Submitted for Publication in {APPLIED ARTIFICIAL INTELLIGENCE, AN INTERNATIONAL JOURNAL}.

Prepared with the T&F Journal Template.

# FSfRT: Forecasting System for Red Tides. An Hybrid Autonomous AI Model.

Florentino Fdez-Riverola
Dpto. de informática, University of Vigo. E.S.E.I., Campus Universitario As Lagoas s/n, 32004, Ourense, Spain. E-mail: riverola@uvigo.es

Juan M. Corchado
Dpto. de Informática y automática, University of Salamanca. Facultad de Ciencias, Plaza de la Merced s/n., 37008, Salamanca, Spain. E-mail: corchado@tejo.usal.es

Juan M. Corchado, Departamento de Informática y Automática, Facultad de Ciencias. Plaza de la Merced s/n, 37008 Salamanca. E-mail: corchado@tejo.usal.es

*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. In such a situation, it has been found that a hybrid case-based reasoning system can provide a more effective means of performing such predictions than other connectionist or symbolic techniques. The 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 at a different stage of the reasoning cycle of the case-based reasoning system to retrieve historical data, to adapt it to the present problem and to review the proposed solution. 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 experiments, in which the system operated in a real environment, are presented.*

Hybrid reasoning system, neural indexing, neural adaptation, extracting fuzzy rules, fuzzy revision, forecasting red tides.

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 the application of a novel hybrid artificial intelligence (AI) model to a forecasting problem over a complex and dynamic environment. The approach, which is discussed, is capable of producing satisfactory results in situations in which neither artificial neural networks nor statistical models have been sufficiently successful.

The oceans of the world form a highly dynamic system for which it is difficult to create mathematical models (Tomczak and Godfrey, 1994). *Red tides* are the name for the discolourations caused by dense concentrations of microscopic sea plants, known as phytoplankton. The discolouration varies with the species of phytoplankton, its pigments, size and concentration, the time of day, the angle of the sun and other factors. Red tides usually occur along the north west coast of the Iberian Peninsula in late summer and autumn (Fernández, 1998). The prevailing southerly winds cause cold, nutrient-rich water to rise up from the deeper regions of the ocean to the surface, a

process known as *upwelling*. Swept along with this upwelled water are dinoflagellate cysts, the resting stages of the organism, which lie dormant in the sediments on the sea floor. The high nutrient concentrations in the upwelled water, together with ideal conditions of temperature, salinity and light, trigger the germination of the cysts, so that the dinoflagellates begin to grow and divide. 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). Concentration of the bloom by wind and currents, as well as the dinoflagellates' ability to swim to the surface, combine to form a red tide. If conditions at the water's surface become unfavourable for the dinoflagellates, for example, if the nutrients are depleted or the bloom is dispersed by wind and currents, the dinoflagellates will again form dormant cysts and sink to the sea floor. This study focusses in the prediction of the evolution of the pseudo-nitzschia spp diatom dinoflagellate, which causes amnesic shellfish poisoning (or ASP).

An artificial intelligence 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 types of standard artificial neural networks (ANN) have been used to forecast the evolution of different oceanographic parameters (Corchado and Fyfe, 1999; Corchado *et al*., 2001a). The reported work shows how difficult it is to train neural networks to successfully forecast time series of oceanographic and/or biological parameters such as the temperature, chlorophyll and salinity of the water. Statistical models such as Auto-Regressive Integrated Moving Averages (ARIMA) have been applied, but the results obtained so far have indicated that they have less facility for forecasting such parameters than neural networks (Corchado *et al*., 2001a).

An important aim in the current work is to develop a universal forecasting mechanism, in the sense that it might operate effectively anywhere, at any point in coastal waters, and at any time of the year without human intervention. The results obtained to date suggest that the approach described in this paper appears to fulfil these aims.

Successful results have been already obtained with hybrid case-based reasoning systems (Corchado and Lees, 2001; Corchado *et al.*, 2001a; Corchado *et al.*, 2001b) 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 a brief overview of the basic concepts that characterize the case-based reasoning model is presented, paying special attention to models constructed to solve prediction problems. Then the red tide problem domain is briefly outlined, the hybrid neuro-symbolic system is explained in detail and finally, the results obtained to date with the proposed hybrid forecasting system are presented and analyzed.

**CBR SYSTEMS OVERVIEW**

Although knowledge-based systems (KBS) represent one of the commercial successes resulting from artificial intelligence research, their developers have encountered several problems (Watson, 1997). Knowledge elicitation, a necessary process in the development of rule-based systems, can be problematic. The implementation of a KBS can also be complex, and once implemented, it may also be difficult to maintain. With the aim of overcoming these problems, Schank (1982) proposed a revolutionary approach: case-based reasoning, which is, in effect, a model of human reasoning. The idea underlying CBR is that people frequently rely on previous problem-solving experiences when solving new problems. This assertion may be verified in many day to day problem-solving situations by simple observation or by psychological experimentation (Klein *et al*., 1988). Since the ideas underlying case-based reasoning were first proposed, CBR systems have been found to be successful in a wide range of application areas (Kolodner, 1993; Watson, 1997; Pal *et al.*, 2000).

A case-based reasoning system solves new problems by adapting solutions that were used to solve previous problems (Riesbeck and Schank, 1989). The case base holds a number of cases, each of which represents a problem together with its corresponding solution. Once a new problem arises, a possible solution to it is obtained by retrieving similar cases from the case base and studying their recorded solutions. A CBR system is dynamic in the sense that, in operation, cases representing new problems together with their solutions are added to the case-base, redundant cases are eliminated and others are created by combining existing cases.

<Insert Figure 1 here>

A CBR system analyses a new problem situation, and by means of indexing algorithms, retrieves previously stored cases together with their solution by matching them against the new problem situation, then adapts them to provide a solution to the new problem by reusing knowledge stored in the form of cases, in the case-base. All of these actions are self-contained and may be represented by a cyclic sequence of processes, in which human interaction may be needed. Case-based reasoning can be used by itself or as part of another intelligent or conventional computing system. Furthermore, case-based reasoning can be a particularly appropriate problem-solving strategy when the

knowledge required to formulate a rule-based model of the domain is difficult to obtain, or when the number or complexity of rules relating to the problem domain is too great for conventional knowledge acquisition methods.

A typical CBR system is composed of four sequential steps which are called into action each time a new problem is to be solved (Watson, 1997; Kolodner, 1993; Aamodt and Plaza, 1994). Figure 1 outlines the basic CBR cycle.

