An Automated Hybrid CBR System for Forecasting *

Florentino Fdez-Riverola[1], Juan M. Corchado[2], and Jesús M. Torres[3]

[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
[3] Dpto. de Física Aplicada, University of Vigo,
Facultad de Ciencias, Lagoas Marcosende, 36200, Vigo, Spain
jesu@uvigo.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 system employs a case-based reasoning model that incorporates 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 if the prediction needs to be achieved in real time. 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. This paper presents a hybrid artificial intelligence (AI) model for forecasting the evolution of complex and dynamic environments. The effectiveness of this model is demonstrated in an oceanographic problem in which neither artificial neural network nor statistical models have been sufficiently successful.

Several researchers [1, 2] have used k-nearest-neighbour algorithms for time series predictions. Although a k-nearest-neighbour algorithm does not, in itself,

* This research was supported in part by PGIDT00MAR30104PR project of Xunta de Galicia, Spain

constitute a CBR system, it may be regarded as a very basic and limited form of CBR operation in numerical domains. [1] uses a relatively complex hybrid CBR-ANN system. In contrast, [2] forecast a data set just by searching in a given sequence of data values for segments that closely match the pattern of the last n measurements and then, by supposing that similar antecedent segments are likely to be followed by similar consequent segments. Other examples of CBR systems that carry out predictions can be found in [3], [4], [5], [6] and [7].

In most cases, the CBR systems used in forecasting problems have flat memories with simple data representation structures using k-nearest-neighbour metric in their retrieve phase. K-nearest-neighbour metric are acceptable if the system is relatively stable and well understood, but if the system is dynamic and the forecast is required in real time, it may not be possible to easily redefine the k-nearest-neighbour metrics adequately. The dominant characteristic of the adaptation stage used in these models are similarity metrics or statistical models, although, in some systems, case adaptation is accomplished manually. If the problem is very complex, there may be no planned adaptation strategy and the most similar case is used directly, but it is believed that adequate adaptation is one of the keys to a successful CBR paradigm. In the majority of the systems surveyed, case revision (if carried out at all) is performed by human expert, and in all the cases the CBR systems are provided with a small case-base. A survey of such forecasting CBR systems can be found in [8].

Traditionally, CBR systems have been combined with other technologies like artificial neural networks, rule-based systems, constraint satisfaction problems and others, producing successful results to solve specific problems [9, 10]. Although, in general each specific problem and domain requires a particular solution, this paper proposes a CBR based solution for forecasting the evolution of a complex problem, with a high degree of dynamism for which there is a lack of knowledge, and for which an adaptive learning system is required. This paper also presents, a method for automating the CBR reasoning process for the solution of problems in which the cases are characterised predominantly by numerical information.

Successful results have been already obtained with hybrid case-based reasoning systems [11–13] and 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 presented in this paper use the previously mentioned technologies to facilitate the CBR adaptation to a wide range of complex problem domains (for instance, the afore-mentioned red tides problem) 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; a case study is then briefly outlined; the results obtained to date with the proposed forecasting system are analyzed, and finally, the conclusions and future work are presented.

2 The Hybrid CBR based Forecasting System

This section proposes a CBR based model for forecasting the evolution of parameters related to problems that can be numerically represented, that evolve with time, for which there is an incomplete knowledge and for which the forecasting system has to be completely automated.

In this context, in order to forecast the value of any variable, a problem descriptor should be generated. A problem descriptor is composed of a vector with the variables that describe the problem and the solution. In this case, this vector holds numerical variables.

Figure 1 illustrates the relationships between the processes and components of the hybrid CBR system. In general, we can say that the forecast values are obtained using a neural network enhanced hybrid case-base reasoning system. The cyclic CBR process shown in the figure has been inspired by the work of [11] and [12]. The diagram shows the technology used at each stage, where the four basic phases of the CBR cycle are shown as rectangles.

