection, $Sel$</td>
<td>Standard Roulette</td>
</tr>
<tr>
<td>Replacement, $Repl$</td>
<td>Children – Parents</td>
</tr>
<tr>
<td>Mutation Probability, $P_m$</td>
<td>0.001</td>
</tr>
<tr>
<td>Crossover probability (one-site), $P_c$</td>
<td>1</td>
</tr>
</tbody>
</table>

*Table 1. GA run parameters*

Running the GA without imposing any constraint on the chromosomes, the algorithm finds an optimal subset of 40 signals ($S_{40}^{opt}$) characterized by a FKNN recognition rate of $0.8355 \pm 0.0353$ on the tuning data of transients at 50% power and of $0.8453 \pm 0.0369$ on the validation data at 80% power, significantly better than the base case and statistically comparable to that achieved by the filter approach. This result remains, however, only “black-box”-appreciable because the feature subset is too wide to interpret from a physical point of view.

To improve this aspect, at first, the same procedure as in [37], of imposing a limit on the number of possible features to be included in the patterns for the FKNN classifier [38], was attempted. The constraint on the maximum number of features is crudely implemented by a procedure which checks whether a chromosome created by crossover or mutation respects the condition and in the negative case it discards it from the population and creates a new one. This, however, has proved to be not applicable because the probability of generating a 123-bit individual with less than 40 bits containing “1” is extremely small so that the GA may not succeed in creating even the first population in acceptable computing time. In order to overcome this problem, a hierarchical 2-stage strategy has been adopted (Figure 3): in the first step, an unconstrained GA search is performed to reduce the dimension of the feature space from $n = 123$ to $m'$ (in our case it turns out $m' = 40$); then, a second (if necessary, constrained) GA run is used to search for an optimal subspace
selected $m^\text{opt}$ among the previously selected $m^\prime$ signals ($m < m^\prime$). The second step has been attempted imposing different constraints on the number of features (40, 20, 10 and 8), producing the results in Table 2.

<table>
<thead>
<tr>
<th>Max. number of features</th>
<th>Number of features in the transformed patterns, $m$ (out of $m^* = 40$)</th>
<th>Recognition rate obtained on the tuning set</th>
<th>Included features</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>28</td>
<td>0.8609 ± 0.0292</td>
<td>$S_{m^\text{opt},2\text{-stage}}$</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>0.8614 ± 0.0364</td>
<td>$S_{m^\text{opt},2\text{-stage}}$</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>0.8519 ± 0.0373</td>
<td>$S_{m^\text{opt},2\text{-stage}}$</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>0.8419 ± 0.0422</td>
<td>$S_{m^\text{opt},2\text{-stage}}$</td>
</tr>
</tbody>
</table>

Table 2. Results of the feature selection performed by the single-objective GA in the second step, searching among the $m^* = 40$ previously selected features

The identification numbers of the features (in Appendix B, the corresponding physical description) constituting the optimal subspaces in Table 2 are:

- $S_{m^\text{opt},2\text{-stage}} = \{59, 60, 76, 77, 100, 107, 109, 140, 155, 157, 165, 168, 191, 194, 195, 196, 197, 223, 229, 242, 281, 320, 321, 324, 325, 332, 357, 359\}$;
- $S_{m^\text{opt},2\text{-stage}} = \{59, 60, 76, 100, 109, 140, 155, 165, 168, 195, 196, 197, 242, 320, 321, 324, 325, 342, 344, 359\}$;
- $S_{m^\text{opt},2\text{-stage}} = \{100, 109, 160, 194, 195, 241, 242, 320, 321, 359\}$;
- $S_{m^\text{opt},2\text{-stage}} = \{77, 160, 194, 195, 242, 320, 321, 359\}$.

$m^\text{opt}_{20}$ turns out to be the best feature subset because it presents the highest classification accuracy with an acceptable number of features. Its validation on the data at 80% of full power gives a classification accuracy of $0.8513 \pm 0.0381$.

![Two-stage hierarchical strategy](image)

Figure 3. Two-stage hierarchical strategy: the first unconstrained GA search reduces the dimension of the feature set from $n = 123$ to $m^\prime < 123$, while the second (possibly constrained) run reduces the dimension of the obtained subset to $m < m^\prime$. 

10
5.2 B: Weighted-sum GA search

In this Section, we investigate the possibility of performing a direct, one-stage, feature selection efficient from the multi-objective point of view of maximum classification performance and minimum number of features. To this aim, a single-objective GA based on the weighted aggregation of the two objectives in a single fitness function (Section 3) has been considered.

Let \( n \) be the total number of parameters in the original data set and \( m(V_i) \) the number of features selected by the transformation vector \( V_i \) of the \( i \)-th chromosome. Two objectives are considered: \( f_1(V_i) \), the FKNN mean recognition rate (Section 4), and \( f_2(V_i) = (n - m(V_i))/n \) in which the numerator is larger the less features are included in \( V_i \), i.e. the smaller is \( m(V_i) \), whereas the normalization with respect to \( n \) transforms also the second objective function in the range \([0, 1]\). These two objectives are combined into the single-valued fitness \( f(V_i) \) through fixed weights \( w_1 \) and \( w_2 \), with \( w_2 = 1 - w_1 \) from (3).

Table 3 shows the GA run parameters adopted for the search. Differently from the previous single-objective search, a replacement procedure promoting the fittest individuals has been found to improve efficiency in the evolution of the population towards the optimum.

| Number of chromosomes in the population, Pop_size | 100 |
| Number of generations (termination criterion), N_gen | 500 |
| Selection, Sel | Standard Roulette |
| Replacement, Repl | Fittest |
| Mutation Probability, \( P_m \) | 0.001 |
| Crossover probability (one-site), \( P_c \) | 1 |

*Table 3. Weighted-sum GA run parameters*

Different combinations of constant weights \( w_1 \) and \( w_2 \) have been tested in the attempt to find a suitable compromise solution. From (3), notice that the combination of the two objective functions into a single one actually rests only on one parameter \( w_1 = w \) (with \( w_2 = 1 - w \) ) which functions as a regularization parameter used to limit the complexity of the problem like in ridge regression and neural network regularization. The results are given in Table 4.
An increasing value of $w_2$ (and, correspondingly, a decreasing value of $w_1$) causes, as expected, a reduction of the number of features in the transformed patterns but this is not necessarily accompanied by a decrease in the classification accuracy, which confirms the general advantage of using a feature selection technique in support to a classification task a priori based on a large number $n$ of available features.

The pair $w_1 = 0.66, w_2 = 0.34$ turns out to be the best combination, since the related optimal feature subspace is characterized by the highest mean recognition rate ($0.8622$) with an acceptable number of features (11). Validation on the set of transients at 80% power level gives a recognition rate of $0.8992 \pm 0.0334$ which compares well to that obtained with the filter approach based on the extended classifiability evaluation function [17].

The identification numbers (Appendix B) of the features constituting the best subspace in Table 4 are:

- $S_{11}^{opt,weight} = \{60, 77, 140, 195, 241, 242, 320, 321, 324, 325, 330\}$.

The features constituting $S_{11}^{opt}$ distinctly regard either only line 1, only line 2 or both lines. Figures 4, 5 and 6 show their corresponding time evolutions at 50% power from 50s to 150s for three transients caused by step faults of the 15 types considered, F1-F5, F7-F9, F11-F15, F17, F18. In each graph, the evaluation of the variation of the feature from its steady state value at the starting of the fault is reported. The identification number of the feature (Appendix B) and the fault number are indicated at the top of the columns and on the left of the rows of the figures, respectively.

Figure 4 refers to faults and features related to line 1 (F1-F5, F7 and features 195, 320, 324). Feature 320 evolves clearly in a characteristic way in case of faults F1, F3, F5, although the evolutions for the latter two are quite similar. Feature 195 is sensitive to faults F2, F7 and somewhat to F4. Feature 324 helps feature 320 in discerning between faults F3 and F5. Globally, the selected features seem indeed capable of distinguishing among the faults concerning line 1.

<table>
<thead>
<tr>
<th>$w_1$</th>
<th>$w_2 = 1-w_1$</th>
<th>Number of features in the transformed patterns, $m$</th>
<th>Recognition rate obtained on the tuning set</th>
<th>Included features</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.999</td>
<td>0.001</td>
<td>28</td>
<td>$0.8568 \pm 0.0221$</td>
<td>$S_{28}^{opt,weight}$</td>
</tr>
<tr>
<td>0.99</td>
<td>0.01</td>
<td>25</td>
<td>$0.8539 \pm 0.0254$</td>
<td>$S_{25}^{opt,weight}$</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>18</td>
<td>$0.8555 \pm 0.0262$</td>
<td>$S_{18}^{opt,weight}$</td>
</tr>
<tr>
<td>0.66</td>
<td>0.34</td>
<td>11</td>
<td>$0.8622 \pm 0.0380$</td>
<td>$S_{11}^{opt,weight}$</td>
</tr>
<tr>
<td>0.50</td>
<td>0.50</td>
<td>9</td>
<td>$0.8363 \pm 0.0433$</td>
<td>$S_{9}^{opt,weight}$</td>
</tr>
<tr>
<td>0.34</td>
<td>0.66</td>
<td>8</td>
<td>$0.8496 \pm 0.0383$</td>
<td>$S_{8}^{opt,weight}$</td>
</tr>
</tbody>
</table>

Table 4. Results of the feature selection performed by a single-objective GA considering an aggregation of objectives for fitness evaluation

PART III

Selected papers
The features selected with respect to line 2 (Figure 5) are symmetric to those identified for line 1: overall, the selected features (242, 321, 325) seem capable of distinguishing the faults F11-F15, F17 relevant to this line.

For what concerns the faults related to the part of the feedwater system common to the two lines, the feature selection GA identifies 5 features (60, 77, 140, 241, 330 Figure 6). Feature 77 and 140 are particularly sensitive to faults F9 and F18 and they are helped by features 241 and 330 in discerning between them; fault F8 is exclusively characterized by the departure of features 60, 77 and 330 from their stationary states.

In conclusion, the wrapper, weighted-sum single-objective GA succeeds in distinguishing different faults without presenting the typical drawback of a filter method, that is the selection of redundant features [17].

Comparing \( S_{11}^{\text{opt}} \) to the best feature subset previously found through the hierarchical two-stage single-objective GA, \( S_{W}^{\text{opt,2-stage}} \), it can be seen that the former achieves a slightly higher recognition rate than the latter with a significant reduction in the number of features selected, from 20 to 11. This result, obtained in a single run of the weighted-sum single-objective GA, proves the superior effectiveness of the approach. On the other hand, it is comforting, from a physical viewpoint, that the two sets found by the two approaches share a significant number of features (8), which indeed play a leading role in the classification task.
PART III

Figure 5. Time evolution of the variations, from their steady state values, of the features contained in $S_{11}^{\text{opt.\ weight}}$, selected by the weighted-sum single-objective GA among the original 123 features with respect to line 2 of the feedwater system.

Figure 6. Time evolution of the variations, from their steady state values, of the features contained in $S_{11}^{\text{opt.\ weight}}$, selected by the weighted-sum single-objective GA among the original 123 features with respect to both lines 1 and 2 of the feedwater system.
5.3 C: Pareto-based multi-objective GA search

The use of constant weight values in the aggregated objective function forces the GA to search for an optimal solution with respect to an a priori fixed preference combination of the two objectives, revealing the inadequacy of this method to find all the nondominated solutions of the multi-objective optimization problem for a posteriori trade-off. In this Section, the potentialities of a Pareto-based approach (Section 3) are investigated with respect to the goal of identifying equivalently optimal feature subsets characterized by different performances in terms of classification accuracy (FKNN mean recognition rate) and number of features.

The performance of a Pareto-based MOGA depends largely on its ability to maintain genetic diversity while attempting to produce a population of individuals uniformly distributed in the vicinity of the various members of the Pareto front itself, so as to achieve a front widely representative of the real nondominated solutions [4, 23]. Thus, the effects on the genetic diversity of different combinations of selection ($Sel$) and replacement ($Repl$) procedures, probabilities of mutation ($P_m$), population sizes ($Pop_size$) and number of generations ($N_{gen}$), have been studied by performing the following MOGA computations:

- A1: $\{Sel: \text{random}; \ Repl: \text{fittest}; \ P_m = 0.001; \ Pop_size = 100; \ N_{gen} = 500\}$;
- A2: $\{Sel: \text{random}; \ Repl: \text{fittest}; \ P_m = 0.008; \ Pop_size = 100; \ N_{gen} = 500\}$;
- A3: $\{Sel: \text{random}; \ Repl: \text{fittest}; \ P_m = 0.001; \ Pop_size = 200; \ N_{gen} = 1000\}$;
- B1: $\{Sel: \text{fit-fit}; \ Repl: \text{children-parents}; \ P_m = 0.001; \ Pop_size = 100; \ N_{gen} = 500\}$;
- B2: $\{Sel: \text{fit-fit}; \ Repl: \text{children-parents}; \ P_m = 0.008; \ Pop_size = 100; \ N_{gen} = 500\}$;
- C1: $\{Sel: \text{fit-fit}; \ Repl: \text{fittest}; \ P_m = 0.001; \ Pop_size = 100; \ N_{gen} = 500\}$;
- C2: $\{Sel: \text{fit-fit}; \ Repl: \text{fittest}; \ P_m = 0.008; \ Pop_size = 100; \ N_{gen} = 500\}$;
- C3: $\{Sel: \text{fit-fit}; \ Repl: \text{fittest}; \ P_m = 0.008; \ Pop_size = 200; \ N_{gen} = 1000\}$.

Most of these MOGA strategies and parameter values are widely adopted in literature [39]: in particular, $P_m = 0.001$ is recommended in [40], whereas $P_m = 0.008$ is the inverse of the number of bits constituting a chromosome, as suggested in [41, 42, 43]. The fraction of the population chosen through elitist selection [24] in each generation is set to 0.25 in all cases.

Table 5 contains the objective function values (number of selected features and mean recognition rates) of the sets of nondominated solutions for the different MOGA settings. With respect to the dimension $m$ of the Pareto solutions, note that it can range from 1 to 13 but only case C2 identifies a complete set from $m = 1$ to $m = 13$. Also, note that two different MOGA computations may reach a solution with an equal number $m$ of different features, to which correspond different mean recognition rates (e.g. cases A1, A2, B1, B2 for $m = 3$). Figures 7 and 8 show the corresponding Pareto frontiers, the final populations and the performance achieved by the reference FKNN classifier which uses all the 123 signals (for graphical reasons, this latter is reported out of scale with respect to the number of features, at $m = 19$ instead of $m = 123$).

From the results obtained, general considerations can be drawn about the effect of different genetic operators on the structure of the Pareto optimal set and on the convergence of both population and archive towards the Pareto front:

- $P_m$: low probabilities of mutation (0.001, as in MOGA configurations A1, A3, B1, C1) allow the population to uniformly converge towards the optimal frontier; higher values of $P_m$ act negatively when combined with fitness-ignoring replacement procedures (children-parents, as in MOGA configuration B2), reducing the extension of the front and producing a scattered final population.
<table>
<thead>
<tr>
<th>Number of features in the transformed patterns, $m$</th>
<th>Mean recognition rates on the tuning set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3431</td>
</tr>
<tr>
<td>2</td>
<td>0.6662</td>
</tr>
<tr>
<td>3</td>
<td>0.7960</td>
</tr>
<tr>
<td>4</td>
<td>0.8310</td>
</tr>
<tr>
<td>5</td>
<td>0.8374</td>
</tr>
<tr>
<td>6</td>
<td>0.8395</td>
</tr>
<tr>
<td>7</td>
<td>0.8444</td>
</tr>
<tr>
<td>8</td>
<td>0.8547</td>
</tr>
<tr>
<td>9</td>
<td>0.8584</td>
</tr>
<tr>
<td>10</td>
<td>0.8598</td>
</tr>
<tr>
<td>11</td>
<td>0.8610</td>
</tr>
<tr>
<td>12</td>
<td>/</td>
</tr>
<tr>
<td>13</td>
<td>0.8613</td>
</tr>
</tbody>
</table>

Table 5. Objective function values of the nondominated solutions found by the Pareto-based MOGA for different combinations of genetic operators

On the other hand, in strongly fitness-oriented searches (e.g. fittest replacement procedures), large values of $P_m$ positively improve genetic diversity, favouring a deep search of the feature space and producing a wide Pareto optimal front (MOGA configurations A2, C2, C3);

- **Repl**: fitness-guided replacement procedures (e.g. fittest) efficiently move the population towards solutions on the optimal frontier; on the other hand, when they are combined with fitness-guided selection methods (e.g. fit-fit), genetic diversity must be ensured by means of a relatively high mutation probability to avoid entrapment in local optima and to obtain an extended front (MOGA configuration C1 vs C2);

- **Sel**: selection seems less determining than replacement; however, it is interesting to observe that the combination of a uniform sampling of the population (random selection) with a fitness-guided replacement procedure (e.g. fittest) helps improving genetic diversity and finding a wide optimal frontier (MOGA configuration A1 vs C1).

MOGA configuration C2 presents the best Pareto set in terms of completeness, homogeneity and classification accuracy of its members: its characteristics are summarized in Table 6.
<table>
<thead>
<tr>
<th>Number of features in the transformed patterns, $m$</th>
<th>Mean recognition rate</th>
<th>Included features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3431 ± 0.0395</td>
<td>$S^*_1$ opt.Pareto</td>
</tr>
<tr>
<td>2</td>
<td>0.6662 ± 0.0380</td>
<td>$S^*_2$ opt.Pareto</td>
</tr>
<tr>
<td>3</td>
<td>0.8159 ± 0.0325</td>
<td>$S^*_3$ opt.Pareto</td>
</tr>
<tr>
<td>4</td>
<td>0.8313 ± 0.0442</td>
<td>$S^*_4$ opt.Pareto</td>
</tr>
<tr>
<td>5</td>
<td>0.8393 ± 0.0379</td>
<td>$S^*_5$ opt.Pareto</td>
</tr>
<tr>
<td>6</td>
<td>0.8455 ± 0.0425</td>
<td>$S^*_6$ opt.Pareto</td>
</tr>
<tr>
<td>7</td>
<td>0.8488 ± 0.0368</td>
<td>$S^*_7$ opt.Pareto</td>
</tr>
<tr>
<td>8</td>
<td>0.8547 ± 0.0344</td>
<td>$S^*_8$ opt.Pareto</td>
</tr>
<tr>
<td>9</td>
<td>0.8584 ± 0.0363</td>
<td>$S^*_9$ opt.Pareto</td>
</tr>
<tr>
<td>10</td>
<td>0.8598 ± 0.0384</td>
<td>$S^*_{10}$ opt.Pareto</td>
</tr>
<tr>
<td>11</td>
<td>0.8622 ± 0.0367</td>
<td>$S^*_{11}$ opt.Pareto</td>
</tr>
<tr>
<td>12</td>
<td>0.8647 ± 0.0345</td>
<td>$S^*_{12}$ opt.Pareto</td>
</tr>
<tr>
<td>13</td>
<td>0.8658 ± 0.0339</td>
<td>$S^*_{13}$ opt.Pareto</td>
</tr>
</tbody>
</table>

Table 6. Detailed description of the best Pareto optimal frontier corresponding to MOGA configuration C2

The features (Appendix B) included in these nondominated solutions are:

- $S^*_1$ opt.Pareto = \{321\};
- $S^*_2$ opt.Pareto = $S^*_1$ opt.Pareto + \{320\};
- $S^*_3$ opt.Pareto = $S^*_2$ opt.Pareto + \{357\};
- $S^*_4$ opt.Pareto = $S^*_3$ opt.Pareto + \{193\};
- $S^*_5$ opt.Pareto = \{59, 192, 194, 320, 321\};
- $S^*_6$ opt.Pareto = $S^*_5$ opt.Pareto + \{193\};
- $S^*_7$ opt.Pareto = $S^*_6$ opt.Pareto + \{240, 241\};
- $S^*_8$ opt.Pareto = \{139, 160, 195, 213, 241, 242, 320, 321\};
- $S^*_9$ opt.Pareto = $S^*_8$ opt.Pareto + \{359\};
- $S^*_{10}$ opt.Pareto = $S^*_9$ opt.Pareto + \{194\};
- $S^*_{11}$ opt.Pareto = \{60, 77, 140, 195, 241, 242, 320, 321, 324, 325, 330\};
- $S^*_{12}$ opt.Pareto = $S^*_{11}$ opt.Pareto + \{359\};
- $S^*_{13}$ opt.Pareto = $S^*_{12}$ opt.Pareto + \{143\}.