The purpose of the retrieval step, is to search the case-base and select one or more previous cases that most closely match the new problem situation, together with their solutions. The selected cases are reused to generate a solution appropriate to the current problem situation. This solution is revised if necessary and finally, the new case (i.e. the problem description together with the obtained solution) is stored in the case-base. Cases may be deleted if they are found to produce inaccurate solutions, they may be merged together to create more generalized solutions, and they may be modified, over time, through the experience gained in producing improved solutions. If an attempt to solve a problem fails and it is possible to identify the reason for the failure, then this information should also be stored in order to avoid the same mistake in the future. This corresponds to a common learning strategy employed in human problem-solving. Rather than creating general relationships between problem descriptors and conclusions, as is the case with rule-based reasoning, or relying on general knowledge of the problem domain, CBR systems are able to utilize the specific knowledge of previously experienced, in the form of concrete problem situations. A CBR system provides an incremental learning process because each time a problem is solved, a new experience is retained, thus making it available for future reuse.

In the CBR cycle there is normally some human interaction. Whilst case retrieval and reuse may be automated, case revision and retention are often undertaken by human experts. This is a current weakness of CBR systems and one of their major challenges. In this paper, a method for automating the CBR reasoning process is presented for the solution of problems in which the cases are characterized predominantly by numerical information.

### *CBR Systems for Forecasting*

Several researchers (Nakhaeizadeh, 1993; Lendaris and Fraser, 1994) have used *k*-nearest-neighbour algorithms for time series predictions. Although a *k*-nearest-neighbour algorithm does not, in itself, constitute a CBR system, it may be regarded as a very basic and limited form of CBR operation in numerical domains. Nakhaeizadeh (1993) uses a relatively complex hybrid CBR-ANN system. In contrast, Lendaris and Fraser (1994) forecast a data set simply by searching a given sequence of data values for segments that closely match the pattern of the last n measurements and then, 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 Faltings (1997), Lekkas *et al*. (1994), Mcintyre *et al*. (1993), Stottler (1994) and Weber-Lee *et al*. (1995).

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 Corchado *et al*. (2001a).

Traditionally, CBR systems have been combined with other technologies like artificial neural networks, rule-based systems, constraint satisfaction problems and others, producing successful results (Corchado and Lees, 2000) and (Pal *et al*., 2000), but the particularities of the problem described mean that these techniques are not the most appropriate for obtaining an accurate prediction.

**THE RED TIDES PROBLEM DOMAIN**

Recently red tides have been very much in the news. Dinoflagellates are usually regarded as the causative organisms, but not all red tides are caused by dinoflagellates and not all dinoflagellates cause red tides. Even the color factor is variable: so-called *red* tides may be brown, yellow, green, etc. Some red tides may be very extensive and several square kilometers of ocean may be affected, even to the extent that satellites have been used to track blooms. 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 and (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.

What causes such blooms?. A range of factors seem to be involved, but very little definite information is available. In some places there seems to be a strong correlation between the occurrence of upwelling (nutrient-rich waters coming in from deep water) and such blooms (Fraga *et al*., 1988). But, in other areas, the blooms have been found to

be associated with tidal turbulence or they seem to be set off by heavy rainfall on the land, the runoff washing phosphates into the sea and also lowering the salinity, all factors which seem to favour dinoflagellate growth. It is also thought that Vitamin B12, which is required by most dinoflagellates, may also be washed into the sea from the soil and salt-marsh areas, where it is produced by bacteria and blue-green algae. Humic substances have also been suggested as possible causative agents.

***Recent Trends***

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 (Hallegraeff, 1993). 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. Possible explanations include:

- Species dispersal through currents, storms, or other natural mechanisms.
- Nutrient enrichment of coastal waters by human activities, leading to a selection for, and proliferation of, harmful algae.
- Increased aquaculture operations which can enrich surrounding waters and stimulate algal growth.
- Introduction of fishery resources (through aquaculture development) which then exposes itself to the presence of indigenous harmful algae in the surrounding waters.
- Dispersal of the species via ship ballast water or shellfish seeding activities.
- Long-term climatic trends in temperature, wind speed, or insolation.
- Increased scientific and regulatory scrutiny of coastal waters and fishery products and improved chemical analytical capabilities that lead to the discovery of new toxins and toxic events (Anderson, 1989).

***Models***

Models of dinoflagellate blooms have been developed from several different perspectives. Kamykowski (1981) examined the response of a swimming dinoflagellate to internal waves and showed that accumulation of motile and non-motile cells may occur due to an internal wave field, with the accumulation of vertically migrating cells being

most significant. These models consider only the physics of the wave field and the swimming behavior of the phytoplankton, without regard to the phytoplankton response to nutrients or light. Others have examined the response of phytoplankton to the flow field of Langmuir cells (Watanabe and Harashima, 1986) or to 2-dimensional, cross-frontal circulation (Franks and Anderson, 1992), to name just two of many physical systems that have been studied in this theoretical context. The growth and accumulation of individual harmful algal species in a mixed planktonic assemblage are exceedingly complex processes involving an array of chemical, physical, and biological interactions. Our level of knowledge about each of the many species varies significantly, and even those most widely studied remain poorly characterized with respect to bloom or population dynamics. Resolution of various rate processes integral to the population dynamics (i.e., input and losses due to growth, grazing, encystment, and physical advection) has not been accomplished, but is fundamental to the long-term management of fishery resources or marine habitats affected by harmful algae. Many of the processes are difficult to quantify in the field because harmful species often represent only a small fraction of the biomass in natural samples. 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 (Anderson, 1995).

**FORECASTING RED TIDES**

In the current work, the aim is to develop a system for forecasting one week in advance and at different geographical points the concentrations (in cells per liter) of the pseudo-nitzschia spp dinoflagellate, the diatom that produces the most harmful red tides. The approach builds on the methods and successful projects previously developed (Corchado and Fyfe, 1999; Corchado and Lees, 2001; Corchado *et al*., 2001a).

The problem of forecasting red tides, 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 additional data derived from satellite images, which is received and processed daily, and other data belonging to ocean buoys that record data on a daily basis.

***Looking into Pseudo-Nitzschia spp***

With the purpose of determining the best methods to carry out an accurate prediction of the variable related with the problem, it has been run several statistical tests. A summary of the obtained results is commented below.

Table 1 shows summary statistics for the variable pseudo-nitzschia spp time series from January 1992 to January 2001. It includes measures of central tendency, variability and shape. Of particular interest here are the standardized skewness and standardized kurtosis, which can be used to determine whether the sample comes from a normal distribution. Values of these statistics outside the range [-2, 2] indicate significant departures from normality, which would tend to invalidate any statistical test regarding the standard deviation. As it can be seen from Table 1, the standardized skewness and the standardized kurtosis values are not within the range expected for data from a normal distribution.

<Insert Table 1 here>

In order to determine if the variable is a random sequence of numbers, three test have been run. A time series of random numbers is often called white noise, since it contains equal contributions at many frequencies. The first test counts the number of times the sequence was above or below the median, the second counts the number of times the sequence rose or fell, while the third is based on the sum of squares of the first 24 autocorrelation coefficients. Since the P-value for the three tests was less than 0,01, it can be rejected the hypothesis that the series is random at the 99% confidence level.

