Classifier-Ensemble Incremental-Learning Procedure for Nuclear Transient Identification at Different Operational Conditions

Piero Baraldi, Roozbeh Razavi-Far, Enrico Zio

Classifier-Ensemble Incremental-Learning Procedure for Nuclear Transient Identification at Different Operational Conditions

Piero Baraldi [1] , **Roozbeh Razavi-Far** [1] , **Enrico Zio** [1,2]

[1] *Dipartimento di Energia - Sezione Ingegneria Nucleare, Politecnico di Milano, Via Ponzio 34/3, 20133 Milano, Italy*

[2] *Ecole Centrale Paris-Supelec, Paris, France*

Abstract: An important requirement for the practical implementation of empirical diagnostic systems is the capability of classifying transients in all plant operational conditions. The present paper proposes an approach based on an ensemble of classifiers for incrementally learning transients under different operational conditions. New classifiers are added to the ensemble where transients occurring in new operational conditions are not satisfactorily classified. The construction of the ensemble is made by bagging; the base classifier is a supervised Fuzzy C Means (FCM) classifier whose outcomes are combined by majority-voting. The incremental learning procedure is applied to the identification of simulated transients in the feedwater system of a Boiling Water Reactor (BWR) under different reactor power levels.

Keywords: Classification; Fuzzy C-Means (FCM) Clustering; Bagging; Ensemble; Incremental Learning; BWR Nuclear Power Plant; Transient Identification.

1. Introduction

Monitoring is a continuous real-time task of determining the conditions of a physical system, by recording information, recognizing and indication anomalies in the behavior (Simani et al., 2002). A fault diagnosis system is a monitoring system that is used to detect faults and diagnose their location and significance in a system (Chen and Patton, 1999). The diagnosis system performs the following tasks: fault detection—to indicate if a fault occurred or not in the system, fault isolation—to determine the kind, location and time of detection, and fault identification—to estimate the size and nature of the fault. The first two tasks of the system: fault detection and isolation are considered the most important. Fault diagnosis is then very often considered as fault detection and isolation (Simani et al., 2002). Here the term diagnosis indicates recognizing and indication transients and anomalies in the system behavior.

A number of diagnostic methods based on the advances of soft computing have been proposed for transient identification in nuclear systems (Hines, et al., 1996; Reifman, 1997; Embrechts and Benedek, 2004; Na, et al., 2004; Evsukoff and Gentil, 2005; Zhao and Upadhyaya 2005; Zio and Baraldi, 2005; Razavi-Far, et al., 2009). However, one factor that has limited their practical application is the difficulty of recognizing transients at different plant operational conditions, e.g. power levels (Uhrig, 1999). The objective of the present work is to overcome this limitation by proposing a procedure of empirical classification by incremental learning of transients at different plant conditions. The procedure is realistically applicable, with new examples of transients in new operational conditions becoming gradually available in time. Since the proposed classification procedure is based on the use of supervised learning algorithms, it requires a training phase in which some examples of patterns formed by the signal measurements observed during the transients (input) and the corresponding class (output) are available. Furthermore, we assume that the classes of the transients do not change at the different plant conditions, i.e. there can be a modification of the relationship between the inputs and the output, but the algorithm cannot be used to classify new classes of patterns.

One approach for learning new data (e.g. coming from new transients of new generated conditions) involves discarding the existing classifier and retraining a new one using all data (e.g. of all transients) that have been accumulated thus far. This approach, however, results in loss of all previously acquired information, a phenomenon known as catastrophic forgetting (Polikar, 2006), and may be infeasible for real diagnostic systems due to the computational and financial efforts necessary for each model retraining. In order to avoid retraining a new model each time a new dataset becomes available, the classification algorithms must be able to learn the novel information content of the new data without forgetting the previously acquired knowledge. A further desiderata, important in those cases in which the datasets previously used for the model training may be lost, corrupted or otherwise unavailable, is the possibility of updating the model without requiring access to the previously seen datasets.

The ability of a classifier to learn under these circumstances is usually referred to as incremental learning (Polikar et al., 2001).

A further challenge comes from the fact that the input (transient data of the measured signals) – output (corresponding fault class that originates the transients) relationships may change in different operational conditions. This problem is usually termed learning in a non-stationary environment. In particular, in the problem addressed in this work, the modification of the environment is assumed to be cyclic, since the plant usually returns to work in one operative condition previously left.

Recently, multiple classifier-based algorithms have been proposed for incremental learning in non-stationary environments. The proposed algorithms generate and then combine an ensemble of classifiers, where each classifier is trained on a different snapshot of the data. In particular, the following types of ensemble-based approaches have been distinguished (Kuncheva, 2004): (i) a fixed ensemble whose combination rules (weights) are changed based on the changing environment (dynamic combiners); (ii) an ensemble where new data are used to update some of the classifiers thanks to an on-line learning algorithm; (iii) a new ensemble structure obtained by altering the old ensemble structure.

