AIChE Wiley InterScience DISCOVER SOMETHING GREAT

Simultaneous Fault Diagnosis in Chemical Plants Using a MultiLabel Approach

Ignacio Yélamos, Moisès Graells, and Luis Puigjaner
Chemical Engineering Dept.–CEPIMA, ETSEIB, Universitat Politècnica de Catalunya (UPC),
Av. Diagonal 647, Barcelona 08028, Spain

Gerard Escudero
Software Dept., EUETIB, Universitat Politècnica de Catalunya (UPC), Comte d'Urgell 187, Barcelona 08036, Spain

DOI 10.1002/aic.11313
Published online September 25, 2007 in Wiley InterScience (www.interscience.wiley.com).

One of the main limitations of current plant supervisory control systems is the reliability and the correct management of simultaneous faults, which is crucial for supporting the plant operators' decision making. In this work, a MultiLabel approach that makes use of support vector machines as the learning algorithm is employed to arrange a novel fault diagnosis system (FDS). The FDS is trained to address a difficult control case study from industry widely studied in the literature, the Tennessee Eastman process. Successful results have been obtained when diagnosing up to four simultaneous faults. These results are very promising since they have been obtained by just using simple training sets consisting of single faults, thus proving a very high learning capacity. © 2007 American Institute of Chemical Engineers *AIChE J,* 53: 2871–2884, 2007
Keywords: simultaneous faults diagnosis, MultiLabel approach, support vector machines, simulation, process, control

Introduction

In attention to the seriousness of accidents that may occur in chemical plants, incipient, accurate, and reliable fault diagnosis are significant requirements for preserving people's safety, as well as for enhancing the economy of the plant.

Data-based diagnosis methods[1] have faced such safety and economic issues with different approaches, offering diverse solutions to the many difficulties arising in this area. Nevertheless, there still are severe limitations on chemical plant fault diagnosis that have not been satisfactorily addressed in the literature. One of these main limitations is the management of multiple process faults.

Correspondence concerning this article should be addressed to L. Puigjaner at luis.puigjaner@upc.edu.

The detection problem is just a binary classification (BC) problem, in which two excluding classes are considered—the normal and the abnormal plant behavior. However, the global fault diagnosis problem is a multiclass (MC) classification problem, since many classes are involved. Furthermore, fault diagnosis problem can be considered as monoLabel [(mL) when more than one fault may happen at a time] or MultiLabel [(ML) when more than one fault may happen at a time. Adopting an mL or an ML view depends on whether or not the assignment of a class (faults) excludes the assignment of other classes. In this sense, the fault diagnosis regarded as ML shares the characteristics of the text classification problem in computational linguistics.[2,3] However, despite being a clear ML problem, the general fault diagnosis of chemical processes has frequently been addressed under an implicit mL view.

Data-based mL diagnosis approaches have solved the problem of simultaneous process faults by creating new faults consisting of the combination of two isolated existing faults,[4,5] which results unfeasible in actual industrial problems. Tackling with a large number of isolated faults and facing combinatorial problems resulting from the presence of simultaneous faults, which must be converted in new classes, leads not only to important classification performance decrease but also makes inaccessible the training of all potential classes' combination.

Another frequent solution found in the literature has been the development of qualitative systems based on the cause effect events relationship[6,7] but such methodologies usually generate too many spurious solutions that make the response unreliable for plant operators. Normally, they require further procedures to reduce the set of fault candidates while maintaining the correct fault candidate.

In this work, the proposed fault diagnosis system (FDS) is based on such ML approach to overcome the mL limitations summarized above. Support vector machines (SVM)[8] were used as the learning algorithm as they have proved to be one of the most efficient techniques developed by the statistical learning theory and allow to properly address the diagnosis problem under the ML approach by using some binarization techniques of the machine learning field. SVM are based on the structural risk minimization principle by the statistical learning theory and have recently been gaining popularity as an alternative learning algorithm in problems of the chemical engineering field.[9,10]

The ML-based FDS has been tested using the Tennessee Eastman process (TEP), proposed by Downs and Vogel,[11] which is a well-known benchmark that allows offering comparable results to the fault diagnosis community. Different examples have been prepared for addressing two simultaneous faults, as well as three and more of whichever original fault's combination. Some works have faced the TEP single[12,13] fault diagnosis problem, but very few the simultaneous[4,5,14] fault diagnosis problem. Moreover, there is a clear lack of quantitative results for comparing the methodologies: different works[6,15,16] address different case studies and those[9,17,18] addressing the same TEP benchmark deal partially with the problem, focusing only in some of the faults and using particular performance measurements. This fact has been remarked and addressed in other works,[19,20] and also in this one, by proposing a complete and quantitative performance assessment indexes for the FDS. Hence, results based on precision and recall indexes confirm the technique presented to be very promising in large and complex case studies, such as addressing the entire set of faults of the TE problem.

Problem Formulation and Results Quantification

A fault diagnosis problem can be formulated as a classification problem. In this work, novel diagnosis problem formulation and performance assessment are developed from that viewpoint to standardize the diagnosis results attained.

mL and ML approaches

There are two kinds of classification problems, the BC problem, which consists of determining if a sample belongs or not to an existing class and the MC classification problem in which more than two classes are involved. Both correspond with the fault detection and diagnosis problems in the fault diagnosis area. The MC classification problem can be faced from two different views: by an mL approach, which consists of classifying a set of patterns into a univocal class, or by the ML approach, which allows assigning each input data to more than just one class.

In the recent literature, pattern recognition techniques have used the mL approach for solving the general fault diagnosis problem.[4,5] Nevertheless, the classification of simultaneous faults requires either the simultaneous class assignment, which is the natural ML view, either the use of more classes representing the occurrence of simultaneous faults (class A + B instead of classes A and B), which is the solution adopted when trying to preserve the mL view. Besides its conceptual and methodological suitability to the problem, the ML approach also offers practical benefits. The basic advantages of the ML approach presented stem from the ability of simply training single classes, being later able to classify the individual classes that compose the current simultaneous faults (combination of single classes). In that way, no artificial classes resulting of single faults combination must be built to address simultaneous faults, thus saving computational cost and increasing learning performance by just training single classes.

A popular and extended machine-learning algorithm called SVM has been adopted in the ML-FDS learning and classification stages. Such algorithm has been efficiently used in different technical areas as in "Natural Language Processing" or "Artificial Vision". However, its potential has not been properly explored facing chemical plant data.

Fault diagnosis performance quantification

To evaluate the performance of an FDS, this work rigorously formulates the diagnosis classification problem and proposes general indexes for evaluating its performance.

Process data are characterized by a matrix X of values x_{vs} for each process variable v and for each sampling s, as well as by a given set of faults $f = 1, 2, \ldots, F$ that may be happening (or not) and that may be eventually diagnosed (or not). Hence, a binary matrix H may be used to indicate which of these faults (in columns) is happening at each sample time (in rows). Accordingly, another binary matrix D may be used to indicate the diagnoses made at each sample time. Both matrices are next shown:

$$H = \begin{pmatrix} h_{11} h_{12} \ldots\ldots h_{1F} \\ h_{21} h_{22} \ldots\ldots h_{2F} \\ \vdots \ddots \vdots \ddots \vdots \\ h_{S1} h_{S2} \ldots\ldots h_{SF} \end{pmatrix}, \; h_{sf} \in \{0,1\}, \forall s,j : 1 \le s \le S, 1 \le f \le F \quad (1)$$

$$D = \begin{pmatrix} d_{11} d_{12} \ldots\ldots d_{1F} \\ d_{21} d_{22} \ldots\ldots d_{2F} \\ \vdots \ddots \vdots \ddots \vdots \\ d_{S1} d_{S2} \ldots\ldots d_{SF} \end{pmatrix}, \; d_{sf} \in \{0,1\}, \forall s,j : 1 \le s \le S, 1 \le f \le F \quad (2)$$

Either a fault f is happening ($h_{sf} = 1$) or not ($h_{sf} = 0$), a good diagnosis is given by $d_{sf} = h_{sf}$. Therefore, the ML classification problem seeks for the matching of matrix D to matrix H when any distribution of binary values is allowed, including null rows for the normal case (no faults) and rows including several non-null values (simultaneous faults).

For determining the matching degree of these matrices, and thus the actual FDS performance, two essential measurements are given in the machine learning literature—precision and recall.