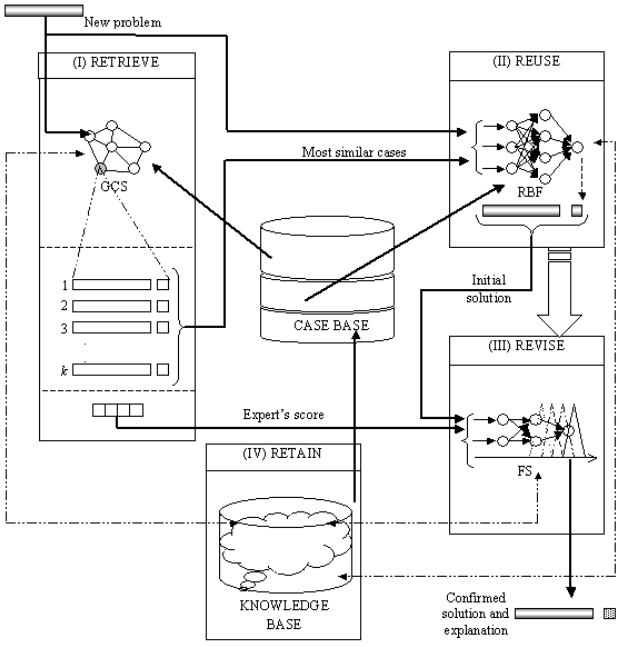

Fig. 1. Hybrid neuro-symbolic system.

The retrieval stage is carried out using a Growing Cell Structures (GCS) ANN [14]. The GCS facilitates the indexing 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 [15], which generates

an initial solution creating a forecasting model with the retrieved cases. The revision is carried out using a group of 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. Now we present the working cycle of the CBR system illustrated in Figure 1.

When a new problem is presented to the system a new problem descriptor (case) is created, and the GCS neuronal network is used to recover from the case-base the k more 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 cases. Expert's score.	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. Expert's score. Initial solution.	Confirmed solution.	Different fuzzy systems 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 [16].

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 [14]. Initially, three vectors are randomly chosen from the training data set. Each cell in the network (representing a generic case), is associated with a weight vector, w, of the same dimension (problem descriptor + solution) that cases stored in the case-base. At the beginning of the learning process, the weight vector of each cell is initialized with random values [17]. The basic learning process in a GCS network consists of topology modification and weight vector adaptations carried out in three steps. The training vectors of the GCS network are the cases stored in the CBR case-base, as indicated in Figure 1.

In the first step of each learning cycle, the cell c, with the smallest euclidean distance between its weight vector, w_c, and the actual case, x, is chosen as the *winner cell* or best-match cell. The second step consists in the adaptation of the weight vector of the winning cell and their neighbours (positioning the vectors more near to the actual case). In the third step, a *signal counter* is assigned to each cell, which reflects how often a cell has been chosen as winner. Growing cell structures also modify the overall network structure by inserting new cells into those regions that represent large portions of the input data (with a higher value of the signal counter), or removing cells that do not contribute to the input data representation.

Repeating this process several times, for all the cases of the case-base, a network of cells will be created. Each cell will have associated cases that have a similar structure and each cell will represent a class. These cells can be seen as a "prototype" that identifies a set of similar problem descriptors.

For each class identified by the GCS neural network a vector of values is maintained (see Figure 1). This "importance" vector is initialized 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. Each value of the importance vector is associated with one of the fuzzy systems. 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.

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 [18].

