Stacking Dynamic Time Warping for the Diagnosis of Dynamic Systems

Carlos J. Alonso[1], Oscar J. Prieto[1], Juan J. Rodríguez[2], Aníbal Bregón[1], and Belarmino Pulido[1]

[1] Intelligent Systems Group (GSI), Department of Computer Science, E.T.S.I Informática, University of Valladolid, Valladolid, Spain
[2] Department of Civil Engineering, University of Burgos, Burgos, Spain

Abstract. This paper explores an integrated approach to diagnosis of complex dynamic systems. Consistency-based diagnosis is capable of performing automatic fault detection and localization using just correct behaviour models. Nevertheless, it may exhibit low discriminative power among fault candidates. Hence, we combined the consistency based approach with machine learning techniques specially developed for fault identification of dynamic systems. In this work, we apply Stacking to generate time series classifiers from classifiers of its univariate time series components. The Stacking scheme proposed uses K-NN with Dynamic Time Warping as a dissimilarity measure for the level 0 learners and Naïve Bayes at level 1. The method has been tested in a fault identification problem for a laboratory scale continuous process plant. Experimental results show that, for the available data set, the former Stacking configuration is quite competitive, compare to other methods like tree induction, Support Vector Machines or even K-NN and Naïve Bayes as stand alone methods.

1 Introduction

Diagnosis of complex dynamic systems is still an open research problem. It has been approached using a wide variety of techniques, [2], being the four main approaches: Knowledge Based —including expert systems—, Case Based Reasoning, Machine Learning and Model Based Systems. Currently, it seems clear that no single technique is capable to claim its success in every field. Therefore, an increasing number of diagnosis systems have opted for hybrid solutions. In this work, we propose a combination of Model Based and Machine Learning methods. Our approach relies primarily upon model-based diagnosis, but it has been enhanced via machine-learning techniques to overcome some drawbacks.

In the Artificial Intelligence field, the DX community has developed Consistency Based Diagnosis, CBD, as the major paradigm for model based diagnosis [5]. CBD can be summarized as an iterative cycle of behavior prediction, discrepancy or conflict detection, fault localization or candidate generation, and candidate refinement by means of new measurements. In this cycle, diagnosis candidates can be automatically obtained from conflicts using a minimal hitting set algorithm.

D. Borrajo, L. Castillo, and J.M. Corchado (Eds.): CAEPIA 2007, LNAI 4788, pp. 11–20, 2007.

Although CBD is able to perform both fault detection and localization with just models for correct behavior, the absence of fault models knowledge is partly responsible of the low discriminative power that CBD may exhibit [8]. Particularly in dynamic systems, with low observability, [3], it is not uncommon to localize a set that involves a large number of components, without been able to discriminate between them. Usually, to solve this drawback, knowledge about fault modes is introduced. We have opted for the predictive approach, which use models of fault modes to estimate faulty behavior, as in Sherlock [6] or GDE+ [17]. Based on such estimation, non-consistent fault modes are rejected. Nevertheless, the increase in the discriminative power has a price. For a system with N components and only two behaviors —ok and faulty—, diagnosis must discriminate between 2^N behavioral mode assignments. When M behavioral models are considered —one correct, $M-1$ faulty—, diagnosis must discriminate among M^N mode assignments. This is the problem faced by any model-based diagnosis proposal which attempts fault identification [8].

For practical reasons, this theoretical approach is infeasible in real systems and many approaches have been proposed in recent years to deal with the complexity issue. However, to the best of our knowledge, there is no general architecture suitable for any kind of system. In fact, many approaches just perform fault detection and localization, or rely upon a combination of some kind of heuristic, which helps focusing the diagnosis task. This will be also our approach.

In the recent past, [13] it has been proposed a diagnosis architecture which combined consistency-based diagnosis with machine learning techniques, maintaining the soundness of the CDB approach. CDB was in charge of fault detection and localization, while machine learning was use for fault identification. The identification problem was approached as a multivariate time series classification task and time series classifiers were induce off line from simulated data.

In this work, this approach is explored further, studying the possibilities of Dynamic Time Warping, DTW, [10] as the basis of induced classifiers. K-Nearest Neighbours, K-NN, using DTW as a dissimilarity measure behaves reasonably well for some univariate problems but degrades in the multivariate case. Although DTW can be easily extended for the multivariate case, these extensions are far from optimal. Instead, we have opted for using univariate classification methods to handle each multivariate time series component —itself a univariate time series— introducing an additional classifier to obtain the final class.