<Insert Figure 2 here>

Figure 2 shows the horizontal time sequence plot of the variable during the whole period of monitoring. At first sight, the graph does not show tendency indications, for what we can suppose that the series presents the horizontal pattern. This fact is reinforced by the form of the autocorrelation function (not shown for abbreviate) that shows a strong fall in the autocorrelation coefficient for the first 4 periods, and which continues with a falling sequence of values as the number of considered retards increases. Figure 3 shows a zoom on the same time series representing only the year 1998 (one year with many blooms). Starting from the two graphs, it can also be appreciated that the series is seasonal, presenting sporadic picks that represent the bloom situations.

<Insert Figure 3 here>

The analysis of crosscorrelation function over other variables does not show any outstanding coefficient, what indicates that does not exist any variable that determines the behavior of the pseudo-nitzschia spp directly.

The main problem with the data is that the sampling interval discrete, and carried out only once a week. This interval may be insufficient to reflect the dynamics of the variable correctly, what presupposes that the standard statistical techniques will not generate accurate results.

***The Hybrid Forecasting System***

In order to forecast the concentration of pseudo-nitzschia spp at a given point a week in advance, a problem descriptor is generated on a weekly basis. A problem descriptor consists of a sequence of *N* sampled data values (filtered and pre-processed) recorded from the water mass to which the forecast will be applied. The problem descriptor also contains various other numerical values, including the current geographical location of the sensor buoys and the collection time and date. Every week, the concentration of pseudo-nitzschia spp is added to a problem descriptor forming a new problem case. In this particular problem a case is composed of a vector with the variables that characterize the problem recorded over two weeks and the concentration of pseudo-nitzschia spp one week later, as indicated in Table 2.

The forecast values are obtained using a neural network and fuzzy model enhanced hybrid case-based reasoning system. Figure 4 illustrates the relationships between the processes and components of the 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.

The retrieval stage is carried out using a Growing Cell Structures (GCS) ANN. The GCS facilitates the indexation of cases and the selection of those that are most similar to the problem descriptor. The reuse of cases is carried out with a Radial Basis Function (RBF) ANN, which generates an initial solution creating a 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 concentration of pseudo-nitzschia spp is measured and the error value is calculated, updating the knowledge structure of the whole system. The cycle of operations of the hybrid system is explained in the following subsection in detail.

<Insert Figure 4 here>

The neural networks and fuzzy systems used in the current study are: a) Growing Cell Structures (GCS) neural network (Fritzke, 1996) which are a variation of Kohonen's Self-Organising Maps and provide the basis for powerful information retrieval applications and similarity visualization tools, b) the Radial Basis Function (RBF)

neural network (Fritzke, 1994), in which the input layer is a receptor for the input data, whilst the hidden layer performs a non-linear transformation from the input space to the hidden layer space, and c) Sugeno-Takagi fuzzy model (TSK) (Takagi and Sugeno, 1985), which is generated from the trained RBF network and provides the basis for the creation of a fuzzy rule-based revision subsystem.

***System Operation***

The forecasting system uses data from two main sources: (i) the data (coming from the buoys and monitoring net) used to create a succession of problem descriptors, characterizing the current forecasting situation and (ii) 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. Some of the cases included in the case memory represents prototypes of red tides. The experts have recommended not to use artificial cases due to the high variability of the problem and the lack of knowledge about it. Table 2 shows the variables that characterize the problem. Data from the previous 2 weeks ($_{Wn-1}$, $W_n$) is used to forecast the concentration of pseudo-nitzschia spp one week ahead ($W_{n+1}$).

<Insert Table 2 here>

Several experiments have been carried out over a testing data set in order to identify the optimum number of weeks for constructing a case. Table 3 shows a summary of the results using the hybrid system to predict the concentration of pseudo-nitzschia spp a week ahead. Each row shows the results obtained when forecasting the concentration of pseudo-nitzschia spp using data of the last 1, 2 and 3 weeks to construct the cases. The best results were obtained using data of 2 weeks previous ($W_{n-1}$, $W_n$).

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 ≥ 100.000 cell/liter) which don't actually materialize (real concentration ≤ 100.000 cell/liter). The latter, more significant occurrence arises when a bloom exists but the model fails to detect it. Another unwelcome situation occurs when the number of predictions exceeds an absolute error of 100.000 cell/liter (labelled as incorrect predictions).

<Insert Table 3 here>

The cycle of forecasting operations (which is repeated every week) proceeds as follows:

First a new problem instance is created from the pre-processed data cited above. When a new problem is presented to the system (Figure 4), the GCS neural network is used to obtain the *k* most similar cases to the given problem (identifying the class to which the problem belongs, see Table 4).

<Insert Table 4 here>

In the reuse phase (Figure 4), the values of the weights and centers of the 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 of the concentration of pseudo-nitzschia spp (see Table 4). During this process the values of the parameters that characterize the network are updated.

The revised forecast is then retained temporarily in the forecast database. When the real value of the concentration of pseudo-nitzschia spp 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 Table 4). 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 of the revision subsystem.

***Growing Cell Structures Operation***

To illustrate the working model of the GCS network for this problem inside the whole system, a two-dimensional space is used, where the cells (neurons) are connected and organized into triangles (Fritzke, 1996). Each cell in the network is associated with a weight vector, *w*, of the same dimension that the problem descriptors stored in the case-base. At the beginning of the learning process, the weight vector of each cell is initialized with random values (Fritzke, 1993). 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 4.

In the first step of each learning cycle, the cell *c*, with the smallest distance between its weight vector, $w_c$, and the actual input vector, $v_x$, is chosen as the *winner cell* or best-match cell. The second step consists in the adaptation of the weight vector of the winning 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. Growing cell structures also modify the overall network structure by inserting new cells into those regions that represent large portions of the input data, or removing cells that do not contribute to the input data representation. In our experiment a new center is inserted into the network after *n* iterations (where *n* is calculated dividing the maximum number of iterations by the maximum number of cells required).

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 representing a class that can be seen as a prototype that identifies a set of similar problem descriptors. The maximum number of cases associated to a cell should be smaller than 750. This number has been empirically identified. When a cell is associated to a high number of cases a new cell is inserted near it and the cases are distributed among them. Since the retrieved cases are used to retrain an RBF neural network, during the reuse stage, and they are the cases associated to the winning cell, we have empirically proved that using more than 750 cases to retrain the RBF neural network do not improve the accuracy of the forecast.

