Prediction of Wastewater Pre-precipitation Variables using Self-Organizing Networks

Padelis Isaias, Sara Nilsson, Anna Stathaki and Robert E. King, *Life Senior Member, IEEE*

Abstract— **This paper describes the derivation and design of an array of self-organizing networks trained by inductive learning for one step ahead prediction of the outputs of the pre-precipitation stage of a wastewater treatment plant with a view to model predictive control of the stage.**

I. Introduction

The influent of any wastewater treatment plant (WWTP) varies significantly over time both in its inflow rate as well as its chemical composition. A WWTP is essentially a cascade of stages, as depicted in Fig. 1, wherein the wastewater undergoes successive processing with the object of satisfying specific quality standards in the process effluent that allow for its safe return to the environment and potential re-use. In general on-line instrumentation at WWTPs is inadequate or limited and as a consequence controlling the quality of the process effluent is by no means an easy task. This field is attracting considerable interest of late because of its urgency.

One of the principal stages of a WWTP is pre-precipitation in which the raw sewage is treated mechanically and chemicals that remove phosphates and organic matter. This technique is widely used for removal of contaminants and the quality of the wastewater at this stage has a direct effect on the effectiveness and efficiency of the overall plant. A pressing need in most WWTPs is improved performance, minimization of the operational costs and environmental impact while stringent standards on the process effluents must be adhered to. Optimization of every stage of the WWTP demands systematic investigation of each stage and determination of control strategies [1] that will lead to minimization of multiple objective criteria. Any such investigation demands determination of models of each stage of the process, a task that is by no means simple. Often, due to the complexity of the physical process *microscopic* models (i.e. mathematical models, typically state equations to account for the dynamic behavior of the process) cannot be established and recourse has to be taken to *macroscopic* models (i.e. black box input-output or *holistic* models). *Deductive neural networks* [2-4] have been extensively used in determining macroscopic models of physical processes.

Another class of neural networks or learning machines [5] that has proved effective for both modeling and prediction uses self-organizing networks, which unlike deductive neural networks, are trained using *inductive learning*. This class of networks possesses not only the optimum weights but the *optimal* structure. Self-organizing networks are also referred to as Group Method of Data Handling or GMDH and were first described by Ivakhnenko [6-9] in the mid 1960's. These networks differ from traditional deductive neural networks in that they use clusters of polynomial networks stacked in layers, each network of which contains only one pair of inputs. The complexity of the network (i.e. its number of layers) is not specified *a priori* but evolves during successive training and selection cycles, increasing its complexity with each training/selection cycle until no further improvement is obtained at which point the learning/ selection cycle is terminated.

This paper presents some results on the design of an inductive predictor of the principal variables of the pre-precipitation stage of a wastewater treatment plant using self-organizing networks with a view to its predictive control.

II. The Wastewater Pre-precipitation Process

A schematic diagram of the various stages of a typical WWTP is shown in Figure 1. Here wastewater inflow is first passed through a coarse screen and a sand trap before entering a pre-aeration basin where a solution of ferrous sulphate ($FeSO_4$) is added. In the presence of oxygen the Fe(II)-ions are oxidized to Fe(III)-ions which form a non-soluble precipitate with the phosphate (PO_4^{3-}) in the water, while simultaneously some iron hydroxide also precipitates to form sludge. Addition of $FeSO_4$ is made proportional to the flow rate of the incoming water. After pre-aeration, water enters a block of parallel primary sedimentation basins where the sludge is separated through sedimentation.

Manuscript received in final form on March 25, 2005. The work reported here was performed as part of an ongoing EU research project 'HIPCON' STRP-505467-1 in which the Environmental Research Institute IVL, Uppsala University, Stockholm Vatten and SSAB from Sweden, the London School of Economics and IMCG from the UK as well as the Computer Technology Institute in Athens, Greece are participating.

P. Isaias (padelis@otenet.gr) and A. Stathaki (stathaki@cti.gr) are in the Data and Knowledge Engineering Group at the Computer Research Institute in Athens, Greece.

S. Nilsson is with the Environmental Research Institute IVL, Stockholm, Sweden (e-mail: sara.nilsson@ivl.se).

R. E. King is Professor of Electrical Engineering and Computer Science at the University of Patras, Greece and Senior Research Associate at the Computer Research Institute in Athens, Greece. (Corresponding author e-mail: reking@cti.gr).

