A Hybrid CBR Model for Forecasting in Complex Domains

Florentino Fdez-Riverola[1] and Juan M. Corchado[2]

[1] Dpto. de Informática, E.S.E.I., University of Vigo,
Campus Universitario As Lagoas s/n., 32004, Ourense, Spain
riverola@uvigo.es
[2] Dpto. de Informática y Automática, University of Salamanca,
Facultad de Ciencias, Plaza de la Merced, s/n., 37008, Salamanca, Spain
corchado@usal.es

Abstract. A hybrid neuro-symbolic problem solving model is presented in which the aim is to forecast parameters of a complex and dynamic environment in an unsupervised way. In situations in which the rules that determine a system are unknown, the prediction of the parameter values that determine the characteristic behaviour of the system can be a problematic task. The proposed model employs a case-based reasoning system to wrap a growing cell structures network, a radial basis function network and a set of Sugeno fuzzy models to provide an accurate prediction. Each of these techniques is used in a different stage of the reasoning cycle of the case-based reasoning system to retrieve, to adapt and to review the proposed solution to the problem. This system has been used to predict the red tides that appear in the coastal waters of the north west of the Iberian Peninsula. The results obtained from those experiments are presented.

1 Introduction

Forecasting the behaviour of a dynamic system is, in general, a difficult task, especially when dealing with complex, stochastic domains for which there is a lack of knowledge. In such a situation one strategy is to create an adaptive system which possesses the flexibility to behave in different ways depending on the state of the environment. An artificial intelligence approach to the problem of forecasting in such domains offers potential advantages over alternative approaches, because it is able to deal with uncertain, incomplete and even inconsistent data numerically represented. This paper presents a hybrid artificial intelligence (AI) model for forecasting the evolution of complex and dynamic environments that can be numerically represented. The effectiveness of this model is demonstrated in an oceanographic problem in which neither artificial neural network nor statistical models have been sufficiently successful.

However, successful results have been already obtained with hybrid case-based reasoning systems [1–3] used to predict the evolution of the temperature of the water ahead of an ongoing vessel, in real time. The hybrid system proposed in this paper presents a new synthesis that brings several AI subfields together (CBR, ANN and Fuzzy inferencing). The retrieval, reuse, revision and learning stages of the CBR system use the previously mentioned technologies to facilitate the CBR adaptation to a wide range of complex problem domains and to completely automate the reasoning process of the proposed forecasting mechanism.

The structure of the paper is as follows: first the hybrid neuro-symbolic model is explained in detail, then a case of study is briefly outlined and finally the results are analyzed together with the conclusions and future work.

2 Overview of the Hybrid CBR based Forecasting Model

In this paper, a method for automating the CBR reasoning process is presented for the solution of complex problems in which the cases are characterised predominantly by numerical information. Figure 1 illustrates the relationships between the processes and components of the proposed hybrid CBR system. The diagram shows the technology used at each stage, where the four basic phases of the CBR cycle are shown as rectangles.

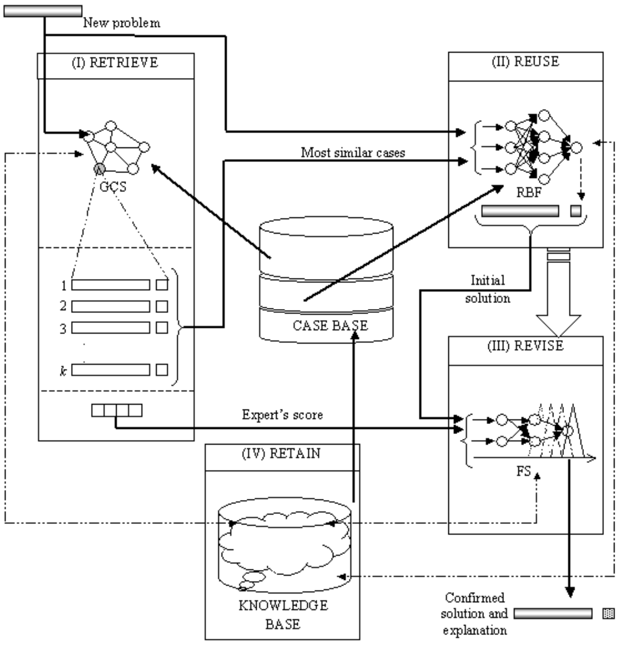
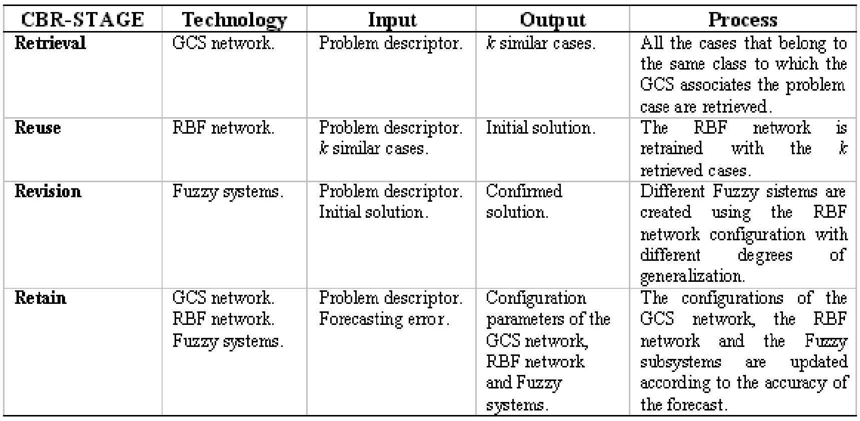