For each class of the GCS neural network a vector of four values is maintained (labeled as *Expert's score* in Figure 4). This vector (to which we will refer as "importance" vector) is initialized with a value of (0.25, 0.25, 0.25, 0.25) and represents the accuracy of each fuzzy system (used during the revision stage) with respect to that class. The sum of the four values of the vector should be. 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 four fuzzy systems. For each forecasting cycle, the value of the importance vector associated with the most accurate fuzzy system is increased and the other three values are proportionally decreased. This is done in order to give more relevance to the most accurate fuzzy system of the revision subsystem.

The automatic discovery of relevance knowledge from cases is particularly useful in CBR schemes based on the *k*-NN algorithm (Dasarathy, 1991), where *k* is typically constant. The proposed GCS scheme does not rely on a predefined, fixed *k*, rather the set of retrieved cases depends on the groupings of cognate cases in the case-base. Thus, this stage could be thought of as a *dynamic k-nearest neighbour* scheme. Figure 5 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.

<Insert Figure 5 here>

The neural network topology of a GCS network is incrementally constructed on the basis of the training data presented to the network. Effectively, such a topology represents the result of the basic clustering procedure (see Figure 5). Such a topology 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 an *indexing*

*structure* which indexes sets of cases in the case-base and a *similarity measure* between case sets (Azuaje *et al.*, 2000).

When dealing with complex, dynamic, stochastic problems that can be numerically represented (as it is the case of the red tides problem), the decision of what retrieval model is better to use for a CBR that need to be completely automated is not a trivial task. GCS neural networks have demonstrated their utility for powerful information retrieval applications and similarity visualization tools (Köhle and Merkl, 1996; Zavrel, 1996). 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 known 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-organizing 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 knowledge representation model.

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.

***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 (Corchado and Lees, 2000). Several hybrid systems have been developed in which CBR components co-operate with one or more reasoning elements (Fdez-Riverola and Corchado, 2000). Most adaptation techniques are based on generalization and refinement heuristics. This subsection proposes an approach based on the radial basis function neural network and their ability to generalize.

The RBF network used in the framework of this experiment, uses 18 input neurons (see Table 2), between three and fifty neurons in the hidden layer and a single neuron in the output layer. When a problem is presented to the network, the output is the concentration of pseudo-nitzschia spp for a given water mass. 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 Fritzke (1994) 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 (¼ of the total of established iterations for the period of training) in order to obtain the best possible values for the weighted connections and the positions of the centers.

A new center is inserted into the network when the average error in the training data set does not fall by more than 15% after *n* iterations (where n is calculated dividing the value that corresponds to the ¾ parts of the total of iterations by the maximum number of centers of the hidden layer, 50). To calculate the place where the new center

will be inserted, the center *C*, with the greatest accumulated error is selected. A new center is then inserted near *C* with an average of the input data vectors of the two near centers.

As it has been mentioned previously, the set of *k* more similar cases retrieved by the GCS network are used to adapt the configuration of the RBF network to the actual problem. Figure 6 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.

<Insert Figure 6 here>

Radial basis function networks have been employed in many different problems. In the literature, the number of applications covered by type of neural network is quite high and can be seen that pattern recognition and time-series analysis are the main fields of interest (Shin, 1996; He and Lapedes, 1991; Kadirkamanathan *et al*., 1991). 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 similar problems to the exposed one. 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 *k* most similar cases retrieved by the GCS network.

RBF networks can also be used to generate Fuzzy inference systems (Jin *et al*., 2000). This characteristic has been used in this model for the automatic generation of the revision subsystem as it will be explained in the following subsection.

***Fuzzy System Operation***

After the case adaptation stage, a crisp value is obtained for the forecasted concentration of pseudo-nitzschia spp. This value is rarely 100% accurate, therefore revision is required to obtain a more realistic output. With this purpose, a set of Sugeno fuzzy models is generated starting from the RBF neural network.

Rule extraction from artificial neural networks is considered to be important due to the following reasons (Jin, 2000):

- 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 network and, to provide a set of simplified fuzzy rule systems that may improve the knowledge that the user has over the problem domain. The construction of the revision subsystem, that allows to achieve both goals, is carried out in two main steps:

1. First, a TSK fuzzy model (Takagi and Sugeno, 1985) is generated using the trained RBF network configuration (centers and weights). In order to transform a RBF neural network to an equivalent well-interpretable fuzzy rule system, the following conditions should be satisfied (Jin *et al*., 2000):

- 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.

2. A measure of similarity is applied to the TSK fuzzy model rule base with the purpose of reducing the number of fuzzy sets describing each variable. Similar fuzzy sets for one oceanographic 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. Figure 7 shows how the fuzzy set generalization is carried out given a variable (i.e. temperature), where S represents the similarity measure used to find similar subsets that can be joined in order to simplify the rule base.

<Insert Figure 7 here>

The theoretical analysis of similarity has been dominated by geometric models. These models represent fuzzy sets as points in a metric space and the similarity between the sets is regarded as an inverse of their distance in this metric space. Denoting the distance between *A* and *B* as *d(A,B)*, the similarity of *A* and *B* can be written as: $S(A,B) = \frac{1}{1+d(A,B)}$, where *A* and *B* are fuzzy sets. The distance between two fuzzy sets is measured calculating the euclidean distance between the Gaussian functions that define each fuzzy set. In our model, four fuzzy systems have been created, starting from the TSK fuzzy model (with no generalization at all), with different thresholds for the value of similarity in order to carry out the revision of the initial prediction (see Figure 7). When similar fuzzy sets are replaced by a common fuzzy one, 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 importance vector as said before. 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 associated to the GCS class to which the problem belongs. The importance value of the fuzzy set that best suits a particular class is increased and the other three are proportionally decreased. This process in the adaptation of the importance vector is carried out because it is difficult to ascertain in advance the optimum level of generalization for a given data set.

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.

Figure 8 provides a more concise description of the Fuzzy-based case revision regime described above, where vx is the value feature vector describing a new problem, P is the importance vector for the identified prototype, confRS represents the configuration of the four fuzzy systems, $f_i$ is the initial forecast generated by the RBF, $f_f$ is the final forecast generated by the revision subsystem and R represents the set of fuzzy rules used during the revision stage.

<Insert Figure 8 here>

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.

***Retain***

Since this is a real time problem, it is not possible to evaluate the outcome of the system before it is used, but when the real value of the concentration of pseudo-nitzschia spp is known, a new case containing the problem descriptor and the solution is stored in the case base. Whenever this happens, the subsystems that compose the hybrid system update their configuration parameters as follows. The structure of the CGS network is modified assigning the new case to an already existing cell (updating the weight vector and the signal counter) and, each certain number of iterations, a new cell is created and one or more of the existing empty cells are deleted as previously mentioned.

The importance vector associated with the retrieved class is modified 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 it. As the sum of the four importance values associated to a class has to be one, the four values are normalized dividing each one by the sum of all of them.

When the new case is added to the case base, it is presented to the RBF network that carries out a learning cycle updating its parameters for future use.

