# Carbon monoxide prediction using novel intelligent network

**[1]M. Abbaspour, [2*]A. M. Rahmani and [3]M. Teshnehlab**

[1]*Department of Mechanical Engineering, Sharif University of Technology, Tehran, Iran*

[2]*Department of Computer Engineering, Science and Research Campus, Islamic Azad University, Tehran, Iran*

[3]*Department of Electronic and Electrical Engineering, Khaje Nasir Toosi University of Technology, Tehran, Iran*

## Abstract

This paper introduces a new structure in neural networks called TD-CMAC, an extension to the conventional Cerebellar Model Arithmetic Computer (CMAC), having reasonable ability in time series prediction. TD-CMAC, the conventional CMAC and a classical neural network model called Multi-Layer Perceptron (MLP) are simulated and evaluated for 1-hour-ahead prediction and 24-hour-ahead prediction of carbon monoxide as one of primary air pollutants. Carbon monoxide data used in this evaluation were recorded and averaged at Villa station in Tehran, Iran from October 3rd. 2001 to March 14[th.] 2002 at one-hour intervals. The results show that the errors made by TD-CMAC is fewer than those made by other models.

**Key words:** *Carbon monoxide, cerebellar model arithmetic computer (CMAC), time series, time delay CMAC (TD-CMAC), carbon monoxide prediction*

*Corresponding Author, E-mail: rahmani74@yahoo.com*

## Introduction

In this changing world, parameters such as air temperature, pressure, pollution, population, traffic, and... are undergoing continuous changes as well. These time altering parameters are known as time series (Shumway and Stoffer, 2000). Time series prediction takes an existing series of data X (t), X (t-1)… X (t-d) and forecasts the values X (t+1), X (t+2)… The goal is to model existing data series to enable accurate prediction of unknown data values.Until now, several methods for prediction of air pollutants (time series data) have been designed. One method for modeling air pollution equations is the analytical method, which is complicated, difficult and needs considerable experience and high expertise (Haase and Schlink, 2001), (Pollack and Stocking, 1989). Another method is applying intelligent systems such as neural networks, fuzzy systems, etc. This method simulates the predictive model based on the observed time series data (Boznar, 1997 and de Castro, *et al.,* 2003).

Numerous researches and studies have been carried out on the subject of nature and dynamic behavior of pollutants, emission, propagation and effects of pollutants. Predicting future dispersion of air pollution is of immense importance since it can provide an effective decision making tool by giving advance warning of excessive pollution beyond the threshold. Also, it enables early air quality control to mitigate the adverse impacts. on the other hand, high dynamism and nonlinear behavior of air pollutant data makes prediction difficult or inaccurate. High capabilities of artificial neural networks e.g. flexible structure and the use of dynamic learning algorithm promote the application of these intelligent systems. The behavioral monitoring of pollutants and modeling the possible through the utilization of a memorizing neural network. Therefore, temporal dimension must be main tined in its structure. In other words, the network must have a dynamic behavior to keep output data continually dependant to present and past inputs.

The aim of this paper is to present a new structure based on neural networks e.g. cerebellar model. This model can be used as the black box approach for prediction of unknown time series data. The model can be modified to predict future behavior of air pollutant density (carbon monoxide) by using the current and past values of observed and collected data.

## Materials and Methods
### Conventional CMAC model

The conventional CMAC was model introduced by Albus (Albus, 1972) for the first time. CMAC is a neural network, which models the structure and operation of the section, part of the brain. The main

function of this part is to the movements of coordinate and control of eyes, hands, fingers, arms, and legs. At first, CMAC model was used in order to control robot arms (Albus, 1975). The model is based on associative memory, which uses a lookup-table technique. The block diagram of CMAC model is shown in Figure 1.

The input variables are divided into distinct states specifying memory cell addresses. Several memory cells (associative memories) are applied to store the data. The output is obtained by summing retrieved data in associative memories. CMAC acknowledges its correct output by changing the contents of memory cells.