Fig. 1. Hybrid neuro-symbolic model.

The retrieval stage is carried out using a Growing Cell Structures (GCS) ANN [4]. The GCS facilitates the indexation of cases and the selection of those that are most similar to the problem descriptor. The reuse and adaptation of cases is carried out with a Radial Basis Function (RBF) ANN [5], which generates an initial solution creating a forecasting model with the retrieved cases. The revision is carried out using a group of pondered fuzzy systems that identify potential incorrect solutions. Finally, the learning stage is carried out when the real value of the variable to predict is measured and the error value is calculated, updating the knowledge structure of the whole system.

When a new problem is presented to the system, a new problem descriptor (case) is created and the GCS neural network is used to recover from the case-base the k most similar cases to the given problem (identifying the class to which the problem belongs, see Figure 2).

In the reuse phase, the values of the weights and centers of the RBF neural network used in the previous forecast are retrieved from the knowledge-base. These network parameters together with the k retrieved cases are then used to retrain the RBF network and to obtain an initial forecast (see Figure 2). During this process the values of the parameters that characterise the network are updated.



CBR-STAGE	Technology	Input	Output	Process
Retrieval	GCS network.	Problem descriptor.	*k* similar cases.	All the cases that belong to the same class to which the GCS associates the problem case are retrieved.
Reuse	RBF network.	Problem descriptor. *k* similar cases.	Initial solution.	The RBF network is retrained with the *k* retrieved cases.
Revision	Fuzzy systems.	Problem descriptor. Initial solution.	Confirmed solution.	Different Fuzzy sistems are created using the RBF network configuration with different degrees of generalization.
Retain	GCS network. RBF network. Fuzzy systems.	Problem descriptor. Forecasting error.	Configuration parameters of the GCS network, RBF network and Fuzzy systems.	The configurations of the GCS network, the RBF network and the Fuzzy subsystems are updated according to the accuracy of the forecast.

Fig. 2. Summary of technologies employed by the hybrid model.

In the revision phase, the initial solution proposed by the RBF neural network is modified according to the response of the fuzzy revision subsystem (a set of fuzzy models). Each fuzzy system has been created from the RBF network using neurofuzzy techniques [6] as it will be seen later.

The revised forecast is then retained temporarily in the forecast database. When the real value of the variable to predict is measured, the forecast value for the variable can then be evaluated, through comparison of the actual and forecast value and the error obtained (see Figure 2). A new case, corresponding to this forecasting operation, is then stored in the case-base. The forecasting error value is also used to update several parameters associated with the GCS network, the RBF network and the fuzzy systems.

2.1 Growing Cell Structures Operation

To illustrate the working model of the GCS network inside the whole system, a two-dimensional space will be used, where the cells (neurons) are connected and organized into triangles [4]. Each cell in the network (representing a generic case), can be seen as a "prototype" that identifies a set of similar problem descriptors. The basic learning process in a GCS network is carried out in three steps.

In the first step, the cell c, with the smallest distance between its weight vector, w_c, and the actual case, x, is chosen as the *winner cell.* The second step consists in the adaptation of the weight vector of the winning cells and their neighbours. In the third step, a *signal counter* is assigned to each cell, which reflects how often a cell has been chosen as winner. Repeating this process several times, for all the cases of the case-base, a network of cells will be created.