Precision for fault f (Eq. 3) can be defined as the conditioned probability of happening fault f, conditioned to fault f has been diagnosed

$$\text{Prec}(f) = \Pr\left[h_{ij} = d_{ij} \middle| d_{if} = 1 : 1 \le i \le S, j = f\right] \tag{3}$$

and recall for fault f (Eq. 4) can be defined as the conditioned probability of the FDS of predicting fault f conditioned to the sample is fault f.

$$\text{Rec}\ (f) = \Pr\left[h_{ij} = d_{ij} \middle| h_{if} = 1 : 1 \le i \le S, j = f\right] \tag{4}$$

The F1 index (widely used in the machine learning area) is a measure that combines both precision and recall. It is used along this work as the measure for evaluating the general performance of the FDS. It is evaluated by the equation:

$$\text{F1} = \frac{2 \times \text{Prec} \times \text{Rec}}{\text{Prec} + \text{Rec}} \tag{5}$$

Accuracy, error, and global measurements are additional and complementary indexes to globally evaluate the system classification general performance.

Accuracy represents the percentage of right assignments, not only the correctly diagnosed but also the correctly not happening and not diagnosed samples. Error is the percentage of wrong assignments and is the accuracy supplementary measure. They are formally defined as follows:

$$\text{Acc}(f) = \Pr\left[d_{ij} = h_{ij} \middle| i : 1 \le i \le S, j = f\right] \tag{6}$$

$$\text{Error}(f) = \Pr\left[d_{ij} \ne h_{ij} \middle| i : 1 \le i \le S, j = f\right] \tag{7}$$

Global index shows how many samples have been completely well diagnosed through all the existing classes, hence giving a very valuable measurement about the system global diagnosis correctness. High values in that measurement are difficult to achieve (as they require a perfect diagnosis for each sample) and not only indicate a good performance diagnosing the isolated classes but also assure good ML characteristics that are crucial facing simultaneous faults (a complete well-diagnosed system also implies no misdiagnosis, which is essential in the ML classification approach). Following the definition procedure it can be expressed as shown in Eq. 8:

$$\text{Global} = \Pr\left[d_{ij} = h_{ij} \middle| i : 1 \le i \le S, \forall j\right] \tag{8}$$

These performance measurements can be calculated considering the FDS responses or votes after testing each of the trained faults. The confusion matrix (Table 1) gathers such information for each fault, showing the occurred fault in columns and the diagnosed fault in rows.

Table 1. Confusion or Contingency Matrix

	Happening Fault (True Class)	
	f	$\neg f$
Diagnosed fault (predicted class)		
f	a	b
$\neg f$	c	d

a: Samples happened and diagnosed; b: samples diagnosed but not happened; c: samples happened and not diagnosed; d: samples not happened and not diagnosed.

Hence, the following equations are proposed for directly evaluating the classification performance indexes from the confusion matrix[21]:

$$\text{Prec}(f) = \frac{a}{a+b} \tag{9}$$

$$\text{Rec}(f) = \frac{a}{a+c} \tag{10}$$

$$\text{Acc}(f) = \frac{a+d}{a+b+c+d} \tag{11}$$

$$\text{Error}(f) = \frac{b+c}{a+b+c+d} = 1 - \text{Acc}(f) \tag{12}$$

As a clarifying example, a fault happening matrix (H) and the FDS corresponding diagnosing matrix (D) are next presented. Three faults and 12 samples (represented in columns and rows, respectively) are assumed.

$$H = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

In H matrix, first two samples correspond to the normal state; second, third, and fourth pairs of samples correspond

Table 2. Indexes Evaluation for the Proposed Example

Index* (%)	Normal	Fault 1	Fault 2	Fault 3	General
Precision	0/(0 + 0) = –	2/(2 + 0) = 100	2/(2 + 2) = 50.0	4/(4 + 2) = 66.7	8/12 = 66.7
Recall	0/(0 + 2) = 0	2/(2 + 0) = 100	2/(2 + 4) = 33.3	4/(4 + 2) = 66.7	8/14 = 57.1
F1	–	100	40.0	66.7	61.5
Accuracy	10/12 = 83.3	12/12 = 100	6/12 = 50.0	8/12 = 66.7	26/36 = 72.2
Error	17.7	0.0	50.0	33.3	27.8
Global			4/12 = 33.3		

General does not include Normal class.

to faults 1, 2, and 3 whereas the last four samples correspond to a simultaneous fault composed of faults 2 and 3. Matrix *D* represents the diagnosis system results facing samples on matrix *H*. For example, the first two rows in matrix *D* show that both samples were classified as fault number 3, as well as the samples in the last four rows. The performance indexes can be calculated by comparing the matching of matrix *D* to matrix *H* by using the details of the individual and global contingency matrices. Next, these matrices for normal state, faults 1–3 and the general results are depicted. Note that class 0 means normal state.

CM	0	¬0
0	0	0
¬0	2	10

CM	1	¬1
1	2	0
¬1	0	10

CM	2	¬2
2	2	2
¬2	4	4

CM	3	¬3
3	4	2
¬3	2	4

ΣCM	yes	no
yes	8	4
no	6	18

Therefore, performance indexes can be easily evaluated by replacing the values from these matrices (*a–d* in Table 1) in Eqs. 5 and 9–12. The results (Table 2) offer a complete and detailed information about the classification performance. Precision gives information about the diagnosed samples, recall gathers the information about the happened faults, and F1 index gives a single normalized measurement for the evaluation of the general system performance. Note that even though the normal state was not trained it was later tested and its performance was also evaluated in the same way. Accuracy and error are supplementary measurements about the correct diagnosed and not diagnosed examples whereas the global measurement is lower just because it focuses on the totally well-diagnosed samples throughout all the existing classes, 4 over the original 12 (check *H* and *D* matrices).

Methodology

The proposed FDS was set to face the diagnosis problem from the ML approach. In that sense, the binarization step was arranged to organize the labeled process information under the ML scheme. Then, learning and classification stages generate the required classifiers and evaluate their performances, respectively. Finally, a de-binarization and evaluation steps were included to decode and show the FDS performance. The procedure to train, test, and evaluate the ML-FDS is shown in Figure 1.

Training and testing sets were random and independently selected to avoid biased results. Thus, each class was first randomly built by sampling original representative data. Then, these new sets were split in two different sets from which training and testing sets may be respectively selected with the desired set size. The complete ML-FDS scheme consists of a core learning algorithm[22] described later on, plus a wrapper implemented in Borland Delphi®, whereas Matlab® was used to make valuable data analysis. The computer used to obtain all the results presented in the article was a Pentium IV (2.6 GHz) operating under Windows 2000®.

Next subsections describe the methodology details sketched in Figure 1 and outline the main differences with the mL approach. The methodology description is illustrated through an example (Figure 2) consisting of two variables (x_1 and x_2), three single classes (including normal class) and a composed class (1–2) and two samples per class.

Training and testing sample coordinates are given in Table 3 (first two rows).

Binarization

The ML-based binarization for each class consisted of labeling as positive all the samples belonging to the class while assigning negative values to all other samples ("one versus all" Ref. 23). In the example, two training binarized sets were built, as just two classes must be trained under the ML approach. Labels for each training set are shown in the third and fourth rows of Table 3. In case an mL approach had been used, two additional classes (normal class and 1 + 2 class) should have been trained, thus complicating the later

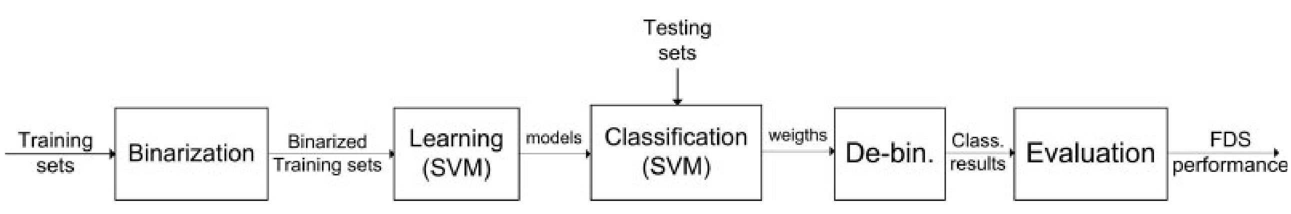

Figure 1. ML-FDS performing scheme.