Samples of the key variables at the inputs and outputs of the pre-precipitation stage that precedes the Activated Sludge Process were taken at regular intervals over a three day period. Precipitation and flocculation in the pre-precipitation stage is directly affected by the *pH* of the water inflow. The Chemical Oxygen Demand or COD (*COD,* g/m^3*)* defines the organic contents of the water and is a measure of the fraction of the pollutant contents of the water that can be oxidized by an agent. When reaching the recipient, the organic material will consume oxygen when degraded, causing oxygen depletion. Phosphorus (*P-tot,* g/m^3) appears in different forms in wastewater. Phosphorus bound in organic compounds is referred to as organic phosphorus while inorganic phosphorus, also called soluble phosphorus, consists of polyphosphate and orthophosphate. *P-tot* comprises the total phosphorus in the water. Discharge of phosphorus can cause eutrophication of the recipient.

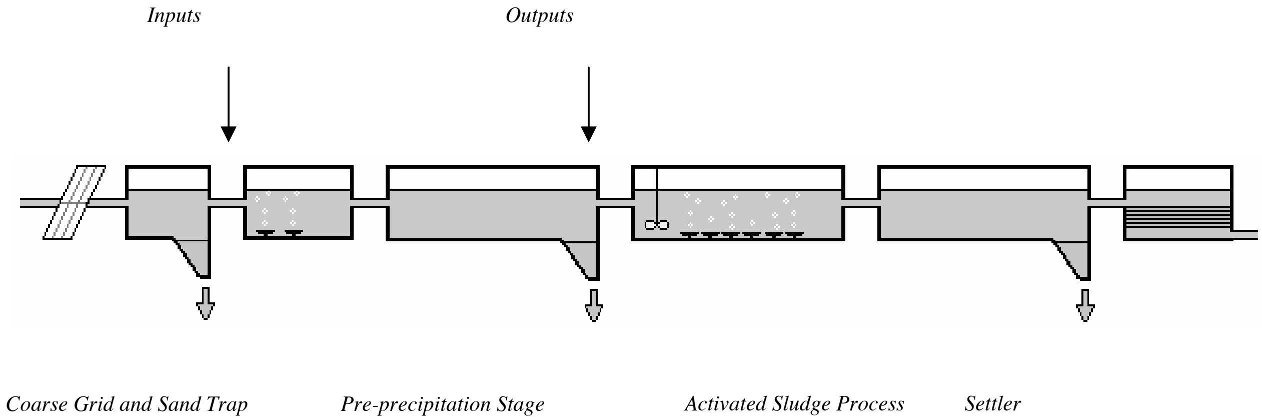


Fig. 1. Schematic diagram of the overall WWTP

Conductivity (*Cond,* μS/cm) is a measure of the ionic contents of the water and indirectly the composition of the water. This is true also for ammonium which constitutes a major part of the total amount of nitrogen in the wastewater. Suspended Solids (*SS,* g/m^3) is a measure of the suspended solids in the water and is defined as the amount of particles larger than 0.45 μm in the water. A major part of the SS is organic material, and can except for turbidity, cause oxygen depletion. Samples were taken 5 times an hour from the water inflow and outflow after the pre-sedimentation stage. The samples were pooled to represent the water for the current hour (i.e. 5 samples per pool). To filter the samples for COD_f, $P\text{-}tot_f$ and SS_f a glass fiber filter was used.

The pre-precipitation stage in a WWTP is a dynamic process whose inputs and outputs are time-varying. The outputs are invariably corrupted by random variations and instrument noise, a fact that makes accurate prediction for subsequent control difficult at best. Dynamic systems whose components and structure are known are most often represented by differential or difference equations where the rate of change of each state is a function of the state and the forcing function. Real world processes are mostly nonlinear and time-varying, compounding the problem of establishing a sufficiently simple *microscopic* model which can be used for simulation, prediction and control. In some cases such models can be derived from first principles [1] but this is a laborious and time-consuming procedure that invariably is followed by exhaustive field tests to verify and validate the model under different operating conditions.

In this paper we focus attention on prediction of the output of the pre-precipitation stage of a WWTP using self-organizing network techniques [10-14] with a view to using the predicted variables for model predictive control of the stage. Since our objective here is to derive a *macroscopic* (i.e. input-output or black-box) model of the pre-treatment process, self-organizing networks were examined at length for.

Unlike classical feed-forward deductive neural networks, whose structure is pre-defined [3,4], self-organizing networks have a layered structure that evolves during the training/selection process [12,14]. Starting with a simple structure, learning is followed by selection in which only those branches of the network that contribute significantly in predicting the output are retained, while those that do are appropriately pruned. Network evolution requires successive addition of new layers of neurons and leads to increasing complexity and thereby increasing fidelity. The evolutionary procedure terminates when the error norm no longer decreases.