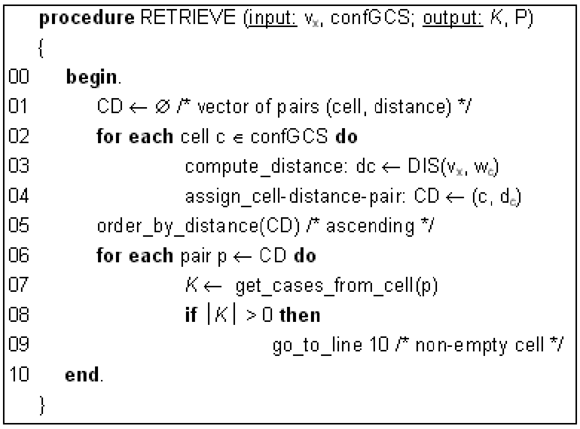
For each class identified by the GCS neural network, a vector of values is maintained (see Figure 1). This vector (to which we will refer as "importance" vector) is initialised with a same value for all its components whose sum is one, and represents the accuracy of each fuzzy system (used during the revision stage) with respect to that class. During revision, the importance vector associated to the class to which the problem case belongs, is used to ponder the outputs of each fuzzy system. For each forecasting cycle, the value of the importance vector associated with the most accurate fuzzy system is increased and the other values are proportionally decreased. This is done in order to give more relevance to the most accurate fuzzy system of the revision subsystem.

Figure 3 provides a more concise description of the GCS-based case retrieval regime described above, where v_x is the value feature vector describing a new problem, confGCS represents the set of cells describing the GCS topology after the training, K is the retrieved set of most relevant cases given a problem and P represents the "importance" vector for the identified prototype.

The neural network topology of a GCS network is incrementally constructed on the basis of the cases presented to the network. Effectively, such a topology represents the result of the basic clustering procedure and it has the added advantage that inter-cluster distances can be precisely quantified. Since such networks contain explicit distance information, they can be used effectively in CBR to represent: (i) an *indexing structure* which indexes sets of cases in the case-base and, (ii) a *similarity measure* between case sets [7].

2.2 Radial Basis Function Operation

Case adaptation is one of the most problematic aspects of the CBR cycle, mainly if we have to deal with problems with a high degree of dynamism and for which there is a lack of knowledge. In such a situation, RBF networks have demonstrated their utility as universal approximators for closely modelling these continuous processes [8].



```
    procedure RETRIEVE (input: $v_x$, confGCS; output: $K$, P)
    {
00    begin.
01       CD ← ∅ /* vector of pairs (cell, distance) */
02       for each cell c ∈ confGCS do
03             compute_distance: dc ← DIS($v_x$, $w_c$)
04             assign_cell-distance-pair: CD ← (c, $d_c$)
05       order_by_distance(CD) /* ascending */
06       for each pair p ← CD do
07             $K$ ← get_cases_from_cell(p)
08             if $|K|$ > 0 then
09                   go_to_line 10 /* non-empty cell */
10    end.
    }
```

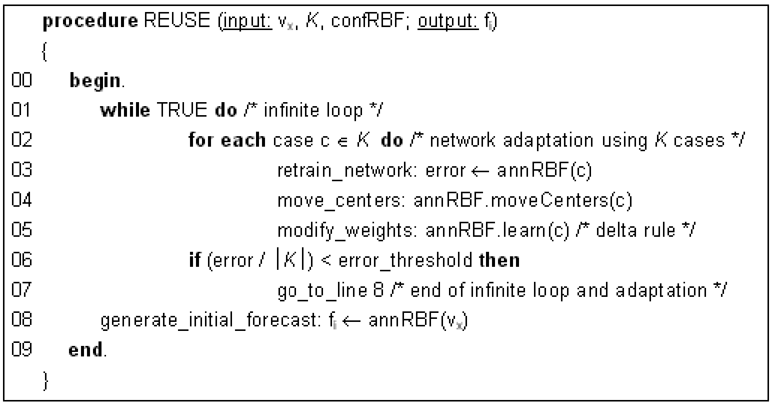
Fig. 3. GCS-based case retrieval.

