Expert Systems with Applications 36 (2009) 6773–6779

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Applying adaptive prediction to sea-water quality measurements

E. Hatzikos [a], J. Hätönen [b], N. Bassiliades [c,*], I. Vlahavas [c], E. Fournou [d]

[a] Department of Automation, Technological Education Institution (TEI) of Thessaloniki, Greece
[b] Department of Automatic Control and Systems Engineering, University of Sheffield, United Kingdom
[c] Department of Informatics, Aristotle University of Thessaloniki, University Campus, 54124 Thessaloniki, Greece
[d] General Department of Sciences, Technological Education Institution (TEI) of Thessaloniki, Greece

ARTICLE INFO

Keywords:
Projection algorithm
Least square algorithm
One-day ahead prediction

ABSTRACT

This study explores the possibility of using adaptive filters to predict sea-water quality indicators such as water temperature, pH and dissolved oxygen based on measurements produced by an under-water measurement set-up. Two alternative adaptive approaches are tested, namely a projection algorithm and a least squares algorithm. These algorithms were chosen for comparison because they are widely used prediction algorithms. The results indicate that if the measurements remain reasonably stationary, it is possible to make one-day ahead predictions, which perform better than the prediction that the value of a certain quality variable tomorrow is going to be equal to the value today.

1. Introduction

Over the last decade an area of considerable importance is that of monitoring water quality and other environmental variables, in an effort to predict their future behaviour and prevent undesirable environmental situations, as well as, to enforce longer term actions for regional growth and development. The ability to predict one or more days ahead the quality of water in an ecosystem is a very important task, giving the possibility to the authorities for the necessary precaution actions on time. More specifically, water quality prediction is of great importance for several commercial applications, such as watering or swimming and piscicultures activities.

The work related to water quality prediction includes a variety of linear and nonlinear modelling techniques. Among the various models used are the early Bayesian probability network models focusing both on their accuracy and the correct characterization of the processes (Reckhow, 1999), the predictive clustering approach using a single decision tree for simultaneous prediction of multiple physico-chemical properties of river water from its current biological properties (Blockeed, Dzeroski, & Crbovic, 1999), the work of regression trees for predicting chemical parameters of river water quality from bioindicator data (Dzeroski, Demsar, & Grbovic, 2000), and the unvaried time series models for determining the long-term and seasonal behaviour of important water quality parameters (Lehmann & Rode, 2001). Nevertheless, similar studies have been done and new models developed, among which are the autonomous case-based reasoning (CBSR) hybrid system that embeds various artificial intelligence tools, such as case-based reasoning, neural networks and fuzzy logic in order to achieve real time forecasting (Fdez-Riverola & Corchado, 2003, 2004), the work of Romero and Shan with a neural network based software tool developed for prediction of the water discharged temperature for industrial purposes in power plant generation units (Romero & Shan, 2005), and the split-step particle swarm optimisation (PSO) model for training perceptrons applied in algal bloom prediction (Chau, 2005).

In this work, we investigate the possibility to predict a number of water quality variables that are obtained by an under-water measurement set-up. Our interest is focused on one-day ahead predictions of certain water quality variables recorded by an under-water set of sensors, such as water temperature, pH, conductivity, salinity, amount of dissolved oxygen and turbidity. The measured data, forming time series, are stored in a database and a number of modelling methods could then be used to reveal any hidden information. However, in this study we deal only with the development of prediction models for the water temperature, pH, amount of dissolved oxygen and turbidity, due to their higher importance in terms of commercial exploitation.

In a previous study (Hatzikos, Anastasakis, Bassiliades, & Vlahavas, 2005), we have used neural networks with active neurons as it was believed to be an appropriate prediction algorithm for noisy and short time series (Ivanhnenko & Moller, 1995; Moller & Lemke, 2003). In the present study, we further investigate the prediction possibility using two alternative adaptive approaches, namely a projection algorithm and a least squares algorithm. Their prediction ability is shown by comparing their performance against the delayed prediction algorithm ($y(t+1) = y(t)$, random

* Corresponding author. Tel.: +30 2310997913; fax: +30 2310998419.
E-mail addresses: hatzikos@teithe.gr (E. Hatzikos), hatonen@sheffield.ac.uk (J. Hätönen), nbassili@csd.auth.gr (N. Bassiliades), vlahavas@csd.auth.gr (I. Vlahavas), efournou@gen.teithe.gr (E. Fournou).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.08.051

walk model), which serves as a benchmark model in prediction tasks. The delayed prediction model states simply that the future value (tomorrow) of a variable will be equal to its current value (today), supporting in that way the unpredictability of the modelling object. However, due to the correlation and interaction between the water quality variables, it is interesting to investigate if there is an underlying mechanism that governs the data and thus will prove the predictability of these variables. The algorithms employed were chosen for comparison because they are widely used prediction algorithms. The identification of such models is particularly useful for ecologists and environmentalists since they will be able to predict in advance the pollution levels in the sea water and thus to instruct all the necessary precaution actions.

2. Description of the water monitoring network

The data used in this study have been produced by the Andromeda network (Hatzikos, 2002), a network of sensors (Fig. 1) plunged into the Thermaikos Gulf that collects aquatic numeric data concerning sea water. After sensor readings are collected, they are transmitted to a main station for processing and storage. An outline of the system's functions is provided within the next sections. A more thorough description can be found in Hatzikos (2002).