III. THE SELF-ORGANIZING PNN PREDICTOR

Self-organizing predictors belong to the class of multi-layered polynomial networks (PNNs) that have proved very effective in practice in both modeling and prediction. Two salient features of this class of networks differentiate them from conventional *deductive* neural networks:

- Their structure evolves during training instead of being predefined. Thus there is no need for initial guessing about the size of the network.
- By their very nature self-organizing networks cannot be over-trained.

To achieve the latter property, the data set is divided into three subsets: the *Training Set*, the *Selection Set* and the *Evaluation Set*. The nodes of a self-organizing polynomial network are N-*Adalines*, i.e. *Adalines* with a nonlinear preprocessor. Six parameters (weights) are needed to define such a function. The main steps required in training this class of networks are:

- Define the signals which will be considered as inputs to the network. When used for prediction, past values of the output could be used only, while for a modeling application signals that have a causal relation with the output should be considered as inputs to the network.
- A self-organizing multi-input network can generate only one output signal. For the multi-output case the number of networks that must be trained is equal to the number of outputs. These networks need not have the same structure and therefore require individual training.
- Preprocessing data affects network training significantly. Instead of using raw data its is often advantages The following normalization is commonly used:

$$x = \frac{m - \overline{m}}{\sigma}$$

  where m is the measured variable, $\overline{m}$ is its mean value and σ is its standard deviation over a finite data set.
- In a real application noise in the measurements has to be taken into account as well. In modeling tasks, noise can be helpful (a well known result from system identification theory) but for prediction in contrast, pre-processing (data smoothing) is essential since random noise in the data leads to misleading results and increased computation.
- From the moment the input signals are specified it is necessary to specify the time span in the past to use. In the self-organizing technique this decision is not critical in principle, since unnecessary nodes are pruned in any case. To be more specific, given a time $x \in \mathbb{R}^N$ it is desired to predict one step ahead using d past samples, define a signal matrix whose first column is $\left[x_1\ x_2\ ...x_{d+1}\right]^T$ and last row is $\left[x_{d+1}\ x_{d+2}\ ...x_N\right]$. All other columns are delayed versions of the first. This structure can be expressed by the *Hankel* matrix:

$$H = \begin{bmatrix} x_1 & x_2 & \cdots & x_{N-d} \\ x_2 & x_3 & & \\ \vdots & \vdots & & \vdots \\ x_{d+1} & x_{d+2} & \cdots & x_N \end{bmatrix}$$

  It is clear that the last row constitutes the desired output array and all other rows are the elements of the predicted array.
- The first layer of the node consisting of PNNs can now be created. One technique for generating these is to use combinations of data values in the form of ordered pairs in a computationally efficient manner. The method used here finds the indices of the non-zero entries of an upper triangular matrix of appropriate dimensions.
- Each node generates the desired output and is trained using the *Widrow-Hoff* learning rule [2-4]. Training is terminated either when some error criterion (e.g. mean square error of the Training Set) is obtained or when the maximum number of epochs is reached.
- Having computed the trained layer, selection of the best neurons follows. The MSE of the Selection Set is subsequently computed for each neuron and those neurons yielding comparatively large error are discarded.
- The outputs of the post-selection neurons (i.e. survivors) constitute the inputs of the next layer. Addition of further layers is not called for when either the minimum MSE increases or when a predefined maximum number of layers has been reached.
- The performance of the network is ascertained using the Evaluation Set. It sis noted that the Selection Set contains new data since it has not hitherto been used for training.

The generalized transfer relationship between the inputs and the output of a MISO self-organizing network can be approximated by an infinite Volterra-Kolmogorov-Gabor (VKG) polynomial, the analog of a continuous Volterra series:

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j + ...$$

where n is the number of inputs. This polynomial can approximate any stationary random sequence of measurements. Ivakhnenko showed that the VKG series could be expressed as a cascade of low order polynomials using only pairs of inputs and employing an iterative perceptron - like procedure to approximate the VKG polynomial to any degree of accuracy by interconnecting an increasing number of low order polynomials. As the procedure proceeds, furthermore, it is found that many node connections do not contribute significantly to the final result and can safely be pruned to permit only the essential relationships to evolve.