**RESULTS**

The hybrid forecasting system 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 monitoring network of the CCCMM consists in 35 oceanographic buoys situated along the coastal waters at different geographical points. In order to prepare the data for the experiment, a case-base was set up with data belonging to 15 monitoring stations containing approximately 6.300 cases. 85% of the cases were included in the case base and 15% of them were presented to the system for evaluating its performance. 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.

The concentration of pseudo-nitzschia spp is not uniform, 87% of the cases do not present red tide.

The average error in the forecast was 38.606,4 cell/liter and only 3,6% 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 ((42º28.90' N, 8º57.80' W) 61 m)), it is believed that these error value results are significant enough to be extrapolated along the whole coast of the North-west of the Iberian Peninsula.

Table 5 shows the percentage of the predictions carried out successfully on the testing data set and the percentage of erroneous predictions differentiating the not detected blooms from the false alarms.

<Insert Table 5 here>

As the results from the experiments indicate, the combination of different techniques in the form of the hybrid CBR system previously presented, produces better results that a RBF neural network alone. 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.

Table 6 shows the same information as the table above but with a RBF neural network. The best results were obtained with a configuration of 50 neurons in the hidden layer, maintaining the input layer (with 18 neurons) and output layer (with 1 neuron) constant.

<Insert Table 6 here>

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 Tables 7 and 8.

Table 7 shows the percentage of successful predictions together with the percentage of blooms not detected and false alarms obtained with a hybrid RBF-GCS neural network for the same data set. The GCS ANN works as a filter in order to select the most similar cases given a problem. These cases are used to adapt the configuration of the RBF ANN before it provides the solution.

<Insert Table 7 here>

Table 8 shows the same information as Table 7 for several statistical methods. The hybrid system is more accurate than any of the other techniques studied during this investigation. The performance of the hybrid system is better than the other methods at each of the individual geographical monitoring points tested.

<Insert Table 8 here>

From Tables 5 to 10 it can be seen that the hybrid CBR-ANN-FS model detects most of the red tides situations, generating a low value for the false alarm rate and that improves the results obtained with other models.

Table 9 shows the percentage of predictions with an absolute error greater than 100.000 cell/liter. As it clearly shows, the hybrid system once again, provides the best results.

<Insert Table 9 here>

Table 10 shows the average error obtained with the hybrid model, a standard RBF network, a hybrid RBF-GCS 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 (Corchado and Fyfe, 1999), which was not able to converge for this type of problem.

<Insert Table 10 here>

Starting from the error series generated by the different models, the Kruskall-Wallis test has been carried out. Since the P-value is less than 0,01, there is a statistically significant difference among the models at the 99,0% confidence level. Table 11 shows a multiple comparison procedure (Mann-Withney test) used to determine which models are significantly different from the others.

<Insert Table 11 here>

The asterisk indicates that these pairs show statistically significant differences at the 99.0% confidence level. It can be seen in Table 11, that the CBR-ANN-FS system presents statistically significant differences with the rest of the

models. The proposed model generates the best results with respect to the correct predictions, not detected blooms, false alarms and incorrect predictions (those with an absolute error value greater than 100.000 cell/liter).

## CONCLUSIONS AND FUTURE WORK

In summary, this paper has presented a problem-solving method that combines a case-based reasoning system integrated with two artificial neural networks and a set of fuzzy systems in order to create a real time autonomous forecasting system. The forecasting system is able to produce a forecast with an acceptable degree of accuracy. Although the accuracy of the forecast depends, to a great extent, on the quality of the cases and the geographical monitoring point, it is believed that good quality forecasts may be obtained even with data collected several years before and belonging to other geographical points.

The method 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 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 system cannot 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 work autonomously.

In conclusion, the hybrid reasoning problem solving approach provides an effective strategy for forecasting in complex environments like the red tides one. 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

Aamodt, A., and Plaza, E. 1994. Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches. *AICOM*, 7(1):39-59.

Anderson, D. M. 1995. Toxic red tides and harmful algal blooms: A practical challenge in coastal oceanography. *Reviews of Geography* (supplement), 1189-1200.

Azuaje, F., Dubitzky, W., Black, N., and Adamson, K. 2000. Discovering Relevance Knowledge in Data: A Growing Cell Structures Approach. *IEEE Transactions on Systems, Man and Cybernetics*, B, 30(3):448-460.

Corchado, J. M., Aiken, J., and Rees, N. 2001a. *Artificial Intelligence Models for Oceanographic Forecasting*. Plymouth Marine Laboratory, U.K. ISBN: 0-951-9618-4-5.

Corchado, J. M., and Fyfe, C. 1999. Unsupervised Neural Network for Temperature Forecasting. *Artificial Intelligence in Engineering*, 13(4):351-357.

Corchado, J. M., and Lees, B. 2000. 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*, (pp. 293-319), London: Springer Verlag.

Corchado, J. M., and Lees, B. 2001. A Hybrid Case-based Model for Forecasting. *Applied Artificial Intelligence*, 15(2):105-127.

Corchado, J. M., Lees, B., and Aiken, J. 2001b. Hybrid Instance-based System for Predicting Ocean Temperatures. *International Journal of Computational Intelligence and Applications*, 1(1):35-52.

Dasarathy, V. (Ed.). 1991. *Nearest Neighbor (NN) Norms NN pattern Classification Techniques*. Los Alamitos, CA: IEEE Computer Society Press.

Faltings, B. 1997. Probabilistic Indexing for Case-Based Prediction. In *Proceedings of Case-Based Reasoning Research and Development, Second International Conference*, ICCBR-97, (pp. 611-622), Providence, Rhode Island, USA.

Fernández, E. 1998. *Las Mareas Rojas en las Rías Gallegas*. Technical Report, Department of Ecology and Animal Biology, University of Vigo.

Fdez-Riverola, F., and Corchado, J. M. 2000. Sistemas Híbridos Neuro-Simbólicos: Una revisión. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 11:12-26.

Fraga, S., Anderson, D. M., Bravo, I., Reguera, B., Steidinger, K. A., and Yetsch. C. M. 1988. *Influence of Upwelling Relaxation on Dinoflagellates and Shellfish Toxity in Ria de Vigo, Spain. Estuarine, Coastal and Shelf Science*, WHOI-R-88-011, 27:349-361.

Franks, P. J. S., and Anderson, D. M. 1992. Toxic phytoplankton blooms in the southwestern Gulf of Maine: testing hypotheses of physical control using historical data. *Marine Biology*, 112:165-174.

Fritzke, B. 1993. *Growing Cell Structures - A Self-organizing Network for Unsupervised and Supervised Learning*. Technical Report, International Computer Science Institute, Berkeley.

Fritzke, B. 1994. Fast Learning with Incremental RBF Networks. *Neural Processing Letters*, 1(1):2-5.