The network consists of:

- Local monitoring stations (LMSs), which record and transmit aquatic data to the main station. The LMS (Fig. 2) consists of:
  1. A buoy that floats on sea surface;
  2. A programmable logic circuit (PLC) by Siemens;
  3. Powerful Radio modems;
  4. A six meter high pillar for the support of the antenna;
  5. Four solar cells;
  6. High-capacity rechargeable batteries.
- Main station (MS), which initiates the communication process with all LMSs and stores the data in the database for future processing.

The LMS incorporates sensors, batteries, solar cells, electronics and the PLC. The necessary power is provided by the batteries and solar cells to the sensors and the electronics. The PLC is responsible for the LMS operation and storage of the measurements to the local memory. In predetermined time intervals, it transmits this data to the Main Station (MS), over a wireless network.

The sensors measure the following hydrological parameters: water temperature, pH, amount of dissolved oxygen (DO), percentage of dissolved oxygen (DO %), conductance, turbidity, sea currents, and salinity. However, in this study we deal only with the

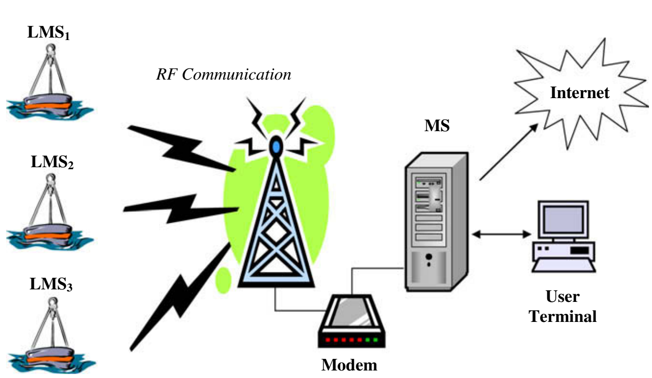

Fig. 1. Architecture of the Andromeda network.

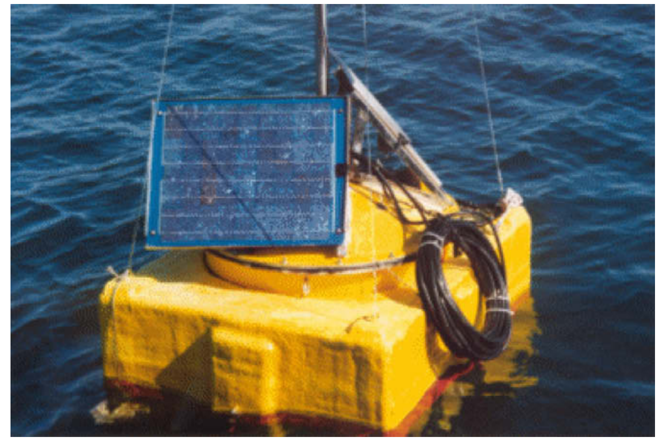

Fig. 2. Andromeda's local monitoring station.

development of prediction models for the water temperature, pH, amount of dissolved oxygen and turbidity, due to their higher importance in terms of commercial exploitation.

More specifically, the temperature of water plays an important role in both environmental and industrial processes. Firstly, it affects the ability of living organisms to resist certain pollutants. Some organisms cannot survive when the water temperature takes a value beyond a specific range. The ability of water to hold oxygen is also affected by water temperature. Finally, low-temperature water is used for cooling purposes in power plants.

pH is a measure of the relative amount of free hydrogen and hydroxyl ions in the water. Water that has more free hydrogen ions is acidic, whereas water that has more free hydroxyl ions is basic. The values of pH range from 0 to 14 (this is a logarithmic scale), with 7 indicating neutral. Values less than 7 indicate acidity, whereas values greater than 7 indicate a base. The presence of chemicals in the water, affects its pH, which in turn can harm the animals and plants that live there. For example, an even mildly acidulous sea water environment can harm shell cultivation. This renders pH an important water quality indicator.

Each molecule of water contains an atom of oxygen. Yet, only a small amount of these oxygen atoms, up to about ten oxygen molecules per million of water molecules, is actually dissolved in the water. This dissolved oxygen is breathed by fish and zooplankton and is necessary for their survival. Rapidly moving water, such as in a mountain streams or large rivers, tends to contain a lot of dissolved oxygen, while stagnant water contains little. Bacteria in water can consume oxygen as organic matter decays. Thus, excess organic material in lakes and rivers can cause an oxygen-deficient situation to occur. Aquatic life can have a hard time in stagnant water that has a lot of rotting, organic material in it, especially in the summer, when dissolved oxygen levels are at a seasonal low.

Turbidity is the amount of particulate matter that is suspended in water. Turbidity measures the scattering effect that suspended solids have on light: the higher the intensity of scattered light, the higher the turbidity. Materials that cause water to be turbid include clay, silt, finely divided organic and inorganic matter, soluble colored organic compounds, plankton, microscopic organisms and others.

Due to its design, the LMS can be easily updated and expanded. It responds well to extreme weather conditions, increased energy requirements, and meets the processing and storage needs for effectively monitoring the sea waters for the purpose of this research.

The main station (MS) of the Andromeda network is a workstation that collects sensor measurements from all the LMSs and visualizes the results in a SCADA environment. The MS is the "master" of the communication process, i.e. it initiates the communication with each of the LMSs in predetermined time intervals using a hand-shake technique. The MS also adjusts the frequency of measurements depending on the situation at hand, i.e. an emergency in the case of pollution. The LMS operates only during the rendezvous. In this way, less energy is consumed. Furthermore, the on-demand measurement policy achieves a higher level of flexibility. The SCADA software exports the measurements in various formats for further processing and long-term storage in databases.