In a single run, the Pareto-based MOGA identifies a range of nondominated solutions with different complexity/performance trade-offs: once such a subset is available, an informed choice can be made of the features to be actually monitored for the diagnostic task, by considering diagnostic classification accuracy and possibly also practical issues related to costs and ease of data acquisition.
Figure 7. Representation, in the objective functions space, of Pareto optimal frontiers, last-generation populations and an out-of-scale indication of the classification accuracy obtained by a FKNN classifier using all the 123 features for configurations A1, A2, B1, B2 of the MOGA.

Figure 8. Representation, in the objective functions space, of Pareto optimal frontiers, last-generation populations and an out-of-scale indication of the classification accuracy obtained by a FKNN classifier using all the 123 features for configurations A3, C1, C2, C3 of the MOGA.
It is significant that $S_{11}^{\text{opt,Pareto}}$ achieves a classification accuracy comparable to those of other nondominated feature subsets characterized by higher numbers of features. The time evolutions of the features contained in $S_{11}^{\text{opt,Pareto}}$, here not reported for brevity, show that feature 357, which evolves in a characteristic way in case of faults F8, F9, F12 and F18, completes the information borne by features 320 and 321 (Section 5.2) giving rise to the efficient recognition of almost all the faults.

Further, it is notable that $S_{11}^{\text{opt,Pareto}}$ coincides with $S_{11}^{\text{opt,weight}}$ and that this core of features is also maintained in $S_{12}^{\text{opt,Pareto}}$ and $S_{13}^{\text{opt,Pareto}}$.

Validation of the classifier applied to the transients at 80% power level confirms the above results.

An analysis of the Pareto optimal solutions also shows that in contrast with heuristic stepwise methods, which add or delete one input at a time, the GA approach may find Pareto-front solutions by an increase/decrease in the number of selected features which not necessarily implies simply addition/deletion of one parameter. Instead, it may be the case, for instance, that some features are removed and two or more others are added: for example, when moving from $m = 4$ to $m = 5$, features 193 and 357 are removed whereas 59, 192 and 194 are added (Table 6).

6. SYNTHESIS AND DISCUSSION OF THE WORK

Three GA-FKNN wrapper approaches have been tested on a case study regarding the selection of features among $n = 123$ initial parameters for the purpose of diagnosing transients in the feedwater system of a Boiling Water Reactor.

First, a single-objective GA search has been performed, leading to an optimal feature subset of $m = 40$ signals ($S_{40}^{\text{opt,2-stage}}$) characterized by a FKNN recognition rate of $0.8355 \pm 0.0353$ on the tuning data of transients at 50% power level and of $0.8453 \pm 0.0369$ on the validation set made of transients at 80% power level not employed in the construction of the classifier. The classification accuracy is significantly improved with respect to the case which employs all the 123 features for the classification and is statistically comparable to the case of an optimal feature subset identified by a filter approach of literature. However, the optimal feature subset is too wide to be interpreted from a physical point of view.

For further dimensionality reduction, a hierarchical 2-stage strategy has been adopted to progressively screen out unimportant features in successive single-objective GA searches: a considerable reduction of the feature set is obtained (from $n = 123$ features to $m = 20$), accompanied by an appreciable improvement in classification accuracy on the tuning set ($0.8614 \pm 0.0364$) and validation set of transients at 80% power level ($0.8513 \pm 0.0381$).

In order to drive the GA towards more parsimonious solutions in terms of number of features retained, a multiobjective problem has been transformed into a single-objective one by a weighted aggregation of two objectives, FKNN mean recognition rate and number of features, for fitness evaluation. The optimal subset found by this method ($S_{11}^{\text{opt,weight}}$) achieves a slightly higher recognition rate ($0.8622 \pm 0.0382$) on the tuning set and a significantly higher one ($0.8992 \pm 0.0334$) on the validation set, with a significant dimensionality reduction (from 20 to 11) and thus a good physical interpretability with respect to the previous optimal subset, $S_{20}^{\text{opt,2-stage}}$. On the other hand, tuning of the aggregating weights has been necessary.

Notice that the recognition rates achieved on the validation set obtained from transients at 80% power level are in general higher than those achieved on the tuning set made of patterns from transients at 50% power. This is due to the fact that signal variations are more relevant when the reactor is working at a higher power level so that the classification task becomes somewhat simpler.
Finally, a multi-objective Pareto-based approach has been embraced to arrive at a set of nondominated feature subsets characterized by different classification performances and dimensionalities, without the need of a priori imposing constraints or arbitrary weights. The performance deriving from such approach has been analyzed in details through a parameter analysis regarding the main characteristics of the GA search procedure. As expected, maintenance of genetic diversity throughout the search has confirmed to be a key issue. In this respect, no attempts has been made in this work to enhance the algorithm with methods for maintaining diversity such as niching and sharing [44]. Research is currently underway in this direction.

From the general point of view of the methodology, a key advantage of the GA feature selection wrapper technique with respect to a filter approach is that the search measures explicitly the goodness of the considered feature subsets on the performance of the classification algorithm (in our case the FKNN) that is used for the diagnostic task.

Additionally, since each of the \( m \) output features of the GA feature selector is related only to one of the \( n \) input parameters, the relationship between the original parameters and the transformed features remains explicit and easy to identify. The analysis of the features that prove to be relevant for the diagnostic classification task can lead to a deeper understanding of the data and underlying physical relationships, allowing feature selection to be used in exploratory data analysis and data mining.

Finally, the GA-FKNN wrapper approach is immune from the “nesting effect” characteristic of “hill climbing” methods, so that the probability of getting stuck in local optima is significantly lower.

**Acknowledgements**

The authors wish to thank Drs. Paolo Fantoni and Davide Roverso of the IFE, Halden Reactor Project for providing the transient simulation data. Also, many thanks go to the reviewers for their constructive comments which have allowed improving the paper.

7. REFERENCES


PART III
Selected papers


APPENDIX A: SKETCH OF THE FEEDWATER SYSTEM OBJECT OF THE CASE STUDY IN SECTION 5

APPENDIX B: FEATURE NUMBER AND NAME WITH REFERENCE TO THE CASE STUDY OF SECTION 5

<table>
<thead>
<tr>
<th>Feature number</th>
<th>Name</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
<td>Temperature high-pressure drain</td>
<td>°C</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>24</td>
<td>Temperature after EA1 (high pressure preheater A1)</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>Pressure drain 6 before VB1</td>
<td>MPa</td>
</tr>
<tr>
<td>74</td>
<td>Pressure drain 5 before VB2</td>
<td>MPa</td>
</tr>
<tr>
<td>75</td>
<td>Temperature drain 6 after VB1</td>
<td>°C</td>
</tr>
<tr>
<td>76</td>
<td>Temperature drain 5 after VB2</td>
<td>°C</td>
</tr>
<tr>
<td>77</td>
<td>Temperature drain 4 before VB3</td>
<td>°C</td>
</tr>
<tr>
<td>100</td>
<td>Pressure before EA1 (preheater A1)</td>
<td>MPa</td>
</tr>
<tr>
<td>107</td>
<td>Temperature after 424EA1 (preheater A1)</td>
<td>°C</td>
</tr>
<tr>
<td>109</td>
<td>Temperature after EA1 train A (preheater A1)</td>
<td>°C</td>
</tr>
<tr>
<td>139</td>
<td>Temperature of condensate after EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>140</td>
<td>Temperature of condensate after EB2 train A</td>
<td>°C</td>
</tr>
<tr>
<td>143</td>
<td>Temperature of condensate after EB3 train B</td>
<td>°C</td>
</tr>
<tr>
<td>155</td>
<td>Temperature of bearing, drive end PB1.M1</td>
<td>°C</td>
</tr>
<tr>
<td>157</td>
<td>Temperature of bearing, drive end PB2.M1</td>
<td>°C</td>
</tr>
<tr>
<td>160</td>
<td>Water level tank TD1</td>
<td>m</td>
</tr>
<tr>
<td>165</td>
<td>Temperature of axial bearing PC3 (pump 3)</td>
<td>°C</td>
</tr>
<tr>
<td>168</td>
<td>Temperature seal water to PC3.E1</td>
<td>°C</td>
</tr>
<tr>
<td>191</td>
<td>Water level in EB1 train B (preheater B1)</td>
<td>m</td>
</tr>
<tr>
<td>192</td>
<td>Water level in EA2 train A (preheater A2)</td>
<td>m</td>
</tr>
<tr>
<td>193</td>
<td>Water level in EB2 train B (preheater B2)</td>
<td>m</td>
</tr>
<tr>
<td>194</td>
<td>Temperature feedwater before EA2 train A</td>
<td>°C</td>
</tr>
<tr>
<td>195</td>
<td>Temperature feedwater after EA2 train A</td>
<td>°C</td>
</tr>
<tr>
<td>196</td>
<td>Temperature of radial bearing PA1.P1 (pump A1)</td>
<td>°C</td>
</tr>
<tr>
<td>197</td>
<td>Temperature of radial bearing PA1.P1 (pump A1)</td>
<td>°C</td>
</tr>
<tr>
<td>223</td>
<td>Vibration PA1.P1 outer vertical</td>
<td>mm/s</td>
</tr>
<tr>
<td>229</td>
<td>Pressure after pump PB1</td>
<td>MPa</td>
</tr>
<tr>
<td>241</td>
<td>Temperature feedwater before EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>242</td>
<td>Temperature feedwater after EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>281</td>
<td>Temperature of radial bearing PD1.P1 (pump D1)</td>
<td>°C</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>%</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>320</td>
<td>Position level control valve for EA1</td>
<td></td>
</tr>
<tr>
<td>321</td>
<td>Position level control valve for EB1</td>
<td></td>
</tr>
<tr>
<td>324</td>
<td>Position level control valve before EA2</td>
<td></td>
</tr>
<tr>
<td>325</td>
<td>Position emergency drain valve EB2</td>
<td></td>
</tr>
<tr>
<td>330</td>
<td>Position valve for level I EB4</td>
<td></td>
</tr>
<tr>
<td>332</td>
<td>Position drain valve TD1</td>
<td></td>
</tr>
<tr>
<td>342</td>
<td>Position valve in drain for TB1</td>
<td></td>
</tr>
<tr>
<td>344</td>
<td>Position valve in drain for TB2</td>
<td></td>
</tr>
<tr>
<td>357</td>
<td>Position steam stop control valve</td>
<td></td>
</tr>
<tr>
<td>359</td>
<td>Position steam stop control valve</td>
<td></td>
</tr>
</tbody>
</table>
A Niched Pareto Genetic Algorithm for Selecting Features for Nuclear Transients Classification

Enrico Zio, Piero Baraldi, Nicola Pedroni

Submitted for publication to
IEEE Transactions on Evolutionary Computations
2006
A Niched Pareto Genetic Algorithm for Selecting Features for Nuclear Transients Classification

E. Zio*, P. Baraldi, N. Pedroni
Department of Nuclear Engineering, Polytechnic of Milan, Via Ponzio 34/3, 20133 Milan, Italy
*Corresponding author. Phone: +39-2-2399-6340; fax: +39-2-2399-6309
E-mail address: enrico.zio@polimi.it

Abstract

Multi-objective genetic algorithms can be effective means for choosing the process features relevant for transient diagnosis. The technique allows identifying a family of equivalently optimal feature subsets, in the Pareto sense. However, difficulties in the convergence of the standard Pareto-based multi-objective genetic algorithm search in large feature spaces may arise in terms of representativeness of the identified Pareto front whose elements may turn out to be unevenly distributed in the objective functions space, thus not providing a full picture of the potential Pareto-optimal solutions. To overcome this problem, a modified Niched Pareto Genetic Algorithm is embraced in this work. The performance of the feature subsets examined during the search is evaluated in terms of two optimization objectives: the classification accuracy of a Fuzzy K-Nearest Neighbors classifier and the number of features in the subsets. During the genetic search, the algorithm applies a controlled “niching pressure” to spread out the population in the search space so that convergence is shared on different niches of the Pareto front which is thus evenly covered. The method is tested on a diagnostic problem characterized by a very large number of process features available for the classification of simulated transients in the feedwater system of a Boiling Water Reactor. The dynamics of the transient signals is captured by wavelet decomposition which actually increases the complexity of the search for the optimal feature subsets by tripling the number of features to be considered.

1. INTRODUCTION

The first step in the development of model-based techniques for the diagnosis of faults in nuclear power plants is the selection of the parameters (hereafter also called features) relevant to the diagnosis. Depending on the application, this may be a crucial task since irrelevant and noisy features unnecessarily increase the complexity of the diagnostic problem and can degrade modeling performance [1]. Further, in nuclear power plants the problem is also critical from the practical point of view, due to the very large number of parameters monitored for operation and safety reasons.

In this paper, the problem of search for an optimal feature subset upon which to perform diagnostics of nuclear transients is carried out via a Multi-Objective Genetic Algorithm (MOGA) within a wrapper approach [2].
The objective functions used for evaluating and comparing the feature subsets during the search are the recognition rate achieved by a Fuzzy K-Nearest Neighbors classifier [3] and the number of features forming the subsets.

Correspondingly, the goal of the MOGA search is to converge on a family of feature subsets representative of the true non-dominated solutions which form the Pareto front in the two-dimensional objective functions space [1, 4]. Pictorially, this amounts to looking for a group of solutions which in the objective functions space are uniformly distributed in the vicinity of the true non-dominated solutions forming the Pareto set.

The success of the search depends largely on the ability of the algorithm of maintaining genetic diversity in the population. In this respect, a standard Pareto-based MOGA [4] may encounter difficulties in maintaining genetic diversity during a search in a high-dimensional feature space [5], so that the solutions found at convergence may not evenly represent the Pareto front, covering only a portion of it [5, 6].

To overcome this problem, in the present paper a modified Niched Pareto MOGA [6] is adopted to exploit its capability of evolving the population towards alternative, equivalent solutions of feature subsets which give a well distributed, representative description of the Pareto front of non-dominated solutions. This is achieved by applying a “niching pressure” in the parents selection step of the algorithm, such that those individuals with less crowded neighborhoods in the objective functions space are preferentially selected as parents, and thus allowed to create more offsprings in the following generations: this results in a population more evenly distributed in the objective functions space [6].

The proposed search scheme is compared with a standard Pareto-based MOGA in a task of classification of simulated transients in the feedwater system of a Boiling Water Reactor [7]. Haar wavelet decomposition [8] is applied to the transient signals for capturing their dynamic behaviour. This actually triplicates the number of features to be initially considered, significantly increasing the complexity of the problem.

The paper is organized as follows. In Section 2, the problem of feature selection is formulated and the techniques to approach it are summarized. In Section 3, a general description of the GA search approach is given with particular emphasis on the Niched Pareto scheme used in this work. In Section 4, the way GA search can be applied to the feature selection task for classification is illustrated. In Section 5, the nuclear case study is presented. Finally, some conclusions are drawn in the last Section.

2. FEATURE SELECTION FOR DIAGNOSTICS

From a mathematical point of view, a diagnostic classifier is a mapping function $\Phi()$ which assigns a vector of measured parameters $\vec{x}$ to a given fault class $i$. If the exact expression of $\Phi()$ were known, the question of which features of $\vec{x}$ to use is not of interest. In fact, adding features cannot degrade the diagnostic performance of $\Phi()$ and, hence, restricting $\vec{x}$ to a subset of features is never advised. However, in the engineering practice it is typically not possible to determine the analytical expression of the function $\Phi()$, due to the complexity of the system under analysis. Hence, one resorts to statistical or empirical
classification techniques in which the diagnostic classifier is built through an iterative process based on a set of examples (patterns) of input vectors \( \mathbf{x} \) labelled with the corresponding class \( i \).

In practical scenarios, at least four reasons call for a reduction in the number of features [9]. First of all, irrelevant, non-informative features result in a classifier model which is not robust. Second, studies have shown that for the success of the classification it is necessary to remove highly correlated features. Third, when the model handles many features, a large number of observation data is required to properly span the high-dimensional feature space for accurate multivariable interpolation. Finally, by eliminating unimportant sensors the cost and time of collecting the data can be reduced.

Feature extraction refers to the task of combining \( n \) features into \( m < n \) features with no or negligible loss of information [10]. All \( n \) measurements are used for extracting the \( m \) features. Feature selection is a special case of feature extraction whereby \( (n - m) \) of the original features are discarded as irrelevant.

For the purposes of classification, feature selection aims at finding a subset of the original features of a dataset such that the classification algorithm applied to the data represented by the selected features achieves the highest possible accuracy [2].

Several methods have been proposed to tackle feature selection. They can be classified into two categories: filter and wrapper methods [2]. In filter methods, the feature selector is independent of the specific algorithm used in the classification and it is used as a filter to discard irrelevant and/or redundant features a priori of the construction of the classification algorithm. A numerical evaluation function is used to compare the feature subsets with respect to the final diagnostic accuracy [10]. On the contrary, in wrapper methods the feature selector behaves as a wrapper around the specific algorithm used to construct the classifier. The feature subsets are compared using as criterium the diagnostic performance achieved by the classification algorithm itself [2].

Both the filter and the wrapper approaches conduct a search for an optimal feature subset in the space of the possible features. The inclusion or not of a feature in the subset can be encoded in terms of a binary variable which takes value 1 or 0, respectively. For \( n \) features, the size of the binary vector search space is \( 2^n \). An exhaustive search is impractical unless \( n \) is small. To overcome this problem, three search approaches are commonly adopted: complete, heuristic and probabilistic [2].

In the complete search approach, the properties of a pre-defined evaluation function are used to prune the feature space to a manageable size, thus avoiding the computational burden of an exhaustive search [11]. Only some evaluation functions give rise to a search that in theory guarantees the optimum feature subset selection without being exhaustive [12].

The heuristic approach does not guarantee that the best feature subset is achieved, but is less time consuming than the complete one and may be employed in combination with any evaluation function [13]. At present, the most popular heuristic methods are greedy search strategies such as the sequential forward selection method (SFS) or the sequential backward elimination method (SBE), both known as “hill climbing” methods, which iteratively add or subtract features. At each iteration the evaluation function is computed to compare
the successive subsets thereby obtained. The forward selection refers to a search that begins with no features and at each step a feature is added to the subspace; on the contrary, the backward elimination refers to a search that begins with the entire \(n\)-dimensional feature space and at each step a feature is removed. At each step, the choice of which feature to add or remove is driven by its effect on the evaluation function so as to climb toward its maximum value. As a remark, note that the hill climbing methods suffer from the so-called “nesting effect” [2]: if the features added cannot be removed, a local minimum of the evaluation function may be found.

The probabilistic approach is based on randomized search algorithms, primarily genetic algorithms, such as the one adopted in this paper, simulated annealing or tabu search algorithms [14].

The filter approach is generally computationally more efficient than the wrapper approach because for each feature subset of trial, the computation of an evaluation function from the available data is less time consuming than the development of a complete classification model. On the other hand, wrapper approaches are more performing since they ensure the selection of the features more suitable for the specific classification algorithm used in the diagnostic task. On the contrary, the filter approach is based on some a priori property of the features as measured on the available data by the evaluation function: in this sense, it totally ignores the effects of the selected feature subspace on the diagnostic performance of the induction algorithm that will be used for the classification.

3. MULTI-OBJECTIVE GENETIC ALGORITHMS: THE NICHED PARETO SEARCH SCHEME

Genetic Algorithms (GAs) are optimization methods aiming at finding the global optimum of a set of real objective functions, \(F=[f(i)]\), of one or more decision variables, \(U=[u]\), possibly subject to various linear or non-linear constraints.

After the pioneering theoretical work by John Holland [15], in the last decade a flourishing literature has been devoted to their application to real problems. The basics of the method may be found in [4]; some applications in various contexts are included in [16].