Again to illustrate how the RBF networks work, a simple architecture will be presented. Initially, three vectors are randomly chosen from the training data set and used as centers in the middle layer of the RBF network. All the centers are associated with a Gaussian function, the width of which, for all the functions, is set to the value of the distance to the nearest center multiplied by 0.5 (see [5] for more information about RBF network).

Training of the network is carried out by presenting pairs of corresponding input and desired output vectors. After an input vector has activated each Gaussian unit, the activations are propagated forward through the weighted connections to the output units, which sum all incoming signals. The comparison of actual and desired output values enables the mean square error (the quantity to be minimized) to be calculated. A new center is inserted into the network when the average error in the training data set does not fall during a given period.

The closest center to each particular input vector is moved toward the input vector by a percentage a of the present distance between them. By using this technique the centers are positioned close to the highest densities of the input vector data set. The aim of this adaptation is to force the centers to be as close as possible to as many vectors from the input space as possible. The value of a is linearly decreased by the number of iterations until its value becomes zero; then the network is trained for a number of iterations (1/4 of the total of established iterations for the period of training) in order to obtain the best possible weights for the final value of the centers.

Figure 4 provides a more concise description of the RBF-based case adaptation regime, where v_x is the value feature vector describing a new problem, K is the retrieved set of most relevant cases, confRBF represents the previously configuration of the RBF network and f_i represents the initial forecast generated by the RBF.



```
    procedure REUSE (input: $v_x$, $K$, confRBF; output: $f_i$)
    {
00    begin.
01      while TRUE do /* infinite loop */
02          for each case $c \in K$ do /* network adaptation using $K$ cases */
03              retrain_network: error ← annRBF(c)
04              move_centers: annRBF.moveCenters(c)
05              modify_weights: annRBF.learn(c) /* delta rule */
06          if (error / $|K|$) < error_threshold then
07              go_to_line 8 /* end of infinite loop and adaptation */
08      generate_initial_forecast: $f_i$ ← annRBF($v_x$)
09    end.
    }
```

Fig. 4. RBF-based case adaptation.

The working model commented above together with their good capability of generalization, fast convergence, smaller extrapolation errors and higher reliability over difficult data, make this type of neural networks a good choice that fulfils the necessities of dealing with this type of problems. It is very important to train this network with a consistent number of cases. Such consistence in the training data set is guaranteed by the GCS network, that provides consistent classifications that can be used by the RBF network to auto-tuning its forecasting model.

2.3 Fuzzy System Operation

The two main objectives of the proposed revision stage are: to validate the initial prediction generated by the RBF and, to provide a set of simplified rules that explain the system working mode. The construction of the revision subsystem is carried out in two main steps:

(i) First, a Sugeno-Takagi fuzzy model [9] is generated using the trained RBF network configuration (centers and weights) in order to transform a RBF neural network to a well interpretable fuzzy rule system [6].

(ii) A measure of similarity is applied to the fuzzy system with the purpose of reducing the number of fuzzy sets describing each variable in the model. Similar fuzzy sets for one parameter are merged to create a common fuzzy set to replace them in the rule base. If the redundancy in the model is high, merging similar fuzzy sets for each variable might result in equal rules that also can be merged, thereby reducing the number of rules as well. When similar fuzzy sets are replaced by a common fuzzy set representative of the originals, the system's capacity for generalization increases.

In our model, the fuzzy systems are associated with each class identified by the GCS network, mapping each one with its corresponding value of the importance vector. There is one "importance" vector for each class or prototype. These fuzzy systems are used to validate and refine the proposed forecast.

The value generated by the revision subsystem is compared with the prediction carried out by the RBF and its difference (in percentage) is calculated. If the initial forecast does not differ by more than 10% of the solution generated by the revision subsystem, this prediction is supported and its value is considered as the final forecast. If, on the contrary, the difference is greater than 10% but lower than 30%, the average value between the value obtained by the RBF and that obtained by the revision subsystem is calculated, and this revised value adopted as the final output of the system. Finally, if the difference is greater or equal to 30% the system is not able to generate an appropriate forecast. This two thresholds have been identified after carrying out several experiments and following the advice of human experts.

The exposed revision subsystem improves the generalization ability of the RBF network. The simplified rule bases allow us to obtain a more general knowledge of the system and gain a deeper insight into the logical structure of the system to be approximated. The proposed revision method then help us to ensure a more accurate result, to gain confidence in the system prediction and to learn about the problem and its solution. The fuzzy inference systems also provides useful information that is used during the retain stage.