The data collection from the sensors and their transmission to the MS is performed via the SCADA software. LabView is responsible for the data reception, visualization and storage at the MS. Additionally, the administrator of the MS can set alarms in LabView (to be received by the user) when certain sensor measurements exceed some predefined limits. However, the check is performed in a rigid fashion, not allowing flexibility.

Furthermore, the central station hosts an intelligent alerting system (Hatzikos, Bassiliades, Asmanis, & Vlahavas, 2007) that monitors sensor data and reasons about the current level of water suitability for various aquatic uses, such as swimming and piscicultures. The aim of this intelligent alerting system is to help the authorities in the "decision-making" process in the battle against the pollution of the aquatic environment, which is very vital for the public health and the economy of Northern Greece. The expert system determines, using fuzzy logic, when certain environmental parameters exceed certain "pollution" limits, which are specified either by the authorities or by environmental scientists, and flags out appropriate alerts.

3. Description of the model

As a starting point it is assumed that a given quality variable $y(t)$ follows a linear multiple-input single-output ARMA (auto-regressive moving-average) model

$$y(t) + \sum_{i=1}^{n} a_i y(t-i) = \sum_{j=1}^{s} \sum_{i=1}^{m-1} b_{ji} u_j(t-i) \tag{1}$$

where a_i, $b_i \in \Re$ are unknown coefficients and these parameters could be also time-varying and $u_j(t)$ are other quality variables that affect the time evolution of quality variable $y(t)$. Thus, the predictive value one-step ahead $y(t+1)$, of the quality variable $y(t)$, based on (1) is given by

$$y(t+1) = -\sum_{i=0}^{n-1} a_i y(t-i) + \sum_{j=1}^{s} \sum_{i=0}^{m-2} b_{ji} u_j(t-i) \tag{2}$$

However, the parameters a_i, b_{ji} are generally unknown, and they have to be estimated somehow from the on-line measurements. In order to do this, Eq. (1) is typically written in the form

$$y(t) = \phi(t)^{\mathrm{T}} \vartheta(t) \tag{3}$$

where

$$\begin{aligned} \varphi(t) &= [-y(t)\ \ldots\ -y(t-n)\ u_1(t-1)\ \ldots\ u_s(t-m)]^{\mathrm{T}} \\ \vartheta(t) &= [-a_1(t)\ \ldots\ -a_n(t)\ b_{11}(t)\ \ldots\ b_{sm}(t)]^{\mathrm{T}} \end{aligned} \tag{4}$$

After that, the parameter vector $\vartheta(t)$ is updated using the following update law

$$\vartheta(t+1) = \vartheta(t) + k(t)(y(t) - \varphi(t)^{T}\vartheta(t)) \tag{5}$$

where $\vartheta(t+1)$ is equal to $\vartheta(t)$ plus a corrective term that is proportional to the prediction error $\varepsilon(t) = y(t) - \varphi(t)^{\mathrm{T}}\vartheta(t)$ and $k(t)$ determines the magnitude of the corrective action. In the following two sections two alternative ways of calculating an 'optimal' $k(t)$ will be shown.

When applying the recursive identification in the one-step ahead prediction context, the prediction process consists of the following steps at time t:

(a) Calculate the prediction error $\varepsilon(t) = y(t) - \varphi(t-1)^{\mathrm{T}}\vartheta(t)$.
(b) Update $\vartheta(t)$ to $\vartheta(t+1)$ using $\vartheta(t+1) = \vartheta(t) + k(t)(y(t) - \varphi(t)^{\mathrm{T}}\vartheta(t))$.
(c) Build the vector $\varphi(t)$.
(d) Predict y(t + 1) using $y(t+1) = \varphi(t)^{\mathrm{T}}\vartheta(t+1)$.
(e) Set $t \to (t+1)$ and go to step a.

Furthermore, an initial condition $\vartheta(0)$ has to be specified in order to initialize the algorithm at $t = 0$.

3.1. Projection algorithm

This section is based on Goodwin and Sin (1984). Projection algorithm is one of several ways of recursively identifying the parameter vector ϑ. As a starting point the following cost function (or performance index)

$$J = \frac{1}{2}\|\vartheta(t) - \vartheta(t-1)\|^2 \tag{6}$$

is proposed, which is minimised subject to

$$y(t) = \varphi(t-1)^{\mathrm{T}}\vartheta \tag{7}$$

The performance index reflects the design criterion of finding an estimate $\vartheta(t)$ which is close to the previous estimate $\vartheta(t-1)$ but at the same time models the current data vector exactly, i.e. $y(t) = \varphi(t-1)^{\mathrm{T}}\vartheta(t)$. This optimisation problem can be solved in the following way: We introduce the modified cost function

$$J_e = \frac{1}{2}\|\vartheta(t) - \vartheta(t-1)\|^2 + \lambda[y(t) - \varphi(t-1)^{\mathrm{T}}\vartheta(t)] \tag{8}$$

where the constrained equation has been adjoined into the original cost function. Then, applying the necessary and sufficient conditions for minimum,

$$\frac{\partial J_e}{\partial \vartheta(t)} = 0, \qquad \frac{\partial J_e}{\partial \lambda} = 0 \tag{9}$$

we obtain, after some mathematical manipulation,

$$\lambda = \frac{y(t) - \varphi(t-1)^{\mathrm{T}}\vartheta(t-1)}{\varphi(t-1)^{\mathrm{T}}\varphi(t-1)} \tag{10}$$

and the algorithm

$$\vartheta(t) = \vartheta(t-1) + \frac{\varphi(t-1)}{\varphi(t-1)^{T}\varphi(t-1)}[y(t-1) - \vartheta(t-1)^{\mathrm{T}}\varphi(t-1)] \tag{11}$$

Because in (11) there is the possibility of division by zero, in practice we use the algorithm

$$\vartheta(t) = \vartheta(t-1) + \frac{a\varphi(t-1)}{c + \varphi(t-1)^{\mathrm{T}}\varphi(t-1)}[y(t-1) - \vartheta(t-1)^{\mathrm{T}}\varphi(t-1)] \tag{12}$$

where $c > 0$ and $0 < a < 1$. In the research literature the modified algorithm (12) is known as the normalized least-mean squares algorithm.