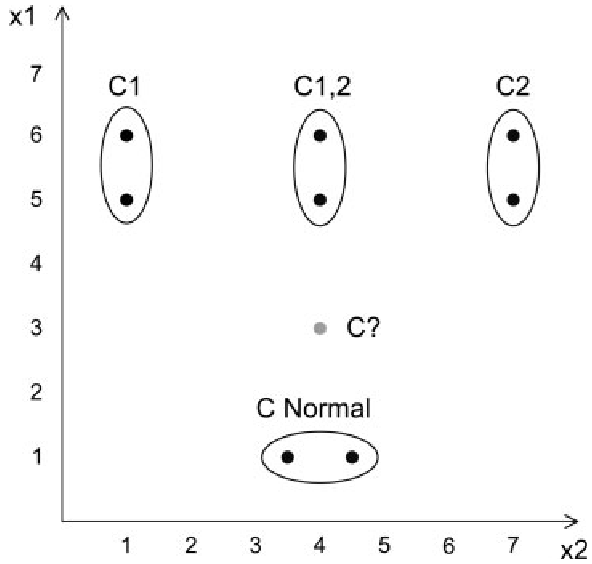

Figure 2. Training and testing example.

Black dots are labeled samples, while the gray dot is the testing sample.

learning step. Four last rows in Table 3 show the mL particular binarization.

For each fault *f*, the binarized data set is given by the following matrix:

$$B_f = \{x_{1s}, x_{2s}, \ldots, x_{vs}, h_{sf} \mid s \in \text{learning}\}, \quad \forall f \tag{13}$$

being learning the training subset of samples.

Learning

These training sets previously binarized (one per fault; two in the example) are used to generate the different models during the learning stage (also one per fault; two in the example, instead of the four that would require the mL approach). Then, SVM are used as the binary learning algorithm which establishes the hyperplanes to separate positive (belonging to the class) from negative samples (not belonging to the considered class). The result of this step, using SVM, is a model for each class *f* given by a vector w_f (with one component for each considered variable) that characterizes the shape of the hyperplane and an offset parameter b_f, which stands for the distance of the hyperplane to the origin (Figure 3).

Next, qualitative details of SVM are given

SVM. SVM are based on the structural risk minimization principle from the statistical learning theory. They learn a linear hyperplane that separates a set of positive examples from a set of negative examples with maximum margin (the margin is defined by the distance of the hyperplane to the nearest of the positive and negative examples). Learning the maximal margin hyperplane is a convex quadratic optimization problem with a unique solution, which supposes an advantage over other conventional algorithms.[24]

When examples are not linearly separable or, simply, a perfect hyperplane is not needed, it may be preferable to allow some errors in the training set so as to maintain a "better" solution hyperplane. This may be achieved by a variant of the optimization problem, referred to as soft margin, in which the contribution to the objective function of the margin maximization and the training errors can be balanced through the use of a parameter called C. Despite the linearity of the basic algorithm, SVM can be also converted into a dual form, allowing the use of kernel functions to produce nonlinear classifiers.

We used SVMlight,[22] an open source implementation of the SVM original mechanism[8] that adopts some practical simplifications to make applicable the theoretical SVM formulation (available at http://svmlight.joachims.org). Just default parameters (degree one polynomial kernels and default soft margin trade-off) were used in the SVM implementation, without applying any kind of tuning procedure.

Classification

During the classification or testing step, new and unlabeled samples (testing samples in Figure 2 and Table 3) are "weighted" using each classifier model (w_f and b_f). The weight for each sample is the result of the sum of b_f, and the inner product of the model vector (w_f) and the testing sample vector. These weights (y_{sf} in Eq. 15) are the basis for the sample classification on the evaluation step.

Being the testing subset of samples:

$$x_s = \{x_{1s}, x_{2s}, \ldots, x_{vs} \mid s \in \text{classification}\} \tag{14}$$

weights are evaluated by:

$$y_{sf} = \langle w_f, x_s \rangle + b_f \tag{15}$$

where "$\langle w_f, x_s \rangle$" represents the inner product of the w_f vector and the tested sample vector (x_s). In that sense, quantitative information for classifying any tested sample is available, and will be used later to make the diagnosis.

Table 3. Problem Class Samples and Binarized Sets Codification (C? Refers to the Unlabeled Sample)

	C1	C2	C1,2	C Normal	C?
Training (X1, X2)	(5,1), (6,1)	(5,7), (6,7)	(5,4), (6,4)	(1,3.5), (1,4.5)	–
Testing (X1, X2)	–	–	–	–	(3,4)
1st ML binarized set	+1, +1	−1, −1	+1, +1	−1, −1	–
2nd ML binarized set	−1, −1	+1, +1	+1, +1	−1, −1	–
1st mL binarized set	+1, +1	−1, −1	−1, −1	−1, −1	–
2nd mL binarized set	−1, −1	+1, +1	−1, −1	−1, −1	–
1+2 mL binarized set	−1, −1	−1, −1	+1, +1	−1, −1	–
Normal mL binarized set	−1, −1	−1, −1	−1, −1	+1, +1	–

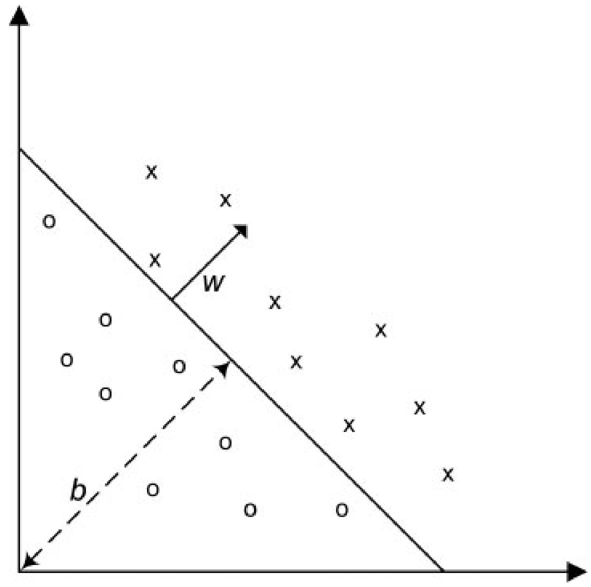

Figure 3. Hyperplane definition.

Following the comparison with the mL view, no difference can be found on the learning or classification steps but that of working with more classes (four, in mL approach) than in the ML approach (just two classes) due to the dragged differences from the binarization.

De-binarization

By this step the system decodes weights given in the classification (y_{sf}) into diagnosis responses (d_{sf}), thus assigning a class to each treated sample. It may be done by simply identifying the positive weights with the Heaviside step function (Eq. 16).

$$d_{sf} = \theta\{y_{sf}\} \tag{16}$$

In our case, the testing sample (3,4) obtained the same negative weights for each classifier (−0.57), hence being classified as normal class (d_{s1}, d_{s2}, and $d_{s3} = 0$), which includes those samples not diagnosed as any of the learned classes.

Under the mL approach a significant difference can be outlined at that point. As just one class is diagnosed for each sample, the diagnosis is made by the "winner takes all" methodology for which the sample is classified as that class with the highest weight. Thus, correlation between y_{sf} and d_s would be expressed by Eq. 17 (note that under the mL view, subindex f is not required).

$$d_s = \arg\max_f\{y_{sf}\} = f \mid y_{sf} \geq y_{sk}, \quad \forall k = 1, \ldots, F \tag{17}$$

The tested sample (Figure 2) was also identified as normal class by the mL approach because of its highest weight, 0.0 versus −1.14, −1.14, and −1.0 of classes 1, 2, and 1 + 2, respectively.

By the de-binarization step, the ML approach allows training each class with specific and differentiated variables, which is a great advantage in terms of computational efficiency and independent class analysis.

Evaluation

Finally, the agreement between the actual sample class membership and the class diagnosed is checked to evaluate the ML-FDS performance. This matching degree from all the existing point of views can be analyzed by the indexes introduced in the previous section.

It is in that stage where the ML approach requires a more complex implementation. Under such an approach, several classes can be correctly diagnosed for the same sample, which allows diagnosing simultaneous faults from single faults learning at the expense of solving a more complex evaluation. In that sense, under the ML approach, the correct sample diagnosis not only requires the right sample classification (mL) but also requires the correct nonassignment of the sample in any of the remaining classes.