Fritzke, B. 1996. Growing Self-Organizing Networks-Why?. In Verleysen, M. (Ed.). *European Symposium on Artificial Neural Networks*, ESANN-96, (pág. 61-72). Brussels.

Hallegraeff, G. M. 1993. A review of harmful algal blooms and their apparent global increase. *Phycologia*, 32:79-99.

He. X. and Lapedes, A. 1991. *Nonlinear Modeling and Prediction by Successive Approximation Using Radial Basis Functions*. Technical Report LA-UR-91-1375. Los Alamos National Laboratory, Los Alamos, NM.

Jin, Y. 2000. Fuzzy Modelling of High-Dimensional Systems: Complexity Reduction and Interpretability Improvement. *IEEE Transactions on Fuzzy Systems*, 8(2):212-221.

Jin, Y., Seelen, W. von., and Sendhoff, B. 2000. *Extracting Interpretable Fuzzy Rules from RBF Neural Networks*. Internal Report IRINI 00-02, Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany.

Kadirkamanathan, V., Niranjan, M., and Fallside, F. 1991. Sequential Adaptization of Radial Basis Function Neural Networks. In *Advances in Neural Information Processing Systems* 3:721-727, San Mateo, CA: MorganKaufmann.

Kamykowski, D. 1981. The simulation of a southern California red tide using characteristics of a simultaneously-measured internal wave field. *Ecol. Model*. 12:253-265.

Klein, G., Whitaker, L., and King, J. A. 1988. Using analogues to predict and plan. In *Proceedings of the DARPA Case-Based Reasoning Workshop*, (pp. 224-232).

Kolodner, J. 1993. *Case-based Reasoning*. Morgan Kaufmann: San Mateo, CA.

Köhle, M., and Merkl, D. 1996. Visualizing similarities in high dimensional input spaces with a growing and splitting neural network. In *Proceedings of International Conference of Artificial Neural Networks*, ICANN-96, (pág. 581-586), Bochum, Germany.

Lekkas, G. P., Arouris, N. M., and Viras, L. L. 1994. Case-Based Reasoning in Environmental Monitoring Applications. *Artificial Intelligence*, 8:349-376.

Lendaris, G. G., and Fraser, A. M. 1994. Visual Fitting and Extrapolation. In Weigend, A. S., and Fershenfield, N. A. (Eds.). *Time Series Prediction, Forecasting the Future and Understanding the Past*, (pp. 35-46), Addison Wesley.

Mcintyre, H. S., Achabal, D. D., and Miller, C. M. 1993. Applying Case-Based Reasoning to Forecasting Retail Sales. *Journal of Retailing* 69(4):372-398.

Nakhaeizadeh, G. 1993. 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, (pág. 65-76), Kaiserslautern, Germany.

Pal, S. K., Dilon, T. S., and Yeung, D. S. 2000. *Soft Computing in Case Based Reasoning*. Springer Verlag: London.

Riesbeck, C. K., and Schank, R. C. 1989. *Inside Case-based Reasoning*. Lawrence Elrlbaum Ass: Hillsdale.

Schank, R. C. 1982. *Dynamic Memory*. Cambridge University Press: Cambridge, UK.

Shin, C. 1996. Radial Basis Function Network Design for Chaotic Time Series Prediction, *Transactions of the Korean Institute of Electrical Engineers*, 45(4):602-611.

Stottler, R. H. 1994. Case-Based Reasoning for Cost and Sales Prediction. *AI Expert*, 25-33.

Takagi, T., and Sugeno, M. 1985. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15:116-132.

Tomczak, M., and Godfrey, J. S. 1994. *Regional Oceanographic: An Introduction.* Pergamon: New York.

Watanabe, M., and Harashima, A. 1986. Interaction between motile phytoplankton and Langmuir circulation. *Ecol. Model*. 31:175-183.

Watson, I. 1997. *Applying Case-based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann: San Mateo, CA.

Weber-Lee, R., Barcia, R. M., and Khator, S. K. 1995. Case-based reasoning for cash flow forecasting using fuzzy retrieval. In *Proceedings of the First International Conference*, ICCBR-95, (pp. 510-519), Sesimbra, Portugal.

Zavrel, J. 1996. Neural navigation interfaces for information retrieval: are they more than an appealing idea?. *Artificial Intelligence Review*, 10:477-504.

**TABLE 1**. Summary of statistics for pseudo-nitzschia spp diatom.

| | | | |
|---|---|---|---|
| Count | 441 | Minimum | 0,0 |
| Average | 48.035,3 | Maximum | 1.376 E6 |
| Median | 5.220,0 | Skewness | 6,2488 |
| Mode | 0,0 | **Stnd. Skewness** | **53,5723** |
| Variance | 1.62138 E10 | Kurtosis | 50,2259 |
| Standard deviation | 127.333,0 | **Stnd. Kurtosis** | **215,299** |
| Range | 1.376 E6 | Coeff. of variation | 265,083% |

**TABLE 2**. 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 | cell/liter | $W_{n-1}$, $W_n$ |
| Pseudo-nitzschia spp | cell/liter | $W_{n-1}$, $W_n$ |
| *Pseudo-nitzschia spp (future)* | *cell/liter* | $W_{n+1}$ |

**TABLE 3**. Summary of results using a RFB with information coming from several weeks.

| **Weeks** | **MAE** | **Incorrect predictions** | **Not detected blooms** | **False alarms** |
|---|---|---|---|---|
| 1 | 36553,06 | 15 | 8 | 4 |
| **2** | **32573,88** | **9** | **5** | **7** |
| 3 | 46798,66 | 15 | 5 | 10 |

**TABLE 4**. Summary of technologies employed by the Hybrid System.

| CBR-STAGE | Technology | Input | Output | Process |
|---|---|---|---|---|
| **Retrieval** | GCS network. | Problem descriptor. | *k* similar cases.<br>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.<br>*k* similar cases. | Initial solution: concentration of pseudo-nitzschia spp. | The RBF network is retrained with the *k* retrieved cases. |
| **Revision** | 4 Fuzzy systems. | Problem descriptor.<br>Expert's score.<br>Initial solution. | Confirmed solution: concentration of pseudo-nitzschia spp. | Four fuzzy systems are created using the RBF network configuration with different degrees of generalization. |
| **Retain** | GCS network.<br>RBF network.<br>4 Fuzzy systems. | Problem descriptor.<br>Forecasting error. | Configuration parameters of the GCS network,<br>RBF network<br>and 4 Fuzzy systems. | The configurations of the GCS network, the RBF network and the Fuzzy subsystems are updated according to the accuracy of the forecast. |

**TABLE 5**. Summary of results using the CBR-ANN-FS Hybrid System.

| OK | Blooms not detected | False alarms |
|---|---|---|
| 98,0% | 1,8% | 0,2% |