3.2. Least squares algorithm

This section is based on Goodwin and Sin (1984) and Bjorck (1997). The least squares algorithm is one of the most commonly used method to iteratively identify the parameter vector ϑ. The staring point is the following cost function

$$J_N(\vartheta) = \sum_{t=1}^{N}(y(t) - \varphi(t-1)^{\mathrm{T}}\vartheta + \frac{1}{2}(\vartheta - \vartheta(0))P_0^{-1}(\vartheta - \vartheta(0)) \tag{13}$$

Consequently, the objective is to find ϑ which minimises the sum of the squares of the estimation error $\varepsilon(t) = y(t) - \varphi(t-1)^{\mathrm{T}}\vartheta$ but is not too far from the initial guess $\vartheta(0)$. Thus, coming up with a reasonable initial guess for ϑ can speed up considerably the convergence rate of the least squares algorithm. In order to solve this optimisation problem recursively, the following data vectors are defined:

$$\begin{aligned} Y_N^{\mathrm{T}} &= [y(1)\, y(2)\, \ldots\, y(N)] \\ \Phi_{N-1}^{\mathrm{T}} &= [\varphi(0)\, \varphi(1)\, \ldots\, \varphi(N-1)] \end{aligned} \tag{14}$$

This allows one to write the cost function (13) in the form

$$J_N(\vartheta) = \frac{1}{2}[Y_N - \Phi_{N-1}\vartheta]^{\mathrm{T}}[Y_N - \Phi_{N-1}\vartheta] + \frac{1}{2}(\vartheta - \vartheta(0))P_0^{-1}(\vartheta - \vartheta(0)) \tag{15}$$

Differentiating with respect to ϑ and setting the result equal to zero we obtain

$$[\Phi_{N-1}^{\mathrm{T}}\Phi_{N-1} + P_0^{-1}]\vartheta = P_0^{-1}\vartheta(0) + \Phi_{N-1}^{\mathrm{T}}Y_N \tag{16}$$

Let $\vartheta(N)$ denotes the vector ϑ that satisfies Eq. (16). Then

$$\begin{aligned} \vartheta(N) &= [\Phi_{N-1}^{\mathrm{T}}\Phi_{N-1} + P_0^{-1}]^{-1}[P_0^{-1}\vartheta(0) + \Phi_{N-1}^{\mathrm{T}}Y_N] \\ &= P(N-1)[P_0^{-1}\vartheta(0) + \Phi_{N-1}^{\mathrm{T}}Y_N] \end{aligned} \tag{17}$$

where $P(N-1)^{-1} = \Phi_{N-1}^{\mathrm{T}}\Phi_{N-1} + P_0^{-1}$. Eq. (14) shows that

$$P(N-1)^{-1} = P(N-2)^{-1} + \varphi(N-1)\varphi(N-1)^{\mathrm{T}} \tag{18}$$

Using (17) gives

$$\begin{aligned} \vartheta(N) &= P(N-1)[P_0^{-1}\vartheta(0) + \Phi_{N-2}^{\mathrm{T}}Y_{N-1} + \varphi(N-1)y(N)] \\ &= P(N-1)[P(N-2)^{-1}\vartheta(N-1) + \varphi(N-1)y(N)] \\ &= P(N-1)[P(N-1)^{-1} - \varphi(N-1)\varphi(N-1)^{\mathrm{T}}]\vartheta(N-1) \\ &\quad + P(N-1)\varphi(N-1)y(N) \\ &= \vartheta(N-1) + P(N-1)\varphi(N-1)[y(N) - \varphi(N-1)^{\mathrm{T}}\vartheta(N-1)] \end{aligned} \tag{19}$$

Continuing recursively, and using the matrix inversion lemma, it can be shown that $P(t-1)$ satisfies

$$P(t-1) = P(t-2) - \frac{P(t-2)\varphi(t-1)\varphi(t-1)^{\mathrm{T}}P(t-2)}{1 + \varphi(t-1)^{\mathrm{T}}P(t-2)\varphi(t-1)} \tag{20}$$

and that this implies the following equality

$$P(t-1)\varphi(t-1) = \frac{P(t-2)\varphi(t-1)}{1 + \varphi(t-1)^{\mathrm{T}}P(t-2)\varphi(t-1)} \tag{21}$$

Applying (21) to the last row of (19) we obtain finally the least squares algorithm

$$\vartheta(t) = \vartheta(t-1) + \frac{P(t-2)\varphi(t-1)}{1 + \varphi(t-1)^{\mathrm{T}}P(t-2)\varphi(t-1)}[y(t) - \varphi(t-1)^{\mathrm{T}} \times \vartheta(t-1)] \tag{22}$$

where the update for $P(t-1)$ is given by (20).