GAs owe their name to their operational similarities with the biological and behavioural phenomena of living beings. Correspondingly, the terminology adopted in GAs contains many terms borrowed from biology, suitably redefined to fit the algorithmic context. Thus, GAs operate on a set of (artificial) chromosomes, representing potential solutions of the optimization problem in terms of strings of numbers, generally sequences of binary digits 0 and 1. If the objective functions of the optimization has many arguments (typically called control factors or decision variables), each string is partitioned in as many substrings of assigned lengths, one for each argument and, correspondingly, we say that each chromosome is partitioned in (artificial) genes. The genes constitute the so called genotype of the chromosome and the substrings, when decoded in real numbers, constitute its phenotype. When the objective functions are evaluated in correspondence of a set of values of the control factors of a chromosome, its values are called the fitness of
that chromosome. Thus, each chromosome gives rise to a trial solution to the problem at hand in terms of a set of values of its control factors.

The GA search is performed by constructing a sequence of populations of \( n_p \) chromosomes, the individuals of each population being the children of those of the previous population and the parents of those of the successive population. The initial population is generated by randomly sampling the bits of all the strings. At each step, the new population is then obtained by manipulating the strings of the old population in order to arrive at a new population hopefully characterized by increased mean fitness. This sequence continues until a termination criterion is reached. As for the natural selection, the string manipulation consists in selecting and mating pairs of chromosomes in order to groom chromosomes of the next population. This is done by repeatedly performing on the strings the four fundamental operations of reproduction, crossover, replacement and mutation, all based on random sampling: the parents’ selection step determines the individuals which participate in the reproduction phase; reproduction itself allows the exchange of already existing genes whereas mutation introduces new genetic material; the substitution defines the individuals for the next population. This way of proceeding enables to efficiently arrive at optimal or near-optimal solutions.

With regards to their performance, it is acknowledged that GAs take a more global view of the search space than many other optimization methods. The main advantages are i) fast convergence to near global optimum, ii) superior global searching capability in complicated search spaces, iii) applicability even when gradient information is not readily achievable.

### 3.1 Multi-Objective optimization

In a multi-objective optimization problem, several possibly conflicting objective functions \( f_p() \), \( p = 1, 2, \ldots, n_f \), must be evaluated in correspondence of each decision variable vector \( U \) in the search space. The goal is to identify the solution vector \( U^* \) which gives rise to the best compromise among the various objective functions. The comparison of solutions is achieved in terms of the concepts of Pareto optimality and dominance [4, 17]: with reference to a maximization problem, solution \( U_a \) is said to dominate \( U_b \) if

\[
\forall p \in \{1, 2, \ldots, n_f\}, f_p(U_a) \geq f_p(U_b)
\]

and

\[
\exists j \in \{1, 2, \ldots, n_f\}, f_j(U_a) > f_j(U_b).
\]

The decision variable vectors which are not dominated by any other of a given set are called nondominated with respect to this set; the decision variable vectors that are nondominated within the entire search space are said to be Pareto optimal and constitute the so called Pareto optimal set or Pareto optimal front.

When tackling a multi-objective problem by GAs, the various approaches to fitness definition may be distinguished into three categories [18]:

- Aggregation methods combine the multiple objectives of the optimization into a scalar fitness function that is used to evaluate the goodness of a solution; an example is represented by the weighted-sum
approach [19, 20], in which the fitness of solution \( U, f(U) \), is computed by the following weighted sum of the individual \( n_f \) optimization objectives:

\[
f(U) = \sum_{p=1}^{n_f} w_p \cdot f_p(U)
\]

(2)

where the arbitrary constant weights \( w_p, p = 1, 2, \ldots, n_f \), satisfy the following relations:

\[
w_p \in [0,1] \text{ and } \sum_{p=1}^{n_f} w_p = 1
\]

(3)

The optimization of a single fitness function, combination of the \( n_f \) objectives has the advantage of producing a single compromise solution, requiring no further selection by the decision maker. However, if the solution were found a posteriori not acceptable as a good compromise of the decision maker preferences, tuning of the aggregating weights may be required, followed by new runs of the optimizer, until a suitable solution is found.

- Population-based non-Pareto approaches are able to evolve multiple nondominated solutions concurrently in a single simulation run: for instance, in [21] sub-populations of the next generation are reproduced from the current population separately for each of the objectives; then, the overall population at each generation is formed by merging and shuffling the sub-populations. The downside of this method is that it achieves a population of individuals that perform well for each objective separately, with no consideration given to trade-offs among them.

- In Pareto-based methods [4], the chromosomes of a population are ranked according to the Pareto dominance criterion applied to the fitnesses \( f_p(U), p = 1, 2, \ldots, n_f \). Firstly, all nondominated individuals are identified and rank 1 is assigned to them. Then, these solutions are virtually removed from the population and the next set of nondominated individuals are identified and assigned rank 2; this process continues until every solution in the population has been ranked. Every solution belonging to the same rank class is Pareto-equivalent to any other of the same class and has the same probability of the others to be selected as a parent for the mating.

During the optimization search, an archive of solution vectors \( U \), each one constituted by a nondominated chromosome and by the corresponding \( n_f \) fitnesses, representing the dynamic Pareto optimality set is recorded and updated [5, 22, 23]. This procedure also allows implementation of elitism in the genetic algorithm: every individual in the archive (or a pre-established number of individuals) is chosen once as a parent in each generation to guarantee a better propagation of the genetic code of nondominated solutions and a more efficient evolution of the population towards Pareto optimality.

At the end of the search procedure the result of the optimization is constituted by the archive itself which hopefully gives the Pareto-optimal set.

The performance of a Pareto-based MOGA depends largely on its ability to maintain genetic diversity through the generations so as to arrive at a population of individuals which uniformly represent the real nondominated solutions of the Pareto set [1, 4]. This can be achieved by resorting to niching techniques such as sharing [4].
In this work, a Niched Pareto GA with random sampling tournament selection is employed within a multi-objective Pareto-based scheme [6].

3.2 The Niched Pareto Genetic Algorithm (NPGA) with random sampling tournament selection

In this Section, the Niched Pareto-based random sampling tournament selection procedure is adopted for selecting from the population the individuals that are going to reproduce [6].

The procedure (Figure 1) is based on the random sampling of two groups of individuals from the entire population. The first one is named dominance tournament group and contains \( n_t \) chromosomes which are the candidates for selection as parents whereas the second one, named dominance tournament sampling group and made of \( n_s \) chromosomes, is used for comparison of the individuals of the first group with respect to dominance.

Each of the \( n_t \) individuals in the dominance tournament group is tested for domination against all the \( n_s \) individuals in the dominance tournament sampling set. Three different situations may occur:

a) only one of the individuals in the dominance tournament group is non-dominated by all the individuals in the dominance tournament sampling set (Figure 2). In this case, the non-dominated individual is selected for reproduction;

b) all individuals in the dominance tournament group are dominated by individuals in the dominance tournament sampling set (Figure 3, left);
c) at least two of the individuals in the dominance tournament group are non-dominated (Figure 3, right).

In cases b) and c) the individual which best seems to maintain diversity is selected for reproduction by using the equivalence class sharing method [6]. This method is based on the selection of the individual with the smallest niche count (see definition below) between all the individuals of the tournament group in case b) and all the non-dominated individuals in case c).

![Figure 2. NPGA domination tournament for selection in a bidimensional objective space where $f_1$ has to be maximized and $f_2$ minimized; only candidate 2 of the dominance tournament group is non-dominated by the individuals of the dominance tournament sampling set and thus it is selected for reproduction.](image)

![Figure 3. NPGA domination tournaments for selection in a bidimensional objective space where $f_1$ has to be maximized and $f_2$ minimized; the candidate with the smallest niche count (candidate 1) is selected as a parent if either all the candidates are dominated (left) or at least two of them are non-dominated (right) (equivalence class sharing method).](image)

The niche count $m_l$ of the $l$-th individual in the tournament group, $l = 1, 2, \ldots, n_p$ is an estimate of how crowded its neighbourhood (niche) is. It is calculated over all the $n_p$ individuals in the current population:
where \( d_{lj} \) is the distance, either in the genotype or phenotype spaces (in the latter case, with respect to either the decision variables or the fitness functions), between the \( l \)-th candidate for selection and the \( j \)-th individual in the population and \( s(d_{lj}) \) is the sharing function. This is a function decreasing with \( d_{lj} \) and such that \( s(0)=1 \) and \( s(d_{lj})=0 \) for \( d_{lj} \geq \sigma_s \), where \( \sigma_s \) is the niche radius, i.e. the distance threshold below which two individuals are considered similar enough to affect the niche count. Typically, a triangular sharing function is used such that \( s(d_{lj}) = 1 - d_{lj} / \sigma_s \) for \( d_{lj} \leq \sigma_s \) and \( s(d_{lj}) = 0 \) for \( d_{lj} > \sigma_s \).

In the application of the method to feature selection for classification, the concern is in finding non-dominated feature subset solutions which differ with respect to the values of the \( n_f = 2 \) objective functions considered, i.e. classification accuracy and number of features. Hence, it makes sense to perform the computation of the distance and niche count in the fitness functions (phenotype) space, as shown in Figure 3, rather than in the space of the decision variables, i.e. the individual features [6, 18 and 24]. To define a meaningful metric in the fitness space, at each generation the objective functions are scaled to the same numerical range \([0,1]\) as follows [4, 18]:

\[
f'(p)(U_j) = \frac{f_p(U_j) - \min_{j=1,...,n} \{f_p(U_j)\}}{\max_{j=1,...,n} \{f_p(U_j)\} - \min_{j=1,...,n} \{f_p(U_j)\}}
\]

where \( f_p(U_j) \) is the value of the \( p \)-th fitness of the \( l \)-th individual in the current population and \( f'(p)(U_j) \) is the corresponding normalized value. Then, the Euclidean distances between individuals in the normalized two-dimensional fitness space are used to compute the niche count.

After performing twice the random sampling tournament for the selection of a pair of parents, and before proceeding to the selection of the next pair, the two selected parents are combined by crossover and the two resulting offsprings are replaced in the population according to the adopted replacement procedure. Then, the population ranking is updated and it is possible to proceed to selecting the successive pair of parents. By so doing, the sampling is performed on a dynamically varying population, with benefits for the convergence towards a diverse Pareto set [24].

4. GENETIC ALGORITHMS FOR FEATURE SELECTION

Given \( n \) features, the problem of selecting a subset of \( m \) relevant ones illustrated in Section 2 can be formulated as an optimization problem. In this view, given a set \( A \) of \( n \)-dimensional input patterns, a GA can be devised to find an optimal binary transformation vector \( V_f \), of dimension \( n \), which maximizes/minimizes a set of optimization criteria, i.e. the objective functions. Let \( m \) be the number of 1’s in \( V_f \) and \( n - m \) the number of 0’s. Then, a modified set of patterns \( B = V_f(A) \) is obtained in an \( m \)-dimensional space \((m < n)\). Figure 4 shows the structure of a multi-objective GA-feature extractor that uses the final
classification accuracy (to be maximized) and the dimension $m$ of the transformed patterns (to be minimized) as optimization criteria. The GA creates a population of competing transformation vectors $V_l$, $l = 1, 2, \ldots$, which are evaluated as follows [25]:

i. The vector $V_l$ is applied to each pattern of set $A$, giving a modified pattern which is then sent in input to the classifier.

ii. The set $B$ of modified patterns thereby obtained is divided into a training set, used to train the classifier, and a testing set, used to evaluate the classification accuracy on new patterns.

iii. The classification accuracy obtained and the number of selected features, $m$, are used by the GA as a measure of the goodness of the transformation vector $V_l$ used to obtain the set of transformed patterns.

iv. On the basis of this feedback, the GA conducts its search for a vector or a set of vectors which give rise to the best compromise among classification accuracy and parsimony in the selection of features.

For the GA feature selector of Figure 4, the organization of the chromosome is quite straightforward [26]: each bit of the chromosome is associated with a parameter (Figure 5) and interpreted such that if the $i$-th bit equals 1, then the $i$-th parameter is included as feature in the pattern for classification or viceversa if the bit is 0. Note that contrary to other GA applications, in this case of feature selection the binary chromosome does not encode real-valued control factors: the information regarding the features presence or absence in the optimal set for classification is included in the bit values themselves so that no decoding is necessary.

Concerning the fitness function of classification accuracy, each subset of features encoded in a chromosome is evaluated on a set of testing data using a fast-running nearest neighbor classifier. More specifically, in the applications which follow, the total number of pre-labelled patterns available is randomly subdivided into training and test sets consisting of 75% and 25% of the data, respectively. The Fuzzy K-Nearest Neighbor algorithm (FKNN) [3], with $K = 5$, has been applied to classify the test data on the basis of the location of the labelled training data. The obtained fuzzy partition of the test data $\{\mu_i(\tilde{x})\}$, where $0 \leq \mu_i(\tilde{x}) \leq 1$ is the membership function of pattern $\tilde{x}$ to class $i$, is converted into a hard partition by assigning each pattern to the class with highest membership value. The random subdivision of the available patterns in training and test sets is repeated 10 times (10 cross-validation): the mean recognition rate, i.e. the average fraction of correct classifications over the 10 cross-validation tests [27], is calculated and sent back to the GA as the fitness value of classification accuracy of the transformation chromosome used to produce the transformed set of patterns $B$. 
5. CLASSIFICATION OF TRANSIENTS IN THE FEEDWATER SYSTEM OF A BOILING WATER REACTOR

The diagnostic problem considered is the early identification of a predefined set of faults in the feedwater system of a Boiling Water Reactor (BWR). The corresponding transients have been simulated by the HAMBO simulator of the Forsmark 3 BWR plant in Sweden [7]. Appendix A shows a sketch of the system [7]. The considered faults occur in the section of the feedwater system 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 [28]. For demonstration purposes, it was decided to limit the diagnosis to only one power operation level, i.e. 50% of full power.

The faults may be divided into three categories:

- F1-F5, F7 regard line 1 of the feedwater system.
- F11-F15, F17 regard line 2 of the feedwater system.
- F6, F8, F9, F10, F16, F18 regard both lines.

Five transients were simulated for each of the 18 faults, for varying degrees of leakage and valve closures and with step and ramp changes at different leak sizes. The data relative to 363 plant parameters were recorded with a sampling frequency of 1 Hz. All transients start after 60 seconds of steady state operation. From the analysis developed in [29], it is clear that faults 6, 10, 16 have no significant consequence on the plant measured parameters because the size of the leakage considered is too small. Hence, these faults were not considered further. Moreover, given that the ramp changes cause variations of the parameters later than the step changes, only the three step changes for each fault were considered.

For the fault classification purpose of this work, the number of parameters was reduced to 123 by combining redundant measurements of the same physical quantity and by eliminating those parameters linearly correlated or carrying basically the same time evolution in all the different faults. Compact wavelet features are then extracted from the 123 measured signals by Haar wavelet decomposition from a sliding window on the actual signal time-series [8]. The selected wavelet features [30] are: the mean residual signal taken at the highest, i.e. coarsest, scale and the minimum and maximum wavelet coefficients over all the scales. The rationale behind this choice is that the first wavelet feature captures the general trend of the signal across the windows in a compact way, being very much related to the average signal value within the analysis window, while the minimum and maximum wavelet coefficients capture important variations in the signal within a single window which would otherwise be severely smoothed out by the compression process. In particular, the maximum coefficient reflects negative trends, step changes and the negative component of spikes, whereas the minimum coefficient reflects the positive trends, step changes and positive components of spikes. The window size is selected so as to correspond to wavelet dyadic decomposition values (i.e. powers of 2) and consecutive windows are chosen with a slight overlap to avoid missing features that might be present at the window borders. Because of its ability of continuously applying the wavelet transform on a sliding window, and since the transform is used as a pre-processing step for the final transient classification, this technique has been named Wavelet On-Line Pre-processing (WOLP) [30].

In a compromise between a high level of transient compression and an acceptable resolution, time windows which are 16 patterns long, with an overlap of 6 seconds, have been chosen. Consequently, the evolution of anyone of the wavelet features in a given transient from t= 58s to t=133s is summarized in 7 points each one representing the segment of dynamics in one time-window. Thus, the application of the WOLP pre-processing on the 123 original plant measured signals generates 369 wavelet coefficients, increasing from $2^{123}$ to $2^{369}$ the dimension of the search space from which the optimal subset of features relevant for the fault classification task is to be selected.
5.1 Comparison between the standard Pareto and the Niched Pareto multi-objective GA searches

As we shall see, given the large number of possible solutions \(2^{369}\), the task of maintaining genetic diversity in the population in order to explore more accurately the search space is crucial for achieving a well-distributed, representative Pareto set.

A standard Pareo-based approach is first applied with the parameters and search strategy reported in Table 1 [31].

<table>
<thead>
<tr>
<th>Selection procedure</th>
<th>fit-fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replacement procedure</td>
<td>50% of the cases fittest, 50% weakest</td>
</tr>
<tr>
<td>Probability of mutation</td>
<td>0.008</td>
</tr>
<tr>
<td>Number of generations</td>
<td>1000</td>
</tr>
<tr>
<td>Population size</td>
<td>200</td>
</tr>
</tbody>
</table>

Figure 6 shows the last population and the Pareto optimal set thereby obtained.

In a single run, the standard Pareto-based MOGA identifies a range of nondominated solutions with different trade-offs of classification performance (FKNN mean recognition rate) and complexity (number of features).

However, the feature selector shows difficulties in exploring the solution space in the region with number of features \(m < 10\): individuals with \(m = 4, 5, 6, 7\) present unsatisfactory recognition rates and individuals with \(m = 0, 1, 2, 3\) are not even found.
The Niched Pareto-based approach of Section 3.2 is then investigated to improve the uniformity of coverage of the Pareto front by the optimal feature subsets at convergence. The following set of parameters has turned out by crude search to give the best results in terms of both classification accuracies and coverage and distribution of the individuals on the Pareto front:

- NPGA best configuration: \(\{n_t = 4; n_s = 20; \sigma_s = 0.1\}\);

The population size \((n_p)\), the number of generations \((n_g)\) and the probability of mutation \((P_m)\) are set respectively to 200, 1000 and 0.008.

The characteristics of the Pareto front found are summarized in Table 2.

<table>
<thead>
<tr>
<th>Number of features in the transformed patterns, (m)</th>
<th>Mean recognition rate</th>
<th>Feature subsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4248</td>
<td>(S_1^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>2</td>
<td>0.8010</td>
<td>(S_2^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>3</td>
<td>0.8733</td>
<td>(S_3^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>4</td>
<td>0.9030</td>
<td>(S_4^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>5</td>
<td>0.9122</td>
<td>(S_5^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>6</td>
<td>0.9174</td>
<td>(S_6^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>7</td>
<td>0.9199</td>
<td>(S_7^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>8</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>9</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>10</td>
<td>0.9268</td>
<td>(S_{10}^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>11</td>
<td>0.9327</td>
<td>(S_{11}^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>12</td>
<td>0.9356</td>
<td>(S_{12}^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
<tr>
<td>13</td>
<td>0.9375</td>
<td>(S_{13}^{\text{opt}, \text{Pareto, Wolp}})</td>
</tr>
</tbody>
</table>
Table 2. Pareto front found by the Niched Pareto Genetic Algorithm at convergence

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0.9383</td>
<td>$S_{14}^{\text{opt, Pareto, wlp}}$</td>
</tr>
<tr>
<td>15</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>16</td>
<td>0.9387</td>
<td>$S_{16}^{\text{opt, Pareto, wlp}}$</td>
</tr>
<tr>
<td>17</td>
<td>0.9391</td>
<td>$S_{17}^{\text{opt, Pareto, wlp}}$</td>
</tr>
<tr>
<td>18</td>
<td>0.9437</td>
<td>$S_{18}^{\text{opt, Pareto, wlp}}$</td>
</tr>
<tr>
<td>19</td>
<td>0.9491</td>
<td>$S_{19}^{\text{opt, Pareto, wlp}}$</td>
</tr>
<tr>
<td>20</td>
<td>0.9516</td>
<td>$S_{20}^{\text{opt, Pareto, wlp}}$</td>
</tr>
<tr>
<td>21</td>
<td>0.9527</td>
<td>$S_{21}^{\text{opt, Pareto, wlp}}$</td>
</tr>
<tr>
<td>22</td>
<td>0.9549</td>
<td>$S_{22}^{\text{opt, Pareto, wlp}}$</td>
</tr>
</tbody>
</table>

The features included in the nondominated sets of Table 2 are reported in Appendix B, with respect to their numbering and description in Appendix C.