In this work, the third approach is embraced within a procedure of modification of the ensemble structure when the classification in the current environment is not satisfactory. Firstly, an ensemble of classifiers is built using the datasets available. The ensemble is built according to the method proposed in (Baraldi et al., 2010): the base classifier is a supervised Fuzzy C Means classifier (Zio and Baraldi, 2005); an ensemble of them is built on different bagging sets of the available data (Breiman, 1996); the single classifiers outcomes are combined using a majority-voting scheme (Parhami, 1994). When the plant starts working in a new operational condition and a corresponding new dataset of data becomes available, the classification performance of the ensemble previously built is verified and, in the case in which it is not satisfactory (i.e. if the fraction of patterns correctly classified is lower than a fixed threshold), the ensemble is updated by adding new classifiers. This is actually done by creating an additional ensemble of classifiers, each base classifier trained by using a different bagging set of the new dataset. Finally, the old and the new ensembles of classifiers are merged into a single ensemble of classifiers formed by all the classifiers of the new and old ensembles. The procedure is repeated each time a new dataset describing a new operational condition becomes available.

The capability of the overall ensemble system to identify faults that initiate from different plant operational conditions has been tested on an application regarding the identification of simulated transients occurring at different reactor power levels in the feedwater system of a Boiling Water Reactor (BWR) (Puska and Norman, 2002).

The remainder of the paper is organized as follows. Section 2 presents a brief description of the problem statement of incremental learning in a non-stationary environment. Section 3 illustrates an ensemble-based scheme for incremental learning, describing the method and algorithm. Section 4 describes how the ensemble-based scheme is used for transient identification in the feedwater system of a BWR at different power levels. Finally, conclusions are drawn in Section 5. For completeness: the procedure of ensemble construction and its algorithm are reported in Appendix A and the supervised, evolutionary-optimized FCM clustering algorithm used to train the base classifiers of the ensemble is briefly described in Appendix B.

2. Incremental learning in a non-stationary environment

Let us consider a plant which can work in several different operational conditions. We assume that at time t_j, $j=1,...,n$, a dataset S^j formed by N^j patterns $\left(\vec{x}_k^j, \Gamma_k^j\right)$ becomes available, with $\vec{x}_k^j$ representing the generic *k*-th signal measurements observed during a transient, and $\Gamma_k^j, =1,\ldots,c$ the label assigned to the corresponding class of the transient. Notice that the total number of possible classes of the patterns $\vec{x}_k^j$ is assumed to be fixed and equal in all the datasets S^j. In general, the unknown mapping function between $\vec{x}_k^j$ and Γ_k^j may vary in different operational conditions, i.e. the class boundaries in the input space may be different in the different datasets S^j, which contain transients occurring with the plant in different operational states.

The final objective of the present work is to develop a classification algorithm able to correctly classify transients of the signal measurement vector $\vec{x}$, independently from the plant operational conditions.

3. An ensemble-based procedure for incremental learning in a non-stationary environment

The idea underlying ensemble-based classification is to create many classifiers and combine their outputs in a way to improve the performance of a single classifier. This requires that individual classifiers perform well in different regions of the feature space and make errors on different patterns, which are balanced out in the combination. Intuitively, if each classifier makes different errors, then a strategic combination of these classifiers can reduce the total error. The overarching principle in ensembles is therefore to make each classifier as unique as possible, particularly with respect to misclassified instances. Specifically, we need classifiers whose decision boundaries are sufficiently different from those of others (Polikar, 2006).

Various techniques have been suggested for obtaining diversity in the base models of an ensemble, e.g. using different training parameters (Hansen and Salamon, 1990), different training patterns (Breiman 1996), different feature subsets (Zio et al., 2008) and different learning methods for each classifier of the ensemble (Xu et al., 1992).

Here, the approach adopted in (Baraldi et al., 2010) where different training patterns are used to train individual classifiers is briefly described. The datasets are obtained through the resampling technique of bagging (Breiman, 1996) from a dataset containing all the available training patterns. Bagging, short for bootstrap aggregating, is one of the earliest ensemble-based algorithms (Breiman, 1996; Breiman, 1999) and is based on the random sampling of the datasets, usually with replacement, from the entire training dataset. The main structure of this ensemble construction scheme is shown in Figure 1. With respect to the construction of the base classifier of the ensemble, the supervised FCM algorithm is considered (Zio and Baraldi, 2005). In this classification algorithm, the information regarding the known, physical class of the *k*-th pattern is used to supervise an evolutionary algorithm for finding *c* optimal Mahalanobis metrics which define *c* geometric clusters as close as possible to the a priori known physical classes (Yuan and Klir, 1997). The Mahalanobis metrics are defined by the matrices , whose elements are identified by the supervised evolutionary algorithm so as to minimize the distances between the patterns belonging to class and the class prototype, i.e. the cluster center . The iterative training scheme is summarized in Figure xyz. Once the classifier is constructed, a new test pattern $\vec{x}$ is classified, in fuzzy terms, by computing its value of membership to the c clusters, based on the Mahalanobis distances of matrices $\underline{\underline{M}}_{i_c}^{+}$. Given the ordered corresponding between classes and clusters the fuzzy membership information is finally used for the crisp assignment of the pattern $\vec{x}$ to the class with the largest value of membership.

The performance of the overall bagging ensemble approach has been verified by comparison with a single supervised, evolutionary-optimized FCM classifier with respect of the task of classifying artificial and nuclear transient datasets. The results obtained indicate that in the cases of datasets of large or very small sizes and/or complex decision boundaries, the proposed bagging ensemble improves the classification accuracy. However, the bagging approach does not allow incremental learning in a non-stationary environment since it requires that all the training patterns, which are used for training the ensemble base classifiers, be available in advance.