The univariate classification method is K-NN with DTW dissimilarity measure; the outputs of each univariate classifier are combined by another classifier to obtain the multivariate time series classifier. This approach is an special case of Stacking [20], a method designed for the combination of classifiers. The classifiers are organized in levels, being the outputs of one level the inputs for the next level. Normally, Stacking is used for combining classifiers obtained with different methods. In the present work, the same method (DTW) is used for all the classifiers in the first level. Nevertheless, each classifier uses a different subset of the input features, the series formed by the values of one of the variables.

The rest of the paper is organized as follows. Next section will introduce the compilation technique used to perform consistency-based diagnosis, which is the basis for our model-based diagnosis system. Section 3 will describe how to induce multivariate time series classifiers based on Stacking and DTW. Section 4 shows how to integrate these classifiers with the consistency based approach to diagnosis. Afterwards, we present some results on a case study plant. Finally, we discuss the results and draw some conclusions.

2 Consistency-Based Diagnosis Using Possible Conflict

CBD generate minimal candidates —i.e., minimal set of faulty components— computing the hitting set of minimal conflicts [14]. Hence the central issue in CBD is computing minimal conflicts from symptoms in an efficient way. Reiter [14] gives a precise definition of the concept of conflict. Intuitively, a conflict is a set of components such that at least one of its elements is faulty: other way, there will be a logical inconsistency between current observations, the system description —i.e., the models of the system— and the assumption that all the components of the conflict work properly.

Although Reiter introduced the theoretical framework of CBD, the computational paradigm is the General Diagnostic Engine [6] proposed by de Kleer and Williams. GDE computes conflicts coupling the simulation process with a dependency recording device, an Assumption based True Maintenance Systems, ATMS. Although this approach is quite efficient in static domains with qualitative variables, it does not scale up to dynamic systems described with quantitative equations. Nevertheless, GDE like conflicts computation may be tackled through compilation techniques, avoiding the need of on line dependency recording.

The computation of possible conflicts is a compilation technique which, under certain assumptions, is equivalent to on-line conflict calculation in GDE. A detailed description of consistency based diagnosis with possible conflicts can be found in [11,12]. For the sake of brevity, we just resume how to perform CBD with possible conflicts.

The main idea behind the *possible conflict* concept is that the set of subsystems capable to generate a conflict can be identified off-line. More over, possible conflicts approach provides a computational technique to automatically obtain, from a graphical representation of the system, the symbolic expression of the models associated to each *possible conflict*.

Those models can be used to perform fault detection. If there is a discrepancy between predictions from those models and current observations, the possible conflict would be responsible for such a discrepancy and should be confirmed as a real conflict. Afterwards, diagnosis candidates are obtained from conflicts following Reiter's theory.

3 Machine Learning Techniques for Fault Identification

There are several works that use machine learning techniques for diagnosis. Those works use methods as Inductive Logic Programming [9], Neural

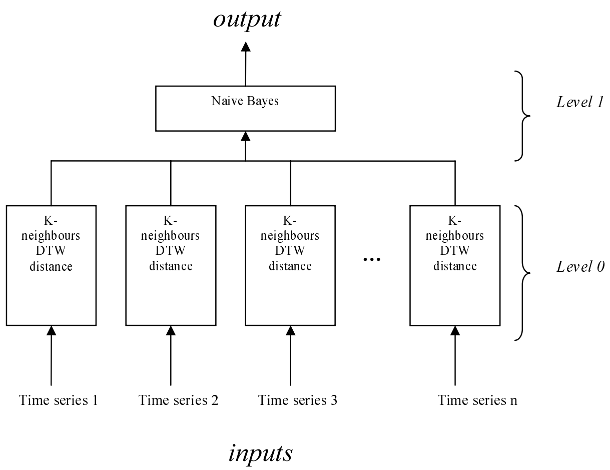

Fig. 1. Schema of the Stacking variant used in this work

Networks [19], KDD techniques [16], decision trees [18], and combination of techniques like recurrent neural networks, Wavelet On-Line Pre-processing (WOLP) and Autonomous Recursive Task Decomposition (ARTD) [15].

To take into account the dynamic nature of the problem, we have approached diagnosis as the task of classifying the recent evolution of the variables involved in the system. Each historic episode of a variable may be considered as a time series. Hence, the evolution of the variables of the systems may be registered as a multivariate time series. In this way, the diagnosis of a dynamic system may be managed as a particular case of multivariate time series classification.