When dealing with complex, dynamic, stochastic problems that can be numerically represented, the decision of what retrieval model is better to use for a CBR that need to be completely automated is not a trivial task. The main characteristics that should have the retrieval model are: an adequate adaptation capacity to the problem domain and strong learning capability. In such a situation, GCS neural networks offer several advantages over other approaches and well know metrics:

- GCS is a neural network which is able to automatically generate a k dimensional network structure highly adapted to a given but not explicitly known probability distribution. The adaptation rules of the GCS also enable the processing of data with changing probability distributions, what helps in the construction of dynamic systems for complex problems.
- Its ability to perform problem-dependent error measures allows the implementation of better adaptive data representations (insertion and deletion of cells) in comparison with static-topology models. This characteristic guarantees the adaptation capacity above mentioned.
- Its ability to interrupt a learning process or to continue a previously interrupted one permits the construction of incremental and dynamic learning systems.
- The GCS self-organising model consists of a small number of constant parameters. There is therefore no need to define time-dependent or decay schedule parameters, what facilitates the implementation of autonomous systems.
- GCS networks have demonstrated their capacity to process both small and high dimensionality data in several application domains and can operate in either unsupervised or supervised learning modes. This characteristic guarantees the construction of dynamic learning systems.

Another specially interesting fact is that the GCS network are structurally similar to the RBF network. The GCS network provides *consistent* classifications that can be used by the RBF network to auto-tune its forecasting model. (given a new problem descriptor the GCS network finds its class and retrieves all cases belonging to that class). Irregularities, discontinuities and exceptions in the data are avoided if the classifications are consistent. In our model, this is achieved by means of taking into account the whole case (problem descriptor + solution) during the training of the GCS network.

In the light of all these reasons, the GCS neural network has been selected to solve the problem of the classification and indexing in our hybrid CBR based forecasting system.

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 [9]. In our system, the term adaptation is used to represent

the process of *adapting* the configuration of the RBF network using the k most similar cases retrieved by the GCS network, instead of the classic meaning in CBR theory.

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 [15] 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.

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.

A new center is inserted into the network when the average error in the training data set does not fall during a given period. There are different methods to identify the place where the new center will be inserted.

The main advantages of this type of networks can be summarized as follows:

- The RBF network is capable of approximating nonlinear mappings effectively.
- The training time of the RBF network is quite low compared to that of other neural network approaches such as the multi-layer perceptron, because training of the two layers of the network is decoupled.
- The RBF networks are successful for identifying regions of sample data not in any known class because it uses a non-monotonic transfer function based on the Gaussian density function.
- RBF network is less sensitive to the order in which data is presented to them because one basis function takes responsibility for one part of the input space.

The above characteristics 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 consistency in the training data set is guaranteed by the GCS network that carries out a classification of

the data using the problem descriptor and the solution of the cases stored in the case-base.

RBF networks can also be used to generate Fuzzy inference systems [16]. This characteristic has been used in this model for the automatic generation of the revision subsystem as it will be explained in the following section.

2.3 Fuzzy System Operation

Rule extraction from artificial neural networks is considered to be important due to the following reasons [16]:

- Rule extraction provides artificial neural networks with an explanation capability, which makes it possible for the user to check on the internal logic of the system.
- Rule extraction helps to discover previously unknown dependencies in data sets and thus new knowledge about the system can be acquired.
- It is believed that a rule system with good interpretability improves the generalization ability of neural networks where training data are insufficient.

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 [19] 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, the following conditions should be satisfied:

- The basis functions of the RBF neural network have to be Gaussian functions.
- The output of the RBF neural network has to be normalized.
- The basis functions may have different variances.
- A certain number of basis functions for the same input variable should share a mutual center and a mutual variance.

(ii) A measure of similarity is applied to the fuzzy system [20] 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 this 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. Given a

problem descriptor and a proposed forecast for it, each of the fuzzy inference systems that compose the revision subsystem generates a solution that is pondered according to the importance vector value associated to the GCS class to which the problem belongs.

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 doesn't differ by more than a certain threshold 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 the defined threshold, 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. This problem dependent threshold must be identified with empirical experiments and following the advice of human experts. In the theoretical CBR cycle, the main purpose of the revise phase is to try out the solution for real and change it before it is retained, making use of knowledge that is *external* to the system. In our system, this is all done in the retain phase, implementing the revise phase as a form of validation and using knowledge that is *internal* to the system.