Validation and Results

The FDS was trained and tested with the original 20 faults proposed by Downs and Vogel[11] and without any additional combination of faults. No preliminary filtering or data pretreatments were applied to the raw plant data. No tuning was applied to the SVM (default soft margin and degree one polynomial kernel are utilized). Despite taking no advantage of these improving opportunities, this work outlines the crucial points of the multiple fault diagnosis analysis and shows its very promising performance.

Case study

The FDS is tested using the TEP which is a well-known benchmark problem in process engineering. Downs and Vogel[11] presented this particular process as a plant-wide control problem that has got to challenge the fault diagnosis community, allowing the comparison of different control strategies and fault diagnosis methodologies.

The process involves the manufacturing of two chemicals, G and H, and one by-product F, from four reactants: A, C, D, and E. The process has five major units: a reactor, a product condenser, a vapor/liquid separator, a recycle compressor, and a product stripper. The process has 41 measurements and 12 manipulated variables. Among the several available simulation schemes, this article follows a plant-wide decentralized control scheme that is based on multiple single-input-single-output control loops.[25] Figure 4 shows the TEP flowsheet.

The TEP simulator can generate 20 types of isolated faults, as well as any multiple combinations of these faults. Once a fault enters the process, it affects different state variables and the identification of many of these faults is highly challenging due to the subtle and similar variable deviations engendered under some specific faults. Other deviations are just caused by random errors making the training data not representative of crucial fault details, which also increase the diagnosis difficulty.

It is a well-known fact that the performance of the statistical methods depends basically on the data sets on which they are trained. So, data training and testing sets were built in

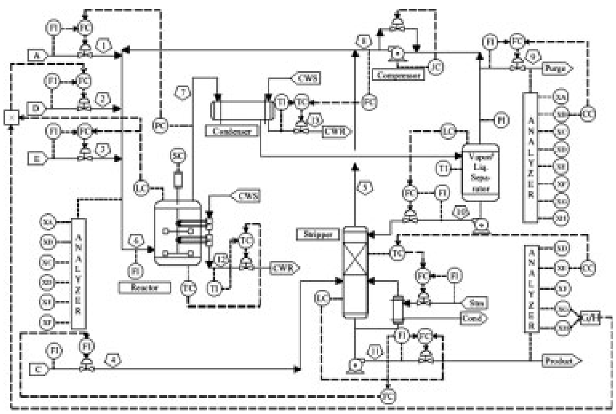

Figure 4. TEP flowsheet under McAvoy and Ye decentralized control scheme.

such a way that all possible state variable dynamics are captured under each fault.

Single fault results

The performances of the mL and ML approaches are first tested and compared. The same training and testing data sets of 351 and 1000 respective samples per single fault were used to check SVM classification capabilities under mL and ML approaches. Additionally, 112 samples under normal operating conditions (no faults) were included in all test data sets to include the normal state in the analysis. Master original data sets were extracted from the simulation of a 20-h operation horizon. Sampling every 10 s resulted in data matrices having 7200 rows (samples) per 52 columns (variables) for each single fault. The mentioned training and testing sets were built by a random sampling of the considered variables for each existing fault, first splitting the simulation samples in training and testing sets, and then fixing the percentage of required samples in each set. The 52 variables considered correspond to the 41 TEP measured variables (XMEAS in the original paper) plus the first 11 manipulated variables (XMV in the original paper). The 12th manipulated variable (XMV12: agitator speed) was not used because it remained constant under any tested operating condition, thus not being relevant from the fault diagnosis viewpoint. The data sets' building procedure is shown in Figure 5.

The diagnosis results obtained for these sets are given in Table 4, which shows exactly the same values of the F1 index for both classification methodologies except for the normal operating condition class (named as 0). Strictly speaking, the normal case should not be considered as a class under the ML approach, since it is an artificial class that excludes all other classes. Therefore, this class (no faults) is better diagnosed by the mL approach because of the advantage of training this class as an additional state and because the ML approach efficiency on the normal class critically depends on the achievement of very high classification performances in the rest of classes. Table 4 also shows a similar classification capability for those cases that revealed more difficulty (faults 3, 5, 9, etc.).

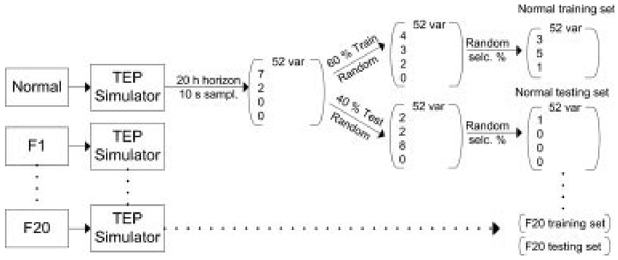

Figure 5. Training and testing data sets building procedure.

Table 4. Single-Fault Diagnosis. F1 Index (%) for the mL and ML Approaches

	mL	ML
0	96.2	13.5
1	96.7	96.7
2	93.3	93.3
3	8.4	8.4
4	89.3	89.3
5	3.7	3.7
6	100	100
7	100	100
8	41.8	41.8
9	0.0	0.0
10	37.0	37.0
11	2.8	2.8
12	0.0	0.0
13	0.0	0.0
14	10.2	10.2
15	0.0	0.0
16	16.1	16.1
17	94.8	94.8
18	78.2	78.2
19	10.2	10.2
20	85.9	85.9
Average	45.9	42.0

Once validated the matching of the mL and ML approaches for the single fault case, the ML approach was checked in regard of the sampling influence. With that purpose, a 50-h operation horizon was simulated collecting data every second (sampling 2, S2). Then, after splitting simulation data on training and testing sets, a random selection allowed building equal-sized but differently sampled training and testing sets (351 and 1000 samples, respectively) than those from first sampling (S1). The overall FDS performance obtained from each sampling is compared in Figure 6.

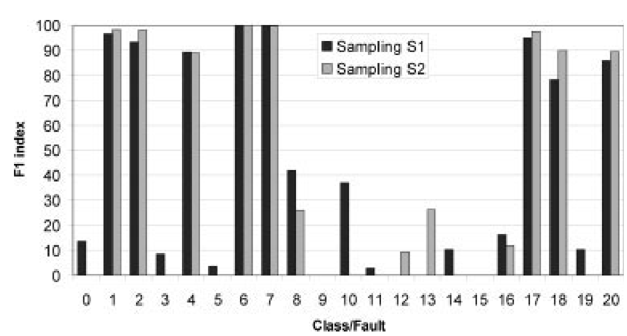

Figure 6. ML approach.

Comparison of F1 index fault by fault using two different data sampling.

Table 5. Multiple-Fault Diagnosis. F1 Index (%) Under ML Approach

	1	2	6	7	17	18
2 Simultaneous faults (1, 2)	97.2	96.1	–	–	–	–
2 Simultaneous faults (6, 7)	–	–	96.6	99.9	–	–
2 Simultaneous faults (17, 18)	–	–	–	–	90.5	64.6
3 Simultaneous faults (2, 7, 17)	–	82.8	–	68.7	91.8	–
4 Simultaneous faults (2, 7, 17, 18)	–	80.0	–	58.8	86.6	17.8

Table 6. Details for the Classification of a Case Consisting of Four Simultaneous Faults (2, 7, 17, 18)

	a	*b*	*c*	*d*	Precision	Recall	F1 (%)
1	0	0	0	1112	–	–	–
2	667	0	333	112	100	66.7	80.0
3	0	147	0	965	0	–	0
4	0	0	0	1112	–	–	0
5	0	1	0	1111	0	–	0
6	0	114	0	998	0	–	0
7	416	0	584	112	100	41.6	58.8
8	0	143	0	969	0	–	0
9	0	99	0	1013	0	–	0
10	0	0	0	1112	–	–	0
11	0	1	0	1111	0	–	–
12	0	0	0	1112	–	–	–
13	0	0	0	1112	–	–	–
14	0	116	0	996	0	–	–
15	0	0	0	1112	–	–	–
16	0	168	0	944	0	–	–
17	763	0	237	112	100	76.3	86.6
18	98	0	902	112	100	9.8	17.8
19	0	75	0	1037	0	–	–
20	0	16	0	1096	0	–	–

a: Samples happened and diagnosed; *b*: samples diagnosed but not happened; *c*: samples happened and not diagnosed; *d*: samples not happened and not diagnosed.