In order to overtake this limitation of the bagging algorithms such as (Baraldi et al., 2010), the basic idea of the procedure proposed in this work for adding the capability of incremental learning is to add new classifiers to an ensemble of classifiers whenever the current classification performance is not satisfactory due to the modification of the environment. This approach allows the ensemble to learn new information, without forgetting the previously acquired knowledge which is contained in the old classifiers which are kept in the ensemble. To control the proliferation of classifiers in the ensemble, new classifiers are added only if the transients occurring in the new operational condition are not satisfactorily classified. According to this procedure, diversity in the base models of the ensemble is obtained by using different training patterns. However, notice that the approach differs from the bagging approach in (Baraldi et al., 2010) since the different training datasets are not all obtained from the same dataset, but they come from different datasets corresponding to different operational conditions.

The diagnostic system is developed according to the following steps (Figure 2.b):

1) Fix the minimum classification performance p^* which is always required to the diagnostic system and a fraction τ indicating the maximum performance reduction which is acceptable when the diagnostic system is used to classify patterns corresponding to different operational conditions from those used to train the ensemble system.

2) At time t_1 when dataset S^1 becomes available, an ensemble system E^1 is built. The ensemble is formed by T^1 base classifiers h_l^1, $l = 1,...,T^1$, built using bootstrapped replicas of the training data S_{train}^1. In particular, T^1 bagging iterations are performed, each one based on:

1.a) Resampling: the creation of a new training dataset by randomly drawing, with replacement, a fraction *F* of the training patterns contained in S^1_{train}. To ensure that there are adequate training samples in each subset, relatively large portions of the samples (*F*=0.75 – 1.00) are drawn into each subset. This causes individual training subsets to overlap significantly, with many of the same instances appearing in most subsets, and some instances appearing multiple times in a given subset. The algorithm used to construct the bagging ensemble is briefly reported in *Appendix A*.

1.b) Training: the building of a supervised, evolutionary-optimized FCM Classifier using the training data obtained in 1.a) and the procedure reported in Figure xyz.

3) The performance p^1 achieved by the first ensemble E^1 on the patterns of the dataset S^1_{test} not used for the training of the ensemble classifiers in step 1 is computed. In this respect, the test patterns are classified by the T^1 supervised FCM classifiers of the ensemble and the response of the single classifiers are aggregated according to the majority voting method (Appendix 1).

4) The achieved performance p^1 is compared to a minimum required classification performance $p^*(1) = p^*$. If the performance p^1 of E^1 is not satisfactory, i.e. $p^1 < p^*$, e.g. due to the presence of insufficient or poor quality data in S^1, at time t_2 when dataset S^2 becomes available go to 7) in order to try to increase the classification performance by adding new classifiers to the ensemble, otherwise the diagnostic algorithm $E = E_1$ can be used for transient classification until time t_2.

5) At each time t_j, $j = 2,\ldots,n$, the classification performance p^j achieved by the previously developed ensemble E^{j-1} on patterns of the new available test dataset S^2_{test} is computed.

6) The achieved performance p^j is compared to $p^*(j) = \max(p^*, (1-\tau)\cdot p^{j-1})$. The term $(1-\tau)\cdot p^{j-1}$ is considered in order to guarantee that the classification performance does not remarkably decrease when the diagnostic system is used for the classification of transient occurring at different operational conditions. If the classification performance p^j is satisfactory, i.e. $p^j \geq p^*(j)$ the ensemble system is left unmodified. In this case at time t_{j+1} when the new dataset S^j becomes available, go to 5). Otherwise, if $p^j < p^*(j)$, go to 7) in order to modify the ensemble. This usually occurs when the operational conditions verified in S^j are significantly different from those previously experimented.

Codice campo modificato

7) The ensemble system is updated by adding T^j base classifiers h^j_l, $l = 1,\ldots,T^j$, trained with bootstrapped replicas of S^j according to the procedure in 1a) and 1b) applied to the data of S^j. Thus, the obtained classification model *E* is an ensemble system formed by the union of the previous classifiers of *E* and the T^j classifiers newly added, i.e. an 'ensemble of ensembles'.

The main structure of the proposed incremental learning scheme is presented in Figure 2.a.

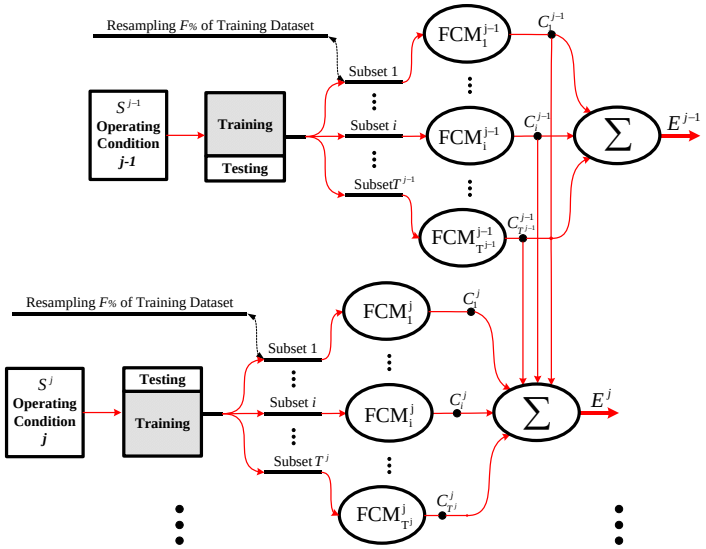


Figure 2.a. Main structure of the 'ensemble of ensembles' scheme for incremental learning