2.4 Retain

As mentioned before, when the real value of the variable to predict is known, a new case containing the problem descriptor and the solution is stored in the case-base. The importance vector associated with the retrieved class is updated in the following way: the error percentage with respect to the real value is calculated, then the fuzzy system that has produced the most accurate prediction is identified and the error percentage value previously calculated is added to the degree of importance associated with this fuzzy subsystem. As the sum of the importance values associated to a class (or prototype) has to be one, the values are normalized. When the new case is added to the case-base, its class is identified. The class is updated and the new case is incorporated into the network for future use.

3 A Case of Study: The Red Tides Problem

The oceans of the world form a highly dynamic system for which it is difficult to create mathematical models [10]. The rapid increase in dinoflagellate numbers, sometimes to millions of cells per liter of water, is what is known as a *bloom* of phytoplankton (if the concentration ascends above the 100.000 cells per liter). The type of dinoflagellate in which this study is centered is the pseudo-nitzschia spp diatom, causing of amnesic shellfish poisoning (known as ASP).

In the current work, the aim is to develop a system for forecasting one week in advance the concentrations (in cells per liter) of the pseudo-nitzschia spp at different geographical points.

The problem of forecasting, which is currently being addressed, may be simply stated as follows:

- **Given**: a sequence of data values (representative of the current and immediately previous state) relating to some physical and biological parameters,
- **Predict**: the value of a parameter at some future point(s) or time(s).

The raw data (sea temperature, salinity, PH, oxygen and other physical characteristics of the water mass) which is measured weekly by the monitoring network for toxic proliferations in the CCCMM (Centro de Control da Calidade do Medio Marino, *Oceanographic environment Quality Control Centre*, Vigo, Spain), consists of a vector of discrete sampled values (at 5 meters' depth) of each oceanographic parameter used in the experiment, in the form of a time series. These data values are complemented by data derived from satellite images stored on a database. The satellite image data values are used to generate cloud and superficial temperature indexes which are then stored with the problem descriptor and subsequently updated during the CBR operation. Table 1 shows the variables that characterise the problem. Data from the previous 2 weeks (W_{n-1}, W_n) is used to forecast the concentration of pseudo-nitzschia spp one week ahead (W_{n+1}).

Table 1. Variables that define a case.

Variable	Unit	Week
Date	dd-mm-yyyy	W_{n-1}, W_n
Temperature	Cent. degrees	W_{n-1}, W_n
Oxygen	milliliters/liter	W_{n-1}, W_n
PH	acid/based	W_{n-1}, W_n
Transmitance	%	W_{n-1}, W_n
Fluorescence	%	W_{n-1}, W_n
Cloud index	%	W_{n-1}, W_n
Recount of diatoms	cel/liter	W_{n-1}, W_n
Pseudo-nitzschia spp	cel/liter	W_{n-1}, W_n
Pseudo-nitzschia spp (future)	*cel/liter*	W_{n+1}

Our proposed model has been used to build an hybrid forecasting system that has been tested along the north west coast of the Iberian Peninsula with data collected by the CCCMM from the year 1992 until the present. The prototype used in this experiment was set up to forecast the concentration of the pseudo-nitzschia spp diatom of a water mass situated near the coast of Vigo (geographical area A0 ((42°28.90' N, 8°57.80' W) 61 m)), a week in advance. Red tides appear when the concentration of pseudo-nitzschia spp is higher than 100.000 cell/liter. Although the aim of this experiment is to forecast the value of the concentration, the most important aspect is to identify in advance if the concentration is going to exceed this threshold.

A case-base was built with the above mentioned data normalized between [-1, 1]. For this experiment, four fuzzy inference systems have been created from the RBF network, which uses 18 input neurons, between three and fifty neurons in the hidden layer and a single neuron in the output layer.

The following section discusses the results obtained with the prototype developed for this experiment as well as the conclusions and future work.

4 Results, Conclusions and Future Work

The hybrid forecasting system has been proven in the coast of north west of the Iberian Peninsula with data collected by the CCCMM from the year 1992 until the present time. The average error in the forecast was found to be 26.043,66 cel/liter and only 5.5% of the forecasts had an error higher than 100.000 cel/liter. Although the experiment was carried out using a limited data set, it is believed that these error value results are significant enough to be extrapolated over the whole coast of the Iberian Peninsula.