Fuzzy systems provide a solution to the revision stage when dealing with complex problems, with a high degree of dynamism and for which there is a lack of knowledge. The exposed revision subsystem improves the generalization ability of the RBF network. Fuzzy models, especially if acquired from data, may contain redundant information in the form of similarities between fuzzy sets. As similar fuzzy sets represent compatible concepts in the rule base, a model with many similar fuzzy sets becomes redundant, unnecessarily complex and computationally demanding. 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. 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 and the sum dividing up accordingly between them. 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 [21]. *Red tides* are the name for the discolourations caused by dense concentrations of microscopic sea plants, known as phytoplankton. The rapid increase in dinoflagellate numbers, sometimes to millions of cells per liter of water, is described as a *bloom* of phytoplankton (concentration levels above the 100.000 cells per liter). This study focusses on the pseudo-nitzschia spp diatom dinoflagellate, which causes amnesic shellfish poisoning along the north west coast of the Iberian Peninsula in late summer and autumn [22].

Surface waters of these blooms are associated with the production of toxins, resulting in mortality of fish and other marine organisms. Toxic blooms of dinoflagellates fall into three categories: (i) blooms that kill fish but few invertebrates; (ii) blooms that kill primarily invertebrates; (iii) blooms that kill few marine organisms, but whose toxins are concentrated within the siphons, digestive glands, or mantle cavities of filter-feeding bivalve mollusc such as clams, oysters, and escallops.

The nature of the red tides problem has changed considerably over the last two decades around the world. Where formerly a few regions were affected in scattered locations, now virtually every coastal state is threatened, in many cases over large geographic areas and by more than one harmful or toxic algal species [23]. Few would argue that the number of toxic blooms, the economic losses from them, the types of resources affected, and the number of toxins and toxic species have all increased dramatically in recent years in all over the world. Disagreement only arises with respect to the reasons for this expansion.

Models of dinoflagellate blooms have been developed from several different perspectives [24–26] but the end result is that despite the proven utility of models in so many oceanographic disciplines, there are no predictive models of population development, transport, and toxin accumulation. There is thus a clear need to develop models for regions subject to red tides, and to incorporate biological behavior and population dynamics into those simulations [27].

An artificial intelligence (AI) approach to the problem of forecasting in the ocean environment offers potential advantages over alternative approaches, because it is able to deal with uncertain, incomplete and even inconsistent data. Several AI techniques have been used to forecast the evolution of different oceanographic parameters [28, 11, 12]. The reported work shows how CBR systems have a greater facility for forecasting oceanographic parameters than other statistical and AI based models [13].

3.1 Forecasting Red Tides

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, the diatom that produces the most harmful red tides, at different geographical points. The approach builds on the methods and expertise previously developed in earlier research.

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, 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. For this experiment, four fuzzy inference systems have been created from the RBF network, and

they were initialised with a value of (0.25, 0.25, 0.25, 0.25) for each class (or prototype) in the GCS network. The RBF network used in the framework of this experiment, uses 18 input neurons, between three and fifty neurons in the hidden layer and a single neuron in the output layer, being the output of the network the concentration of pseudo-nitzschia spp for a given water mass.

The following section discusses the results obtained with the prototype developed for this experiment.

4 Results

The average error in the forecast was found to be 26.043,66 cell/liter and only 5.5% of the forecasts had an error higher than 100.000 cell/liter. Although the experiment was carried out using a limited data set (geographical area A0 ((42°28.90' N, 8°57.80' W) 61 m)), it is believed that these error value results are significant enough to be extrapolated along the whole coast of the Iberian Peninsula.

Two situations of special interest are those corresponding to the *false alarms* and the *blooms not detected*. The former refers to predictions of bloom (concentration of pseudo-nitzschia $\geq$ 100.000 cell/liter) which don't actually materialize (real concentration $\leq$ 100.000 cell/liter). The latter, more significant occurrence arises when a bloom exists but the model fails to detect it.