In practical applications of least squares it is common to use a slightly modified algorithm

$$\vartheta(t) = \vartheta(t-1) + \frac{P(t-2)\varphi(t-1)}{c(t-1) + \varphi(t-1)^{\mathrm{T}}P(t-2)\varphi(t-1)}[y(t) - \varphi(t-1)^{\mathrm{T}}\vartheta(t-1)] \tag{23}$$

where the update for $P(t-1)$ is given by

$$P(t-1) = P(t-2) - \frac{P(t-2)\varphi(t-1)\varphi(t-1)^{\mathrm{T}}P(t-2)}{c(t-1) + \varphi(t-1)^{\mathrm{T}}P(t-2)\varphi(t-1)} \tag{24}$$

where $c(t) = a_o c(t-1) + (1 - a_o)$, with a_o having a value between 0 and 1. It can be shown that this exponential weighting puts more emphasis on the more recent data, and therefore allows the algorithm to perform better when the underlying process is either time-varying or nonlinear.

4. Experimental results

4.1. Preliminary data processing and analysis

The data used in this study are produced by the Andromeda-analyser, which measures water temperature, pH, conductivity, salinity, amount of oxygen (%) and turbidity (FNU) of sea water. For details see Hatzikos (1998) and Hatzikos (2002). The original data had a sampling time of 9 s. However, due to the fact that there was a large number of outliers in the data, and that the data did not vary too much on hourly basis, it was decided that the data should be averaged over one day. Furthermore, in this study it was decided that the prediction models for temperature, pH, oxygen and turbidity would be the important ones, mainly because they are the most interesting in terms of commercial exploitation. Consequently, for the time being, the rest of the variables are omitted from the analysis.

Table 1 shows the cross-correlations between the four variables. The correlations are calculated from one-month data sets collected during July 2004.

This table demonstrates that pH, oxygen and turbidity are correlated with each other to a moderate degree. Fig. 3 shows the autocorrelation sequences for temperature, pH, oxygen and turbidity.

This figure demonstrates that temperature, oxygen and turbidity are correlated with past values, and therefore it is possible to use them in prediction. The autocorrelation sequence for pH, on the other hand, is very sharply peaked around time lag $t = 0$, which implies that it could be very difficult to construct one-day ahead predictions for this variable.

4.2. Water temperature

In the water temperature prediction only its previous values were used. This was due to the fact that the water temperature is mainly influenced by external variables such as the amount of radiation from the sun, wind direction etc. Based on the autocorrelation analysis, it was decided that the current and two previous values of water temperature should be used to predict the water temperature next day. The initial estimate for the parameter vector was taken to be $\vartheta(0) = [1;\ 0\ \ 0]^{\mathrm{T}}$ – this reflects the reasoning that in most cases the temperature tomorrow should be more or less equal to the temperature today. Before building the prediction model, the data were scaled to zero mean and unity variance. Fig. 4 shows the measured and the predicted temperature by use of the Projection algorithm and of the Least squares algorithm, with a data set taken during July of 2004.

Table 1
Cross-correlated matrix for temperature, pH, oxygen and turbidity

	Temperature	pH	Oxygen	Turbidity
Temperature	1			
pH	0.50	1		
Oxygen	0.68	0.76	1	
Turbidity	0.37	0.78	0.74	1

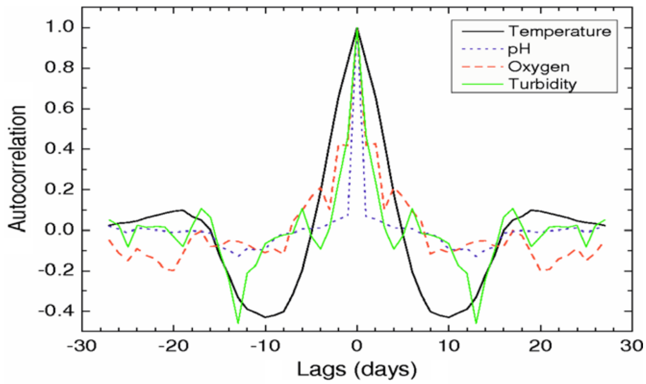

Fig. 3. Autocorrelation sequences for temperature, pH, oxygen and turbidity.

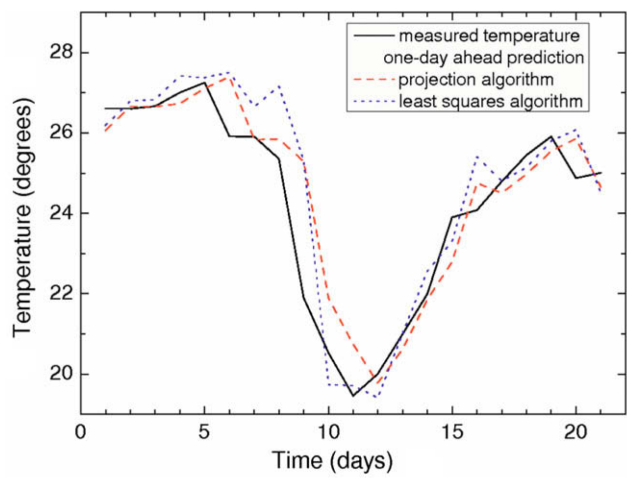

Fig. 4. Temperature prediction by use of the projection algorithm and least square algorithm one-day ahead with July 2004 data set.

4.2.1. Projection algorithm

The parameters c and a in the projection algorithm were taken to be $c = 6$ and $a = 0.5$. These parameter values were chosen using the 'trial and error method'. The prediction is surprisingly accurate, in particular after day twelve. The algorithm converges to the parameter vector $\vartheta = [0.911 \quad -0.1011 \quad -0.1634]^T$ (for scaled data), which shows that the algorithm bases its prediction mostly on the current temperature, but corrects the value using previous temperatures as well.