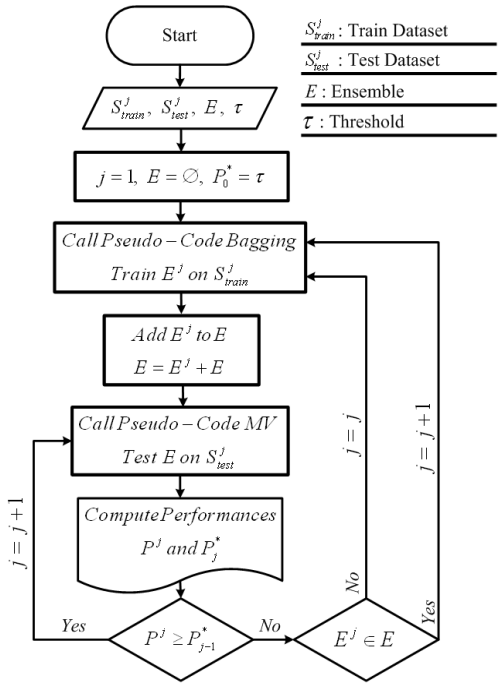


Figure 2.b. Flowchart of the classifier-ensemble incremental-learning procedure

Considering a generic time instant t at which an incremental learning ensemble system has been developed, the classification of an incoming new test pattern $\vec{x}$ is done by using the majority voting method, i.e. the class label which is supported by the majority of the individual classifiers is assigned to $\vec{x}$. In case the number of votes to different classes is equal, the class is assigned randomly among those classes with largest total votes.

4. Application to nuclear transient identification

In this Section, the capability of the proposed procedure is tested with respect to the classification of transients in the feedwater system of a BWR. The diagnosis considers three power operational levels, i.e. 50%, 80%, and 108% of full power. The corresponding transients have been simulated by the HAMBO simulator of the Forsmark 3 BWR plant in Sweden (Puska and Noemann, 2002).

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 3 shows a sketch of the system. A set of six faults, F1-F6, that are generally hard to detect for an operator have been chosen for this application (see (Roverso 2004) for their description).

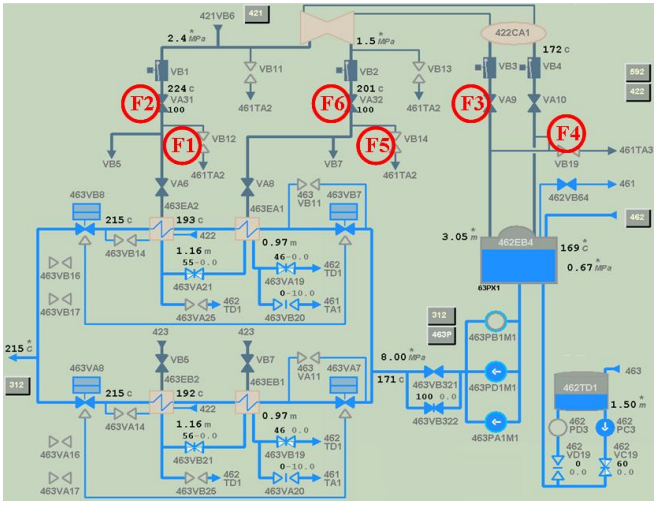


Figure 3: A sketch of the feedwater system of the BWR

Among the 363 measured signals, only the 5 reported in Table 1 have been used for the fault classification in the two case studies here considered. These signals have been chosen considering the results of the application of a feature selection algorithm and some benchmark tests (Zio et al., 2006).