Table 2 shows the predictions carried out with success (in absolute values and %) and the erroneous predictions differentiating the not detected blooms from the false alarms.

Table 2. Summary of results using the CBR-ANN-FS Hybrid System.

OK	OK (%)	Not detect.	False alarms
191/200	95,5%	8	1

Further experiments have been carried out to compare the performance of the CBR-ANN-FS hybrid system with several other forecasting approaches. These include standard statistical forecasting algorithms and the application of several neural networks methods. The results obtained from these experiments are listed in Table 3.

Table 3 shows the number of successful predictions (in absolute value and %) as well as the blooms not detected and false alarms for each method. As it indicates, 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 and any of the other techniques studied during this investigation. This is due to the effectiveness of the revision subsystem and the re-training of the RBF neural network with the cases recovered by the GCS network. The

Table 3. Summary of results using statistical techniques.

Method	OK	OK (%)	N. detect.	Fal. alarms
RBF	185/200	92,5%	8	7
ARIMA	174/200	87%	10	16
Quadratic Trend	184/200	92%	16	0
Moving Average	181/200	90,5%	10	9
Simp. Exp. Smooth.	183/200	91,5%	8	9
Lin. Exp. Smooth.	177/200	88,5%	8	15

performance of the hybrid system is better than the other methods at each of the individual geographical monitoring points.

Table 4 shows the average error obtained with the hybrid model, a standard RBF network, an ARIMA model, a Quadratic Trend, a Moving Average, a Simple Exp. Smoothing, a Brown's Linear Exp. Smoothing and a Finite Impulse Response ANN [28], which was not able to converge for this type of problem.

Table 4. Average error in the forecast with other techniques and the CBR-ANN-FS Hybrid System.

Method	Type	Aver. error (cel/liter)
CBR-ANN-FS	Hybrid System	26.043,66
RBF	ANN	45.654,20
FIR	ANN	–
ARIMA	Statistics	71.918,15
Quadratic Trend	Statistics	70.354,35
Moving Average	Statistics	51.969,43
Simple Exp. Smoothing	Statistics	41.943,26
Brown's Linear Exp. Smoothing	Statistics	49.038,19

5 Conclusions and Future Work

In summary, this paper has presented an automated hybrid CBR system that combines a case-based reasoning system integrated with two artificial neural networks and a set of fuzzy inference systems in order to create a real time autonomous forecasting system. The model employs a case-based reasoning model that incorporates 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 system thus combines complementary properties of both connectionist and symbolic AI methods.

The developed prototype is able to produce a forecast with an acceptable degree of accuracy. The results obtained may be extrapolated to provide forecasts further ahead using the same technique, and it is believed that successful results may be obtained. However, the further ahead the forecast is made, the less accurate the forecast may be expected to be. The developed prototype can not be used in a particular geographical area if there are no stored cases from that area. Once the system is in operation and it is forecasting, a succession of cases will be generated, enabling the hybrid forecasting mechanism to evolve and to work autonomously.