Although some differences can be appreciated between performances obtained from sampling, F1 index shows a similar trend for each class, thus validating the original performance result on S1. Particularly, faults 1, 2, 4, 6, 7, 17, 18, and 20 are very efficiently classified in both cases, while faults, 3, 5, 9, 11–16, and 19 are badly classified despite of the sampling. Performance differences in faults 8 or 10, which are also misclassified, are due to differences in specific dynamic features captured by each sampling (that comes from the random nature of these faults).

Quantitative results regarding the normal class should be considered as merely illustrative and nonrelevant. The identification of the normal state (a case excluding the rest of the fault classes in the ML approach considered) is a detection problem that can be more easily identified by using a parallel detection system that is simpler and normally more robust than a diagnosis focused tool. The results on that normal class may be given as additional information to check the FDS does not diagnose any of the trained faults. Good results in normal class under the ML approach could be just expected in case the remaining states were properly addressed.

Simultaneous faults results

Considering the equivalent results produced (Table 4) and the evident formulation advantages for the diagnosis of simultaneous faults (section "mL and ML approaches"), it is clear that the ML view must be adopted in a classification problem such as fault diagnosis in chemical processes. Therefore, the ML approach is exploited in the rest of the article.

Thus, different simultaneous diagnosis tests were arranged. Combinations of two, three, and up to four simultaneous faults were simulated and results obtained when diagnosing them are shown in Table 5. Just faults that were individually well diagnosed (Table 4) were selected to build simultaneous faults because they cannot be identified whether composing faults are not properly recognized.

The first three tested pairs that were successfully diagnosed as both single faults were identified when happening simultaneously with very high performance (F1). Very good performance is also obtained when diagnosing three and four simultaneous faults, although it is not surprising to get lower F1 values when more faults arise simultaneously. When faults 2, 7, and 17 are simulated at the same time, the FDS is able to identify correctly the isolated faults that the system has been trained with. In the four simultaneous faults case, the F1 index just suffers a significant decrease for the 18th fault, while faults 2, 7, and 17 are being correctly diagnosed by the system.

Table 6 shows the performance details for the case of four simultaneous faults, including precision, recall, and the individual class assignment, allowing thorough analysis about its goodness. One of the most significant points to be highlighted from these detailed results is the low misdiagnosis rate (b in Table 6) of the FDS. Note that the sum of samples in any column of Table 6 is 1112, which corresponds to 1000 samples belonging to that fault, plus 112 samples belonging to normal operating conditions. For comparative purposes again, simultaneous fault results in Table 5 are confronted with those obtained using the second sampling described in the previous subsection (Figure 7). Equivalent

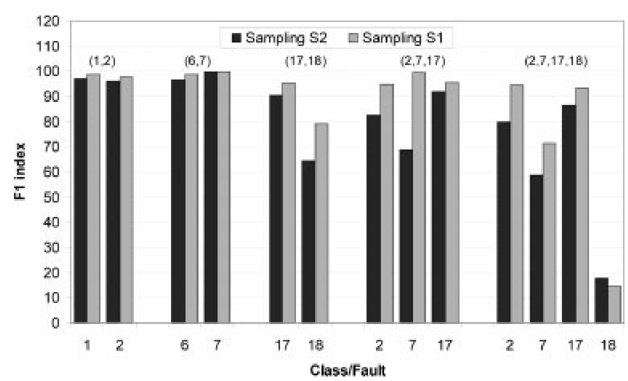

Figure 7. Comparison of F1 index diagnosing simultaneous faults using two different data sampling.

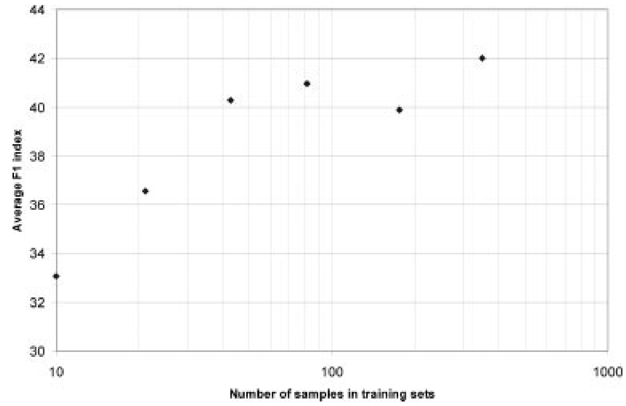

Figure 8. F1 average learning curve in logarithmic scale.

results are achieved again, following the expected predictions.

The results presented show a very high diagnosis performance facing two, three, and up to four simultaneous faults, given that the classes are those correctly trained and diagnosed (Table 4). Since it is not previously reported, they are very remarkable from different viewpoints:

(1) The entire set of faults (20) in the original TEP is addressed considering nonexcluding classes.

(2) The results include the complete and required information to evaluate quantitatively the FDS performance in all senses (accuracy, reliability, misdiagnosis, etc.).

(3) They show correct two, three, and up to four simultaneous faults diagnoses.

(4) Moreover, these results are obtained under a flexible methodology that only requires training single classes, thus:

- Without any need for defining artificial classes for multiple faults, thus considering any possible faults combination.
- Without any need for training these extra classes, which results in higher computational efficiency and an increase of the overall classification performance.

Nevertheless and despite of these excellent results coming from the SVM classification capabilities and from the ML approach considered, certain faults (3, 9–12, etc.) are badly identified (Table 4), and their possible simultaneous combinations will not be properly diagnosed either. For that reason, and considering the high dependence of machine learning results on the information provided, a data analysis was made to improve the information representation utilized as input data by the system.

Performance analysis

The appropriate sizes for training and testing data sets were selected from a preliminary study on the F1 index response. The learning curve in Figure 8 plots F1 index averaged for all the considered faults versus different training data set sizes. All these training sets were extracted from the master split training set (section "Single fault results") in such a way that each greater set includes samples of smaller sets. As a result, the set containing 351 samples for each single fault was selected as the trade-off between the system classification performance and the computational cost [training set: (351×20)]. One thousand-sample testing set previously used was used for obtaining these results (files are available at http://ciao.euetib.upc.es/).

Splitting the global learning curve into its individual classes (Figure 9) permits focusing on diverse individual classification behaviors. In that sense, two different groups can be identified in Figure 9.

A first group consists of certain faults properly classified for any training set size, increasing their performance by asymptotes that offer an adequate training size (faults 1, 2, 7, etc.). The second group is conformed by those faults badly

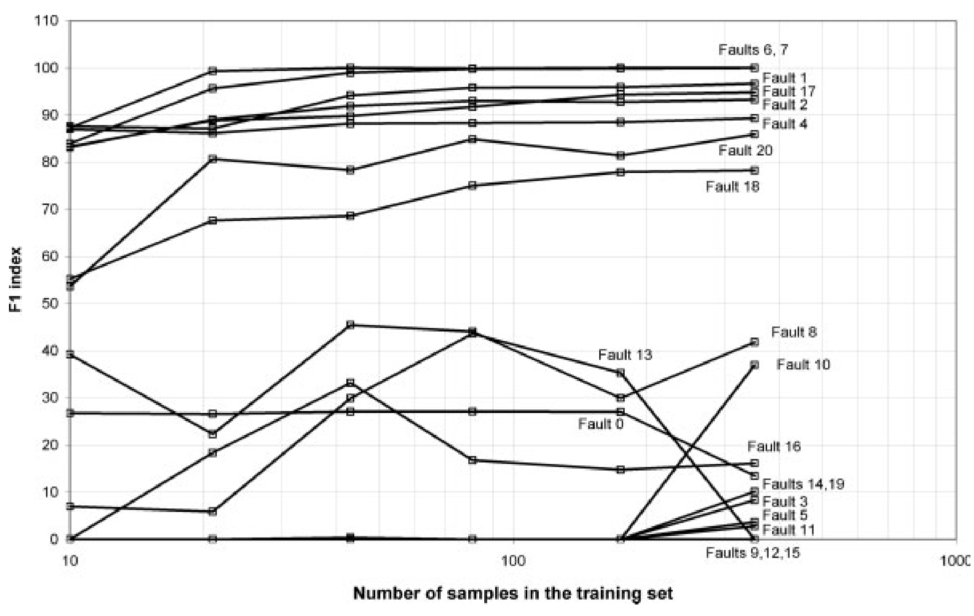

Figure 9. Individual learning curves in logarithmic scale.