Based on these results, an informed choice has then to be made on the features to be actually monitored for the diagnostic task, by considering the diagnostic classification accuracy objective and also the practical issues of costs and ease of data acquisition related to complexity. The analyst, provided with the whole spectrum of performances with respect to the objectives, must ultimately select the preferred solution according to some subjective preference values. Thus, the closure of the problem must still rely on techniques of decision analysis such as utility theory, multi-attribute value theory or fuzzy decision making, to name a few. In alternative, a popular criterion for choosing a “single best compromise solution” is the so called “min-max method” [32]. Let $Q = \{f_1, f_2, \ldots, f_n\}$ denote a generic point on the $n$-dimensional Pareto surface, and $f_p^{\text{max}}, p=1, 2, \ldots, n_P$, the maximum value of the $p$-th objective function on such surface. For each point $Q$ we calculate the relative deviations $z_p = (f_p^{\text{max}} - f_p)/f_p^{\text{max}}, p=1, 2, \ldots, n_P$ and take as a representative value $z_Q = \min_p (z_p)$. By definition, the best compromise solution is the point $Q^*$ on the Pareto surface corresponding to the maximum $z_Q$.

The application of the min-max method to the Pareto Surface obtained in the present case study select the feature subset $S^*_3$ as the optimal compromise solution. A set of patterns taken from transients at 80% power level, never employed during the feature selection phase, has then been used to cross-validate the performance of the FKNN classifier resting upon the optimal compromise feature subset $S^*_3$. The resulting recognition rate of $0.9060 \pm 0.0301$ is very satisfactory.
Figure 7 shows the comparison between the Pareto sets obtained by the Niched Pareto and the previous standard Pareto-based MOGA.

![Figure 7](image-url)

**Figure 7. Comparison between the Pareto optimal fronts obtained by the Niched Pareto (o) and the standard Pareto-based MOGA (◊)**

The niching “pressure” applied by the equivalence class sharing method succeeds in spreading the population out along the Pareto optimal front: indeed, the NPGA Pareto solutions cover from $m = 0$ to $m = 22$, missing only individuals with $m = 8, 9, 15$; on the contrary, the standard Pareto-based MOGA front extends from $m = 4$ to $m = 25$, with elements with $m = 15, 18, 19, 22-24$ missing. Moreover, all the elements of the NPGA set have larger recognition rates than those of the set found by the standard Pareto-based MOGA, with particularly significant differences for $m = 4-7$. These results prove that the Pareto domination tournament and equivalence class sharing are efficient in preserving good individuals and maintaining genetic diversity in the population throughout. This combined action leads to a dense and uniform distribution of the final population along a well distributed Pareto optimal set, as shown in Figure 8 (to be compared with Figure 6).

![Figure 8](image-url)

**Figure 8. Pareto front and final population found by the NPGA at convergence**
The performance of the NPGA on the feature selection task proposed, in terms of the structure of the Pareto front and of the convergence of the population towards it, depends on the number of individuals in the dominance tournament group, \( n_t \), the number of individuals in the dominance tournament sampling set, \( n_s \), and the niche radius, \( \sigma_s \).

In [1] and [6], a tournament group of two candidates is suggested; however, higher values of \( n_t \) in principle sample the population more deeply, increasing the probability to find a candidate which is non-dominated by the comparison set and hence pressing the algorithm to move faster towards the Pareto front. This has been verified by a thorough parametric analysis with respect to \( n_t \) of which for brevity sake we only report the results for \( n_t = 2 \) and \( n_t = 5 \) in Figure 9. The analysis also shows that, if the size of the tournament group \( n_t \) exceeds 1/4 or 1/5 of \( n_p \), the population tends to prematurely converge to a small portion of the Pareto front.

![Figure 9. Evolution of the maximum recognition rate (left) and of the minimum number of selected features (right) in the individuals of the Pareto front for \( n_t = 2 \) and \( n_t = 5 \)](image)

The dimension of the dominance tournament sampling set, \( n_s \), gives control over the “selection pressure”: in general, increasing the relative size of the dominance tournament sampling set seems to increase pressure towards the front. The following empirically-derived guidelines suggested in [6] have been confirmed by experimentation on the present feature selection task for fault diagnosis:

a) too little dominance pressure (e.g. \( n_s \approx 0.01n_p \), in Figure 10, top) limits the capability of the algorithm in forcing the population towards the Pareto front, thus producing a quite scattered final population with a limited extension of the Pareto front identified (the Pareto solutions range from \( m = 3 \) to \( m = 14 \));

b) for \( n_s \approx 0.1-0.2n_p \) (Figure 10, middle), a quite dense and complete distribution of the population approaches a well distributed and extended Pareto front (the solutions range from \( m = 0 \) to \( m = 22 \));
c) for higher domination pressure (e.g. \( n_s > 0.2n_p \) in Figure 10, bottom) the population moves more effectively towards the Pareto optimal front, so that almost all its individuals are uniformly distributed over it; but it also seems that the exploration of the search space is somewhat curtailed: in the case of Figure 10, bottom, individuals with \( m = 0-4 \) are not found, whereas those with \( m = 5-9 \) bear a worse classification accuracy than the corresponding ones of cases a) and b) above; only for the solutions with \( m \) ranging from 10 to 23 the algorithm seems to have deeply searched the space, finding individuals with satisfactory recognition rates, comparable with those in a) and b).

![Figure 10. Pareto front and final population for different sizes of the dominance tournament sampling set, \( n_s \).](image)

a): \( n_s \approx 0.01n_p \) (top);  b): \( n_s \approx 0.1n_p \) (middle);  c): \( n_s > 0.2n_p \) (bottom)

Finally, the niche radius \( \sigma_s \) can be interpreted as a qualitative measure of the minimal separation desired or expected between optimal solutions on the Pareto front [24]. As a consequence, for a given Pareto front, the number of distinct solutions that form it and whose niches do not interfere with one another increases continuously as the dimension of the niche decreases: the smaller is the niche radius, the denser is the coverage of the Pareto front. In our case, since the number of selected features, \( m \), is an integer, the distances between Pareto solutions along this axis in the objective functions space can only assume discrete values, so that the effect of \( \sigma_s \) on the performance of the NPGA remains hidden. Yet, an important consideration about the effect of the niche dimension on the evolution of the population and on the final Pareto front can still be drawn: large niches tend to perform a more sparse and thus ineffective sampling of the population. To explain this pictorially, Figure 11, left, shows that although the actual neighbourhood situation of the two candidates 1 and 2 is quite different, if the niches are big their counts \( m_1 \) and \( m_2 \) may be almost the same because candidate 2 has even more individuals in its niche, although they are more distant. On the contrary, small niches are more selective as shown pictorially in Figure 11, right: the niche count associated to candidate 1, with more closer neighbours, is higher.
6. CONCLUSIONS

A Niched Pareto-optimal tournament selection genetic algorithm search has been embraced for the selection of the features relevant to a nuclear transient classification task. Two objectives, i.e. the maximization of the classification accuracy in terms of FKNN mean recognition rate and the minimization of the number of features forming the subsets, have been used to drive the algorithm towards the identification of a representative and evenly distributed set of alternative, equivalent feature sets offering different trade-offs in terms of diagnostic power and complexity.

The Niched Pareto GA has been tested and compared to a standard MOGA, with respect to its ability of maintaining genetic diversity by means of niches during the search and with particular emphasis on its capability of finding feature sets made of a low number of features. The case study considered for the analysis has concerned the problem of selecting among \( n = 369 \) features those relevant for diagnosing transients in the feedwater system of a Boiling Water Reactor. The 369 features considered are the compact wavelet coefficients extracted by Haar decomposition from 123 measured signals and capture the dynamic behaviour of the system during the transients. This leads to an increase of the dimension of the search space from \( 2^{123} \) to \( 2^{369} \).

The results prove that the Niched Pareto approach is more effective than the standard Pareto-based MOGA in the selection of features in a high-dimensional space. In the case study analyzed, the number of features contained at convergence in the NPGA Pareto solutions ranges from \( m = 0 \) to \( m = 22 \) and the corresponding recognition rates from 0.0667 to 0.9549 whereas for the standard Pareto-based MOGA \( m \) goes from 4 to 25 with recognition rates from 0.7506 to 0.9329. Thus, the NPGA is superior in producing a diverse set of solutions with differing performance versus complexity trade-off characteristics. This is achieved by applying a niching pressure to spread the population out along the Pareto front. Moreover, the incorporation of the concept of Pareto domination in the selection operator allows the algorithm to preserve good individuals, improving the propagation of the genetic code.

A key advantage of the Niched Pareto GA is the possibility to regulate the domination pressure and, hence, the speed of convergence by means of the parameters \( n_s \) and \( n_t \) which represent the dimensions of the dominance tournament sampling set and tournament group, respectively. A parametric analysis has
confirmed the findings reported in literature, that $n_s$ should be about 10-20 % of the population size $n_p$ to obtain a dense and complete distribution of the population on the Pareto front. Concerning $n_s$, an increasing value of this parameter forces the algorithm to move faster towards the Pareto set, whereas if it exceeds 1/4 or 1/5 of $n_s$, the population tends to prematurely converge to a small portion of it.

Acknowledgements

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

7. REFERENCES


8. **APPENDIX A: SKETCH OF THE FEEDWATER SYSTEM OBJECT OF THE CASE STUDY IN SECTION 5**
9. **APPENDIX B: FEATURES IN THE PARETO SET**

- \( S_{1, \text{opt, Pareto, wlp}} = \{321\} \);
- \( S_{2, \text{opt, Pareto, wlp}} = S_{1, \text{opt, Pareto, wlp}} + \{320\} \);
- \( S_{3, \text{opt, Pareto, wlp}} = S_{2, \text{opt, Pareto, wlp}} + \{357\} \);
- \( S_{4, \text{opt, Pareto, wlp}} = S_{3, \text{opt, Pareto, wlp}} + \{194\} \);
- \( S_{5, \text{opt, Pareto, wlp}} = S_{4, \text{opt, Pareto, wlp}} + \{325\} \);
- \( S_{6, \text{opt, Pareto, wlp}} = S_{5, \text{opt, Pareto, wlp}} + \{324\} \);
- \( S_{7, \text{opt, Pareto, wlp}} = S_{6, \text{opt, Pareto, wlp}} + \{72\} \);
- \( S_{10, \text{opt, Pareto, wlp}} = \{160, 194*, 194**, 195, 242, 314, 320, 321, 338, 357\} \);
- \( S_{11, \text{opt, Pareto, wlp}} = S_{10, \text{opt, Pareto, wlp}} + \{100\} \);
- \( S_{12, \text{opt, Pareto, wlp}} = \{74, 76, 100, 160, 194*, 195, 242, 314, 320, 321, 338, 357\} \);
- \( S_{13, \text{opt, Pareto, wlp}} = S_{12, \text{opt, Pareto, wlp}} + \{141\} \);
- \( S_{14, \text{opt, Pareto, wlp}} = S_{13, \text{opt, Pareto, wlp}} + \{191*\} \);
- \( S_{16, \text{opt, Pareto, wlp}} = \{68, 74*, 76, 100, 191*, 194, 194*, 195, 229, 242, 251, 314, 320, 321, 324, 325\} \);
- \( S_{17, \text{opt, Pareto, wlp}} = \{14*, 68, 74*, 76, 100, 191**, 194, 194*, 195, 229, 242, 251, 314, 320, 321, 324, 325\} \);
- \( S_{21, \text{opt, Pareto, wlp}} = S_{20, \text{opt, Pareto, wlp}} + \{73\} \);

---

1 No asterisk: mean residual
* : minimum wavelet coefficient
** : maximum wavelet coefficient
10. **APPENDIX C: FEATURE NUMBER AND NAME WITH REFERENCE TO THE CASE STUDY OF SECTION 5**

<table>
<thead>
<tr>
<th>Feature number</th>
<th>Name</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
<td>Temperature high-pressure drain</td>
<td>°C</td>
</tr>
<tr>
<td>60</td>
<td>Temperature after EA1 (high pressure preheater A1)</td>
<td>MPa</td>
</tr>
<tr>
<td>73</td>
<td>Pressure drain 6 before VB1</td>
<td>MPa</td>
</tr>
<tr>
<td>74</td>
<td>Pressure drain 5 before VB2</td>
<td>MPa</td>
</tr>
<tr>
<td>75</td>
<td>Temperature drain 6 after VB1</td>
<td>°C</td>
</tr>
<tr>
<td>76</td>
<td>Temperature drain 5 after VB2</td>
<td>°C</td>
</tr>
<tr>
<td>77</td>
<td>Temperature drain 4 before VB3</td>
<td>°C</td>
</tr>
<tr>
<td>100</td>
<td>Pressure before EA1 (preheater A1)</td>
<td>MPa</td>
</tr>
<tr>
<td>107</td>
<td>Temperature after 424EA1 (preheater A1)</td>
<td>°C</td>
</tr>
<tr>
<td>109</td>
<td>Temperature after EA1 train A (preheater A1)</td>
<td>°C</td>
</tr>
<tr>
<td>139</td>
<td>Temperature of condensate after EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>140</td>
<td>Temperature of condensate after EB2 train A</td>
<td>°C</td>
</tr>
<tr>
<td>143</td>
<td>Temperature of condensate after EB3 train B</td>
<td>°C</td>
</tr>
<tr>
<td>155</td>
<td>Temperature of bearing, drive end PB1.M1</td>
<td>°C</td>
</tr>
<tr>
<td>157</td>
<td>Temperature of bearing, drive end PB2.M1</td>
<td>°C</td>
</tr>
<tr>
<td>160</td>
<td>Water level tank TD1</td>
<td>m</td>
</tr>
<tr>
<td>165</td>
<td>Temperature of axial bearing PC3 (pump 3)</td>
<td>°C</td>
</tr>
<tr>
<td>168</td>
<td>Temperature seal water to PC3.E1</td>
<td>°C</td>
</tr>
<tr>
<td>191</td>
<td>Water level in EB1 train B (preheater B1)</td>
<td>m</td>
</tr>
<tr>
<td>192</td>
<td>Water level in EA2 train A (preheater A2)</td>
<td>m</td>
</tr>
<tr>
<td>193</td>
<td>Water level in EB2 train B (preheater B2)</td>
<td>m</td>
</tr>
<tr>
<td>194</td>
<td>Temperature feedwater before EA2 train A</td>
<td>°C</td>
</tr>
<tr>
<td>195</td>
<td>Temperature feedwater after EA2 train A</td>
<td>°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>--------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>196</td>
<td>Temperature of radial bearing PA1.P1 (pump A1)</td>
<td>°C</td>
</tr>
<tr>
<td>197</td>
<td>Temperature of radial bearing PA1.P1 (pump A1)</td>
<td>°C</td>
</tr>
<tr>
<td>223</td>
<td>Vibration PA1.P1 outer vertical</td>
<td>mm/s</td>
</tr>
<tr>
<td>229</td>
<td>Pressure after pump PB1</td>
<td>MPa</td>
</tr>
<tr>
<td>241</td>
<td>Temperature feedwater before EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>242</td>
<td>Temperature feedwater after EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>281</td>
<td>Temperature of radial bearing PD1.P1 (pump D1)</td>
<td>°C</td>
</tr>
<tr>
<td>320</td>
<td>Position level control valve for EA1</td>
<td>%</td>
</tr>
<tr>
<td>321</td>
<td>Position level control valve for EB1</td>
<td>%</td>
</tr>
<tr>
<td>324</td>
<td>Position level control valve before EA2</td>
<td>%</td>
</tr>
<tr>
<td>325</td>
<td>Position emergency drain valve EB2</td>
<td>%</td>
</tr>
<tr>
<td>330</td>
<td>Position valve for level I EB4</td>
<td>%</td>
</tr>
<tr>
<td>332</td>
<td>Position drain valve TD1</td>
<td>%</td>
</tr>
<tr>
<td>342</td>
<td>Position valve in drain for TB1</td>
<td>%</td>
</tr>
<tr>
<td>344</td>
<td>Position valve in drain for TB2</td>
<td>%</td>
</tr>
<tr>
<td>357</td>
<td>Position steam stop control valve</td>
<td>%</td>
</tr>
<tr>
<td>359</td>
<td>Position steam stop control valve</td>
<td>%</td>
</tr>
</tbody>
</table>
Evolutionary fuzzy clustering for the classification of transients in nuclear components

Enrico Zio and Piero Baraldi

Progress of Nuclear Energy, Vol. 46, No. 3-4, pp. 282-296
2005
PART III

Selected papers
EVOLUTIONARY FUZZY CLUSTERING FOR THE CLASSIFICATION OF TRANSIENTS IN NUCLEAR COMPONENTS

E. Zio, P. Baraldi

ABSTRACT

The classification of objects or patterns is an important area of research with practical applications in a variety of fields. In this paper we are interested in the classification of signal transients for the reliable monitoring and timely diagnosing of nuclear components and systems. These represent fundamental tasks for the operation, control and accident management of nuclear power plants. The problem is tackled within a fuzzy clustering approach. The choice of the metrics upon which the clustering is based is critical for obtaining geometric clusters in the features space as close as possible to the real physical classes. In this respect, here the a priori known information regarding the true classes to which the objects belong will be exploited to select, by means of an evolutionary algorithm of literature, an optimal metrics for the clustering. In case the classification thereby obtained were still unsatisfactory, an iterative procedure is used to split the less compact physical classes in further subclasses.

KEYWORDS

Transient classification; fuzzy clustering; evolutionary optimization algorithm; steam generator

1. INTRODUCTION

The early identification of the causes for the onset of a meaningful departure from steady state behaviour is an essential step for the operation, control and accident management of nuclear power plants. The basis for the identification of a change in the system behaviour is that different system faults and anomalies lead to different patterns of evolution of the involved process variables. Given the safety and economical importance of the problem, several approaches for signal trend identification have been investigated and many efforts are continuously devoted to the improvement of the results thus far obtained.
The approach to the classification of the causes responsible of a pre-defined set of transients is typically performed through a sequence of steps which include the selection of the signals relevant to the classification, the validation and reconciliation (when needed) of the measured signals, the reduction of the dimensionality of the signals space by means of a process of features extraction, the identification of the clusters in the features space associated to the different physical behaviours, e.g. in our case the different causes of transients. Further, in practice, often the complexity of the problem is such that the clusters identified do not bear a one-to-one correspondence with the physical classes (a given cluster may be “ambiguous” in that it contains data from different causes) so that an additional modelling step is necessary which regards the development of tailored classification models, one for each cluster, which disentangle the clusters by mapping the features to the classes representing the different transient causes.

In this work, the focus is on the clustering task. Cluster analysis is one of the main techniques in pattern recognition. The application of cluster analysis in various disciplines is well documented (see, for example, (Duran and Odell, 1974) and (Jain and Dubes, 1988)). The conventional (hard) clustering methods constrain each pattern to belong to one cluster only. In practice, however, the clusters may not be completely disjoint and patterns could be classified as belonging to one cluster almost as well as to another. In these cases, one must assign to each pattern a set of membership values, one for each class identified. The implication of this is that the class boundaries are not 'hard' but rather 'fuzzy' and the representation of the data structure can be more accurately handled by fuzzy clustering methods (Bezdek, 1981),(Yang, 1993).

When cluster analysis is to be used within the framework of classification of data belonging to physically different classes (e.g. transient data generated by different causes), an obvious desiderata is that the clusters eventually identified be indeed representative of the underlying physical differences (e.g. that the transient data forming a cluster be generated by the same physical cause).

In this work, the feasibility of building a fuzzy classifier by means of an evolutionary procedure (Yuan and Klir, 1997) applied to the well known Fuzzy C Means (FCM) algorithm (Duran and Odell, 1974), (Bezdek, 1981), (Gustafson and Kessel, 1979), (Jain and Dubes, 1988), (Yang, 1993) is investigated. The evolutionary algorithm searches for the optimal metrics to be used by the FCM to achieve clusters as close as possible to the real physical classes. To enhance the performance of the algorithm when dealing with classes of data characterized by little compactness, an iterative, supervised procedure is introduced which amounts to splitting the less compact physical classes into subclasses, as a manner for achieving a clustering of the data which more closely adheres to the known physical classes. The approach is verified with respect to an ancillary classification problem built with artificial data and then applied to the case of classification of transients of a U-Tube Steam Generator (UTSG) in a Pressurized Water Reactor (PWR). The transients are generated by means of the code UTSG which is part of the code ALMOD-2, which simulates the non-linear behavior of a PWR and is provided by the Nuclear Energy Agency (NEA) data bank of Paris (Höld, 1978).