4.2.2. Least squares algorithm

The weighted version of the least squares algorithm were used with parameter values $P(0) = 0.1I$, $c(0) = 0.1$ and $c(t) = 0.95c(t-1) + (1-0.95)$. The prediction accuracy here is reasonable. The algorithm converges to $\vartheta = [1.1062 \quad -0.07 \quad 0.1634]^T$, which shows that the resulting prediction model uses more heavily the current temperature than the prediction model from the projection algorithm.

4.2.3. Comparison of the results

Table 2 shows the l_2-norm $(\sum_{i=1}^{N} e(i)^2)$ and l_∞-norm$(\max_{i=1,\dots,N}|e(i)|)$ of the prediction error for the projection algorithm, the least squares algorithm and the 'delayed prediction algorithm' $y(t+1) = y(t)$ (i.e. temperature tomorrow is equal to the temperature today). Prediction models from both the projection algorithm and least squares algorithm give better prediction accuracy, which is a remarkable result. Furthermore, the prediction model from the projection algorithm is the most accurate one.

Table 2
Prediction accuracy of temperature

Prediction method	l_2	l_∞
Projection algorithm	2.0	1.47
Least square algorithm	2.0	1.5
Delayed prediction algorithm	2.2	5.5

4.3. pH

In the pH prediction previous values of both pH and turbidity were used. This was due to the fact that turbidity and pH seem to be correlated with each other, at least to certain extent. Based on the autocorrelation analysis, it was decided that the current and two previous values of pH and turbidity measurements should be used to predict the pH value for the next day. The initial estimate for the parameter vector was taken to be $\vartheta(0) = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$; this reflects again the reasoning that in most cases pH tomorrow is should be more or less equal to pH today. Before building the prediction model, the data were scaled to zero mean and unity variance. Fig. 5 shows the measured and the predicted pH by use of the Projection algorithm and of the Least squares algorithm, with a data set taken during July of 2004.

4.3.1. Projection algorithm

The parameters c and a in the projection algorithm were taken to be c=6 and a=0.5. The algorithm converges to the parameter vector $\vartheta = [0.99 \quad -0.0092 \quad -0.0127 \quad 0.0028 \quad -0.0146 \quad -0.0178]^T$ (for scaled data), which shows that the algorithm bases its prediction almost only on the current pH. Consequently, the best one can do is to predict that pH tomorrow is equal to the pH today.

4.3.2. Least squares algorithm

The weighted version of the least squares algorithm were used with parameter values $P(0) = 0.1I$, $c(0) = 0.1$ and $c(t) = 0.95c(t-1) + (1-0.95)$. The algorithm converges to $\vartheta = [0.987 \quad -0.0019 \quad -0.0076 \quad 0.0013 \quad -0.0258 \quad -0.0096]^T$, which shows that the resulting prediction model uses heavily the current pH value.

4.3.3. Comparison of the results

Table 3 shows the l_2-norm $(\sum_{i=1}^{N} e(i)^2)$ and l_∞-norm$(\max_{i=1,\dots,N}|e(i)|)$ of the prediction error for the projection algorithm, the least squares algorithm and the 'delayed prediction algorithm' $y(t+1) = y(t)$. Prediction models from both the projection algorithm and least squares algorithm give more or less the same accuracy as

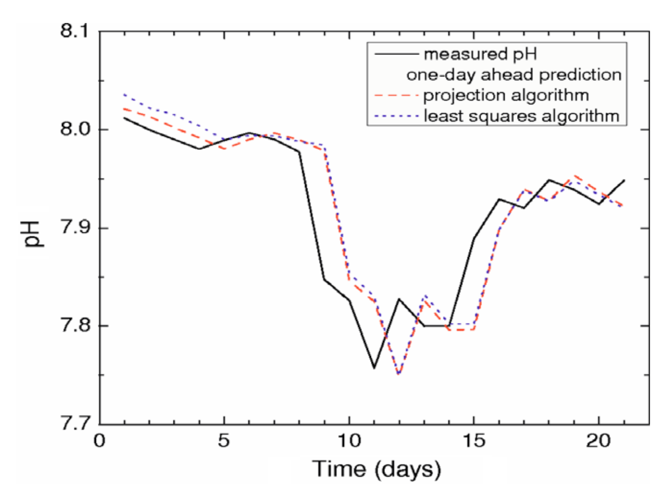

Fig. 5. pH prediction by use of the projection algorithm and least square algorithm one-day ahead with July 2004 data set.

Table 3
Prediction accuracy of pH

Prediction method	l_2	l_∞
Projection algorithm	0.60	0.39
Least square algorithm	0.64	0.41
Delayed prediction algorithm	0.60	0.39

the delayed prediction algorithm, which shows that best one can do is to predict that pH tomorrow is equal to the pH today. This could be due to the fact that the chemical reactions determining pH are typically nonlinear, and therefore linear models used in this study are not adequate to predict pH.

4.4. Oxygen

In the oxygen prediction both previous values of oxygen and pH were used. This was due to the fact oxygen and pH seem to be correlated with each other to a moderate degree. Based on the autocorrelation analysis, it was decided that the current and two previous values of oxygen and pH measurements should be used to predict the oxygen value for the next day. The initial estimate for the parameter vector was taken to be $\vartheta(0) = [1\ 0\ 0\ 0\ 0\ 0]^T$; this reflects again the reasoning that in most cases the amount of oxygen tomorrow should be more or less equal to the amount of oxygen today. Before building the prediction model, the data were scaled to zero mean and unity variance. Fig. 6 shows the measured and the predicted oxygen by use of the Projection algorithm and of the Least squares algorithm, with a data set taken during July of 2004.