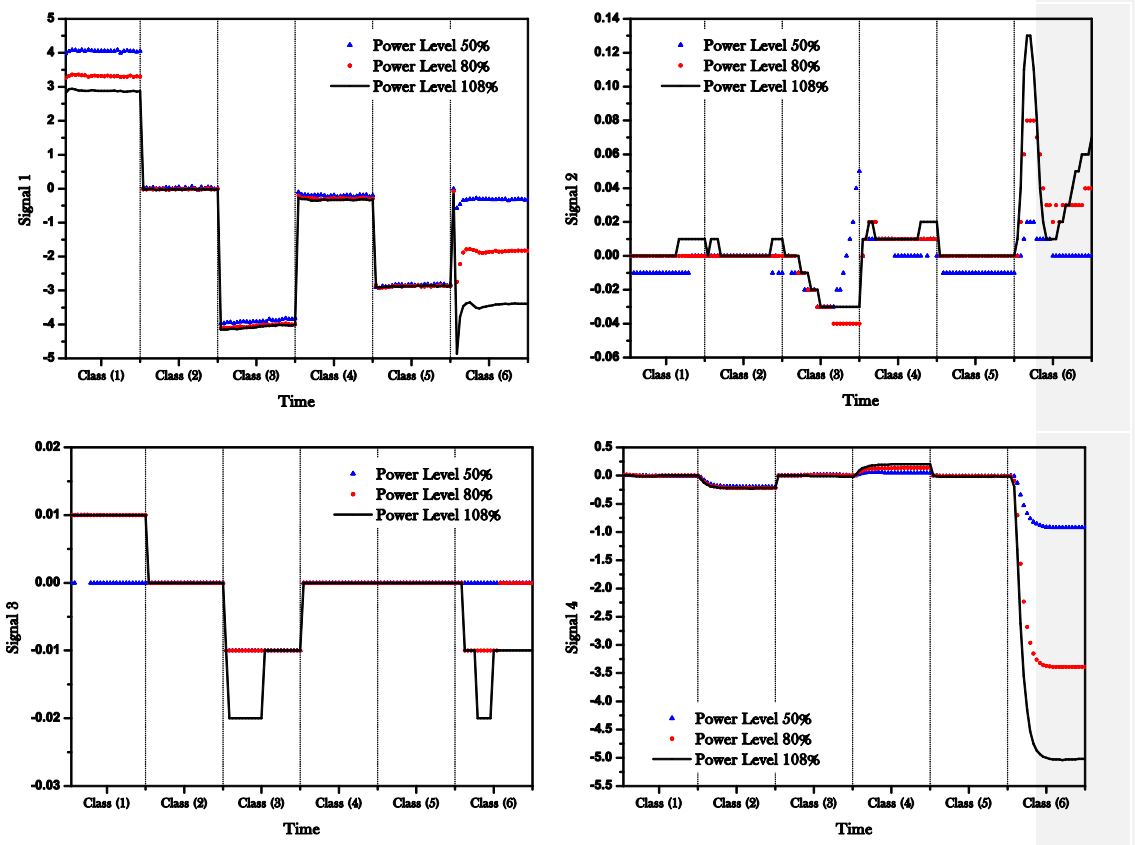
Table 1: Signals selected for the fault classification.

Signal Number – Name	Unit
1- Temperature drain 4 before VB3	$^{\circ}C$
2- Temperature feedwater before EB2 train B	$^{\circ}C$
3- Temperature after EA1 (high pressure preheater A1)	MPa
4- Temperature of condensate after EB2 train A	$^{\circ}C$
5- Position valve for level I EB4	%

Three datasets, S^{50}, S^{80}, S^{108} have been considered, containing patterns taken from transients simulated with the plant working at 50%, 80% and 108% of full power, respectively. More specifically, each dataset is formed by 1800 patterns taken from three transients for each of the 6 faults, differing in the degrees of leakage and valve closure. The data relative to the

selected 5 signals were recorded with a sampling frequency of 1 Hz. All transients start after 60 seconds of steady state operation. Given that the goal is early fault diagnosis, only the data from 70 seconds after the beginning of the transients have been considered for each transient.

Figure 4 shows the time evolution of the 5 features in transients of the 6 classes at the three power levels. Notice that signal variations are different at the different power levels, and more pronounced when the reactor is working at high power.



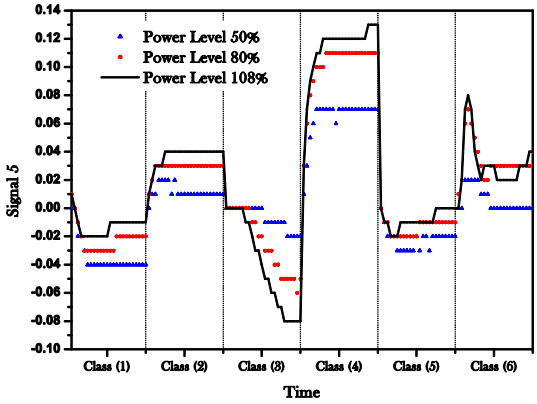


Figure 4: Time evolution of the signals at different power levels in case of transients of classes F1 – F6.

To test the incremental learning capability of the proposed algorithm under different operational conditions, it has been supposed that the datasets S^{50}, S^{80}, S^{108} become available at different time instants t_{50}, t_{80}, t_{108}; a fraction equal to 75% of the patterns of each dataset ($S^{50}_{train}, S^{80}_{train}, S^{108}_{train}$), has been used to train the models whereas the remaining 25% has been used to test the classification performance ($S^{50}_{test}, S^{80}_{test}, S^{108}_{test}$).

4.1. Case study 1: increasing power level

In this case study, the power level is increased firstly from 50% to 80% of the full power and then from 80% to 108%, at $t_{50} < t_{80} < t_{108}$.

The procedure for incremental learning in a non-stationary environment is applied as follows. At t_{50}, an ensemble E^1 is constructed using the data in S^{50}_{train}.

Although an overall investigation of the influence of the parameters used to build the ensemble on the classification performance is outside the scope of the present work, some considerations on the possible choices are here given. The two parameters of the ensemble $\{F, T\}$, the fraction of the total number of training patterns in S^{50}_{train} randomly drawn to create the single classifier training set and the number of ensemble classifiers have been fixed following a trial and error procedure. The results of tests performed by the authors have shown that the key issue to guarantee high performance of the ensemble is the diversity between the ensemble classifiers. In particular, since a low value of F leads to training sets with few common patterns, high performances can be obtained by reducing F, and, at the same time, increasing the number of classifiers T in order to properly cover all the training space. Notice, however, that since the computational efforts necessary to develop the diagnostic system is directly proportional to T, the choice of the parameters $\{F, T\}$ results from a compromise between high performance (low F, high T) and reduced computational effort (high F, low T).