The paper organization is as follows. The next Section formally introduces the classification problem, in both crisp and fuzzy terms, and presents an index to measure the performance of classification algorithms (Yuan and Klir, 1997). Section 3 illustrates the evolutionary Fuzzy C Means algorithm employed in this work and its application to the well known benchmarking Iris data (Fisher, 1936), (Keller et al., 1985), (Gath and Geva, 1989) and to the steam generator transients. The limitations of the approach with respect to classes characterized by little compactness are also discussed, with the aid of an example artificially built to fit the purpose. In Section 4 a quantitative measure of the compactness of a class is introduced (Xie and Beni, 1991). This measure is the basis for the extension of the algorithm presented in Section 5 and which is founded on the splitting of the less compact classes. Section 6 presents the improvements achieved by the extended evolutionary FCM algorithm on both the artificial example and the case regarding the classification of steam generator transients. The overall findings of the works are sinthetized and discussed in the Conclusions Section.
2 DEFINITION OF THE CLASSIFICATION PROBLEM

The task of pattern classification may be viewed as a problem of partitioning of objects (hereafter also called data patterns) into classes. From a mathematical point of view, a classifier is a mapping function $\Phi(\cdot)$ which assigns an object $\mathbf{x}$ in an $h$-dimensional domain $\Omega \subset \mathbb{R}^h$ to a given class $i$. Often in engineering, due to the complexity of the problems, it is not possible to determine the analytical expression of the function $\Phi$, i.e. to individuate the exact boundaries of the classes in the $\Omega$ space. Hence, one resorts to empirical classification techniques in which the classifier is built through a process of learning based on a set of classification examples. In other words, in the learning phase, labelled patterns belonging to the different classes are utilized to estimate the values of the parameters of the empirical classification function. This kind of technique is termed “supervised” and the available pre-classified data are termed “training” data (Gustafson and Kessel, 1979).

2.1 Crisp classification

Suppose that the physical problem requires to build a classifier for assigning patterns to a number $c$ of physical classes. To tackle the classification problem with a supervised technique we must have available a set $X$ of $n$ “training” data grouped into $c$ subsets representing the classes $\Gamma_i \equiv \{x_i^{(i)}\}, i = 1, 2, \ldots, c$, with $\sum_{i=1}^c N_i = n$ and $\bigcup_{i=1}^c \Gamma_i = \Gamma$, the partition of the data into the $c$ physical classes. In other words, the number of classes $c$ into which patterns are to be classified is known and $c$ groups of correctly pre-classified data are available to build the classifier, i.e. to find the mapping function $\Phi(\cdot)$ which assigns the membership of $x_i^{(i)}$ to class $i$, i.e. $\Phi(x_i^{(i)}) = i$.

In a crisp approach, the classes $\Gamma_i$ are regarded as crisp sets and the membership of an object to a class is binary (Yes/No): each object is assigned only to one class and the classes are regarded as disjoint gatherings of the data set (Duda and Hart, 1973), (Jain and Dubes, 1988). A crisp partition $\Gamma = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_c\}$ into $c$ classes is then characterized by a set $W$ of $c$ Boolean functions $w_i(x), i = 1, 2, \ldots, c$, on the feature space $\Omega \subset \mathbb{R}^h$ which satisfy:

$$w_i(x) = 0 \quad \text{or} \quad 1 \forall x \in \Gamma, \ i = 1, 2, \ldots, c$$  \hspace{1cm} (1)

$$\sum_{i=1}^c w_i(x) = 1 \quad \forall x \in \Gamma$$  \hspace{1cm} (2)

In eq.(1), $w_i(x) = 1$ means that $x \in \Gamma_i$ and eq.(2) ensures that $x$ is a member of the $i$-th class only.

2.2 Fuzzy classification

In many practical instances, however, the above Boolean restriction to class membership is not a realistic one since many data points may share characteristics common to several classes. In other words, there are cases in which the classes are not completely disjoint and data could be classified as belonging to one class almost as well as to another. Such situations cannot be described by a crisp classifying process. In these cases, it comes more natural to assign to each data point a set of membership values, one for each class identified. The implication is that the class boundaries are not ‘hard’ but rather ‘fuzzy’ and the representation of the data structure can be more accurately handled by fuzzy sets (Zadeh, 1965).

It is possible to move from the formal definition of a hard partition as given by (1) and (2) to a corresponding fuzzy partition by retaining condition (2) but replacing (1) with the relaxed condition:

$$0 \leq w_i(x) \leq 1, \quad i = 1, 2, \ldots, c$$  \hspace{1cm} (3)
Following the conventional notation of fuzzy sets, the function \( w_i(\bar{x}) \) represents the membership function of \( \bar{x} \) to class \( i \) and will be denoted as \( \mu_i(\bar{x}), \bar{x} \in \Gamma, i = 1, 2, ..., c \). The values of the memberships found can serve as a confidence measure in the classification (Keller et al., 1985): for example, if a pattern is assigned 0.9 membership in one class and 0.05 membership in two other classes we can be reasonably sure that the class of 0.9 membership is the class to which it belongs. On the other hand, if a pattern is assigned 0.55 membership in class one, 0.44 membership in class two, and 0.01 membership in class three, then we should be hesitant to assign it to a specific class based on these results.

2.3 A measure of the ‘distance’ between fuzzy partitions

Let \( \Gamma = \{ \Gamma_1, \Gamma_2, ..., \Gamma_c \} \) be the a priori known crisp partition in physical classes of the \( n \) training patterns \( \bar{x}_k, k = 1, ..., n \) and \( \Gamma^* = \{ \Gamma_1^*, \Gamma_2^*, ..., \Gamma_c^* \} \) the partition obtained by a crisp classifier. \( W \) and \( W^* \) are the corresponding sets of Boolean functions satisfying conditions (1) and (2). As index of the performance of the classification algorithm we consider the fraction of the \( n \) training patterns correctly assigned to the right physical class

\[
I_c(\Gamma, \Gamma^*) = 1 - \sum_{k=1}^{n} \sum_{i=1}^{c} \frac{|w_i(\bar{x}_k) - w_i^*(\bar{x}_k)|}{2n}
\]  

(4)

where the subscript \( c \) stands for ‘crisp’, \( w_i(\bar{x}_k) \in W \) is the a priori known Boolean membership value of the \( k \)-th pattern to the \( i \)-th class and \( w_i^*(\bar{x}_k) \in W^* \) is the corresponding membership value computed by the classifier. The performance index \( I_c(\Gamma, \Gamma^*) \) takes on values in \([0,1]\), being equal to 0 if none of the \( n \) patterns are correctly classified and to 1 if all the \( n \) patterns are correctly classified, i.e. \( \Gamma^* \equiv \Gamma \). Note that the misclassification of a pattern leads to two unitary contributions in the sum on the \( c \) classes, one from the actual class of the pattern and one from the erroneously assigned class: hence the division by 2. The definition of the performance index can be naturally extended to the case of fuzzy classification and, more generally, to the case of both fuzzy partitions \( \Gamma \) and \( \Gamma^* \):

\[
I_f(\Gamma, \Gamma^*) = 1 - \sum_{k=1}^{n} \sum_{i=1}^{c} \frac{\mu_i(\bar{x}_k) - \mu_i^*(\bar{x}_k)}{2n}
\]  

(5)

where the subscript \( f \) stands for ‘fuzzy’, \( 0 \leq \mu_i(\bar{x}) \leq 1 \) is the a priori known fuzzy membership of the \( k \)-th pattern to the \( i \)-th class and \( \mu_i^*(\bar{x}) \leq 1 \) is the fuzzy membership computed by the classifier. This definition of performance index is related to the concept of ‘distance’ between fuzzy partitions introduced in (Yuan and Klir, 1997):

\[
D(\Gamma, \Gamma^*) = \sum_{k=1}^{n} \sum_{i=1}^{c} \frac{\mu_i(\bar{x}_k) - \mu_i^*(\bar{x}_k)}{2n} = 1 - I_f(\Gamma^*, \Gamma)
\]  

(6)

3 AN EVOLUTIONARY FUZZY C MEANS CLUSTERING ALGORITHM

Fuzzy clustering algorithms have been widely studied and applied in a variety of substantive areas such as taxonomy, medicine, geology, business, engineering, image processing and others. A general classification of these algorithms is offered in (Yang, 1993) in terms of three categories: fuzzy clustering based on fuzzy relations, fuzzy clustering based on the minimization of an objective function, and the class of nonparametric classifiers based on the fuzzy generalized k-nearest neighbors rule. The interested reader is referred to (Yang, 1993) for a detailed discussion of the three categories of
In this section, we investigate the feasibility of building a fuzzy classifier by using the well known Fuzzy C Means (FCM) algorithm originally defined by Dunn (Dunn, 1974), then generalized by Bezdek (Bezdek, 1981) and nowadays available in a variety of variations and generalizations (Yang, 1993). Such algorithm is based on the minimization of an objective function which in its most common version relates to an Euclidean distance (or similarity) measure and leads to a search for hyper-spherical clusters in the $h$-dimensional space $\mathbb{R}^h$ hereafter called the features space.

The approach to the classification problem offered by the FCM algorithm is unsupervised because the algorithm makes no use of the a priori known information on the true physical classes of the training data and so the clustering is based only on the geometric grouping of the data in $\mathbb{R}^h$. Then, the geometric clusters obtained by the FCM algorithm do not necessarily yield the actual physical classes.

Furthermore, in (Yuan and Klir, 1997) it is shown that different partitions of a given set of data are obtained by using different metrics in the FCM algorithm. Based on these results, here we investigate whether the search for physical classes of objects, performed as a search for geometric clusters in the features space, can be improved by choosing an appropriate metrics. The target is that of identifying the optimal metrics, as usual defined by its geometric distance function (not to be confused with the partition distance previously introduced in eq.6), such that the geometric clusters found by the FCM algorithm are as close as possible to the a priori known classes of data.

A natural way to achieve this is to supervise the choice of the metrics by making use of the a priori known information on the true physical classes of membership of the training data.

In this view, the overall clustering task can be framed within an optimization problem in which the metrics, i.e. the geometric distance function, becomes an additional parameter to be determined besides the fuzzy partition. The target of the optimization is to minimize the partition distance $D(\Gamma, \Gamma^*)$ between the a priori known physical partition $\Gamma$ and the obtained partition $\Gamma^*$ (eq.6). This corresponds to maximizing the fuzzy classification performance index $I_f(\Gamma, \Gamma^*) = 1 - D(\Gamma, \Gamma^*)$ of (eq.5).

For the optimization, we employ the approach proposed in (Yuan et al., 1995) which amounts to using an evolutionary algorithm for determining the optimal geometric distance function and the FCM algorithm, based on such distance, to determine the optimal fuzzy partition. The overall structure of the algorithm is depicted in Figure 1. The closed loop iterates until an acceptable clustering of the training data is found, i.e. the obtained partition $\Gamma^*$ is “close” to the a priori known partition $\Gamma$ or, in

**Fig. 1.** Evolutionary FCM scheme: $M$ is a positive definite matrix which defines the metrics; $I$ is the identity matrix; $\varepsilon$ is a predefined threshold.
terms of the algorithm, the distance $D(\Gamma, \Gamma^*)$ is smaller than a predefined threshold $\epsilon$.

3.1 The evolutionary algorithm

The evolutionary algorithm employed for the optimization of the metrics used in the FCM clustering algorithm is a particular form of a genetic algorithm in which the only reproductive operations allowed are mutations. In (Angeline and Saunders, 1994) it has been shown that this kind of iterative search algorithm works very well when no suitable encoding scheme can be found for a regular genetic algorithm with crossover operation.

With respect to the FCM algorithm, the geometric distance between two points, $\vec{x}_1, \vec{x}_2$ in the $h$-dimensional domain $\Omega \subset \mathbb{R}^h$ may be any Mahalanobis distance computed by:

$$
d(\vec{x}_1, \vec{x}_2) = \left( (\vec{x}_1 - \vec{x}_2)^T M (\vec{x}_1 - \vec{x}_2) \right)^{\frac{1}{2}}
$$

(7)

where $M$ is a definite positive matrix of dimensions $h \times h$. A particular case is the Euclidean distance which is obtained using the identity matrix, $M = I$.

To find the optimal Mahalanobis distance, we need to update the positive definite matrix $M$ by which the distance is defined in eq.(7). To do this, we exploit the fact that any positive definite matrix $M$ can be uniquely decomposed into its Cholesky factors (Yuan and Klir, 1997). That is: $M = G^T G$ where $G$ is a lower triangular matrix with positive entries on the main diagonal. Therefore, to update the matrix $M$ we can update its Cholesky factor matrix $G$. This is done as follows. At the generic generation $t$ the evolutionary algorithm manipulates a working matrix which contains the entries $g_{ij}$ of the Cholesky factor $G$. At the first generation, the working matrix is initialized to the Cholesky factor of the Identity matrix, i.e the first distance considered for the FCM is the Euclidean distance. The FCM algorithm partitioning, based on this distance, is then performed up to convergence, using as initial partition of the $n$ training data a “supervised initialization” which sets the initial clusters assignments coincident to the a priori known classes. With this “supervised initialization”, the FCM has been found, in all our test cases, to converge in less iterations than with a random initialization.

By construction, the obtained clusters $\Gamma^*_1, \Gamma^*_2, ..., \Gamma^*_c$ correspond in the ordering to the a priori known physical classes $\Gamma_1, \Gamma_2, ..., \Gamma_c$ so that the partition distance $D(\Gamma, \Gamma^*)$ is computed by comparing, in the sense of eq. (6), individually the class $\Gamma_i$ with the cluster $\Gamma^*_i$, $i = 1, ..., c$.

At the next generation, the entries of the working matrix are then updated as follows:

$$
g_{ij}(t+1) = g_{ij}(t) + N_{ij}(0, \delta(t)) \quad \text{if } i \neq j
$$

$$
g_{ii}(t+1) = \max \left( 10^{-5}, g_{ii}(t) + N_{ii}(0, \delta(t)) \right)
$$

(8)

(9)

where $\delta(t) = \alpha D(\Gamma, \Gamma^*(t))$, $\alpha$ is a parameter that controls the size of the random step of modification of the Cholesky factor entries $g_{ij}$, $N_{ij}(0, \delta)$ denotes a Gaussian noise with mean 0 and standard deviation $\delta$, and eq.(9) ensures that all entries in the main diagonal of the matrix $G(t+1)$ are positive numbers. From the definition of the parameter $\delta(t)$, the smaller the distance between the computed and a priori known partitions, the smaller the value of $\delta$ and hence the smaller the random step of modification of the $g_{ij}$ elements.

Once the updating of the Cholesky factor has been performed, the FCM is run using the newly generated matrix $M(t+1) = G(t+1)^T G(t+1)$ and the corresponding new distance $D(\Gamma, \Gamma^*(t+1))$ is computed. The modified Cholesky factor matrix $G(t+1)$, is accepted only if its corresponding partition distance $D(\Gamma, \Gamma^*(t+1))$ is less than that of the parent $G(t)$, i.e. $D(\Gamma, \Gamma^*(t))$; otherwise we take again $G(t+1) \equiv G(t)$ as parent of the next generation.

In the end, at convergence, the evolutionary FCM algorithm provides the centers $\vec{\mu}_i, i = 1, 2, ..., c$ of
the c clusters and the membership values of the n training data to the c clusters. A new pattern $\tilde{x}^*$ is classified, in fuzzy terms, by computing its values of membership to the c clusters, based on the Mahalanobis distance of matrix $M$ at convergence, knowing the centers $\tilde{u}_i$, i = 1, 2, ..., c.

Given the ordered correspondence between classes and clusters, the fuzzy membership information may be further used for the crisp assignment of the patterns to the various classes by setting a classification threshold $\gamma$; if the largest value of membership of the generic pattern $\tilde{x}^*$ to any of the c clusters is greater than $\gamma$ then the pattern is assigned to the class corresponding to the cluster of largest membership; otherwise, if none of the membership of $\tilde{x}^*$ is larger than $\gamma$, the pattern remains ambiguous, i.e. it is not assigned to any of the c classes. In this sense, the classification threshold $\gamma$ regulates the confidence in the crisp assignments to classes, along the views illustrated in Section 2.2.

3.2 Application of the evolutionary FCM classifier to the Iris data

The Iris data, a well known classification benchmark (Fisher, 1936), (Keller et al., 1985), (Duran and Odell, 1974), (Gath and Geva, 1989), have been used to verify the performance of the evolutionary FCM classifying algorithm.

In the present work, 120 randomly selected flowers have been used to train the evolutionary FCM classifier, and the remaining 30 flowers to test the performance of the classifier on new data. The obtained results have been compared with the classification obtained by using the standard FCM algorithm based on the Euclidean metrics. With respect to the standard algorithm, the performance index $I_f(\Gamma, \Gamma^*)$ of the evolutionary FCM classifier increases from 0.8757 to 0.9816 on the 120 flowers used for training and from 0.8609 to 0.9377 on the 30 flowers of test. These results confirm that the evolutionary FCM algorithm effectively modifies the metrics to obtain geometric clusters close to the physical classes.

Figure 2 compares the performances of the two classifiers on all the 150 flowers in terms of their final crisp assignment to a class, as explained at the end of the previous Section 3.1. In this respect, in Section 2.2 we have observed that the membership value with which an object is assigned to a class may be interpreted as a measure of the confidence of its classification. In our case, if we want high confidence we assign a flower to a class only if its membership computed by the algorithm is close to 1 (e.g. $\gamma = 0.99$), but in this case many flowers may not be assigned to any class by the algorithm (in the case of $\gamma = 0.99$, 86.65% of the patterns are assigned to the right class, 13.35% are not assigned to any class and no misclassifications occur). On the other hand, if we want to assign all the flowers to one of the classes, we would use a lower membership value as classification threshold $\gamma$ (e.g. $\gamma = 0.55$), but the number of flowers incorrectly assigned might increase unacceptably (in the case of $\gamma = 0.55$, all patterns are classified, 98.00% to the correct class and 2.00% to the wrong class). For the case study at hand, Figure 2 shows how the performance of the classifier evaluated on all the 150 flowers changes for different values of the threshold $\gamma$. As expected, decreasing the confidence threshold required for assigning a pattern to a class, both the numbers of correct and incorrect assignments increase, and correspondingly the number of flowers not assigned decreases.

3.3 Application of the evolutionary FCM classifier to nuclear transients data

In this Section, we tackle the issue of classifying four kinds of transients occurring in the steam generator (SG) of a PWR on the basis of the available temperature, pressure and flow measurements. The steam generator is a component of the Pressurized Light and Heavy Water Reactors and operates as an interface between the nuclear heat generating part and the conventional power conversion part of the plant. Among the existing configurations, we consider here the well-known standard recirculation U-tubes type. A description of this component can be found in (Collier, 1998) whereas a detailed model of its physical functioning is given in (Marseguerra et al., 1996).
Fuzzy clustering for the classification of transients

Fig. 2. Fraction of flowers assigned correctly (top), incorrectly (middle) and not assigned (bottom) by the FCM classifier based on the Euclidean metrics (dashed lines) and by the evolutionary FCM classifier (solid lines), as a function of the classification threshold $\gamma$. The vertical solid lines at $\gamma = 0.95$ and $\gamma = 0.55$ indicate the confidence levels for the assignment to a class used for the numerical examples reported in the text.

Table 1
Forcing functions: reference values and ranges of variability of $F_{\text{fin}}$.