In conclusion, the hybrid reasoning problem solving approach provides an effective strategy for forecasting in an environment in which the raw data is derived from different sources and it can be represented by means of a vector of numeric values. This model 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 model 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. Nakhaeizadeh, G.: Learning prediction of time series. A theoretical and empirical comparison of CBR with some other approaches. In Proceedings of First European Workshop on Case-Based Reasoning, EWCBR-93. Kaiserslautern, Germany.(1993) 65–76
2. Lendaris, G. G., and Fraser, A. M.: Visual Fitting and Extrapolation. In Weigend, A. S., and Fershenfield, N. A. (Eds.). Time Series Prediction, Forecasting the Future and Understanding the Past. Addison Wesley. (1994) 35–46
3. Faltings, B.: Probabilistic Indexing for Case-Based Prediction. In Proceedings of Case-Based Reasoning Research and Development, Second International Conference, ICCBR-97. Provindence, Rhode Island, USA. (1997) 611–622
4. Lekkas, G. P., Arouris, N. M., Viras, L. L.: Case-Based Reasoning in Environmental Monitoring Applications. Artificial Intelligence, 8, (1994) 349–376
5. Mcintyre, H. S., Achabal, D. D., Miller, C. M.: Applying Case-Based Reasoning to Forecasting Retail Sales. Journal of Retailing, 69, num. 4, (1993) 372–398
6. Stottler, R. H.: Case-Based Reasoning for Cost and Sales Prediction. AI Expert, (1994) 25–33
7. Weber-Lee, R., Barcia, R. M., and Khator, S. K.: Case-based reasoning for cash flow forecasting using fuzzy retrieval. In Proceedings of the First International Conference, ICCBR-95. Sesimbra, Portugal, (1995) 510–519
8. Fyfe C., and Corchado J. M.: Automating the construction of CBR Systems using Kernel Methods. International Journal of Intelligent Systems, 16, num. 4, (2001) 571–586

9. 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
10. Pal, S. K., Dilon, T. S., and Yeung, D. S.: Soft Computing in Case Based Reasoning. Springer Verlag: London, (2001)
11. Corchado, J. M., Lees, B.: A Hybrid Case-based Model for Forecasting. Applied Artificial Intelligence, 15, num. 2, (2001) 105–127
12. 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
13. Corchado, J. M., Aiken, J., Rees, N.: Artificial Intelligence Models for Oceanographic Forecasting. Plymouth Marine Laboratory, U.K., (2001)
14. Fritzke, B.: Growing Self-Organizing Networks-Why?. In Verleysen, M. (Ed.). European Symposium on Artificial Neural Networks, ESANN-96. Brussels, (1996) 61–72
15. Fritzke, B.: Fast learning with incremental RBF Networks. Neural Processing Letters, 1, num. 1, (1994) 2–5
16. 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)
17. Fritzke, B.: Growing Cell Structures - A Self-organizing Network for Unsupervised and Supervised Learning. Technical Report, International Computer Science Institute. Berkeley, (1993)
18. 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
19. 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
20. Setnes, M., Babuska, R., Kaymak, U., and van Nauta, H. R.: Similarity measures in Fuzzy Rule Base Simplification. IEEE Transactions on systems, Man, and Cybernetics, 28, num. 3, (1998) 376–386
21. Tomczak, M., Godfrey, J. S.: Regional Oceanographic: An Introduction. Pergamon, New York, (1994)
22. Fernández, E.: Las Mareas Rojas en las Rías Gallegas. Technical Report, Department of Ecology and Animal Biology. University of Vigo, (1998)
23. Hallegraeff, G. M.: A review of harmful algal blooms and their apparent global increase. Phycologia, 32, (1993) 79–99
24. Kamykowski, D.: The simulation of a southern California red tide using characteristics of a simultaneously-measured internal wave field. Ecol. Model., 12, (1981) 253–265
25. Watanabe, M., Harashima, A.: Interaction between motile phytoplankton and Langmuir circulation. Ecol. Model., 31, (1986) 175–183
26. Franks, P. J. S., Anderson, D. M.: Toxic phytoplankton blooms in the southwestern Gulf of Maine: testing hypotheses of physical control using historical data. Marine Biology, 112, (1992) 165–174
27. Anderson, D. M.: Toxic algal blooms and red tides: a global perspective. In Okaichi, T., Anderson, D. M., and Nemoto, T. (Eds.). RedTides: Biology, Environmental Science and Toxicology. New York: Elsevier, (1989) 11–16
28. Corchado, J. M., Fyfe, C.: Unsupervised Neural Network for Temperature Forecasting. Artificial Intelligence in Engineering, 13, num. 4, (1999) 351–357