In this work we propose to use Stacking for combining several univariate time series classifiers to obtain the classification of multivariate time series. Each of these classifiers is K-NN using DTW distance. The outputs of these classifiers are combined using Naïve Bayes. The schema is showed in the figure 1.

When we use stacking, classification is achieved using a multilevel architecture. Stacking uses a first layer called level 0 that is composed of the *base classifiers*, in our case, the k-neighbors with DTW distance classifier. The inputs of this level are the time series we classify and we have one classifier for each univariate time series. The output of this layer is the input of the second layer called level 1. This layer is composed of the meta-learner that learns how to combine the decisions of the base classifiers. The output of the level 1 layer is the target class.

Naïve Bayes has been selected for the level 1 classifier because it is a global and continuous classifier; these are desirable properties for the level 1 classifier. Against it, it is the fact that the univariate series are not independent. For the domain problem that we are interested in, it seems that some subset of components is adequate to predict some classes and other subsets to predict another. Hence, it is reasonable to expect some independence between different subsets of the multivariate time series components. On the contrary, some dependence must exits among the components of each subset. Nevertheless, the fact of training the level 0 classifiers with different and disjoint data gives a chance to increase independence. Although usually Stacking applies different level 0 classifiers to the same learning set, in this work we propose to use different learning sets with the same level 0 classifier. We can do this because of the nature of the process we are classifying. This approach tries to offer an alternative of multivariate DTW.

4 Integration Proposal

Consistency-based diagnosis automatically provides fault isolation based on fault detection results. Using possible conflicts, consistency-based diagnosis can be easily done without on-line dependency recording. The proposed diagnosis process will incrementally generate the set of candidates consistent with observations. In the off-line stage, we initially analyze the system and find out every possible conflict, pc_i. Then, we build an executable model, SD_{pc_i}, for each pc_i.

In the on-line stage, we perform a semi-closed loop simulation with each executable model SD_{pc_i}:

1. *repeat*
   (a) *simulate*$(SD_{pc_i}; OBS_{pc_i}) \rightarrow PRED_{pc_i}$.
   (b) *if* $|PRED_{pc_i} - OBS_{Opc_i}| > \delta_{pc_i}$ *confirm* pc_i *as a real conflict.*
   (c) *update(set of candidates, set of activated pcs)*
2. *until every* pc_i *is activated or time elapsed.*

Where OBS_{pc_i} denotes the set of input observations available for SD_{pc_i}; $PRED_{pc_i}$ represents the set of predictions obtained from SD_{pc_i}; OBS_{Opc_i} denotes the set of output observations for SD_{pc_i}; and δ_{pc_i} is the maximum value allowed as the dissimilarity value between OBS_{0pc_i} and $PRED_{pc_i}$.

Without further information about fault modes, consistency-based diagnosis will just provide a list of feasible faulty candidates. In recent works, [1,13,3] it has been proposed a diagnosis architecture which combines consistency based diagnosis with possible conflicts with induced multivariate time series classifiers. These classifiers provide a ranking of fault modes compatible with consistency based diagnosis candidates. In this way, the logical soundness of consistency based diagnosis is preserved, because fault models are not used to propose non consistent behaviors. Nonetheless, the ranking information may improve fault isolation accuracy and may provide some clue towards fault identification.

Let's *CLASSIFIER_StackDTW(t; c)* denote an invocation of the classifier induced using stacking univariate DTWs, with a fragment of series from t to the min(current time, t+maximum series length), and with the set of candidates c.

With this notation, the integration of the fault mode knowledge in the consistency based diagnosis cycle may be simply stated. Just add:

(d) CLASSIFIER_StackDTW $(t_0$, *set of candidates)*

to the on-line simulation loop, with t_0 the starting time of the series, prior to the first conflict confirmation. In this way, the diagnostician may provide fault isolation a la consistency based, ordering fault candidates according to the confidence assigned to them by the classifiers and providing fault identification information.