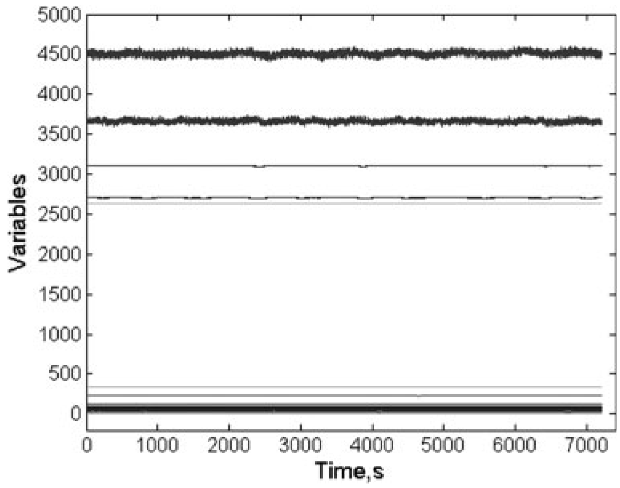

Figure 10. TEP measured and manipulated variables.

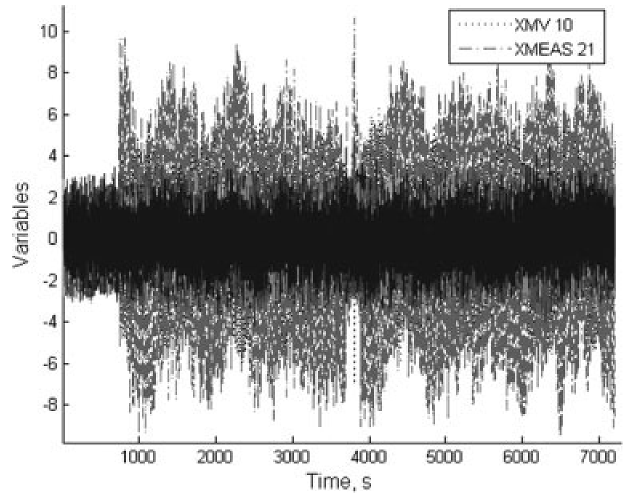

Figure 11. TEP measured and manipulated normalized variables to 0 mean and 1 variance.

classified by the system (8, 9, etc). For these faults, performance classification seems not to depend on the training set size. Decrease on normal class performance as well as random F1 index variation of these second group faults come from the respective classifier deficiencies that make the performance independent of the training set sizes.

Figures 8 and 9 cannot be considered a true learning curve since there is no asymptotic behavior for some faults, thus being impossible to determine the best training set size. This is the reason for adopting the largest of the set sizes evaluated (351 samples) for the subsequent experiments. Figures 8 and 9 are very useful for comparing the different faults, for example, for identifying the random behavior of some faults that could be mistaken with a higher or lower performance depending on the experiment selected.

Enhancing information representation

To increase the classification performance of this second group, the original data and their characteristics were studied. The measurements utilized to characterize these faults, process variables and manipulated variables, are not significant enough and cannot differentiate some faults from the rest. However, the variances of these process variables were able to capture representative features of these faults. For example, original variables XMEAS21 and XMV10[11] were noticeably and randomly modified when fault 14 occurs, whereas other original variables were not altered. For that reason it was decided to adjust those variables and use them as new derived variables. The procedure followed to take advantage of that characteristic is detailed next.

Figure 10 shows the 41 process variables and 11 first manipulated variables given in the TEP[11] after simulating a 20-h operation horizon under the control strategy previously specified (section "Case study"). Fault 14 was introduced at second operating hour and the examples shown are those sampled each 10 s. Note that in Figure 10, any abnormal situation may be clearly observed from the view of variables XMEAS21 and XMV10. Figure 11 shows the same variables normalized to 0 mean and unit variance to adjust all the variables at the same scale and facilitate the analysis.

Figure 10 does not show evidences of deviations from the normal operating regime while Figure 11 shows a variance increase in at least two of the original variables at 720 sample time (fault arising time). This variance increase is not properly characterized by the depicted variables as it just shows random variable variations around the normalized variable mean, which is not significant enough to describe the fault from a machine learning algorithm point of view.

To measure the data variance, the standard deviation of each variable was evaluated for a given moving window. Twenty samples were selected to configure the window length, which demonstrated to accurately capture the process variance changes. The result was an additional set of 52 variables capturing the changes of standard deviation in any of the original variables.

Figure 12 shows how two of these new built variables (StdXMEAS21 and StdXMV10) suddenly went off the normal operating regime at sample 720. By a later filtering pro-

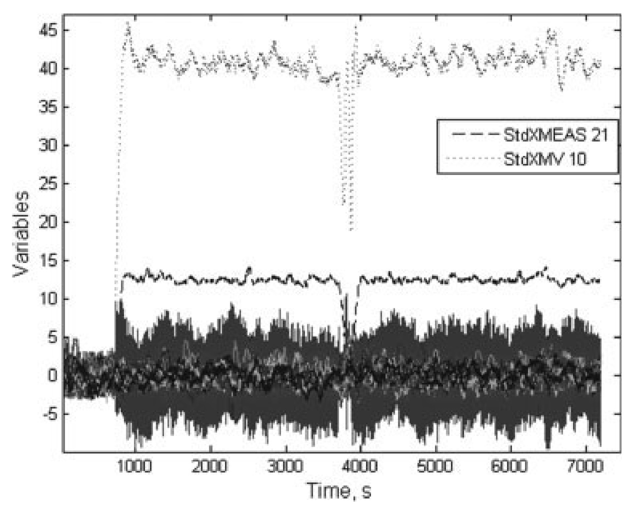

Figure 12. TEP normalized variables plus their standard deviation built variables.

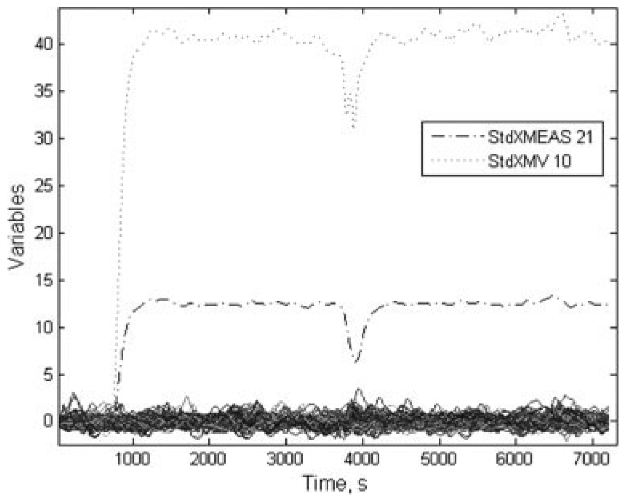

Figure 13. TEP normalized and filtered variables plus their standard deviation built variables.

cedure, a more homogeneous behavior was achieved in these new variables, facilitating the fault feature extraction and then improving the machine learning algorithm results (Figure 13). The filter utilized was the standard exponential weighted moving average with $\alpha = 0.01$.

Results including new information representation: single faults

Without applying any SVM tuning procedure and including as new variables, the standard deviation on those faults badly classified (3, 5, 8–16, and 19) the F1 average raised from 42.0 to 76.2%. Just using specified variables to train each fault represents an additional advantage of the ML regarding the mL approach, as it allows the use of different variables to deal with challenging states avoiding to increase the input information to the already solved classes.

Figure 14 compares each fault/class that the individual F1 indexes obtained when using raw data with those attained when incorporating standard deviations. Faults correctly diagnosed (F1 $> 50\%$) using the original process variables (faults 1, 2 6, 7, 17, 18, and 20) were trained without including standard deviations to avoid increasing unnecessarily data processing. Figure 14 shows the same results for these classes.

Once again, the mL and ML approaches are compared. The results achieved including standard deviation are detailed in Table 7 for both cases.

Similarly, as demonstrated in the beginning of this section, the difference between both approaches comes from the attempt to infer the diagnosis of the normal state, which would only achieve good quantitative results when the rest of states are perfectly addressed. The normal class difficulties could be managed using a reliable fault detection system from the wide variety of reliable systems available in the literature.[26–28] Note that the inference of the normal class is now enhanced (F1 $= 13.5$ in Table 4 to F1 $= 46.6$ in Table 7) due to the improvements attained in the classification of the rest of considered classes.