Two situations of special interest are those corresponding to the *false alarms* and the *not detected blooms*. The first one happens when the system predicts bloom (concentration of pseudo-nitzschia $\geq$ 100.000 cel/liter) and this doesn't take place (real concentration $\leq$ 100.000 cel/liter). The second, more important, arise when bloom really exists and the system doesn't detect it.

Table 2 shows the predictions carried out with success (in absolute value and %) and the erroneous predictions differentiating the not detected blooms and the false alarms. This table also shows the average error obtained with several techniques. As it can be shown, the combination of different techniques in the form of the hybrid CBR system previously presented, produces better results that a RBF neural network working alone or anyone of the tested statistical techniques. This is due to the effectiveness of the revision subsystem and the retrained of the RBF neural network with the cases recovered by GCS network. The hybrid system is more accurate than any of the other techniques studied during this investigation.

Table 2. Summary of results forecasting pseudo-nitzschia spp.

Method	OK	OK (%)	N. detect.	Fal. alarms	Aver. error (cel/liter)
CBR-ANN-FS	**191/200**	**95,5%**	**8**	**1**	**26.043,66**
RBF	185/200	92,5%	8	7	45.654,20
ARIMA	174/200	87%	10	16	71.918,15
Quadratic Trend	184/200	92%	16	0	70.354,35
Moving Average	181/200	90,5%	10	9	51.969,43
Simp. Exp. Smooth.	183/200	91,5%	8	9	41.943,26
Lin. Exp. Smooth.	177/200	88,5%	8	15	49.038,19

In summary, this paper has presented an automated hybrid CBR model that employs case-based reasoning to wrap a growing cell structures network (for the index tasks to organize and retrieve relevant data), a radial basis function network (that contributes generalization, learning and adaptation capabilities) and a set of Sugeno fuzzy models (acting as experts that revise the initial solution) to provide a more effective prediction. The resulting hybrid model thus combines complementary properties of both connectionist and symbolic AI methods in order to create a real time autonomous forecasting system.

In conclusion, the hybrid reasoning problem solving approach may be used to forecast in complex situations where the problem is characterized by a lack of knowledge and where there is a high degree of dynamism. The prototype presented here will be tested in different water masses and a distributed forecasting system will be developed based on the model in order to monitor 500 km. of the North West coast of the Iberian Peninsula.

This work is financed by the project: *Development of techniques for the automatic prediction of the proliferation of red tides in the Galician coasts, PGIDT-00MAR30104PR*, inside the Marine Program of investigation of Xunta de Galicia. The authors want to thank the support lent by this institution, as well as the data facilitated by the CCCMM.

References

1. Corchado, J. M., Lees, B.: A Hybrid Case-based Model for Forecasting. Applied Artificial Intelligence, 15, num. 2, (2001) 105–127
2. Corchado, J. M., Lees, B., Aiken, J.: Hybrid Instance-based System for Predicting Ocean Temperatures. International Journal of Computational Intelligence and Applications, 1, num. 1, (2001) 35–52
3. Corchado, J. M., Aiken, J., Rees, N.: Artificial Intelligence Models for Oceanographic Forecasting. Plymouth Marine Laboratory, U.K., (2001)
4. Fritzke, B.: Growing Self-Organizing Networks-Why?. In Verleysen, M. (Ed.). European Symposium on Artificial Neural Networks, ESANN-96. Brussels, (1996) 61–72
5. Fritzke, B.: Fast learning with incremental RBF Networks. Neural Processing Letters, 1, num. 1, (1994) 2–5
6. Jin, Y., Seelen, W. von., and Sendhoff, B.: Extracting Interpretable Fuzzy Rules from RBF Neural Networks. Internal Report IRINI 00-02, Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany, (2000)
7. Azuaje, F., Dubitzky, W., Black, N., and Adamson, K.: Discovering Relevance Knowledge in Data: A Growing Cell Structures Approach. IEEE Transactions on Systems, Man and Cybernetics, 30, (2000) 448–460
8. Corchado, J. M., and Lees, B.: Adaptation of Cases for Case-based Forecasting with Neural Network Support. In Pal, S. K., Dilon, T. S., and Yeung, D. S. (Eds.). Soft Computing in Case Based Reasoning. London: Springer Verlag, (2000) 293–319
9. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics, 15, (1985) 116–132
10. Tomczak, M., Godfrey, J. S.: Regional Oceanographic: An Introduction. Pergamon, New York, (1994)