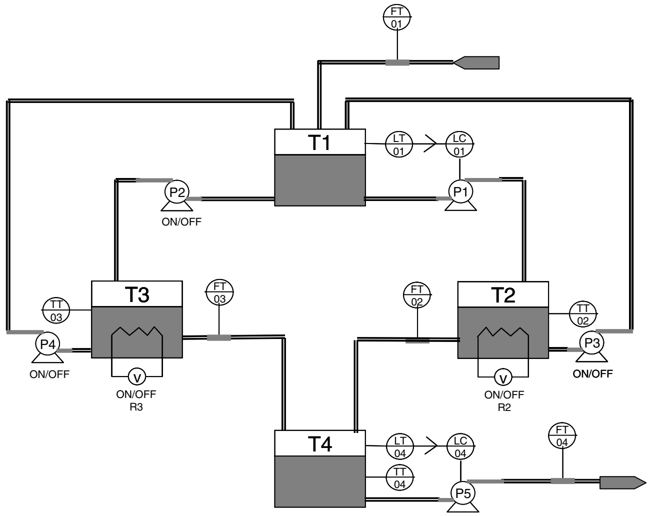

Fig. 2. The diagram of the plant

5 Case Study

5.1 The System to Be Diagnosed

For this work, we have used the laboratory scale plant shown in figure 2. Although a laboratory plant, its complexity is comparable to the one encountered in several subsystems of real processes. It is made up of four tanks $\{T_1, \ldots, T_4\}$, five pumps $\{P_1, \ldots, P_5\}$, and two PID controllers acting on pumps P_1, P_5 to keep the level of $\{T_1, T_4\}$ close to the specified set point. To control temperature on tanks $\{T_2, T_3\}$ we use two resistors $\{R_2, R_3\}$, respectively.

In this plant we have eleven different measurements: levels of tanks T_1 and T_4 –$\{LT01, LT04\}$–, the value of the PID controllers on pumps $\{P_1, P_5\}$ –$\{LC01, LC04\}$–, in-flow on tank T_1 –$\{FT01\}$–, outflow on tanks $\{T_2, T_3, T_4\}$ –$\{FT02, FT03, FT04\}$–, and temperatures on tanks $\{T_2, T_3, T_4\}$ –$\{TT02, TT03, TT04\}$–. Action on pumps $\{P_2, P_3, P_4\}$, and resistors –$\{R_2, R_3\}$– are also known.

The plant may work with different configurations and a simple setting without recirculation —pumps $\{P_3, P_4\}$ and resistor R_2 are switch off— has been chosen.

5.2 Possible Conflicts for the System

We have used common equations in simulation for this kind of process.

1. t_{dm}: mass balance in tank t.
2. t_{dE}: energy balance in tank t.
3. t_{fb}: flow from tank t to pump.
4. t_f: flow from tank t through a pipe.
5. r_p: resistor failure.

Based on these equations we have found the set of possible conflicts shown in table 1. In the table, second column shows the set of constraints used in

Table 1. Possible conflicts found for the laboratory plant; constraints, components, and the estimated variable for each possible conflict

	Constraints	*Components*	*Estimate*
PC_1	$t1_{dm}, t1_{fb1}, t1_{fb2}$	T_1, P_1, P_2	$LT01$
PC_2	$t1_{fb1}, t2_{dm}, t2_f$	T_1, T_2, P_1	$FT02$
PC_3	$t1_{fb1}, t2_{dm}, r2_p$	T_1, P_1, T_2, R_2	$TT02$
PC_4	$t1_{fb2}, t3_{dm}, t3_f$	T_1, P_2, T_3	$FT03$
PC_5	$t1_{fb2}, t3_{dm}$	T_1, P_2, T_3	$TT03$
PC_6	$t4_{dm}$	T_4	$LT04$
PC_7	$t4_{fb}$	T_4, P_5	$FT04$

Table 2. Fault modes considered

Class	*Component*	*Description*
f_1	T_1	*Small leakage in tank* T_1
f_2	T_1	*Big leakage in tank* T_1
f_3	T_1	*Pipe blockage* T_1 (*left outflow*)
f_4	T_1	*Pipe blockage* T_1 (*right outflow*)
f_5	T_3	*Leakage in tank* T_3
f_6	T_3	*Pipe blockage* T_3 (*right outflow*)
f_7	T_2	*Leakage in tank* T_2
f_8	T_2	*Pipe blockage* T_2 (*left outflow*)
f_9	T_4	*Leakage in tank* T_4
f_{10}	T_4	*Pipe blockage* T_4 (*right outflow*)
f_{11}	P_1	*Pump failure*
f_{12}	P_2	*Pump failure*
f_{13}	P_5	*Pump failure*
f_{14}	R_2	*Resistor failure in tank* T_2

each possible conflict, which are minimal with respect to the set of constraints. Third column shows those components involved. Fourth column indicates the estimated variable for each possible conflict.

5.3 Experimental Design

We have considered the fourteen fault modes shown in table 2.

Possible conflicts related to fault modes are shown in the following theoretical fault signature matrix shown in table 3.