**TABLE 6**. Summary of results using a RBF neural network.

| OK | Blooms not detected | False alarms |
|---|---|---|
| 89,1% | 5,8% | 5,1% |

**TABLE 7**. Summary of results using a hybrid RBF-GCS neural network.

| OK | Blooms not detected | False alarms |
|---|---|---|
| 90,6% | 2,9% | 6,5% |

**TABLE 8**. Summary of results using statistical techniques.

| Method | OK | Blooms not detected | False alarms |
|---|---|---|---|
| ARIMA | 81,7% | 5,5% | 12,8% |
| Quadratic Trend | 88,2% | 11,8% | 0,0% |
| Moving Average | 87,5% | 5,9% | 6,6% |
| Simple Exp. Smoothing | 86,2% | 5,7% | 8,1% |
| Brown's Lin. Exp. Smooth. | 85,8% | 5,7% | 8,8% |

**TABLE 9**. Number of predictions with an error ≥ 100.000 cell/liter.

| Method | Incorrect Predictions |
|---|---|
| CBR-ANN-FS | 3,6% |
| RBF | 9,8% |
| RBF-GCS | 8,3% |
| ARIMA | 19,3% |
| Quadratic Trend | 7,9% |
| Moving Average | 11,6% |
| Simple Exp. Smoothing | 12,9% |
| Brown's Lin. Exp. Smooth. | 14,0% |

**TABLE 10**. Average error in the forecast with other techniques and the CBR-ANN-FS Hybrid System.

| Method | Type | Average error (cell/liter) |
|---|---|---|
| CBR-ANN-FS | Hybrid System | 38.606,4 |
| RBF | ANN | 40.973,8 |
| RBF-GCS | Hybrid System | 33.260,0 |
| FIR | ANN | - |
| ARIMA | Statistics | 76.725,0 |
| Quadratic Trend | Statistics | 62.489,9 |
| Moving Average | Statistics | 45.365,1 |
| Simple Exp. Smoothing | Statistics | 47.141,3 |
| Brown's Lin. Exp. Smooth. | Statistics | 52.159,8 |

**TABLE 11**. Multiple comparison procedure among the models.

| | CBR-ANN-FS | RBF | RBF + EAN | ARIMA | Quadratic Trend. | Moving Average | Simp. Exp. Smooth. | Br. Lin. Exp. Smo. |
|---|---|---|---|---|---|---|---|---|
| CBR-ANN-FS | | | | | | | | |
| RBF | * | | | | | | | |
| RBF+EAN | * | = | | | | | | |
| ARIMA | * | * | * | | | | | |
| Quadratic Trend | * | * | * | * | | | | |
| Moving Average | * | * | * | * | * | | | |
| Simp. Exp. Smo. | * | = | = | * | * | = | | |
| Br. Lin. Exp. Smo. | * | = | = | * | * | = | = | |

**FIGURE 1**. The classic CBR cycle.

**FIGURE 2**. Time sequence plot for pseudo-nitzschia spp from the year 1992 to 2001.

**FIGURE 3**. Time sequence plot for pseudo-nitzschia spp during the year 1998.

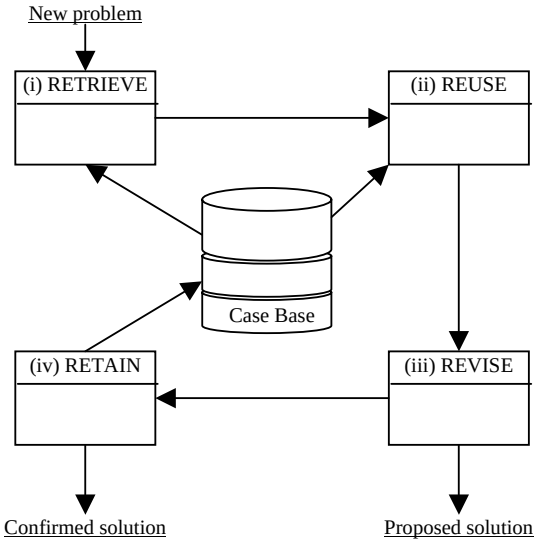
**FIGURE 4**. Hybrid neuro-symbolic system.

**FIGURE 5**. GCS-based case retrieval.

**FIGURE 6**. RBF-based case adaptation.

**FIGURE 7**. Different levels of generalization in a fuzzy set.

**FIGURE 8**. Fuzzy-based case revision.

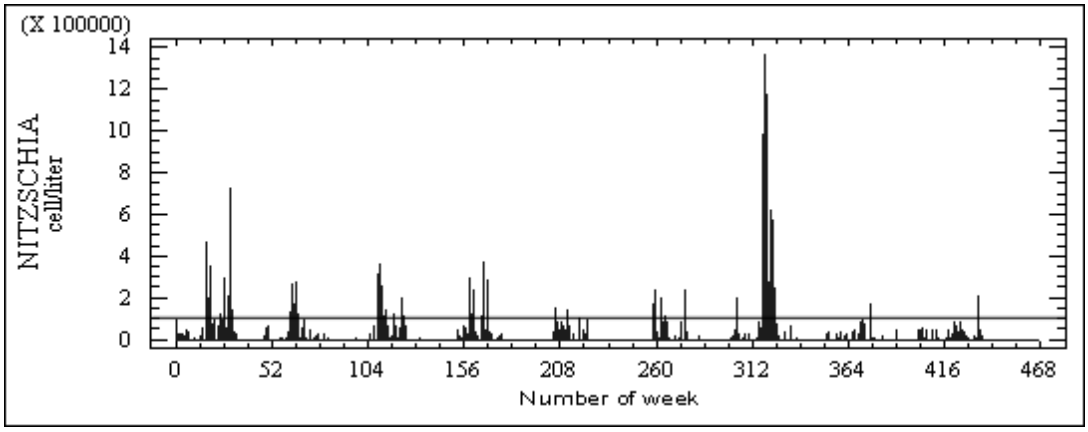


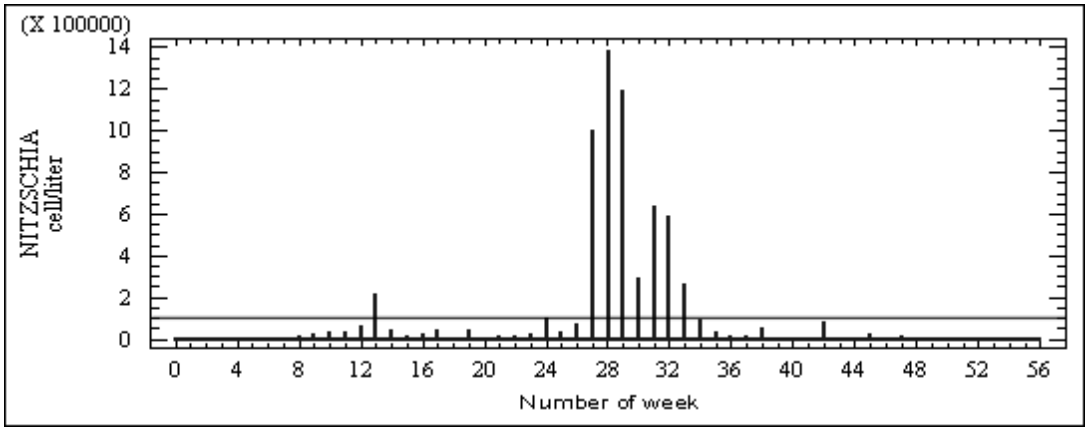

(i) ***Retrieve*** the most relevant case(s).