To simplify the procedure, consider here the case of a single polynomial network that involves only a single pair of inputs (y_i, y_j) to generate an output equal to the inner product $y = < W, Y >$, where

$$Y = \left[1 \; y_i \; y_i^2 \; y_i y_j \; {y_j}^2 \; y_j\right]^T$$

is the output array while the weight vector is given by

$$W = \left[w_0 \; w_1 \; w_2 \; w_3 \; w_4 \; w_5\right]^T$$

During each phase of evolution, the weights of the networks in each layer are determined and those branches that lead to high mean squared errors are pruned, the 'best' branches that yield minimum MSE being retained. Interestingly, the computational effort and speed of learning of this class of networks is much less than that required to train a corresponding conventional feed-forward ANN due of the polynomial nature of the optimization problem.

Using only two variables as inputs per neuron requires computation of 6 weights per neuron in every layer. Training such a network consists of configuring the network by starting from the input layer which involves a total of $3n(n-1)$ weights, where n is the number of input variables. It is clear that the number of neurons in each layer increases roughly as the square of the number of inputs. A finite training set, which is sufficiently long to represent the underlying physical process, but not too long as to lead to over-training, is used in the training/selection procedure. The length of this time series is a result of experimentation and is heuristic. The optimum weights for each neuron at each layer are then computed using any training algorithm. The delta rule of *Widrow and Hoff* is commonly used because of its simplicity. The speed of convergence can be increased significantly using more sophisticated methods. The training algorithm is given by the recursion:

$$W_{k+1} = W_k + \mu \frac{X_k}{|X_k|^2}(y_k^d - W_k^T X_k)$$

where W_k is the neuron weight vector and X_k is the total neuron input vector at time $k = 1,2,...K$. The desired output is given by y_k^d and $\mu \approx 0.001$ is the learning rate. When the error norm of the output of the training set reaches a minimum then the weights of the neuron in that layer are held constant and training for that layer is terminated.

Following learning, a new data set, often shorter than the *training* set and termed the *selection* set, is fed to the network and the error norm for each neuron is recomputed. Those neurons that yield inordinately large errors are subsequently pruned as they are deemed to contribute to the end result insignificantly. This is an essential difference between self-organizing and conventional deductive neural networks. Network complexity is then increased by two degrees by adding a second layer and the procedure of training and selection is repeated once more. This procedure continues as many times as the overall error norm no longer decreases or when only one neuron remains in the current layer. At this point the evolutionary procedure is terminated.

For the case of one step ahead prediction only the current value the output and its most recent past values need only be used. The time span into the past required to predict the next value could also be determined inductively using this class of networks.

In the case study, samples of measurements were taken from the outflow of the pre-precipitation stage every hour over a 3 day measurement campaign. The physical measurements are shown as continuous lines while the values predicted by the self-organizing network are shown as dashed lines. It is seen that even with two or three layer networks and two previous data values, fidelity is quite acceptable.

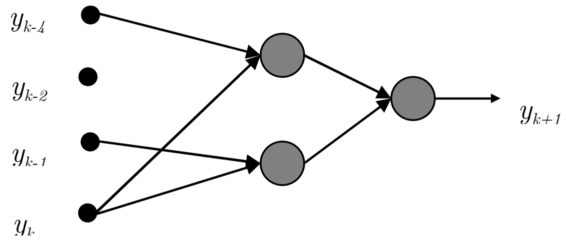
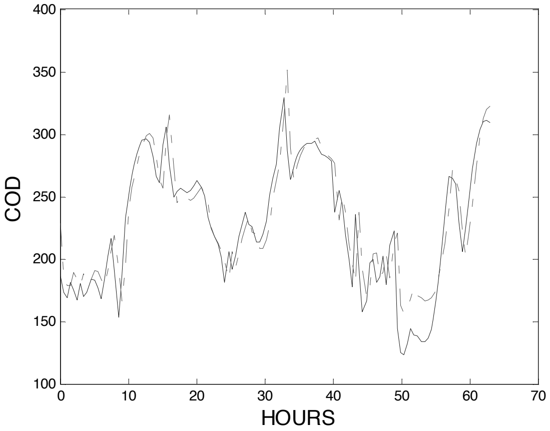


Fig. 2. Self-organizing network for one step ahead prediction of *Cond*.

Examples of the inductive networks that evolved during training are shown in Figs. 3 and 4. The first shows a simple two layer predictor of the variable *Cond*. It is seen that the last measurement y_k and only the previous values y_{k-1} and y_{k-3} are required to predict the next output. Interestingly, the data value two time samples in the past y_{k-2} does not enter the expression for the predicted output.