It should be noticed that these are the fault modes classes which can be distinguished for fault identification. In the fault localization stage, the following pair of faults $\{f_1, f_2\}$, $\{f_4, f_{11}\}$, and $\{f_3, f_{12}\}$, and $\{f_{10}, f_{13}\}$ can not be separately isolated.

Due to the cost of obtaining enough data for a fourteen classes classification problem from the laboratory plant, we have resorted to a detail, non linear quantitative simulation of the plant. We have run twenty simulations for each class, adding noise in the sensors readings. We have modeled each fault class with a parameter in the $[0, 1]$ range. We have made twenty simulations for each

Table 3. PCs and their related fault modes

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
PC_1	1	1	1	1							1	1		
PC_2				1			1	1			1			
PC_3				1			1				1			1
PC_4			1		1	1						1		
PC_5			1		1							1		
PC_6									1					
PC_7										1			1	

class of fault. Each simulation lasted 900 seconds. We randomly generate the fault magnitude, and its origin, in the interval $[180, 300]$. We also have assumed that the system is in stationary state before the fault appears.

The data sampling was one data per second. However, due to the slow dynamics in the plant, we can select one data every three seconds without losing discrimination capacity. Since we just have eleven measures, then each simulation will provide eleven series of three hundred numeric elements.

5.4 Results

In this section, the results from the proposed method are compared to some standard machine learning methods: Decision Trees, Naïve Bayes Classifiers and Support Vector Machines (with the linear kernel).

Moreover, the results for Nearest Neighbor method, for different values of the number of neighbors, are included. They are from [4]. For this method, DTW is used considering that the distance between two multivariate series is the sum of the distances for each variable.

The methods are used with series of different lengths, because the classifiers are going to be used for early classification. We consider some significative length values: 30, 40, 50 and 100% of the series. The length of the full series is 15 minutes.

The results were obtained using 10-fold stratified cross-validation. Moreover, the Stacking method uses another internal cross-validation, also with 10 folds.

Table 4 shows the results obtained using different methods for different percentages of the series length. Stacking DTW classifiers has better results than any of the other considered methods, for all the considered lengths.

Table 4 also shows the average rank of each method. For each method, the average rank is calculated from its ranks in the different folds. For each fold, the methods are ranked. The best method in the fold is assigned the number 1, the second the number 2, and so on. The average rank of the proposed method is always smaller than 2.0. According to Friedman test [7] these average ranks are, for all the considered lengths, significantly different from the mean rank.

The second best method is decision trees. If we compare the results of the two best methods for the different folds, using a paired t-test the differences are significant when using half-length and full series.

Table 4. Results of the different methods for different lengths of the series

	Series Length	Decision Tree	Naïve Bayes	SVM	DTW 1-NN	DTW 3-NN	DTW 5-NN	Stacking DTW+NBC
Accuracy	30%	68.57	59.64	44.64	56.07	57.86	53.21	73.93
(percentage)	40%	94.29	87.50	80.71	87.86	84.29	83.21	95.36
	50%	91.79	91.79	84.64	91.07	87.14	83.57	96.79
	100%	93.93	83.57	92.14	91.43	88.57	85.00	98.57
Average	30%	2.35	3.60	6.45	4.60	3.75	5.30	1.95
ranks	40%	2.05	3.70	6.05	3.60	5.35	5.75	1.50
	50%	3.25	3.25	6.00	3.00	4.90	6.00	1.60
	100%	3.05	5.75	3.50	3.30	4.95	6.10	1.35

6 Conclusions

This work further explores an integrated approach to diagnosis that pretends to be effective in complex dynamic systems, combing Consistency Based Diagnosis with machine learning techniques.

The main contribution of this work is the proposal of Stacking to address multivariate time series classification from univariate time series classifiers induced for each component of the original time series.This new proposal improves previous results because of the better performance of the induced classifier. With 40% of the series, long before the system reaches another stationary state, the new method provides a 95% success rate. The only drawback is the need to train the meta level learner with different lengths of the time series.

The results using Stacking with DTW and Naive Bayes are much better than the results from DTW and Naive Bayes. Hence, the success of the method is not a consequence of combining classifiers that work well isolated. The proposed method has also better results than other standard machine learning methods, such as decision trees and support vector machines.

Although the proposed method was designed for the diagnosis of dynamic systems, it can be used for other multivariate time series classification tasks. The method will be tested with data sets from other domains.