The choice of the minimum classification performance, p^*, is usually guided by requirements of the diagnostic system users. In this application, since the diagnostic system is devoted to the classification of faults which mainly produce efficiency losses if undetected, p^* is set to 0.95. With respect to the parameter τ indicating the maximum fraction of performance reduction which is acceptable when the diagnostic system is used to classify patterns corresponding to different operational conditions from those used to train the ensemble system, notice that a too low value of τ will risk to cause the updating of the ensemble each time a new dataset becomes available with consequent high computational effort. In this respect, a value of τ equal to 0.05 has been used.

Table 2 reports the basic parameters used in this work to build the ensemble of classifiers. The obtained performance in the classification of the test patterns of S^{50} is 96.67% (Table 3).

Table 2: basic parameters used to build the ensembles of classifiers.

T^j	"Number of base classifiers in the ensemble"	10
F	"fraction of the total number N^j of training patterns which constitute each bootstrapped replica of S^j_{train}"	0.75
$\tau_{\max}$	"Number of iterations of the supervised algorithm used to train the single base classifiers of the ensemble"	500

At t_{80}, the dataset S^{80} becomes available and the performance of the previously developed ensemble E^1 is tested with the patterns of S^{80}_{test}: the fraction of patterns of S^{80}_{test} correctly classified is satisfactory (Table 3, first row, fifth column), so that it is not necessary to add classifiers to the ensemble structure i.e. $E^2 = E^1$. The same occurs at t_{108} when the dataset S^{108} becomes available, i.e. $E^3 = E^2 = E^1$ (Table 3, first row, sixth column).

Thus, in this case, it has not been necessary to update the first ensemble to learn the newly arriving information under different operational conditions: the ensemble constructed with data taken from transients occurring when the plant is working at 50% of full power is satisfactorily performing on transients at 80% and 108% of full power.

Table 3: Performances and training computational time of the ensemble-based approaches in the classification of the test patterns: proposed approach for incremental learning in a non-stationary environment (first row), ensemble obtained by retraining all the base classifiers (second and third rows).

Approach	*Train Dataset*	*Training time (min)*	*Test Dataset* S^{50}_{test}	S^{80}_{test}	S^{108}_{test}
Incremental Learning Ensemble - $E^1 = E^2 = E^3$	S^{50}_{train}	**51**	**0.9666**	**0.9606**	**0.9659**
Retraining - $E^{1,2}$	$S^{50}_{train}, S^{80}_{train}$	**100**		**0.9694**	
Retraining - $E^{1,2,3}$	$S^{50}_{train}, S^{80}_{train}, S^{108}_{train}$	**150**			**0.9715**

The second row of Table 3 reports the classification results that would be obtained if the previously developed ensemble E^1 were discarded at t_{80} and a new ensemble $E^{1,2}$ formed by $T = 10$ classifiers built using all the patterns of $S^{50}_{train}, S^{80}_{train}$ is constructed: the performance of the latter ensemble is slightly better, but at the cost of high computational efforts since all the classifiers have to be retrained from scratch on an enlarged dataset (Table 3, third column). The training time on a Pentium IV 2.2 MHz PC is 51 minute and in case of retraining raises to 100 minute. A similar situation occurs at t_{108} if an ensemble $E^{1,2,3}$ of $T = 10$ classifiers is built using the patterns of $S^{50}_{train}, S^{80}_{train}, S^{108}_{train}$.

4.2. Case study 2: decreasing power level

In this case study the power level is decreased from 108% to 80% of full power and then from 80% to 50%, with datasets S^{108}, S^{80}, S^{50} becoming available at times $t_{108} < t_{80} < t_{50}$.

The first developed ensemble E^1 formed by $T^1 = 10$ classifiers trained at t_{108} using only the patterns of dataset S^{108}_{train} gives a satisfactory performance in the classification of the patterns of S^{108}_{test} and S^{80}_{test} i.e. $E^2 = E^1$ (Table 4, first row, fourth and fifth column). However, when at t_{50} the plant starts working at 50% of full power the performance of E^1 decreases to 71.52% of correctly classified patterns in S^{50}_{test} (Table 4, first row, sixth column). In the procedure proposed in this work, at t_{50}, $T^3 = 10$ new classifiers are trained with bagging of the dataset S^{50}_{train} and added to the previously constructed ensemble. The performance of the obtained new ensemble E^3 rises to 94.93% (Table 4, second row, sixth column).

Furthermore, if the power plant returns to work at 80% and 108% of full power, the performance of E^3 remains still satisfactory (Table 4, second row, fourth and fifth columns), this shows that the ensemble E^3 has incrementally learned the new information in S^{50} without forgetting what it has learned before (S^{80} and S^{108}).

The performance of E^3 is compared with those of the ensembles $E^{1,2}$ and $E^{1,2,3}$ that would be obtained by discarding the previously constructed ensembles at t_{80} and t_{50} and retraining new ensembles of classifiers with all data in $S^{108}_{train}, S^{80}_{train}$ and $S^{108}_{train}, S^{80}_{train}, S^{50}_{train}$, respectively. Again, retraining leads to slightly improved performances, but at high computational costs (Table 4, third column).