The CMAC model uses the following three mapping relations:

• Quantized input state to Associative memory cells

$$S \rightarrow A$$

• Associative memory cells to Physical memory addresses $\qquad A \rightarrow P$

• Physical memory addresses to CMAC output vector $\qquad P \rightarrow Y$

Figure 2 illustrates an example of CMAC structure with two $S_1$ and $S_2$ input variables. Each input variable is divided into a number of discrete sections called blocks. In this Figure, these sections
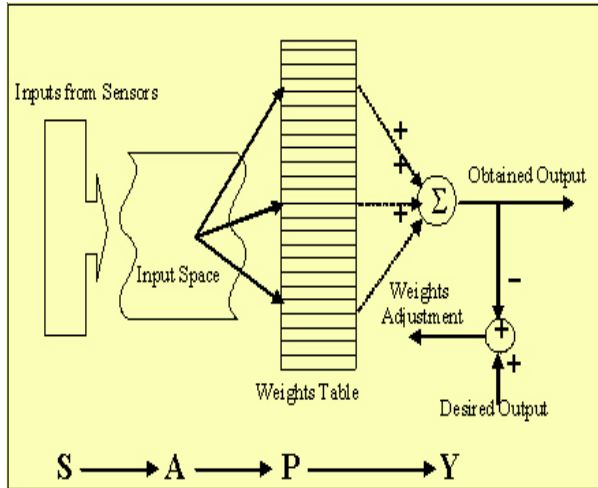


Figure 1: Block diagram of CMAC

are shown as A, B and C for $S_1$ and a, b and c for $S_2$. Several blocks in input state form a space called hypercubes. These spaces are named Bb, Aa, Ac and Ca… Each hypercube is a memory cell, which stores and retrieves data. By shifting each block a small distance (called an element), different blocks are created in different layers. For example: for A, B and C of $S_1$ input variable, D, E and F were in the next layer and G, H and I in the last layer. Hence, the hypercubes Bb, Fe, Hh, Gi … are created. In

conventional CMAC, only blocks in similar layers will create hypercubes.

In other words, the available parameters in CMAC network are defined as follows:

$N_v$: Number of input variables (here two),

$N_b$: Number of blocks in one layer (here three),

$N_e$: Number of layers or number of complete elements occupying one block (here three),

S: Input vector, e.g. State $(S_1, S_2) = (4, 3)$,

A: Associative cells or hypercubes, e.g. Bb, Fe, Hh,

P: Physical address of $N_e$ hypercubes of set A,

Y: Related output to S, or stored data in those hypercubes, which cover S.

The output value is equal to the sum of stored data (weights) in associative cell of $N_e$ hypercubes of set A. The outputs are calculated by the following equation:

$$y(s) = a^T(s)W = \sum_{j=1}^{M} a_j(s)w_j \qquad (1)$$

Where, s is the input vector. M is memory size equal to the number of hypercubes ( $M = N_e \times N_b^2$ ). W is weight vector or stored data in memories and if the memory location of j is covered by one of the hypercubes then $a_j(s) = 1$, otherwise, $= 0$.
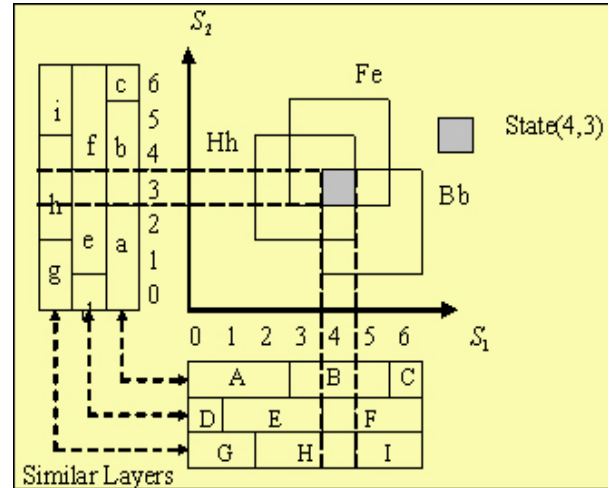