4.4.1. Projection algorithm

The parameters c and a in the projection algorithm were taken to be $c = 5$ and $a = 0.1$. The algorithm in this case converges to the parameter vector $\vartheta = [0.95\ \ -0.0322\ \ -0.0229\ \ 0.0032\ \ -0.0010\ \ -0.0055]^T$ (for scaled data), which shows that the algorithm bases its prediction mostly on the current oxygen content.

4.4.2. Least squares algorithm

The weighted version of the least squares algorithm were used with parameter values $P(0) = 0.1I$, $c(0) = 0.95$ and $c(t) = 0.95c(t-1) + (1-0.95)$. The algorithm converges to $\vartheta = [0.7891\ \ -0.0633\ \ -0.0755\ \ 0.0102\ \ 0.0231\ \ -0.0113]^T$, which shows again that the resulting prediction model uses heavily the current oxygen measurement.

Table 4
Prediction accuracy of oxygen

Prediction method	l_2	l_∞
Projection algorithm	3.17	1.92
Least square algorithm	3.23	1.88
Delayed prediction algorithm	3.18	1.96

4.4.3. Comparison of the results

Table 4 shows the l_2-norm $(\sum_{i=1}^{N} e(i)^2)$and l_∞-norm$(\max_{i=1,\dots,N}|e(i)|)$ of the prediction error for the projection algorithm, the least squares algorithm and the 'delayed prediction algorithm' $y(t+1) = y(t)$. Prediction models from both the projection algorithm and least squares algorithm give slightly better results than the delayed prediction algorithm, so the prediction of oxygen works better than pH but worse than water temperature.

4.5. Turbidity

In the turbidity prediction both previous values of turbidity and pH were used. This was due to the fact that turbidity and pH seem to be correlated with each other to a moderate degree. Based on the autocorrelation analysis, it was decided that the current and two previous values of turbidity and pH measurements should be used the predict turbidity value for the next day. The initial estimate for the parameter vector was taken to be $\vartheta(0) = [1\ 0\ 0\ 0\ 0\ 0]^T$ (turbidity tomorrow is should be more or less equal to turbidity today). Before building the prediction model, the data were scaled to zero mean and unity variance. Fig. 7 shows the measured and the predicted turbidity by use of the projection algorithm and of the least squares algorithm, with a data set taken during July of 2004.

4.5.1. Projection algorithm

The parameters c and a in the projection algorithm were taken to be $c = 5$ and $a = 0.5$. The algorithm converges to the parameter vector $\vartheta = [0.95\ \ -0.0322\ \ -0.0229\ \ 0.0032\ \ -0.0010\ \ -0.0055]^T$ (for scaled data), which shows that the algorithm bases its prediction mostly on the current turbidity.

4.5.2. Least squares algorithm

The weighted version of the least squares algorithm were used with parameter values $P(0) = 0.1I, c(0) = 0.95$ and $c(t) = 0.95c(t-1) + (1-0.95)$. The algorithm converges to $\vartheta = [0.9345\ \ -0.0498$

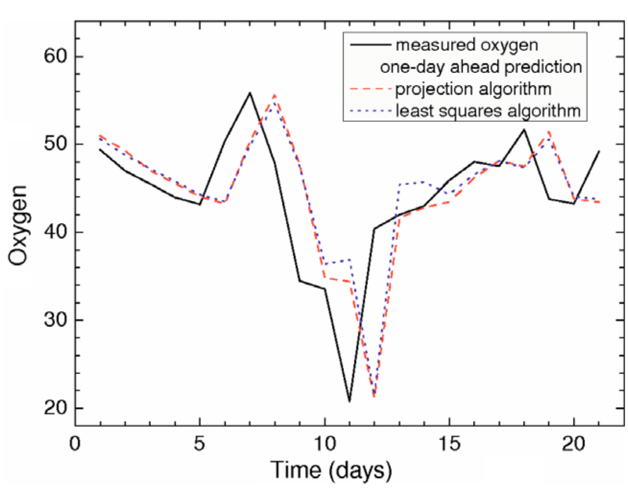

Fig. 6. Oxygen (%) prediction by use of the Projection algorithm and least square algorithm one-day ahead with July 2004 data set.

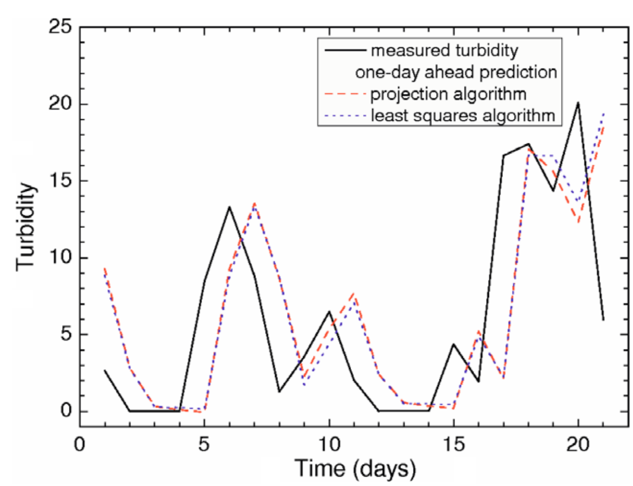

Fig. 7. Turbidity (FNU) prediction by use of the projection algorithm and least square algorithm one-day ahead with July 2004 data set.