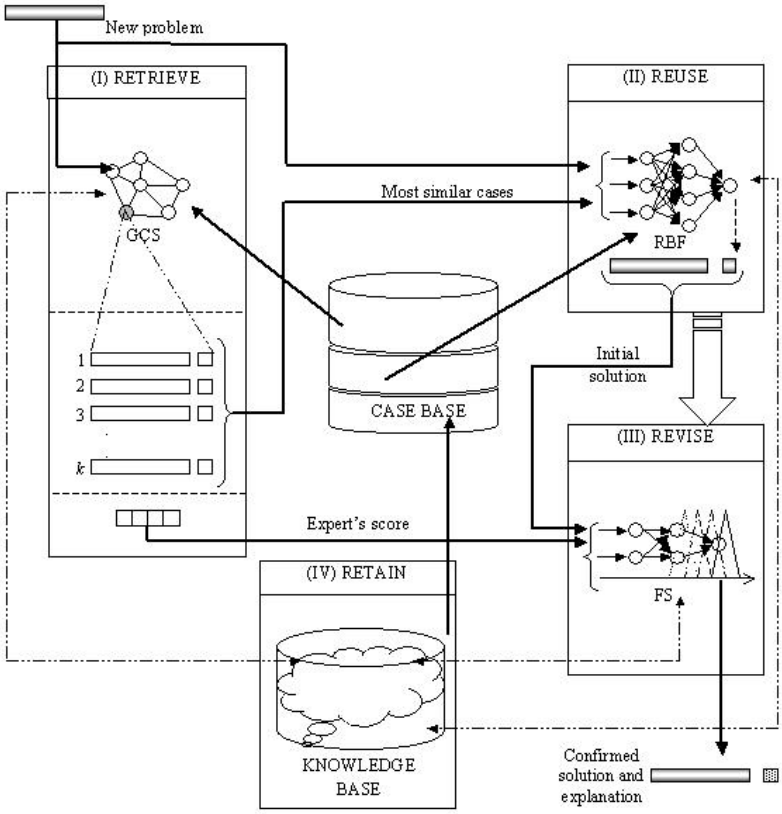
(ii) ***Reuse*** the case(s) to attempt to resolve the problem.

(iii) ***Revise*** the proposed solution if necessary.

(iv) ***Retain*** the new solution as a part of a new case.

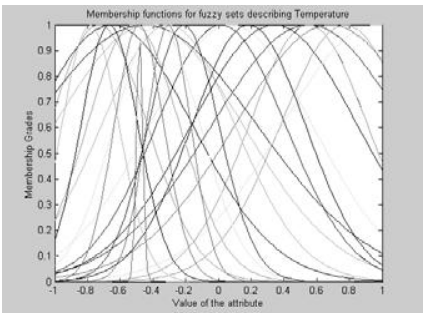


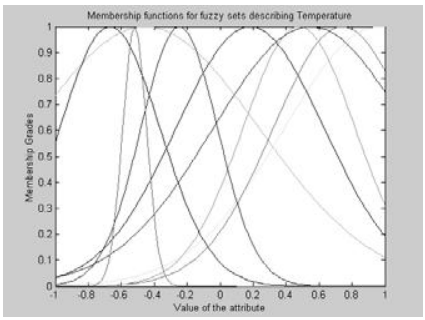



```
    procedure RETRIEVE (input: v_x, confGCS; output: K, P)
    {
00      begin.
01          CD ← ∅ /* vector of pairs (cell, distance) */
02          for each cell c ∈ confGCS do
03                      compute_distance: dc ← DIS(v_x, w_c)
04                      assign_cell-distance-pair: CD ← (c, 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.
    }
```

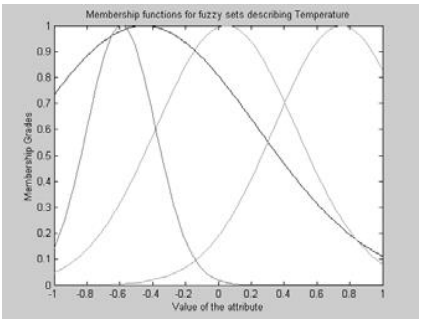
```
    procedure REUSE (input: v_x, K, confRBF; output: f_i)
    {
00      begin.
01          while TRUE do /* infinite loop */
02                  for each case c ∈ K do /* network adaptation using K cases */
03                          retrain_network: error ← annRBF(c)
04                          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.
    }
```



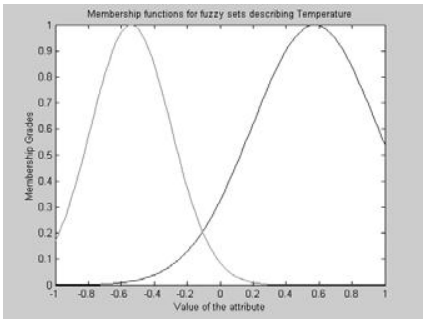

a. $S = 0.9$




b. $S = 0.8$




c. $S = 0.7$




d. $S = 0.6$

```
   procedure REVISE (input: v_x, P, confRS, f_f; output: f_f, R)
   {
00    begin.
01        SOL ← ∅ /* vector of pairs (solution, importance) */
02        for each fuzzy system  fs ∈ confRS do /* each one generate a different prediction */
03                generate_solution: inicForecast ← fs(v_x)
04                assign_solution-importance_pair: SOL ← (inicForecast, P_fs)
05        calculate_final_solution: finalForecast ← ponder(SOL) /* each forecast is pondered by its importance */
06        if (inicForecast * 100) / finalForecast <= lower_limit then /* if predictions don't differ in more than 10% */
07                f_f ← inicForecast /* the initial prediction is supported */
08        if (inicForecast * 100) / finalForecast >= upper_limit then /* if predictions differ in more than 30% */
09                f_f ← NULL /* the system is not able to respond */
10        if not
11                f_f ← (inicForecast + finalForecast) / 2 /* the mean of the two predictions is calculated */
12    end.
   }
```