Figure 2: The CMAC structure with two inputs

The CMAC model uses iterative supervised learning algorithm based on global error minimization to adjust the stored weights in hypercubes. It has two steps: 1- forward step: applying inputs and obtaining outputs, 2- backward step: adjusting weights by comparing the obtained outputs with the desired out comes. The first step is carried out by equation (1), which calculates the output mapping. The second step is performed by the following equation:

$$W_{new} = W_{old} + \frac{\alpha}{N_e} a(s)(\hat{y}(s) - a^T(s)W_{old}) \qquad (2)$$

Where $\alpha$ is the learning rate, $\hat{y}(s)$ is the desired output, $N_e$ is the number of elements and $\hat{y}(s) - a^T(s)W_{old}$ is the amount of error at the backward step. Then, the amount of error is multiplied in $a(s)$ in order to adjust the weights that are covered by hypercubes. Generally, learning algorithms have two main capabilities:

1. Generalization: In regard to input data, the network has the capability of adjusting memory weights. It has the ability to learn and experience new conditions. So that, by applying new inputs not experienced before, it can produces proper outputs. The CMAC model can divide the input space into suitable blocks and layers to reach good generalization ability (Gonzalez-Serrano, 1998).

2. Identification capability: By means of the Kolmogorov Theory, it could be proven that the CMAC model can be used as universal approximator of non-linear mappings or as a general class for approximation of any type of functional models [Cotter and Guillerm, 1992] such as the time series prediction.

**TD-CMAC (Time delay CMAC)**

TW-CMAC (Time window CMAC) (Sarmadi and Teshnehlab, 2002) was used to memorize the conventional CMAC model for time series prediction. In this method, both present and previous values of time series (X (t), X (t-1), X (t-2), …, X (t-d)) were used as input variables of the CMAC model in order to predict its output (X (t+1)), which is 1-step-ahead prediction.

**TD-CMAC structure**

In this paper, the new structure based on CMAC (called TD-CMAC) is used for time series prediction. The block diagram of new model and its structure details are shown in Figure 3 and Figure 4.In TD-CMAC model, the output value or X (t+1) depends both on the present value (X (t)) and on the previous values or X (t-1), ..., X (t-d). Therefore:

$$X(t+1) = F_0\big(X(t)\big) + F_1\big(X(t-1)\big) + ... + F_d\big(X(t-d)\big) \qquad (3)$$

TD-CMAC structure consists of d (d+1)/2 number of two input conventional CMACs which d is the number of delays. TD-CMAC structure shown in Figure 4, block $B_1$ consists of d number of two-input CMACs. The first CMAC holds X (t) and X (t-1) as inputs and the second CMAC uses X (t) and X (t-2) as inputs. Finally, the last CMAC has two input variables of X (t) and X (t-d).