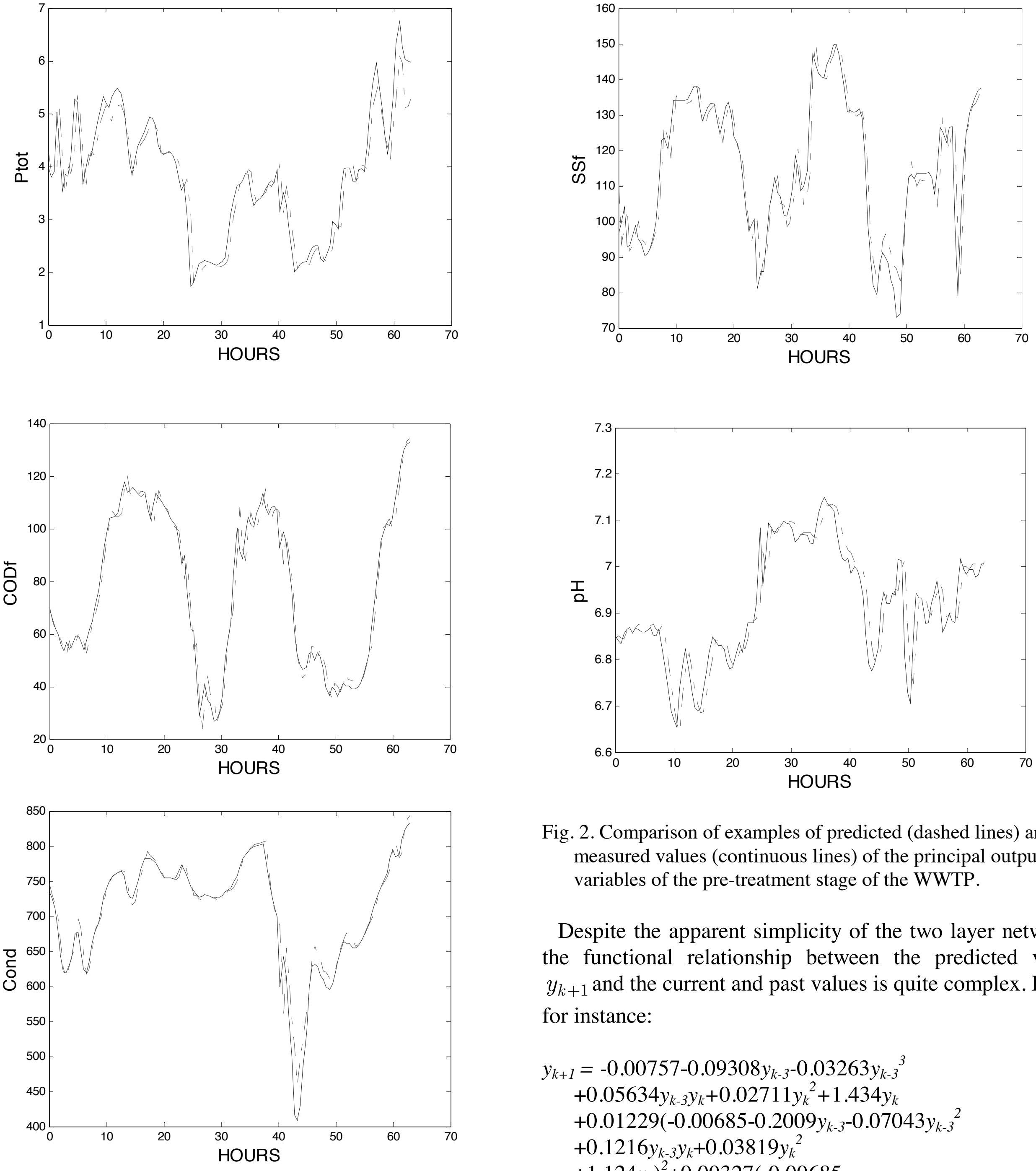


Fig. 2. Comparison of examples of predicted (dashed lines) and measured values (continuous lines) of the principal output variables of the pre-treatment stage of the WWTP.