Next, the learning curves for these novel solutions are presented (Figure 15). Comparing the classification performance versus that shown in Figure 9, an evident change of behavior is observed. Many of the faults badly classified with the original process variables (faults 4, 10, 11, etc.) now belong to these faults following asymptotic learning curves. Nevertheless, several classes (3, 9, and 15) remain poorly identified, thus requiring a different information representation that is able to gather their specific features by means of alternative variables.

The CPU time required for training each data set size is also depicted in Figure 15 right-hand vertical axis. It exponentially increases with doubling number of samples in training sets. Comparing the F1 asymptotes regarding the CPU

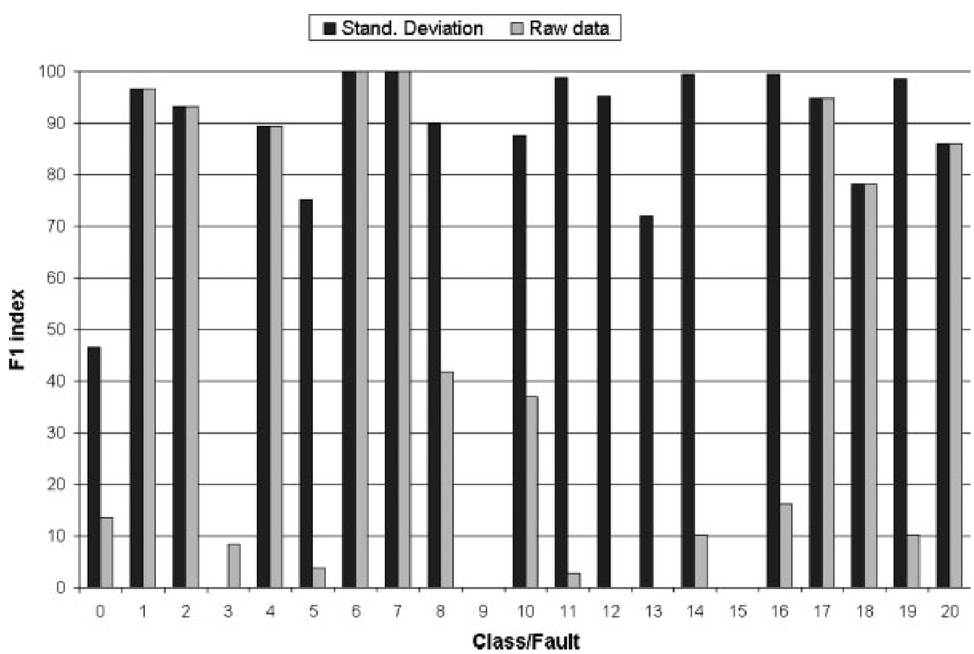

Figure 14. F1 indexes obtained using original process variables including standard deviation.

Table 7. Single-Fault Diagnosis. F1 Index (%) for the mL and ML Approaches Including Standard Deviation Variables

	mL	ML
0	96.2	46.6
1	96.7	96.7
2	93.3	93.3
3	0.0	0.0
4	89.3	89.3
5	75.2	75.2
6	100	100
7	100	100
8	89.8	89.8
9	0.0	0.0
10	87.7	87.7
11	98.8	98.8
12	95.3	95.3
13	72.0	72.0
14	99.3	99.3
15	0.0	0.0
16	99.4	99.4
17	94.8	94.8
18	78.2	78.2
19	98.6	98.6
20	85.9	85.9
Average	78.6	76.2

time, an intelligent trade-off decision between training calculation time and classification performance can be made. In that case, the 351-sample training set is selected because of its better performance and attainable training calculation time.

Results including new information representation: simultaneous faults

Including the new deviation variables, most of the faults are correctly diagnosed, while just three classes remain being poorly identified (faults 3, 9, and 15). Therefore, the new information representation allows extending the simultaneous fault diagnosis to other combinations. Table 8 summarizes some two, three, and four simultaneous faults diagnosed by the ML-based FDS when including the new data representation.

Not only double and triple simultaneous faults shown in Table 8 are identified with a high performance, but even combinations of more faults are well diagnosed.

Four simultaneous faults were also considered. The first example (faults 2, 7, 17, and 18) consists of those faults previously tested. Same results were obtained since the same variables are contemplated (see Table 5).

The second and third set of the four simultaneous faults correspond to novel fault combinations involving faults that were not individually identified initially but whose classification performance was notably increased later by the new information representation (section "Enhancing information representation"). Now, these faults are not only just identified when they occur individually but also when they arise simultaneously with others.

However, some faults cannot be simultaneously identified in the presence of other faults due to the overlapping of fault contributions. Such effect becomes more evident when fault simultaneity increases. A clear example is given by the fourth four-simultaneous fault shown in Table 8. It includes fault 6, which biases, more heavily, variables 1, 7, or 16 of the TEP than the other faults. These variables are key variables, which the system uses to identify faults 1 or 2, and the contribution of these faults to the simultaneous fault is unobserved by the effect of fault 6. Identification of faults 18, 11, and 14 is strongly penalized in first three four-simultaneous faults respectively because of that effect.

Such overlapping deficiencies can be viewed as a problem or as a natural and correct FDS response. In that sense, when a higher sensitivity fault (biases the same variables deeper) is happening, other faults which affect much less the same variables may be considered negligible and should not necessarily be identified.

Next, details of the same tested four simultaneous faults diagnosed initially (2, 7, 17, and 18) are shown in Table 9.

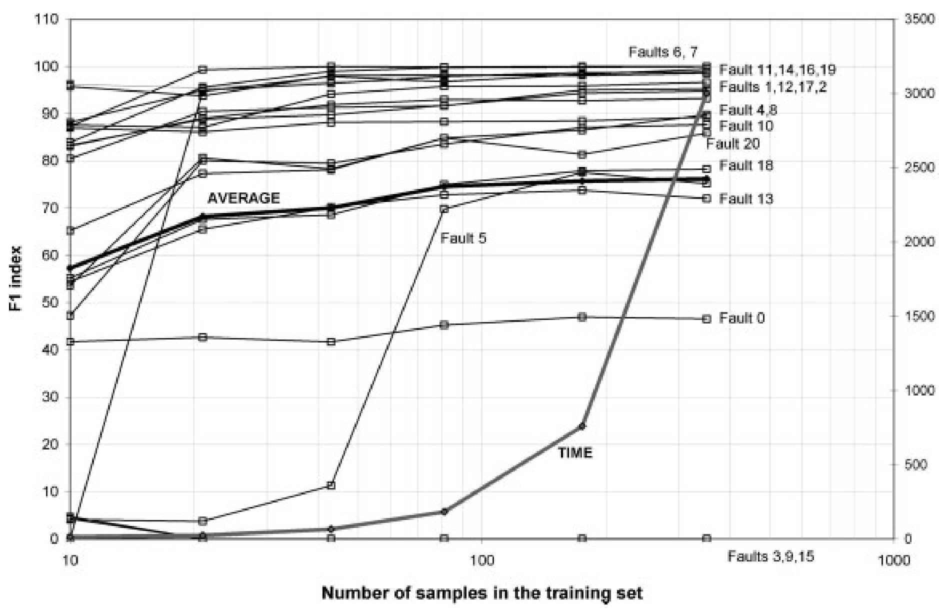

Figure 15. Individual and average learning curves in logarithmic scale join to their calculation times.

Table 8. Multiple-Fault Diagnosis. F1 Index for Different Simultaneous Faults

	1	2	6	7	11	14	17	18
2 Simultaneous faults (1, 2)	97.2	96.1	–	–	–	–	–	–
2 Simultaneous faults (2, 7)	–	76.8	–	82.1	–	–	–	–
2 Simultaneous faults (6, 7)	–	–	96.6	99.9	–	–	–	–
2 Simultaneous faults (11, 14)	–	–	–	–	97.2	75.8	–	–
2 Simultaneous faults (17, 18)	–	–	–	–	–	–	90.5	64.6
3 Simultaneous faults (2, 7, 17)	–	82.8	–	68.7	–	–	91.8	–
3 Simultaneous faults (2, 11, 14)	–	94.2	–	–	94.9	76.9	–	–
3 Simultaneous faults (7, 11, 14)	–	–	–	99.9	34.0	57.8	–	–
4 Simultaneous faults (2, 7, 17, 18)	–	80.0	–	58.8	–	–	86.6	17.8
4 Simultaneous faults (2, 7, 11, 14)	–	77.1	–	84.9	30.5	59.6	–	–
4 Simultaneous faults (2, 11, 14, 17)	–	94.2	–	–	20.8	9.2	94.7	–
4 Simultaneous faults (6, 7, 1, 2)	0	0	100	100	–	–	–	–

Although the same F1 indexes are obtained for each fault (see Table 6), since the same variables are contemplated for those classes, however, additional information is now provided. By using this additional information, representation for those faults poorly classified, a much lower false diagnosis occurs, thus increasing the operators' trust on the FDS response. It can be easily verified by comparison of column "b" contents in Tables 6 and 9. The 880 wrongly diagnosed samples (Table 6) have come down to 138, thus revealing this additional advantage on simultaneous fault diagnosis by increasing the diagnosis performance of the remaining faults.