Table 4: Performances and training computational time of the ensemble-based approaches in the classification of the test patterns: proposed approach for incremental learning in a non-stationary environment (first and second row), ensemble obtained by retraining all the base classifiers (third and fourth rows).

Approach	*Train Dataset*	*Training time (min)*	*Test Dataset*		
			S^{108}_{test}	S^{80}_{test}	S^{50}_{test}
Incremental Learning Ensemble - $E^1 = E^2$	S^{108}_{train}	**51**	**0.9638**	**0.9576**	**0.7152**
Incremental Learning Ensemble - E^3	$S^{108}_{train} + S^{50}_{train}$	**51**	**0.9645**	**0.9652**	**0.9493**
Retraining - $E^{1,2}$	$S^{108}_{train}, S^{80}_{train}$	**100**		**0.9653**	
Retraining - $E^{1,2,3}$	$S^{108}_{train}, S^{80}_{train}, S^{50}_{train}$	**150**		**0.9722**	**0.9611**

4.3. Discussion

The above results show that the structure of the proposed ensemble is influenced by the order in which the datasets become available. In case study 1, the first ensemble of classifiers built on the information in S^{50}_{train} is able to classify the upcoming datasets with good accuracy, whereas in case study 2, the first ensemble of classifiers built using the information in S^{108}_{train} needs to be updated when the plant starts working at 50% of full power.

5. Conclusions

In this work, a realistic situation in which transient examples of plant behavior in different operational conditions become available in successive datasets has been considered. A practical procedure has been proposed based on the addition of classifiers to an ensemble, for incrementally learning new situations while keeping the computational efforts under control.

The approach used to construct the ensemble is bagging; the base classifier is a supervised Fuzzy C Means (FCM) classifier; the individual base classifiers outcomes are combined using a majority-voting scheme.

The novelty of the procedure is that it allows learning the new information contained in the data becoming available during the plant life without forgetting the previously acquired knowledge. This incremental learning capability is obtained by adding new classifiers to the ensemble if the transients occurring in the new operational conditions are not satisfactorily classified by the current ensemble model.

The procedure has been applied to the identification of simulated transients in the feedwater system of a Boiling Water Reactor (BWR) at different power levels. The proposed classification scheme has been compared with the classical approach which requires that the existing classification model is discarded when new data become available and a new one is retrained from scratch using all data that have been accumulated thus far. The obtained results show that the performance of the proposed procedure is comparable to that achieved by complete retraining of the models, but with the advantage of significant savings in computational efforts. Furthermore, the proposed procedure is suitable to be used in cases in which the datasets previously used for model training are lost, corrupted or otherwise unavailable.

One limitation of the proposed approach which will be object of future work is that the proposed diagnostic system cannot be used for the classification of new classes of faults for which transient examples are not available in the first dataset.

Acknowledgments: 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.

Appendix A: Algorithms for the ensemble of classifiers

Let S be a training dataset formed by N patterns $\vec{x}_k$ whose known physical classes are $\Gamma_k^{true}, k = 1,..., N$. The subscript *true* indicates that Γ_k^{true} represent the true, a priori known physical class of $\vec{x}_k$. The total number of classes of the *N* patterns $\vec{x}_k$ is c and thus Γ_k^{true} assumes a value in $(1,\cdots,c)$.

Bagging and training:

Figure 1 shows the bagging algorithm used to train an ensemble E formed by T classifiers. A weak learning algorithm "*WeakLearn"* is used to train the individual base classifiers h_i of the ensemble. The maximum number of iterations of the supervised algorithm used to train the single base classifiers of the ensemble is determined by rule of thumb. The flowchart of the training algorithm is shown in the Figure 5.

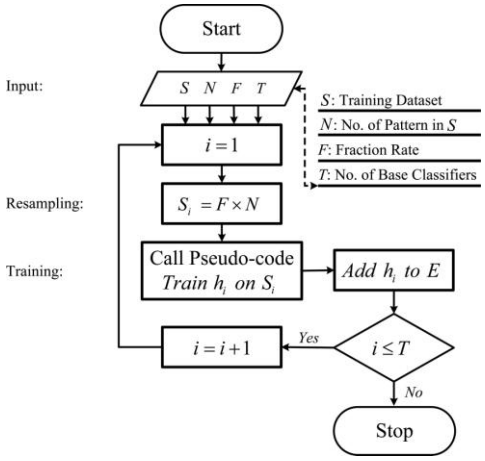


Figure 5. Flowchart for ensemble bagging and training

Majority voting aggregation:

Majority voting is one of the simplest and most intuitive methods to combine classification decisions. The majority voting method consists in assigning to $\vec{x}$ the class label which is supported by the majority of individual classifiers. Let Γ_i be the class assigned by classifier h_i of the ensemble to an unlabeled test pattern $\vec{x}$ and V_{Γ_i} be the vote given to the different classes. In this algorithm, the class that receives the largest total vote is assigned as final decision; in case the number of votes to different classes is equal, the final class is assigned randomly among these classes with largest total vote. The flowchart diagram of the majority voting algorithm is shown in Figure 6.