Despite the apparent simplicity of the two layer network, the functional relationship between the predicted value y_{k+1} and the current and past values is quite complex. Here, for instance:

$$
\begin{aligned}
y_{k+1} = {} & -0.00757-0.09308y_{k-3}-0.03263y_{k-3}^{3} \\
& +0.05634y_{k-3}y_k+0.02711y_k^{2}+1.434y_k \\
& +0.01229(-0.00685-0.2009y_{k-3}-0.07043y_{k-3}^{2} \\
& +0.1216y_{k-3}y_k+0.03819y_k^{2} \\
& +1.124y_k)^{2}+0.00327(-0.00685 \\
& -0.2009y_{k-3}-0.07043y_{k-3}^{2}+0.1216y_{k-3}y_k \\
& +0.03819y_k^{2}+1.124y_k)(-0.02712 \\
& -0.7772y_{k-1}-0.09284y_{k-1}^{2}+0.1605y_{k-1}y_k \\
& +0.01809y_k^{2}+1.754y_k)-0.01840(-0.02712 \\
& -0.7772y_{k-1}-0.09284y_{k-1}^{2}+0.1605y_{k-1}y_k \\
& +0.01809y_k^{2}+1.754y_k)^{2}-0.4047y_{k-1} \\
& -0.04834y_{k-1}^{2}+0.08357y_{k-1}y_k
\end{aligned}
$$

The network for predicting the COD_f is shown in Figure 4 and is considerably more complex than the first case requiring three layers before acceptable fidelity is achieved. It is interesting to note that only a very small fraction of the possible branch combinations are used here. The relationship between the current and past inputs and the predicted value is far too complex to present and its inclusion here would serve little purpose.

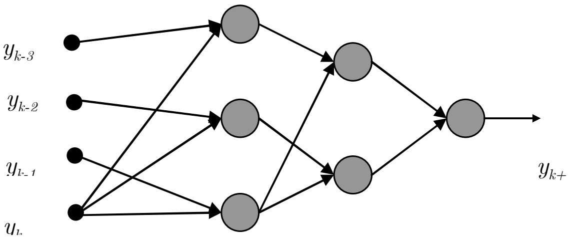


Fig. 4. Self-organizing network for one step ahead prediction of *CODf*.

IV. Conclusions

Prediction using time-series analysis of complex physical processes is a resort that is taken when all other methods of simulation have failed. The approach followed in this paper, that of inductive learning using self-organizing networks has proved useful in prediction in a variety of fields. In this paper it has been used to predict the outputs of the pre-precipitation process in a wastewater treatment plant a process that has proved too difficult to model using first principles. Following an extensive study, deductive self-organizing networks were found to yield improved results over conventional adaptive neural networks.

References

[1] Olsson G. and B. Newell, *Wastewater Treatment Systems,* IWA Publishing, London, 1999.

[2] Jang J.-S.R., C.-T. Sun, and E. Mitzutani, *Neuro-Fuzzy and Soft Computing – a Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, NJ, 1997.

[3] Kosko B., *Neural Networks and Fuzzy Systems,*Prentice Hall, NJ, 1992.

[4] Norgaard M., O. Ravn, N. K. Paulsen and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems,* Springer-Verlag, London, 2000.

[5] Nilsson N. J., *Learning Machines,* McGrraw Hill, N.Y. 1965.

[6] Madala H. R. and A. G. Ivakhnenko, *Inductive Learning Algorithms for Complex Systems Modeling,* CRC Press, Boca Raton, 1994.

[7] Pham D. T. and Liu X., “Modeling and Prediction using GMDH Networks of Adalines with Nonlinear Processors”, Int. J. Systems Science, Vol. 25, No. 11, pp. 1743-1759, 1994.

[8] Pham D. T. and L. Xing, *Neural Networks for Identification, Prediction and Contro,,* Springer-Verlag, N.Y., 1997.

[9] Farlow S. J. *et al., Self-organizing methods in Modeling GMDH-type Algorithms,* Marcel Decker, N.Y. 1984.

[10] Gilles G., Takacs P. and Takacs I., “Simulation: A key component in the development of an integrated computer-based approach to waste water treatment plant control”, *Proc. IEE Conference on Control Applications,* Glasgow, pp. 1023-1028, 1994.

[11] Rosen-Zvi M., I. Kanter and W. Kinzel, “Time series prediction by feed-forward networks”, J. of Physics A: Math. Gen., Vol. 36, pp. 4543-4550, 2003.

[12] Mueller J-A. and Lemke F., “Self-Organizing Data Mining”, 1999, www.knowledgeminer.net

[13] Cherkassky V. and F. Muiler, “Learning from Data”, J. Wiley & Sons, N.Y. 1998.

[14] Mueller, J.-A., F. Lemke and A. G. Ivakhnenko: “GMDH algorithm for complex systems modeling”, Mathematical Modeling of Systems, 1997.