<table>
<thead>
<tr>
<th>Forcing Function</th>
<th>Reference Value</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{1E}$ [°C]</td>
<td>312.8</td>
<td>27</td>
<td>330</td>
</tr>
<tr>
<td>$G_{1E}$ [kg/s]</td>
<td>5259</td>
<td>4759</td>
<td>6759</td>
</tr>
<tr>
<td>$T_{FW}$ [°C]</td>
<td>212</td>
<td>162</td>
<td>262</td>
</tr>
<tr>
<td>$G_{FW}$ [kg/s]</td>
<td>467.3</td>
<td>420</td>
<td>510</td>
</tr>
</tbody>
</table>

In our analysis, we have assumed that the pressurizer imposes a constant primary system pressure ($P_1$) of 154.4 bar. Among the possible transient-initiating perturbations (hereafter called forcing functions) the ones here considered are: the inlet water temperature ($T_{1E}$) and the inlet water mass flow ($G_{1E}$) on the primary side, the feed water temperature ($T_{FW}$) and the feed water mass flow ($G_{FW}$) on the secondary side. These four forcing functions may vary as a consequence of an operator action or because of plant's anomalies or faults.

The transient responses of the SG were obtained with the code UTSG provided by the NEA data bank of Paris. The code has been developed by the "Gesellschaft für Reaktorsicherheit (GRS)" in Garching bei München, and it properly accounts for the feedback due to the heat removal system. It is part of the code ALMOD-2, which simulates the non-linear behavior of a PWR. UTSG computes the solution of the system of equations in two steps (Höld, 1978): (1) at first, the time derivatives in the base equations are set to 0 and the initial values are calculated by solving the resulting system of algebraic equations with recursive techniques; (2) then the stationary part of the equations is eliminated and the solution is obtained by an explicit integration procedure based on a method by Burlisch and Stör (Höld, 1978). The UTSG code was used to generate 529 transients, each one 50s long, obtained by randomly varying the kind and intensity of the forcing function. More specifically, the generic forcing function $F(t)$ is assumed to vary in 8s according to the following sigmoidal shape:

$$F(t) = F_o + \frac{F_{\text{fin}} - F_o}{1 + e^{-m(t-\tau)}}$$

where $F_o$ is its initial value at time $t = 0s$, $F_{\text{fin}}$ is its final value for $t = 8s$, $\tau$ is the delay time and $m$ is the characteristic frequency of the transient, here set to $5/t$. Table 1 reports the intervals of variation of the forcing functions considered in this work for the generation of the transients of interest, as derived from engineering considerations.

Regarding the monitored signals, they have been selected taking into account their physical mea-
surability. Out of the 15 signals provided by the UTSG code, we considered the following 8: primary outlet water temperature and mass flow \((T_{1A} \text{ and } G_{1A})\); secondary inlet water temperature and mass flow \((T_{2E} \text{ and } G_{2E})\); secondary system pressure \((P_2)\); secondary water level \((Z_{WL})\); total steam mass flow \((G_{ST})\); generator power \((Q_{ON})\). The generic measurement vector is composed of the values of the 8 measurable signals taken at the same instant. Along the 50s-long transients, we have considered 10 sampling instants, taken every 4s, from 7s to 43s. For improved classification performance, the dimensionality of the measurements vector has been reduced to four features by means of a non-linear principal component analysis performed by means of an autoassociative artificial neural network (Marseguerra et al., 2004). Hence, the transient classifying algorithm aims at establishing, through an appropriate clustering procedure, a mapping between the four-dimensional input space of the features and the one dimensional discrete output space of the forcing functions responsible of the transients. All the available 3500 data have been used for training the evolutionary FCM classifier in the task of assigning a pattern to one of the four a priori known classes. The resulting classification performance index is equal to 0.5211. With a degree of confidence \(\gamma = 0.9\), for example, only 35.59% of the transients are correctly assigned to the right class (23.60% with the standard, Euclidean-based FCM algorithm), whereas 29.00% (12.43% with the standard, Euclidean-based FCM algorithm) are assigned to the wrong class and 35.41% (69.97% with the standard, Euclidean-based FCM algorithm) are not assigned to any class. Evidently, the four features which are taken to physically characterize the data give an insufficient description of the four classes to which the data belong. This will be discussed in more details in the next Section and we shall see how the classification performance can be significantly improved by a supervised procedure which splits the data in further geometric clusters, without loosing the identity of the four classes. In other words, several geometric clusters may be set up to correspond to a single physical class.

3.4 Limitations of the evolutionary FCM classification approach

The reason for the unsatisfactory results obtained in the evolutionary fuzzy clustering classification of the transients considered in the previous Section is the little compactness of the a priori known physical classes in the features space \(\mathbb{R}^4\). The examination of Figure 3 shows that the four classes are not compact and that each class tends to be split in at least two subclasses, like the two wings of a butterfly. Physically, this is due to the fact that the forcing functions causing the transients
Fuzzy clustering for the classification of transients

may vary in two directions, above or below the nominal values, leading to different and possibly opposite consequences on the behavior of the plant signals in the features space.

To support the above claim, we consider the artificial two-dimensional data set of Figure 4. These data comprise three physical classes A,B,C two of which, A and B, appear completely disjoint into two pairs of clusters, A₁ - A₂ and B₁ - B₂, when represented in the two-dimensional features space. The application of the evolutionary FCM classifier to this data set gives unsatisfactory performance index values of 0.7198 on 3000 training data and of 0.7239 on 750 test data. For example, with a degree of confidence of γ = 0.85, the algorithm correctly classifies 67.3% of the patterns, misclassifies only 18.1% and does not assign to any class 14.6%. It is clear that the evolutionary FCM algorithm performs poorly when the physical classes are characterized by small compactness of their corresponding geometric clusters in the features space.

4 A MEASURE OF CLASS COMPACTNESS

In the previous Section, we have qualitatively introduced a concept of compactness to interpret the classification performance of the evolutionary FCM algorithm in an intuitive way. We now introduce a corresponding quantitative index that measures the compactness of a class.

Consider the quadratic distance \( d^2(\vec{x}_k, \vec{u}_i) = (\vec{u}_i - \vec{x}_k)^T \Sigma (\vec{u}_i - \vec{x}_k) \) with respect to the \( M \) metrics, between the center \( \vec{u}_i \) of the \( i \)-th class and the \( k \)-th pattern \( \vec{x}_k \), \( k = 1, 2, \ldots, n \). The quantity \( \sigma_{fi} = \sum_{k=1}^{n} \mu_i^2(\vec{x}_k)d^2(\vec{x}_k, \vec{u}_i) \) can be taken as representing the (fuzzy) variation, or dispersion, of the data in cluster \( i \) (Xie and Beni, 1991). Then, to measure the compactness of class \( i \), the quantity \( \sigma_i \) is introduced:

\[
\sigma_i = \frac{\sigma_{fi}}{N_{fi}} = \frac{\sum_{k=1}^{n} \mu_i^2(\vec{x}_k)d^2(\vec{x}_k, \vec{u}_i)}{\sum_{k=1}^{n} \mu_i(\vec{x}_k)} \quad i = 1, 2, \ldots, c
\]

where \( N_{fi} = \sum_{k=1}^{n} \mu_i(\vec{x}_k) \) is the fuzzy cardinality of class \( i \), i.e. the fuzzy number of samples in class \( i \). The less compact (more dispersed) the classes are, the greater \( \sigma_i \) is. For example, in the data set of Figure 4, the two “split” classes A and B have a significantly larger fuzzy dispersion (2.55E-02) than class C (4.93E-03).

5 A SUPERVISED ALGORITHM WITH SPLITTING OF THE LESS COMPACT CLASSES

In Sections 3.3 and 3.4 it is shown that the evolutionary classifying algorithm performs poorly in the classification of data characterized by physical classes which are not sufficiently compact in the features space. To improve the classification performance, a new iterative approach is investigated, based on the splitting of the most dispersed class into two more compact subclasses. The method proceeds as follows:

(1) First the least compact physical class \( \Gamma_i \), i.e. that with highest \( \sigma_i \), is identified. Note that \( \Gamma_i \) is a crisp class so that in the evaluation of \( \sigma_i \) the membership values \( \mu_i(\vec{x}_k) \) of the patterns \( \vec{x}_k, k = 1, 2, \ldots, n \) are equal to 1 when \( \vec{x}_k \in \Gamma_i \) and 0 when \( \vec{x}_k \notin \Gamma_i \).

(2) Then, the standard, Euclidean-based FCM algorithm is applied to the data belonging to such class in order to split it into two fuzzy sub-classes, \( \Gamma_{i1} \) and \( \Gamma_{i2} \), such that: \( \Gamma_i = \Gamma_{i1} \cup \Gamma_{i2} \) where the union of the two subclasses \( \Gamma_{i1}, \Gamma_{i2} \) may be defined as (Bojadziev and Bojadziev, 1995):

\[
\Gamma_{i1} \cup \Gamma_{i2} \longrightarrow \mu_{\Gamma_{i1} \cup \Gamma_{i2}}(\vec{x}_k) = \max\{1, \mu_{\Gamma_{i1}}(\vec{x}_k) + \mu_{\Gamma_{i2}}(\vec{x}_k)\}
\]

Note that in this way we obtain two fuzzy subclasses from one crisp class and the a priori known crisp partition \( \Gamma = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_c\} \) is substituted by the fuzzy partition \( \Gamma(\vec{x}) = \Gamma_{i1} \cup \Gamma_{i2} \).
Merge the two subsets $F_1$ and $F_2$ into one:

$$\Gamma / F_1 , F_2 . . . , F_h , \Pi_1 , . . . , \Pi_c$$

where for clarity we have explicitely the number of classes $c = c + 1$ as argument of the partition. This new partition is also a priorly known, with the data of subclasses $\Gamma_1$ and $\Gamma_2$ actually belonging to the same physical class $\Gamma_i$.

(3) The FCM evolutionary algorithm described in Section 3.1 is used to partition the data in $c+1$ subclusters, i.e. to determine the optimal partition in $c$ clusters $\Gamma^*(c) = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_i, \Gamma_j, \ldots, \Gamma_h, \Pi_1, \Pi_2, \ldots, \Pi_c\}$.

(4) However, for our classification purposes we are interested in a partition $\Gamma^*$ "as close as possible" to $\Gamma$, and not in the obtained, enlarged partition $\Gamma^*(c)$, so that in the end we have to merge the obtained subclusters $\Gamma_1$ and $\Gamma_2$ by means of the union operator (eq.12), i.e. $\Gamma_i = \Gamma_1 \cup \Gamma_2$

(5) At this point, the value of the partition distance $D(\Gamma, \Gamma^*)$ can be computed.

This procedure is repeated iteratively until a satisfactory classification of the training data is reached, i.e., until the partition distance exceeds a given threshold $\epsilon$ (Figure 5).

Formally, after $L$ splitting, the $i$-th physical class is subdivided in $l_i$ subclusters and the evolutionary algorithm performs the partition $\Gamma^*_i(c) = \{\Gamma_{i1}, \Gamma_{i2}, \ldots, \Gamma_{il}, \Gamma_{i1}, \Gamma_{i2}, \ldots, \Gamma_{il}, \Pi_1, \Pi_2, \ldots, \Pi_c\}$ with $c = \sum_{i=1}^l l_i$. Then, the partition $\Gamma^*$ is obtained by merging accordingly the sets of $l_i$ subclusters, i.e. $\Gamma_i = \bigcup_{j=1}^{l_i} \Gamma_{ij}$

$$i = 1, 2, \ldots, c$$

5.1 Application to the artificial two-dimensional data set

The developed algorithm has been applied to the artificial two-dimensional, three clusters data set of Figure 4. Figure 6 shows the evolution with $\bar{c}$ (the number of clusters) of $D(\Gamma, \Gamma^*)$, the distance between the a priori known physical partition, $\Gamma$, of the $n = 3000$ training data and the corresponding partition obtained by the iterative splitting algorithm, $\Gamma^*$. The minimum of the distance is reached when the fuzzy clustering algorithm handles $\bar{c} = 5$ clusters obtained by splitting the 2 disjointed, and thus most dispersed, classes, $A, B$, in 4 more compact subclusters, $A_1, A_2, B_1, B_2$. This means that if one of the 5 obtained subclasses $A_1, A_2, B_1, B_2, C$ is further split, two less compact subclusters are obtained and the evolutionary FCM algorithm is not able to find a partition $\Gamma^*$ with $\bar{c} = 6$ closer to $\Gamma$ than the one found with $\bar{c} = 5$.

Figure 7 compares the performance on the 750 test data of the standard FCM, the evolutionary FCM, and the iterative splitting algorithm with $\bar{c} = 5$ subclusters properly merged back into 3. The best results are obtained by the new proposed iterative algorithm that, for example, with a degree of
Fuzzy clustering for the classification of transients

5.2 Application to the nuclear transients data

Finally the developed algorithm has been applied to the nuclear transients data of section 3.3. A set of 2800 transients has been used to train the algorithm while the remaining 700 patterns have been left for testing its performance.

Figure 8 shows the distance, $D(\Gamma, \Gamma^*)$, between the physical classes and the obtained corresponding clusters as a function of the number of clusters $\gamma$. The minimum value of $D(\Gamma, \Gamma^*)$ is reached with $\gamma = 8$ clusters.

Figure 9 shows a significant improvement in the classification of the 700 test transients by the iterative splitting algorithm with $\gamma = 8$ compared with the standard and evolutionary FCM classifiers. For example, the proposed algorithm with a degree of confidence of 0.85, correctly classifies 70.0% of the transients (27.3% with the standard FCM and 37.4% with the evolutionary FCM), misclassifies 7.7% (33.6% with the standard FCM and 31.0% with the evolutionary FCM) and does not assign to any class 22.3% of the patterns (39.1% with the standard FCM and 31.6% with the evolutionary FCM).

Table 2 shows the number of test transients of each physical class assigned with a degree of confidence of 0.85 to each of the four clusters obtained after merging. Note that the transients forming the clusters $\Gamma_1$, $\Gamma_2$ and $\Gamma_3$ are all coherent, i.e. they actually have been univocally caused by the corresponding associated forcing function (respectively $T_{LE}$, $G_{LE}$ and $T_{FW}$). In other words, when a transient is assigned to one of these clusters we have 0.85 confidence that the corresponding associated forcing function is the cause for such transient. The three clusters $\Gamma_1$, $\Gamma_2$ and $\Gamma_3$ can then be said to be “unambiguous” in that they meet the required property of being representative of a single cause of transients. On the other hand, cluster $\Gamma_4$ is “ambiguous” because it contains transients caused by all
Table 2

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Class F1 ($T_{1E}$)</th>
<th>Class F2 ($G_{1E}$)</th>
<th>Class F3 ($T_{FW}$)</th>
<th>Class F4 ($G_{FW}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>111</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>0</td>
<td>156</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>0</td>
<td>0</td>
<td>74</td>
<td>0</td>
</tr>
<tr>
<td>Cluster 4</td>
<td>23</td>
<td>1</td>
<td>30</td>
<td>152</td>
</tr>
<tr>
<td>Not Assigned</td>
<td>58</td>
<td>51</td>
<td>39</td>
<td>5</td>
</tr>
</tbody>
</table>

Fig. 8. Distance $D(\Gamma, \Gamma^*)$ between the a priori known partition $\Gamma$ of the $n = 2800$ training data and the corresponding partition $\Gamma^*$ obtained by the iteratively splitting algorithm.

Fig. 9. Fraction of the 700 test transients assigned correctly (top), incorrectly (middle) and not assigned (bottom) by the standard FCM classifier (dashed line), the evolutionary FCM classifier (dotted line) and the iterative splitting classifier (solid line) as a function of the classification threshold $\gamma$. The vertical solid line indicates the confidence level of $\gamma = 0.85$ for the assignment to a class, as used for the numerical example in the text.

the four forcing functions so that we cannot be sure of the cause of a transient assigned to this cluster, although all the transients caused by a variation of the fourth forcing function, $G_{FW}$, are correctly included. To understand the situation in greater details, let us examine the geometric partition of the case with $c = 7$ subclasses which is similar in performance to that of $c = 8$ and it is more illustrative because the 7 subclasses are obtained by splitting once the clusters $F_1$, $F_2$ and $F_3$ with no splitting of $F_4$. In Figure 10 we report, in the subspace defined by the features 1 and 2, the partition of the points that belong to the 7 clusters with a membership value larger than 0.85. Cluster $F_4$ lies in the middle of the others corresponding exclusively to the first three forcing function $F_1$, $F_2$, $F_3$ ($T_{1E}$, $G_{1E}$, $T_{FW}$) so that it seems reasonable that some transients caused by forcing functions other than $F_4$ ($G_{FW}$) may be assigned to this cluster. Physically this is due to the fact that a variation of $G_{FW}$ within the range of Table 1 has little effect on the 4 input signals upon which the clustering is based. It can then be argued that the transient caused by a variation of this forcing function is very close to a steady-state and thus it can be easily confused with those arising from small, quasi-steady variations of any of the other three forcing functions.

As a final remark, the results obtained lead us to suggest the application of a hierarchical model for the classification problem at hand, as follows. At the first stage of the hierarchical model, the evolutionary FCM algorithm with splitting is applied to build the unambiguous clusters $F_1^*$, $F_2^*$ and $F_3^*$; at a second stage, two dedicated classifiers (e.g. neural network or neuro-fuzzy models) are built to map the transients not assigned by the first classifier and those assigned to the ambiguous cluster.
Fig. 10. Partition of the points with membership larger than 0.85 into $c = 7$ clusters in the subspace of features 1 and 2.

Although the optimal development of these classifiers is not in the scope of the current work, here we are reporting the results obtained by a classifier based on fuzzy logic rules automatically constructed from numerical input/output data pairs (Wang and Mendel, 1992), (Marseguerra et al., 2003). At the first stage of the hierarchical classification model, 359 of the 700 test patterns are not assigned, or assigned to the ‘ambiguous’ cluster $F_4^*$ by the evolutionary FCM, while at the second stage 26 patterns out of the 359 not classified by the first stage are misclassified so that, globally, the hierarchical classification model misclassifies only 3.71% of the 700 test patterns.

6 CONCLUSIONS

The reliable and timely classification of signal transients is a matter of paramount importance for the safe operation of nuclear power plants and components. The complexity and nonlinearity of the signal patterns followed by the measurable process variables renders the classification task difficult, if not prohibitive, by means of standard pattern recognition techniques. For this reason, powerful soft computing techniques, such as neural networks, fuzzy logic and genetic algorithms, are being extensively investigated.

In the present work we have embraced the fuzzy clustering approach to the classification problem and exploited an evolutionary algorithm of literature for determining the optimal Mahalanobis metrics to be used within a generalized Fuzzy C Means clustering method. To overcome the limitations of the approach with respect to the analysis of dispersed classes we have introduced an innovative algorithm which iteratively splits the less compact class into two more compact subclasses. At each iteration, the extended physical partition is treated by the evolutionary FCM algorithm to identify the corresponding fuzzy clusters.

The complete algorithm has been verified first on an artificial test case and then applied to simulated nuclear transients data related to the operation of a steam generator of a pressurized water reactor. The classification results obtained with the proposed algorithm can be considered satisfactory and spark further enthusiasm for research in this area towards the practical application of these methods. Indeed, as future research step in this direction we plan to optimize a hierarchical classification model in which the first stage is constituted by the FCM evolutionary classifier with iterative splitting. Then, tailored mapping algorithms can be employed to classify the patterns not assigned with sufficient membership to any cluster or assigned to ambiguous clusters. The aim of this future work is to further improve the classification accuracy of the algorithm. The mapping algorithms which will be investigated for this task are envisioned to be empirical ones such as multilayered, feed forward neural networks, fuzzy logic systems and neuro-fuzzy techniques.
REFERENCES


PART III

Selected papers
Identification of Nuclear Transients via Optimized Fuzzy Clustering

Enrico Zio and Piero Baraldi

2005
Identification of nuclear transients via optimized fuzzy clustering

E. Zio *, P. Baraldi

Department of Nuclear Engineering, Polytechnic of Milan, Via Ponzio 34/3, 20133 Milano, Italy

Received 22 December 2004; accepted 23 February 2005
Available online 31 March 2005

Abstract

In this paper, we look into the issue of using cluster analysis for transient classification in nuclear components and systems. In general, the choice of the metrics upon which clustering is based can be critical for obtaining geometric clusters as close as possible to the real physical classes in the feature space. The complexity and variety of cluster shapes and dimensions which can be expected in the transient classification of interest lead us to take an approach based on a different Mahalanobis metric for each cluster. The a priori known information regarding the true classes to which the patterns belong is exploited to select, by means of a supervised evolutionary algorithm, the different optimal Mahalanobis metrics. Further, the diagonal elements of the matrices defining the metrics can be taken as measures of the relevance of the features employed for the classification of the different patterns.