Table 5
Prediction accuracy of turbidity

Prediction method	l_2	l_∞
Projection algorithm	1.82	0.99
Least square algorithm	1.80	0.97
Delayed prediction algorithm	1.84	0.99

$0.0048 \ \ -0.0055 \ \ 0.0046 \ \ -0.0174]^T$, which shows again that the resulting prediction model uses heavily the current turbidity measurement.

4.5.3. Comparison of the results

Table 5 shows the l_2-norm $(\sum_{i=1}^{N} e(i)^2)$ and l_∞-norm $(\max_{i=1,\dots,N}|e(i)|)$ of the prediction error for the projection algorithm, the least squares algorithm and the 'delayed prediction algorithm' $y(t+1) = y(t)$. Prediction models from both the projection algorithm and least squares algorithm give slightly better results than the delayed prediction algorithm, so the prediction of turbidity works similarly to the case of oxygen.

5. Conclusions and future work

This study has concentrated on building one-day ahead predictions for sea quality measurements. Two different algorithms were used to build prediction models, namely the projection algorithm and the least squares algorithm. The prediction accuracy from the two algorithms were quite similar, but experimental work pointed out that it is slightly easier to tune the projection algorithm than the least squares algorithm.

The results show that for water temperature it is possible to come up with predictors that are 'substantially' better than the delayed prediction algorithm. For oxygen and turbidity the difference between adaptive and delayed prediction is far less substantial, while for pH, both predictions give more or less the same accuracy.

Future study will concentrate on the exploration of the possibility to construct prediction models for the other variables on shorter time-scales than the one-day ahead prediction. For example, the autocorrelation series for pH shows that the dynamics of this variable change at a quicker rate than 24 h. Therefore, it could be possible to use for example an 1-h ahead prediction for this variable.

Furthermore, we plan to integrate the water quality prediction algorithms we presented in this paper within the fuzzy intelligent alerting system of Hatzikos et al. (2007), so that the alerting system will be able to issue early warnings based on predicted hydrological parameters values. To this end, we have to extend our least square algorithms using fuzzy regression techniques (Hong & Hwang, 2004; Wang, Zhang, & Mei, 2007).

Acknowledgement

The project is co-funded by the European Social Fund & National Resources - EPEAEK II - ARCHIMEDES.

References

Bjorck, E. (1997). *Numerical methods for least squares problems*. SIAM.
Blockeed, H., Dzeroski, S., & Crbovic, J. (1999). Simultaneous prediction of multiple chemical parameters of river water quality with tilde. *Proceedings of the 3rd European conference on principles of data mining and knowledge discovery. LNAI* (Vol. 1704). Springer-Verlag.
Chau, K. (2005). A split-step PSO algorithm in prediction of water quality pollution. In *Proceedings of the 2nd international symposium of neural networks*.
Dzeroski, S., Demsar, D., & Grbovic, J. (2000). Predicting chemical parameters of river water quality from bioindicator data. *Applied Intelligence, 13*, 7–17.
Fdez-Riverola, F., & Corchado, J. M. (2003). BR based system for forecasting red tides. *Knowledge-Based Systems, 16*, 321–328.
Fdez-Riverola, F., & Corchado, J. M. (2004). FSfRT: Forecasting system for red tides. *Applied Intelligence, 21*, 251–264.
Goodwin, G. C., & Sin, K. S. (1984). *Adaptive filtering prediction and control*. Prentice Hall.
Hatzikos, E. V. (1998). The Andromeda network for monitoring the quality of water and air elements. In *Proceedings of the 2nd conference on automation and technology, Thessaloniki, Greece, October*.
Hatzikos, E. V. (2002). A fully automated control network for monitoring polluted water elements. In *Proceedings of the 4th conference on automation and technology, Thessaloniki, Greece, October* (pp. 443–448).
Hatzikos, E. V., Anastasakis, L., Bassiliades, N., & Vlahavas, I. (2005). Applying neural networks with active neurons to sea-water quality predictions. In *Proceedings of the 2nd international scientific conference on computer Science* (pp. 114–119). Bulgarian Section: IEEE Computer Society.
Hatzikos, E., Bassiliades, N., Asmanis, L., & Vlahavas, I. (2007). Monitoring water quality through a telematic sensor network and a fuzzy expert system. *Expert Systems, 24*(3), 143–161.
Hong, D. H., & Hwang, C. (2004). Extended fuzzy regression models using regularization method. *Information Sciences, 164*(1–4), 31–46.
Ivanhnenko, A. G., & Moller, J. A. (1995). Self-organisation of nets of active neurons. *Systems Analysis Modelling Simulation, 20*(1–2), 93–106.
Lehmann, A., & Rode, M. (2001). Long-term behaviour and cross-correlation water quality analysis of the river Elbe, Germany. *Water Research, 35*(9), 2153–2160.
Moller, J. A., & Lemke, F. (2003). *Self-organising data mining: An intelligent approach to extract knowledge from data*. Canada: Trafford Publishing.
National Instruments, LabView. <http://www.ni.com/labview/>.
Reckhow, K. (1999). Water quality prediction and probability network models. *Canadian Journal of Fisheries and Aquatic Sciences, 56*, 1150–1158.
Romero, C., & Shan, J. (2005). Development of an artificial neural network-based software for prediction of power plant canal water discharge temperature. *Expert Systems with Applications, 29*, 831–838.
Wang, N., Zhang, W. X., & Mei, C. L. (2007). Fuzzy nonparametric regression based on local linear smoothing technique. *Information Sciences, 177*(18), 3882–3900.