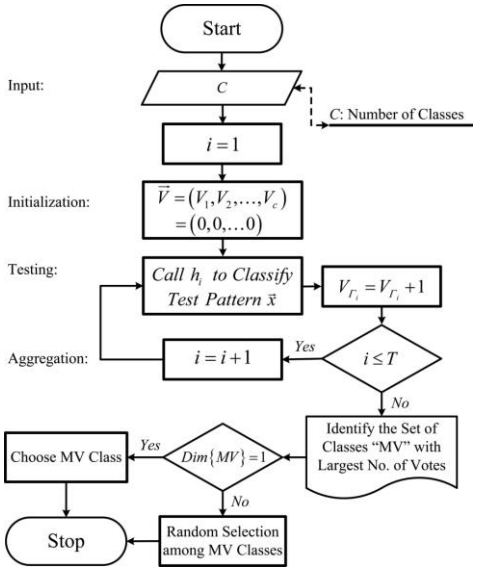


Figure 6. Flowchart of the majority voting algorithm

References:

Baraldi, P., Razavi-Far, R., Zio, E., 2010. Bagged Ensemble of FCM Classifier for Nuclear Transient Identification. Submitted to Annals of Nuclear Energy.

Breiman, L., 1996. Bagging predictors. Machine Learning. 24 (2), pp. 123–140.

Breiman, L., 1999. Combining predictors, in Combining Artificial Neural Nets. A. J. C. Sharkey (ed.), Springer.

Chen, J., & Patton, R.J. (1999). Robust model-based fault diagnosis for dynamic systems. Asian studies in computer science and information science. Boston, USA: Kluwer Academic Publishers.

Embrechts, M.J., Benedek, S., 2004. Hybrid identification of nuclear power plant transients with artificial neural networks. IEEE Transactions on Industrial Electronics. 51, pp. 686–693.

Evsukoff, A., Gentil, S., 2005. Recurrent neuro-fuzzy system for fault detection and isolation in nuclear reactors, Advanced Engineering Informatics. 19 (1), pp. 55-66.

Hansen, L.K., Salamon, P., 1990. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence. 12 (10), pp. 993–1001.

Hines, J.W., Miller, D.W., Hajek, B K., 1996. A hybrid approach for detecting and isolating faults in nuclear power plant interacting systems. Nuclear Technology, 115(3).

Kuncheva, L.I., 2004. Classifier Ensembles for Changing Environments. Book series lecture notes in computer science, Springer Berlin / Heidelberg, Volume 3077, pp. 1-15.

Na, M.G., Shin, S.H., Lee, S.M., Jung, D.W., Kim, S.P., Jeong, J.H., Lee, B.C., 2004. Prediction of major transient scenarios for severe accidents of nuclear power plants. IEEE transactions on nuclear science, 51(2), pp. 313-321.

Parhami, B., 1994. Voting algorithms. IEEE Transactions on Reliability, 43 (4), pp. 617–629.

Polikar, R., 2006. Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine, 6 (3), pp. 21-45.

Polikar, R., Udpa, L., Udpa, S.S., Honavar, V., 2001. Learn++: An incremental learning algorithm for supervised neural networks. IEEE transactions on systems, man, and cybernetics-part C: Applications and reviews, Vol. 31, No. 4.

Puska, E., Normann, S., 2002. 3-d core studies for hambo simulator. In proceedings of presentations on man-machine system research, Enlarged Halden programme group meeting, vol. 2.

Reifman, J., 1997. Survey of artificial intelligence methods for detection and identification of component faults in nuclear power plants. Nuclear Technology 119, pp. 76–97.

Razavi-Far, R., Davilu, H., Palade, V., Lucas, C., 2009. Model based fault detection and isolation of a steam generator using neuro-fuzzy networks. Neurocomputing Journal, 72, pp. 2939–2951.

Roverso, D., 2004. On-line early fault detection and diagnosis with the alladin transient classifier. Proceedings of PNPIC and HMIT-2004, the 4th American Nuclear Society, International Topical Meeting on Nuclear Plant Instrumentation Control and Human-Machine Interface Technologies, Columbus, Ohio, September 19-22.

Simani, S., Fantuzzi, C., Patton, R.J., 2002. Model-based fault diagnosis in dynamic systems using identification techniques. Springer-Verlag.

Uhrig, R.E., 1999. Soft computing technologies in nuclear engineering applications. Progress in Nuclear Energy, Vol. 34, No 1, pp. 13-75.

Xu, L., Krzyzak, A., Suen, C.Y., 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Transactions on Syst., Man, and Cybern., 22, pp. 418–435.

Yuan, B., Klir, G., 1997. Intelligent Hybrid Systems Fuzzy Logic, Neural Network, and Genetic Algorithms, Kluwer Academic Publishers.

Zhao, K., Upadhyaya, B.R., 2005. Adaptive fuzzy inference causal graph approach to fault detection and isolation of field devices in nuclear power plants. Progress in Nuclear Energy, 46 (3-4), pp. 226-240.

Zio, E., Baraldi, P., 2005. Identification of nuclear transients via optimized fuzzy clustering. Annals of Nuclear Energy, 32, pp. 1068-1080.

Zio, E., Baraldi, P., Gola, G., 2008. Feature-based classifier ensembles for diagnosing multiple faults in rotating machinery. Applied Soft Computing, 8, pp. 1365–1380.

Zio, E., Baraldi, P., Pedroni, N., 2006. Selecting features for nuclear transients classification by means of genetic algorithms. IEEE transactions on nuclear science, 53 (3), pp. 1479-1493.