The efficiency of the approach is verified with respect to a literature problem and then applied to the case of classification of transients in a nuclear component.

© 2005 Elsevier Ltd. All rights reserved.

1. Introduction

The basis for the identification of a change in the functioning of a system is that different system faults and anomalies lead to different patterns of evolution of the

* Corresponding author. Tel.: +39 02 2399 6340; fax: +39 02 2399 6309.
E-mail address: enrico.zio@polimi.it (E. Zio).

0306-4549/S - see front matter © 2005 Elsevier Ltd. All rights reserved.
involved process variables or features. Given the safety and economical importance of the problem, methods are sought for performing the mapping from the features to the different system behaviors, for diagnostic purposes.

In particular, a large number of diagnostic methods for nuclear power plant sensors and components has been proposed in the past decade, based on the advances of soft computing techniques such as artificial neural networks and fuzzy logic based techniques (Reifman, 1997). These methods differ on their diagnostic scope and on the type of computing approach used. With regards to the former, the main tasks that these diagnostic systems tackle are fault detection (i.e., establishing the existence of a plant anomaly based on observed trends of measured signals) and identification (i.e., establishing the cause of the detected anomaly). With regards to the latter, artificial neural networks and fuzzy logic systems are the most widely used. Artificial neural networks are data-processing systems consisting of a number of simple processing elements interconnected in an architecture inspired by the cerebral cortex of the brain. When used as diagnostic tools, properly trained artificial neural networks offer the capability of providing real-time responses and of recognizing faults even when fed with noisy or partially incomplete signals (Uhrig, 1991b). Further, their empirical nature allows modelling complex systems and processes when only input and output data are available (Uhrig, 1991a). Fuzzy logic systems are also widely proposed in modern diagnostic technologies. Based on the principles of Zadeh’s fuzzy set theory, fuzzy logic provides a formal mathematical framework for dealing with the vagueness of everyday reasoning (Zadeh, 1965). As opposed to binary reasoning based on ordinary set theory, fuzzy logic allows for classification into multiple classes with different degrees of membership (Klir and Yuan, 1995). Further, measurement uncertainty and estimation imprecision can be properly accommodated within the fuzzy logic framework.

In this work, we present an approach to transient identification based on pattern classification by fuzzy clustering (Jain and Dubes, 1988). In general, the task of pattern classification may be viewed as a problem of partitioning of objects (hereafter also called data, patterns) into classes. In particular, a fuzzy partition \( \Gamma \) into \( c \) subsets \( \Gamma_i \) is characterized by a set of \( c \) membership functions \( \{ \mu_1(\bar{x}), \ldots, \mu_c(\bar{x}) \} \) where \( \mu_i(\bar{x}) \) denotes the membership of pattern \( \bar{x} \) into class \( i \) (Gustafson and Kessel, 1979).

Fuzzy clustering algorithms have been widely studied and applied in a variety of substantive areas such as taxonomy, medicine, geology, business, engineering, image processing and others. A general classification of these algorithms is offered in Yang (1993) in terms of three categories: fuzzy clustering based on fuzzy relations, fuzzy clustering based on the minimization of an objective function, and the class of non-parametric classifiers based on the fuzzy generalized \( k \)-nearest neighbors rule. The interested reader is referred to Yang (1993) for a detailed discussion of the three categories of algorithms and an extensive literature review of works in the field.

In this work, we investigate the feasibility of building a fuzzy classifier by means of a supervised evolutionary procedure (Yuan et al., 1995) applied to the well known Fuzzy C Means (FCM) clustering algorithm (Bezdek, 1981). To tackle the classification problem with a supervised technique we assume to have available a partition \( \Gamma \), in general fuzzy, of \( n \) “training” labelled data, \( \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n \) in \( c \) subsets. The
evolutionary algorithm searches for the optimal metrics to be used by the FCM so as to achieve clusters as close as possible to the real physical classes.

The traditional, unsupervised FCM algorithm leads to spherical clusters which rarely are adequate to represent the data partition in practice. A partial improvement is achieved by considering a Mahalanobis distance as in the supervised algorithm proposed in Zio and Baraldi (2004). This technique leads to ellipsoidal clusters that more adequately fit the a priori known data partition. However, by this approach all the clusters have the same shape. To increase the classification performance of the clustering algorithm, in this work we investigate the possibility of optimizing a different metrics for each cluster so as to allow for different shapes in the feature space.

The approach is verified with respect to an ancillary classification problem built with artificial data and applied to the case of classification of simulated transients in a U-tube steam generator (Collier, 1998; Marseguerra et al., 1996).

The paper is organized as follows. The next section illustrates the supervised evolutionary Fuzzy C Means algorithm. In Sections 3 and 4, the approach is applied, respectively, to a two-dimensional artificial case study and to the case of classification of transients in a U-tube steam generator of a pressurized water reactor (PWR). In Section 5, a possible interpretation of the diagonal elements of the matrix defining the Mahalanobis metric, in terms of feature importance, is discussed. The overall findings of the work are discussed in Section 6.

2. The supervised evolutionary optimization of the Fuzzy C Means algorithm

The FCM technique is unsupervised since it makes no use of a priori known information on the true physical classes of the training data. The clustering is based on the minimization of a weighed sum of the distances \( d(\bar{x}_k, \bar{v}_i) \) between the patterns \( \bar{x}_k \) and the cluster centers \( \bar{v}_i \),

\[
J = \sum_{i=1}^{c} \sum_{k=1}^{n} \mu_i(\bar{x}_k)^m d^2(\bar{x}_k, \bar{v}_i),
\]

where the weight \( \mu_i(\bar{x}_k) \) denotes the membership of \( \bar{x}_k \) into class \( i \) and \( m \) is a parameter which controls the degree of fuzziness of the clusters (often a value of 2 has been found suitable). In the traditional algorithm (Dunn, 1974), the distance is Euclidean,

\[
d^2(\bar{x}_k, \bar{v}_i) = d_I^2(\bar{x}_k, \bar{v}_i) = (\bar{x}_k - \bar{v}_i)^T I (\bar{x}_k - \bar{v}_i),
\]

where \( I \) is the identity matrix. This leads to spherical clusters which rarely are adequate to represent the data partition in practice.

An improvement is offered in Gustafson and Kessel (1979), where each cluster \( i \) is defined by an individual distance through a dedicated Mahalanobis, definite positive, matrix \( M_i \),

\[
d^2(\bar{x}_k, \bar{v}_i) = d_{M_i}^2(\bar{x}_k, \bar{v}_i) = (\bar{x}_k - \bar{v}_i)^T M_i (\bar{x}_k - \bar{v}_i).
\]
From the minimization of $J$ (Eq. (1)), it turns out that $M_i$ is proportional to the fuzzy covariance matrix of the $i$th cluster (Gustafson and Kessel, 1979).

In an attempt of improving the classification performance of the clustering algorithm, we investigate the feasibility of finding the metrics of each cluster $i = 1, 2, \ldots, c$ in a supervised manner so as to achieve geometric clusters as close as possible to the a priori known $c$ classes of the available labelled data. In this view, the classification task can be framed within an optimization problem in which the metrics, i.e., the geometric distance functions, become additional parameters to be determined besides the fuzzy partition. The supervised target of the optimization is that of minimizing the partition distance $D(\Gamma, \Gamma^*)$ between the a priori known physical partition $\Gamma$ and the obtained partition $\Gamma^*$ (Zio and Baraldi, 2004),

$$D(\Gamma, \Gamma^*) = \sum_{k=1}^{n} \sum_{i=1}^{c} \frac{|\mu_i(\bar{x}_k) - \mu_i^*(\bar{x}_k)|}{2n},$$

(4)

where $0 \leq \mu_i(\bar{x}) \leq 1$ is the a priori known (possibly fuzzy) membership of the $k$th pattern to the $i$th class and $0 \leq \mu_i^*(\bar{x}) \leq 1$ is the fuzzy membership computed by the FCM algorithm.

For the optimization, we integrate an evolutionary algorithm for determining the $c$ optimal geometric distance functions (Yuan et al., 1995) with the FCM algorithm for determining the optimal fuzzy partition based on such distance. The evolutionary algorithm used in this work is a particular form of a genetic algorithm in which the only reproductive operations allowed are mutations. In Angeline et al. (1994), it has been shown that this kind of iterative search algorithm works very well when no suitable encoding scheme can be found for a regular genetic algorithm with crossover operation. With respect to the approach in Yuan et al. (1995) the evolutionary algorithm must find a different optimal metrics $M_i$ for each cluster $i = 1, 2, \ldots, c$.

The overall structure of the algorithm is depicted in Fig. 1. The closed loop iterates until an acceptable clustering of the training data is found, i.e., until the obtained partition $\Gamma^*$ is "close" to the a priori known partition $\Gamma$ (where "close" means that the distance $D(\Gamma, \Gamma^*)$ is smaller than a predefined threshold $\epsilon$).

The overall scheme can be summarized as follows:

1. At the first iteration ($t = 1$), initialize the metrics of all the $c$ clusters to the Euclidean metrics, i.e., $M_i(1) = I_i$, $i = 1, 2, \ldots, c$.
2. Perform the FCM partitioning of the $n$ training data into $c$ clusters $\Gamma^*(t) = \{\Gamma^*_1(t), \ldots, \Gamma^*_c(t)\}$, based on the fixed metrics $M_i(t)$ and using a "supervised" initial partition which sets the initial clusters assignments coincident to the a priori known classes.
3. Compute the distance $D(\Gamma, \Gamma^*(t))$ between the a priori known physical classes and the geometric FCM clusters by Eq. (4). At the first iteration ($t = 1$) initialize the best distance $D^+$ to $D(\Gamma, \Gamma^*(1))$ and the best metrics $M^+_i$ to $M_i(1)$ and go to step 5.
4. If \( C^* (t) \) is close to \( C \), i.e., \( D( C, C^* (t)) \) is smaller than the predefined threshold \( \epsilon \), or if the number of iterations \( t \) is greater than the maximum allowed number of iteration \( t_{max} \), stop; otherwise, if \( D( C, C^* (t)) \) is less than \( D^+ \) upgrade \( D^+ \) to \( D( C, C^* (t)) \) and \( M^+ \) to \( M(t) \).

5. Increment \( t \) by 1. Update each matrix \( M^+ \) by exploiting its unique decomposition into Cholesky factors (Yuan and Klir, 1997), \( M^+ = \{ G^+_i \}^T G^+_i \), where \( G^+_i \) is a lower triangular matrix with positive entries on the main diagonal (see also Section 5 below). More precisely, at iteration \( t \), the entries \( g_{l_1, l_2}^i (t) \) of the Cholesky factor \( G_i(t) \) are updated as follows:

\[
g_{l_1, l_2}^i (t) = g_{l_1, l_2}^{+ i} + N_{l_1, l_2}^i (0, \delta^+) \text{ if } l_1 < l_2,
\]

\[
g_{l_1, l_2}^i (t) = \max \left( 10^{-5}, g_{l_1, l_2}^{+ i} + N_{l_1, l_2}^i (0, \delta^+) \right) \text{ if } l_1 = l_2,
\]

where \( \delta^+ = \alpha D(\Gamma, \Gamma^+) \), \( \alpha \) is a parameter that controls the size of the random step of modification of the Cholesky factor entries \( g_{l_1, l_2}, N_{l_1, l_2}^i (0, \delta) \) denotes a gaussian noise with mean 0 and standard deviation \( \delta \), and Eq. (6) ensures that all entries in the main diagonal of the matrices \( G_i(t) \) are positive numbers and so \( M(t) \) are definite positive distance matrices.

6. Return to step 2.

In the end, at convergence, the evolutionary FCM algorithm provides the centers \( \bar{v}_i, i = 1, 2, \ldots, c \) of the \( c \) clusters and the membership values of the \( n \) training data to the \( c \) clusters. A new pattern \( \bar{x} \) is classified, in fuzzy terms, by computing its values of...
membership to the $c$ clusters, based on the Mahalanobis distances of matrix $M^+$ at convergence, knowing the centers $\tilde{v}_i, i = 1, 2, \ldots, c$.

Given the ordered correspondence between classes and clusters, the fuzzy membership information may be further used for the crisp assignment of the patterns to the various classes by setting a classification threshold $\gamma$: if the largest value of membership of the generic pattern $\tilde{x}$ to any of the $c$ clusters is greater than $\gamma$ then the pattern is assigned to the class corresponding to the cluster of largest membership; otherwise, if none of the memberships of $\tilde{x}$ is larger than $\gamma$, the pattern remains ambiguous, i.e., it is not assigned to any of the $c$ classes.

3. Application of the extended evolutionary FCM algorithm to an artificial two-dimensional data set

The developed algorithm has been applied to the artificial, two-dimensional data set of Fig. 2. These data comprise three physical classes A, B, C two of which, A and B, appear disjoint into two pairs of clusters, $A_1-A_2$ and $B_1-B_2$, when represented in the two-dimensional features space $(\tilde{x}_1, \tilde{x}_2)$.

In Zio and Baraldi (2004), these data were shown to be poorly classified by the original optimization algorithm of Yuan et al. (1995), based on a single Mahalanobis metrics for the three clusters to be associated to the three classes, whereas a

![Fig. 2. Artificial two-dimensional data set.](image-url)
significant improvement was obtained by splitting the less compact classes (A and B) into two subclasses each, thereby obtaining a total of five clusters.

The application of the new evolutionary optimized FCM classifier presented in Section 2, with \( t_{\text{max}} = 1000 \) iterations and \( \alpha = 0.08 \), leads to significantly improved results, with only three clusters. The distance of Eq. (4) between the a priori known data partition \( I \) and the optimal one \( I^* \) on the 3000 training data is of 0.1309. This result compares well with the values of 0.3641 and 0.1866 obtained in Zio and Baraldi (2004), for three and five clusters (after splitting), respectively, by means of the previous evolutionary algorithm with the same value of \( t_{\text{max}} \) and \( \alpha \) and only one Mahalanobis distance. When testing the performance of the new classifier on 750 data points different from those of training, the distance between the a priori known partition \( I \) and the optimal one is 0.1571 compared to a value of 0.3575 (with three clusters) and of 0.2073 (with five clusters, after splitting) obtained in Zio and Baraldi (2004).

Fig. 3 compares the performance on the 750 test data of the evolutionary FCM algorithm with one Mahalanobis metrics for all clusters, proposed in Yuan et al. (1995) and the extended version with one Mahalanobis metrics for each cluster here proposed. The efficiency of correct classification appears to be significantly improved: for example, with a degree of confidence \( \gamma = 0.85 \) the extended algorithm correctly classifies 65.87% of the test data (42.66% and 48.12% by the original algorithm with three and five clusters, respectively), misclassifies 1.47% (20.13% and 0.40% by the original algorithm with three and five clusters, respectively), and does not assign 32.66% (37.21% and 51.48% by the original algorithm with three and five clusters, respectively).

Fig. 4 shows the optimal geometric partition of points that are assigned to one of the three clusters with a membership value larger than 0.85.

It is interesting to analyze the cluster centers and metrics found by the algorithm. In particular, the centers of clusters A and B, \( \bar{v}_A \equiv (-0.03, 0.02) \) and \( \bar{v}_B \equiv (0.00, 0.01) \), respectively, turn out to be very close to each other and to the origin \( O(0,0) \) of the feature space. What distinguishes the different partition of the data points into clusters A and B are then the different metrics used for the two clusters. Indeed, for example, the distance between a generic point \( P \) and the origin \( O \) depends on which metrics is used: for example, the distance of the point \( P = (1,1) \) from \( O \) is approximately 0.39, when computed with the metrics associated to cluster A and 2.57 when computed with the distance associated to cluster B. Moreover, the off-diagonal terms of the matrix defining the metrics of cluster A are negative (−1.48) so that the corresponding cluster shape is rotated clockwise, whereas the off-diagonal terms of the matrix of cluster B are positive (1.60), with the cluster shape rotated correspondingly counterclockwise. This peculiar partition cannot be obtained by an unsupervised Fuzzy C Mean algorithm because, from a purely geometric point of view, points of subcluster A_1 have nothing in common with points of subcluster A_2 and, analogously, points of subcluster B_1 have nothing in common with points of subcluster B_2.

Concerning the metrics associated with cluster C, it is interesting to note that in the associated Mahalanobis matrix found by the optimization the lower-right
diagonal term corresponding to the input feature $x_2$ turns out to be nearly 0. From Eq. (3), this means that the distance of a point $\bar{x}$ to the cluster-C center $\bar{v}_C$ basically depends only on feature $x_1$. This is physically reasonable since, as it can be seen from Fig. 2, the feature $x_2$ is not discriminant for the assignment of a point $\bar{x}$ to cluster C rather than to A or B. This result suggests an interesting interpretation of the diagonal elements of the Mahalanobis matrix defining the metrics of a given cluster.

Indeed, as we shall see in the following Section 5, the value of these elements can be taken as a relative quantitative measure of the ‘importance’ of the associated feature for the clustering task at hand (Yuan et al., 1995). In particular, if a feature turns out to be not important for the definition of any of the clusters, it can be eliminated. In other words, the non-negligible values of the diagonal elements of the Mahalanobis metrics of the different clusters can guide a feature selection procedure (Kohavi and John, 1997) aiming at reducing the dimensionality of the patterns to be classified.

Fig. 3. Fraction of the 750 test data points correctly (top), incorrectly (middle) and not assigned (bottom) by the optimizing evolutionary FCM classifier with three clusters and a single Mahalanobis metrics (dashed line), the evolutionary FCM classifier with splitting into a total of five clusters (dotted line) and the extended classifier with three clusters and three Mahalanobis metrics (solid line), as a function of the classification threshold $\gamma$. 
4. Application of the extended evolutionary FCM classifier to nuclear transients data

Consider the problem of classifying transients occurring in the steam generator (SG) of a Pressurized Water Reactor (PWR). The well-known standard recirculation U-tubes type steam generator is considered. A detailed model of the physical functioning of this component is given in Marseguerra et al. (1996). Four kind of transients are considered, depending on the variation of the initiating forcing function. The four forcing functions here considered are: the inlet water temperature ($T_{1E}$) and the inlet water mass flow ($G_{1E}$) on the primary side, the feed water temperature ($T_{FW}$) and the feed water mass flow ($G_{FW}$) on the secondary side. These four forcing functions may vary as a consequence of an operator action or because of plant anomalies or faults. The transient classification is to rely on temperature, pressure and flow measurements by the available plant sensors. A total of 3500 labelled patterns have been simulated with a validated code (Höld, 1978), varying the kind and intensity of the forcing functions within the ranges and in the same manner as in Marseguerra et al. (2004). Each pattern consists of four features and is labelled with the corresponding forcing function whose variation has caused the transient.

A set of 2800 patterns has been used to train the classifier while the remaining 700 patterns have been left for testing its performance. On this basis, the classification results obtained in Zio and Baraldi (2004) by the original optimization algorithm of Yuan et al. (1995), based on a single Mahalanobis metrics for all the clusters were totally unsatisfactory. A significant improvement, but still insufficient for classification purposes was achieved by the evolutionary splitting FCM with eight clusters.

Fig. 4. Loci of points assigned with membership larger than 0.85 to clusters A, B and C.
On the contrary, the application of the extended evolutionary optimized FCM classifier with $t_{\text{max}} = 7000$, $\alpha = 0.09$ and four clusters, each one defined by its Mahalanobis metrics (Section 2), leads to an efficient clustering algorithm suitable for the direct classification of the transients. Fig. 5 compares the performance, on the 700 test patterns, of the evolutionary FCM algorithm with one Mahalanobis metrics for all clusters (Yuan et al., 1995) and the extended version with one Mahalanobis metrics for each cluster here proposed. The efficiency of correct classification appears to be significantly improved: for example, with a degree of confidence $\gamma = 0.85$ the extended algorithm correctly classifies 98.43% of the test data (37.40% and 70.00% by the original algorithm with three and eight clusters, after splitting, respectively), misclassifies 0.29% (70.00% and 7.70% by the original algorithm with three and eight clusters, after splitting, respectively), and does not assign 1.28% (31.60% and 22.30% by the original algorithm with three and eight clusters, after splitting, respectively).
5. A feature importance measure

In practical model-building scenarios, various reasons seem to justify a reduction in the number of features used for building a classification model (Na et al., 2002). First of all, irrelevant, non informative inputs may result in a model which is not robust. Second, studies have shown that for better classification performance it is necessary to remove highly correlated variables. Third, when the model is characterized by many input variables, a large number of observation data are required to properly span the high-dimensional input space for accurate multivariable interpolation. Finally, by eliminating unimportant sensors the cost and time of collecting the data can be reduced.