Conclusions

This article deals with the simultaneous fault diagnosis problem from the machine-learning viewpoint by using an ML approach, which shows superior performance with respect to mL approaches considered in the literature. Benefits of ML approach are clarified theoretically and practically using a flexible and robust machine-learning algorithm called SVM.

Table 9. Details for the Classification of the Improved Four Simultaneous Faults (2, 7, 17, 18) Case

	a	*b*	*c*	*d*	Precision	Recall	F1 (%)
1	0	0	0	1112	–	–	–
2	667	0	333	112	100	66.7	80.0
3	0	0	0	1112	–	–	0
4	0	0	0	1112	–	–	0
5	0	0	0	1112	–	–	0
6	0	114	0	998	0	–	0
7	416	0	584	112	100	41.6	58.8
8	0	0	0	1112	–	–	0
9	0	0	0	1112	–	–	0
10	0	8	0	1104	0	–	0
11	0	0	0	1112	–	–	0
12	0	0	0	1112	–	–	0
13	0	0	0	1112	–	–	0
14	0	0	0	1112	–	–	0
15	0	0	0	1112	–	–	0
16	0	0	0	1112	–	–	0
17	763	0	237	112	100	76.3	86.6
18	98	0	902	112	100	9.8	17.8
19	0	0	0	1112	–	–	0
20	0	16	0	1096	0	–	0

a: Happening and diagnosed; *b*: diagnosed but not happening; *c*: happening and not diagnosed; *d*: not happening and not diagnosed.

To evaluate the quality of the ML approach diagnosing single and simultaneous faults, the fault diagnosis was formulated as a classification problem and several measurements were proposed to quantify the FDS performance. In that sense, all the required variables to evaluate the performance of a classification tool were adapted to be used on fault diagnosis, thus giving a rigorous frame for future diagnosis comparison.

Single and simultaneous faults generated from a challenging benchmark case study (TEP) were tested and diagnosed by the ML-SVM system offering very good results even when dealing with four simultaneous faults. By means of improving the fault information representation, a process deeper knowledge was achieved resulting in a remarkable FDS performance increase. Note that these noticeable results were achieved without the need of training simultaneous faults, thus enabling any possible fault combination. In contrast with the mL view, the ML approach allowed to train any class with specific variables, which contributed to reduce computational cost and to focus efforts on challenging faults.

To completely solve the TEP single and simultaneous fault diagnosis problem, further study on information representation is required for the few classes (3 over 20) having the lowest diagnosis performance. This should be done following a similar analysis to the one presented in this article. Besides that, performance increase can be reached by tuning the SVM parameters, by selecting key input variables (feature selection), by adding new attributes (codifying the available data in different ways), or by adjusting the quality of input data by means of data filtering or any other statistical technique. Furthermore, the proposed ML approach could implement diverse learning algorithms other than the SVM adopted in this work, thus allowing further tuning opportunities.

Acknowledgments

Financial support from "Spanish Ministerio de Educación y Ciencia, MEC" through the FPI program and project no. DPI 2006-05673 is fully appreciated. Support received from the European Union through contracts MRTN-CT-2004-512233 and RFC-CR-04006 is also acknowledged.

Literature Cited

1. Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: Part I. Quantitative model-based methods. *Comput. Chem. Eng.* 2003;27:293–312.
2. Joachims T. Learning to classify text using support vector machines. Dissertation, Kluwer. 2002.
3. Schapire Y, Singer, RE. Boostexter: a boosting-based system for text categorization. *Machine Learn.* 2000;39:135–168.
4. Raich A, Çinar A. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *AIChE J.* 1996;42: 995–1009.
5. Raich A, Çinar A. Diagnosis of process disturbances by statistical distance and angle measures. *Comput. Chem. Eng.* 1997;21:661–673.
6. Vedam H, Venkatasubramanian V. PCA-SDG based process monitoring and fault diagnosis. *Control Eng. Pract.* 1998;7:903–917.
7. Maurya MR, Rengaswamy R, Venkatasubramaian V. Application of signed digraphs based analysis for fault diagnosis of chemical flowsheets. *Eng. Appl. Artif. Intell.* 2004;17:501–518.
8. Boser B, Guyon I, Vapnik V. A training algorithm for optimal margin classifiers. In: *Proceedings of the Workshop on Computational Learning Theory*, COLT, Pittsburgh, PA, 1992.
9. Kulkarni A, Jayaraman VK, Kulkarni BD. Knowledge incorporated support vector machines to detect faults in TE Process. *Comput. Chem. Eng.* 2005;29:2128–2133.
10. Chiang LH, Kotancheck ME, Kordon AK. Fault diagnosis based on Fisher discriminant analysis and support vector machines. *Comput. Chem. Eng.* 2003;28:1389–1401.
11. Downs J, Vogel E. A plant-wide industrial process control problem. *Comput. Chem. Eng.* 1992;17:245–255.
12. Chen J, Howell J. Towards distributed diagnosis of the Tennessee Eastman process benchmark. *Control Eng. Pract.* 2002;10:971–987.
13. Chiang LH, Russell EL, Braatz RD. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemom. Intell. Lab. Syst.* 2000;50:243–252.
14. Yélamos I, Escudero G, Graells M, Puigjaner, L. Simultaneous fault diagnosis in chemical plants using support vector machines. In: Plesu V, Agachi PS, editors. *Proceedings of the Seventeenth European Symposium on Computer Aided Process Engineering*, Bucharert, Romania, 2007. 1253–1258. ISBN: 978-0-444-53157-5.
15. Dash S, Rengaswamy R, Venkatasubramanian V. Fuzzy-logic based trend classification for fault diagnosis of chemical processes. *Comput. Chem. Eng.* 2003;27:347–362.
16. Leung D, Romagnoli J. An integration mechanism for multivariate knowledge-based fault diagnosis. *J. Process Control* 2006;12:15–26.
17. Gertler J, Weihua L, Huang Y, McAvoy T. Isolation enhanced principal component analysis. *AIChE J.* 1999;45:323–334.
18. Wachs A, Lewin DR. Improved PCA methods for process disturbance and failure identification. *AIChE J.* 1999;45:1688–1700.
19. Yélamos I, Escudero G, Graells M, Puigjaner L. Performance assessment of a novel fault diagnosis system based on support vector machines. *Comput. Chem. Eng.* Submitted.
20. Yélamos I, Escudero G, Graells M, Puigjaner L. Fault diagnosis based on support vector machines and systematic comparison to existing approaches. In: Marquardt W, Pantelides C, editors. Sixteenth ESCAPE and 9th International Symposium on PSE, ISBN: 0-444-52969-1, 2006:2009–2014.
21. Manning C, Schütze H. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.
22. Joachims T. Making large-scale SVM learning practical. In: Schölkopf B, Burges C, and Smola A, editors. *Advances in Kernel Methods—Support Vector Learning*. Cambridge, MA: MIT Press, 1999.
23. Allwein EL, Schapire RE, Singer Y. Reducing multiclass to binary: a unifying approach of margin classifiers. *J. Machine Learn. Res.* 2000;1:113–141.
24. Vapnik VN. *Statistical Learning Theory*. New York: Wiley, 1998.
25. McAvoy TJ, Ye N. Base control for the Tennessee Eastman problem. *Comput. Chem. Eng.* 1994;18:383–413.
26. Yoon S, MacGregor JF. Statistical and causal model-based approaches to fault detection and isolation. *AIChE J.* 2000;46:1813–1824.
27. Bakshi BR. Multiscale PCA with application to multivariate statistical process monitoring. *AIChE J.* 1998;44:1596–1610.
28. Gertler J. *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, 1998. ISBN: 0-8247-9427-3.

Manuscript received Apr. 30, 2007, and revision received Aug. 4, 2007.