Normally, Stacking is using for combining several methods, while in the presented variant it is used with the same method with different inputs. The two approaches can be used in conjunction, so we plan to test the method using several methods for the first level.

Acknowledgments. This work has been partially funded by Spanish Ministry of Education and Culture, through grant DPI2005–08498, and Junta Castilla y León VA088A05.

References

1. Alonso, C., Rodríguez, J.J., Pulido, B.: Enhancing consistency based diagnosis with machine learning techniques. In: Conejo, R., Urretavizcaya, M., Pérez-de-la-Cruz, J.-L. (eds.) Current Topics in Artificial Intelligence. LNCS (LNAI), vol. 3040, pp. 312–321. Springer, Heidelberg (2004)

2. Balakrishnan, K., Honavar, V.: Intelligent diagnosis systems. Journal of Intelligent Systems 8 (1998)
3. Bregón, A., Pulido, B., Simón, M.A., Moro, I., Prieto, O., Rodríguez, J., González, C.A.: Focusing fault localization in model-based diagnosis with case-based reasoning. In: 17th International Workshop on Principles of Diagnosis (2006)
4. Bregón, A., Simón, M.A., Rodríguez, J.J., Alonso, C., Pulido, B., Moro, I.: Early fault classification in dynamic systems using case-based reasoning. In: Marín, R., Onaindía, E., Bugarín, A., Santos, J. (eds.) CAEPIA 2005. LNCS (LNAI), vol. 4177, Springer, Heidelberg (2006)
5. de Kleer, J., Mackworth, A.K., Reiter, R.: Characterising diagnosis and systems. In: Readings in Model Based Diagnosis, pp. 54–65. Morgan-Kauffman, San Francisco (1992)
6. de Kleer, J., Williams, B.C.: Diagnosing with behavioral modes. In: Eleventh International Joint Conference on Artificial Intelligence (IJCAI 1989) (1989)
7. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
8. Dressler, O., Struss, P.: The consistency-based approach to automated diagnosis of devices. In: Brewka, G. (ed.) Principles of Knowledge Representation, pp. 269–314. CSLI Publications, Stanford (1996)
9. Feng, C.: Inducting temporal fault diagnostic rules from a qualitative model. In: Muggleton, S. (ed.) Inductive Logic Programming, Academic Press, London (1992)
10. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowledge and Information Systems 7(3), 358–386 (2005)
11. Pulido, B., Alonso, C.: An alternative approach to dependency-recording engines in consistency-based diagnosis. In: Cerri, S.A., Dochev, D. (eds.) AIMSA 2000. LNCS (LNAI), vol. 1904, pp. 111–120. Springer, Heidelberg (2000)
12. Pulido, B., Alonso, C., Acebes, F.: Lessons learned from diagnosing dynamic systems using possible conflicts and quantitative models. In: Monostori, L., Váncza, J., Ali, M. (eds.) IEA/AIE 2001. LNCS (LNAI), vol. 2070, pp. 135–144. Springer, Heidelberg (2001)
13. Pulido, B., Rodríguez, J.J., Alonso, C., Prieto, O.J., Gelso, E.R.: Diagnosis of continuous dynamic systems: Integrating consistency based diagnosis with machine learning techniques. In: 16th IFAC World Congress, Prague, Czech Republic (2005)
14. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32, 57–95 (1987)
15. Roverso, D.: Fault diagnosis with the aladdin transient classifier. In: System Diagnosis and Prognosis: Security and Condition Monitoring Issues III, AeroSense2003, Aerospace and Defense Sensing and Control Technologies Symposium (2003)
16. Sleeman, D., Mitchell, F., Milne, R.: Applying KDD techniques to produce diagnostic rules for dynamic systems. Technical Report AUCS/TR9604, Department of Computing Science. University of Aberdeen (1996)
17. Struss, P., Dressler, O.: Physical negation: Introducing fault models into the general diagnostic engine. In: Eleventh International Joint Conference on Artificial Intelligence (IJCAI 1989), Detroit, Michigan, USA (1989)
18. Suárez, A.J., Abad, P.J., Ortega, J.A., Gasca, R.M.: Diagnosis progresiva en el tiempo de sistemas dinámicos. In: IV Jornadas de ARCA, Sistemas Cualitativos y Diagnosis, JARCA'02 (2002)
19. Venkatasubramanian, V., Chan, K.: A neural network methodology for process fault diagnosis. AIChE J. 35(12), 1993–2001 (1989)
20. Wolpert, D.H.: Stacked generalization. Neural Networks 5(2), 241–260 (1992)