In Section 3, it was shown that there is a connection between the value of the diagonal elements of the matrices defining the cluster metrics and the “importance” of the features for the clustering task. In this Section, this point is further investigated in order to explicitly define a quantitative measure of the importance of a feature with respect to the classification task (Yuan and Klir, 1997).

Let $\tilde{x}_i, \tilde{x}_j$ be two vectors in the space $\Omega$ of features $x_1, x_2, \ldots, x_h$. As mentioned in Section 2, the positive definite matrix $M$, defining the Mahalanobis metrics used for clustering, can be uniquely decomposed in its Cholesky factors $G$:

$$M = G^T G.$$  \hspace{1cm} (7)

where $G = [g_{l_1,l_2}], l_{1,2} = 1, 2, \ldots, h$, is a lower triangular matrix with positive entries on the main diagonal. The Mahalanobis distance between $\tilde{x}_i$ and $\tilde{x}_j$ is expressed, using Eqs. (3) and (7), by

$$d_M(\tilde{x}_i, \tilde{x}_j) = [((\tilde{x}_i - \tilde{x}_j)^T M (\tilde{x}_i - \tilde{x}_j))]^{1/2} = [((\tilde{x}_i - \tilde{x}_j)^T G^T G (\tilde{x}_i - \tilde{x}_j))]^{1/2}. \hspace{1cm} (8)$$

By introducing a linear transformation, defined by the matrix $G$, from the space $\Omega$ to the space $\Omega'$,

$$\Omega \rightarrow \Omega', \quad \tilde{x} \rightarrow \tilde{x}' = G\tilde{x}. \hspace{1cm} (9)$$

Eq. (8) becomes

$$d_M(\tilde{x}_i, \tilde{x}_j) = [((\tilde{x}'_i - \tilde{x}'_j)^T (\tilde{x}'_i - \tilde{x}'_j))]^{1/2} = d_I(\tilde{x}'_i, \tilde{x}'_j). \hspace{1cm} (10)$$

Hence, the Mahalanobis distance between $\tilde{x}_i, \tilde{x}_j$ in the space $\Omega$ becomes the Euclidean distance between $\tilde{x}'_i, \tilde{x}'_j$ in the linearly transformed space $\Omega'$.

The components of the generic vector $\tilde{x}'$ in the space $\Omega'$ are given in terms of the components of $\tilde{x}$ in the space $\Omega$ by

$$x'_1 = g_{1,1}x_1, \quad x'_2 = g_{2,1}x_1 + g_{2,2}x_2, \quad \ldots$$

$$x'_h = g_{h,1}x_1 + g_{h,2}x_2 + \ldots + g_{h,h}x_h. \hspace{1cm} (11)$$
It can be seen that the role played by a feature $x_l$, $l = 1, \ldots, h$ of $\Omega$ in the transformed space $\Omega'$ is determined by the weighing coefficients $g_{r,l}, r = l, l+1, \ldots, h$, of the Cholesky factor $G$. For instance, if all the coefficients $g_{r,2}, r = 2, \ldots, h$ are approximately 0, the feature $x_2$ has no influence on the location of $\tilde{x}'$ in $\Omega'$ and, correspondingly, the Mahalanobis distance between two points in $\Omega$ (i.e., the Euclidean distance in $\Omega'$) does not depend on the values of feature $x_2$. Hence, it seems reasonable to measure the importance $I_M(x_l)$ of the feature $x_l$ by Yuan et al. (1995)

$$I_M(x_l) = \sum_{r=l}^{h} g_{r,l}^2,$$

where the square takes care of possible negative values. From the computational point of view, by construction, the quantities $I_M(x_l)$ are the $l$th entries on the main diagonal of the matrix $M$ defining the metrics.

Given that the classification procedure proposed in this work is driven by the Mahalanobis distances of a point to the cluster centers, with each cluster $i$ having its own Mahalanobis metrics $M_i$, if a feature $x_l$ turns out to be not important (low values of $I_M(x_l), i = 1, 2, \ldots, c$) for all Mahalanobis metrics $M_i$ associated to the clusters, the feature can be regarded as not important for the classification task and thus eliminated.

6. Conclusions

In the present work, we have extended an evolutionary algorithm for defining the optimal Mahalanobis metrics to be used within a generalized Fuzzy C Means clustering method. The proposed, supervised approach allows the identification of clusters of different shapes and orientations through the definition of individual Mahalanobis matrices, one for each cluster.

The algorithm has been verified with respect to an artificial classification problem in which two classes are actually completely disjoint into two pairs of clusters in a two-dimensional feature space. The algorithm is capable of representing each of the two dispersed classes by a single, appropriately oriented cluster.

The methodology has, then, been applied to the classification of simulated transients of a U-tube steam generator of a pressurized water reactor. The classification results have been very satisfactory.

Moreover, from the analysis of the identified optimal metrics it is possible to extract information on the importance of the features upon which the classification is based. This information may aid the selection of the important features to be considered for the classification task. Concerning this latter point, the identification of those measured plant parameters (features) which are most sensitive to faults and malfunctions is particularly important for nuclear power plants where hundreds of parameters are monitored for operation and safety reasons. A future research step in this direction concerns the investigation of the possibility of developing a feature selection algorithm based on the characteristics of the metrics selected by the
supervised optimal Fuzzy C Means, to remove irrelevant, non informative inputs to the classification model, thus improving its performance.

References


Identification of transients in nuclear systems by a supervised evolutionary possibilistic clustering approach

Enrico Zio, Piero Baraldi and Davide Mercurio

Accepted for publication on the Proceedings of the 7th International FLINS Conference on Applied Artificial Intelligence
29-31 August 2006
IDENTIFICATION OF TRANSIENTS IN NUCLEAR SYSTEMS
BY A SUPERVISED EVOLUTIONARY POSSIBILISTIC
CLUSTERING APPROACH

E. ZIO, P. BARALDI AND D. MERCURIO
Department of Nuclear Engineering, Polytechnic of Milan,
Via Ponzio 34/3, Milan, 20133, Italy

In this paper, the task of identifying transients in nuclear systems is tackled by means of a possibilistic fuzzy classifier devised in such a way to recognize the transients belonging to a priori foreseen classes while filtering out unforeseen plant conditions, independently from the operational state of the plant before the transient occurrence. The classifier is constructed through a supervised evolutionary procedure which searches geometric clusters as close as possible to the real physical classes.

The proposed approach is applied to the classification of simulated transients in the feedwater system of a boiling water reactor.

1. Introduction

Two important issues for the practical implementation of model-based fault diagnostic systems in Nuclear Power Plants (NPPs) regard the possibility of defining and controlling the boundaries of their utilization and their capability to diagnose a fault independently from the plant operational state before its occurrence [1].

In this work, these issues are tackled by means of a novel possibilistic clustering classifier. The possibilistic viewpoint considers the memberships to a given cluster as degrees of compatibility, or ‘typicality’ measured with respect to the cluster prototypical members [2, 3]. In this view, the memberships of representative (typical) patterns are high, while unrepresentative (atypical) points bear low membership to all clusters.

The approach embraced in this work exploits i) a possibilistic clustering algorithm for classifying the transients or labeling them as “unknown” if the associated feature values are located far away, in the feature space, from those characteristics of the training data; ii) a supervised evolutionary procedure for optimizing a different Mahalanobis metric for each of the possibilistic clusters by exploiting a priori known information regarding the true classes which a set of available labeled patterns belong to [4], [5].
The proposed possibilistic clustering scheme is verified with respect to a problem regarding the early identification of a predefined set of faults in a Boiling Water Reactor (BWR). The corresponding transients have been simulated by the HAMBO simulator of the Forsmark 3 BWR plant in Sweden [6].

2. The supervised evolutionary possibilistic clustering algorithm for classification

In this Section, a possibilistic clustering algorithm is developed to perform the diagnostic identification of transients.

The traditional, unsupervised possibilistic algorithm based on a Euclidean metric to measure compatibility leads to spherical clusters that rarely are adequate to represent the data partition in practice. A significant improvement in classification performance is achieved by considering a different Mahalanobis metric for each cluster, thus obtaining different ellipsoidal shapes and orientations of the clusters that more adequately fit the a priori known data partition [4,5].

The information on the membership of the available patterns $x_k$, $k=1,...,N$, to the $c$ a priori known classes, can be used to supervise the algorithm for finding the optimal Mahalanobis metrics such as to achieve geometric clusters as close as possible to the a priori known physical classes. Correspondingly, the possibilistic clustering algorithm is said to be constructed through an iterative procedure of ‘training’ based on a set of available patterns, pre-labeled with their possibilistic memberships to the a priori classes. The training procedure for the optimization of the metrics is carried out via an evolutionary procedure, presented in the literature within supervised fuzzy clustering schemes [4] and further extended to diagnostic applications [5]. Here, the procedure is employed within the possibilistic clustering scheme [2].

The target of the supervised optimization is the minimization of the distance $D(\Gamma', \Gamma)$ between the a priori known physical class partition $\Gamma' = (\Gamma'_1, \Gamma'_2, ..., \Gamma'_c)$ and the obtained geometric cluster partition $\Gamma = (\Gamma_1, \Gamma_2, ..., \Gamma_c)$:

$$D(\Gamma', \Gamma) = \sum_{i=1}^{c} \frac{D(\Gamma'_i, \Gamma_i)}{c} = \sum_{i=1}^{c} \sum_{k=1}^{N} \frac{|\mu'_{ik} - \mu_{ik}|}{N \cdot c}$$  (1)\

where $0 \leq \mu'_{ik} \leq 1$ is the a priori known (possibilistic) membership of the $k$-th pattern to the $i$-th physical class and $0 \leq \mu_{ik} \leq 1$ is the possibilistic membership to the corresponding geometric cluster in the feature space.
The overall iterative training scheme can be summarized as follows:

1. At the first iteration ($\tau = 1$), initialize the metrics of all the $c$ clusters to the Euclidean metrics, i.e. $M_i(l) = I$, $i = l, 2, ..., c$, where $I$ is the identity matrix.

2. At the generic iteration step $\tau$, run the possibilistic clustering algorithm [2] to partition the $N$ training data into $c$ clusters of memberships $\Gamma (\tau) = \{ \Gamma_1(\tau), ..., \Gamma_c(\tau) \}$, based on the current metrics $M_\tau$ and on the “supervising” initial partition $\Gamma^*$ which sets the initial memberships of the $N$ patterns to $c$ clusters equal to the true memberships to the a priori known classes.

3. Compute the distance $D(\Gamma^*, \Gamma(\tau))$ between the a priori known physical classes and the geometric possibilistic clusters by eq.(1). At the first iteration ($\tau = 1$) initialize the best distance $D^+$ to 0, $D_i^+$ to $iM_\tau$ and the best metrics $M_i^+$ to $M_1(1)$ and go to step 5.

4. If $\Gamma(\tau)$ is close to $\Gamma^*$, i.e. $D(\Gamma^*, \Gamma(\tau))$ is smaller than a predefined threshold $\varepsilon$, or if the number of iterations $\tau$ is greater than the predefined maximum allowed number of iterations $\tau_{\max}$, stop: $\Gamma(\tau)$ is the optimal cluster partition $\Gamma^*$; otherwise, if $D(\Gamma^*, \Gamma(\tau))$ is less than $D^+$ upgrade $D^+$ to $D(\Gamma^*, \Gamma(\tau))$, $M_i^+$ to $M_\tau$ and $D_i^+ = D(\Gamma^*, \Gamma_i(\tau))$.

5. Increment $\tau$ by 1. Update each matrix $M_i^+$ by exploiting its unique decomposition into Cholesky factors [5], $M_i^+ = (G_i^+)^T G_i^+$, where $G_i^+$ is a lower triangular matrix with positive entries on the main diagonal. More precisely, at iteration $\tau$, the entries $g_{i,l}^\tau (\tau)$ of the Cholesky factor $G(\tau)$ are updated as follows:

\[
g_{i,l}^\tau (\tau) = g_{i,l}^{i+} + N_{i,l}^\tau (0, \delta_i^\tau) \quad \text{if } l_1 < l_2
\]

\[
g_{i,l}^\tau (\tau) = \max \left( 10^{-5} g_{i,l}^{i+} + N_{i,l}^\tau (0, \delta_i^\tau) \right) \quad \text{if } l_1 = l_2
\]

where $\delta_i^\tau = \alpha D_i^\tau$, $\alpha$ is a parameter that controls the size of the random step of modification of the Cholesky factor entries $g_{i,l}^{i\tau}$, $N_{i,l}^\tau$ denotes a Gaussian noise with mean 0 and standard deviation $\delta$, and eq.(3) ensures that all entries in the main diagonal of the matrices $G_i(\tau)$ are positive.
numbers and so $M_i(\tau)$ are definite positive distance matrices. Notice that
the elements of the $i$-th Mahalanobis matrix are updated proportionally to
the distance $D_i^\tau$ between the $i$-th a priori known class and the $i$-th cluster
found. In this way, only the matrices of those clusters which are not
satisfactory for the classification purpose are modified.

6. Return to step 2.
At convergence, the supervised evolutionary possibilistic clustering algorithm
provides the $c$ optimal metrics $M_i^*$ with respect to the classification task, the
possibilistic cluster centers $\hat{v}_i^*$ and the possibilistic membership values $\mu_ik^*$ of
the patterns $\hat{x}_k$, $k=1,..,N$, to the clusters $i=1,2,..,c$.

When fed with a new pattern $\hat{x}$, the classification algorithm provides the
values of the membership functions $\mu_i^*(\hat{x})$, $i=1,2,..,c$, to the possibilistic
clusters. These values give the degree of compatibility or “typicality” of $\hat{x}$ to
the $c$ clusters. In practice, three situations may arise: i) $\hat{x}$ does not belong to any
cluster with enough membership, i.e. all the membership values $\mu_i^*(\hat{x})$ are
below a given threshold $\epsilon_f$ (degree of ignorance): this means that $\hat{x}$ is an
atypical pattern with respect to the training patterns; ii) at least two membership
values are above the threshold $\epsilon_f$ (degree of confidence): $\hat{x}$ is thus ambiguous.
In this case, the ambiguity must be regarded as “equal evidence”, i.e. the pattern
is typical of more than one class and thus cannot be assigned to a class with
enough confidence. This situation occurs if $\hat{x}$ is at the boundary between two
classes. iii) $\hat{x}$ belongs only to a cluster with a membership value greater than
the threshold $\epsilon_f$: in this case, it is assigned to the corresponding class.

3. Application of the possibilistic classifier to the identification of
nuclear transients

In this Section, the possibilistic classifier described in the previous Section is
applied to the early identification of a predefined set of faults in the feedwater
system of a Boiling Water Reactor (BWR) (see [7] for a detailed description of
the faults). The corresponding transients have been simulated by the HAMBO
simulator of the Forsmark 3 BWR plant in Sweden [6].

Here, the diagnosis considers three power operation levels, i.e. 50%, 80%
and 108% of full power. Transient data were made available for each of the
fault types, with varying degrees of leakage and valve closures. All transients
start after 60 seconds of steady state operation.
Among the 363 measured signals, only 5 signals, i.e. Temperature of drain 4 before valve VB3, Water level of tank TD1, feedwater temperature after preheater EA2, feedwater temperature after preheater EB2, Position level of control valve for preheater EA1, have been chosen for the transient classification using the feature selection algorithm proposed in [8].

3.1. Case study 1: filtering out unknown transients

In this analysis, the patterns used for building the classification system have been taken from the six faults F1, F2, F3, F4, F5 and F7 that regard line 1 of the feedwater system [7]. For each type of fault, the simulated transients with the plant at 80% of full power have been considered, taking patterns every 6 seconds from $t = 80s$ to $t = 200s$.

After the training of the possibilistic classifier, its performance has been tested using patterns taken every second from $t = 0s$ to $t = 300s$ from both the training transients and from an unknown transient caused by F13.

Figure 1 shows the obtained transient classification as time progresses. Considering a degree of confidence $\varepsilon_c = 0.7$ and a degree of ignorance $\varepsilon_f = 0.2$, the results are quite satisfactory, even though at the beginning of the transient the possibilistic classifier assigns the steady state patterns to the class of fault F2 albeit with low membership. This is explained by the fact that for transients of class F2 there are no significant effects on the selected input signals so that understandably the steady state may be confused with a fault of class 2.
Also note that, the possibilistic classifier is able to assign to the right class the foreseen transients also at times well beyond the temporal domain of training of 200s, due to the increased significance of the signals as the transients continue evolving away from their initial steady state.

Finally, the algorithm is very efficient in filtering out the patterns of the unknown fault F13 as atypical, by assigning them membership values to all classes less than $\varepsilon_f = 0.2$ (Figure 1, bottom).

3.2. **Case study 2: classification of transients at different power levels**

In this Section, the capability of the classifier to identify faults that initiate from different plant operational conditions is investigated. In this respect, the possibilistic classifier is trained using patterns taken from classes F1, F2, F3, F4, F5 and F7 at 50 % and 108 % power whereas in the test phase also patterns taken at 80 % power are considered.

From each of the 12 training transients considered (6 transients for each of the 2 power levels), patterns taken every 6 seconds from 80s to 200s have been used, for a total of 252 patterns.

The performance of the classifier has been tested using the training transients (belonging to the foreseen classes of faults at 50 % and 108 %) and the new transients (belonging to the same classes of faults but at 80 % power), taking a pattern every second from 0s to 300s. The behavior of the memberships with respect to all the classes at 80% power is shown in Figure 2.

Considering a degree of confidence $\varepsilon_c = 0.7$ and a degree of ignorance $\varepsilon_f = 0.2$, the performance of the classification at the new power level 80 % is very satisfactory for the first five classes of faults whereas the transients caused by the fault F7 at 80 % is filtered as atypical at all times. This happens even for the training patterns of class F7 taken at 50 % of full power.

To further investigate this situation, a sensitivity analysis has been performed, based on the technique reported in [5]. Figure 3 shows the disposition of the patterns in the subspace formed by two of the three signals identified as most important (signal 320 and signal 195).
In this case, the evolutionary algorithm cannot find an optimal metric $M^*$ that results in a cluster that contains exclusively all the patterns of class F7 without containing those patterns of the other classes which are close to the patterns of class F7 at 50% power level. The reason for this is that two patterns having nearly the same distance from the center of cluster 7 (+), for example pattern A of class F7 and pattern B of class F5 in Figure 3, have nearly the same membership values to class F7, measuring their compatibility with it (eq. A2). In this situation, it is impossible to have possibilistic clusters with sharp borders and the target of the evolutionary algorithm of minimizing the distance...
between the true known physical class memberships \( \Gamma_{\gamma} \), and the possibilistic cluster memberships \( \Gamma_{\gamma} \), is better satisfied by a small cluster centred on the patterns of class F7 at 108 % than by a big cluster that contains all patterns of class F7 but also patterns of other classes.

4. **Conclusions**

In the present paper a supervised, evolutionary possibilistic clustering algorithm is proposed for building a diagnostic system for the classification of transients in nuclear systems. Within the possibilistic clustering scheme, a supervised evolutionary algorithm finds a Mahalanobis metric for each cluster which is optimal with respect to the classification of an available set of labeled patterns.

The approach distinguishes from the existing ones because it allows identifying the boundaries of application of the classification model so that ‘unbeknownst’ transients are filtered out. Also it is capable of correctly classifying faults that occur with the plant in operating conditions different from those of the patterns used for the construction of the classifier model.

The approach has been successfully verified with respect to the classification of simulated nuclear transients in the feedwater system of a BWR plant.

**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**