Again, block $B_2$ consists of d-1 number of two-input CMACs. Like block $B_1$, the first CMAC holds X (t-1) and X (t-2) as inputs, the second CMAC uses X (t-1) and X(t-3) as inputs and finally, the last CMAC has two inputs as X (t-1) and X (t-d).
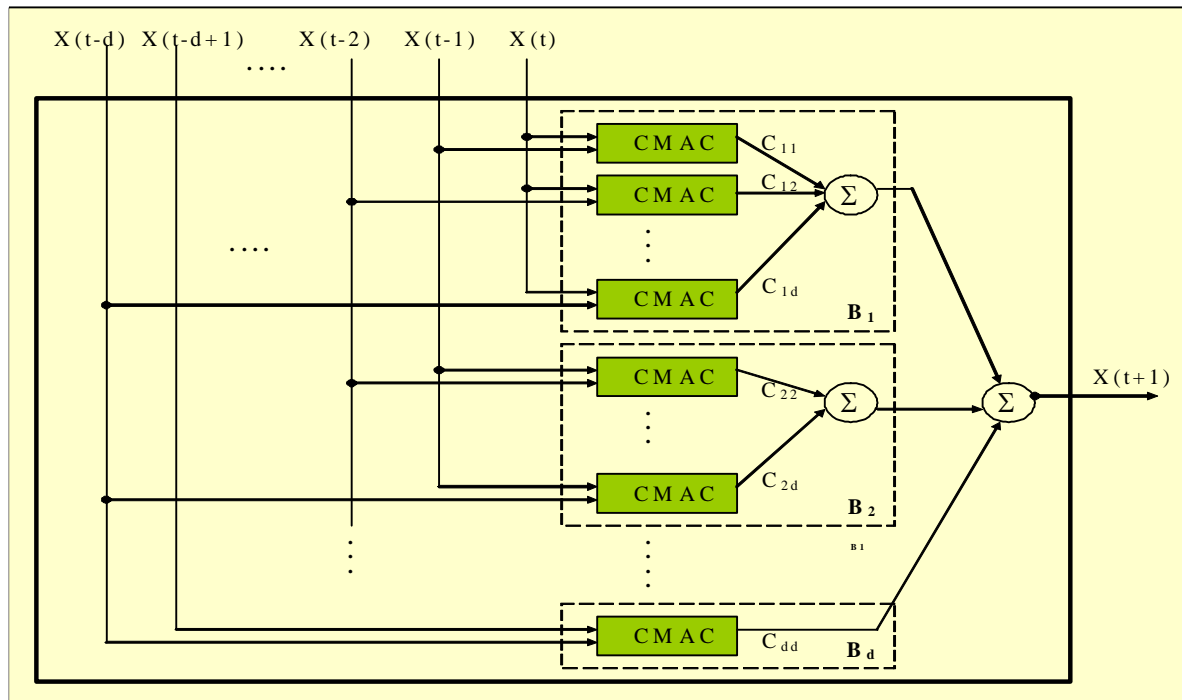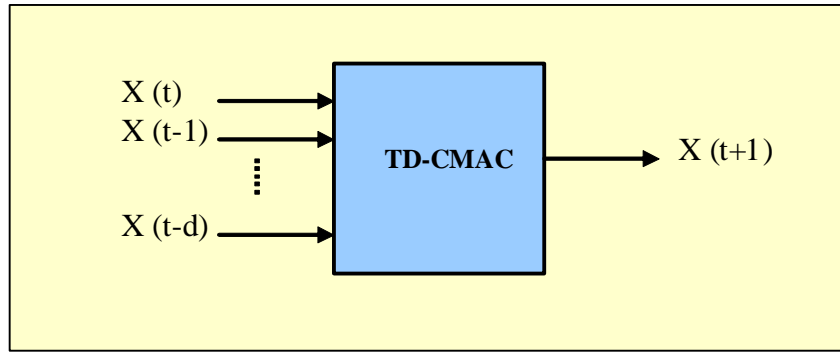


Figure 3: The structure details of TD-CMAC model

Figure 4: Block diagram of TD-CMAC model

Moreover, the last block shown in Figure 4 or $B_d$ consists of X (t-d+1) and X (t-d) as inputs.

**Output mapping**
According to the following equation, X (t+1) or the output value which is used for prediction of time series is equal to the sum of the CMAC outputs each multiplied by a factor:

$$y(s) = c(s)a(s)W = \sum_{i=1}^{d}\sum_{j=i}^{d} c_{ij}(s) \sum_{k=1}^{M} a_{ijk}(s)w_{ijk} \quad (4)$$

Where,

$$c_{ij}(s) = \frac{1}{ij} \quad (5)$$

In the above equation, S is the determined input vector, W is the weight vector, and d is the number of delays. Each CMAC has two inputs, namely X (t-i+1) and X (t-j) for (i=1, 2, .., d) and (j=i, i+1, …, d). M is the number of hypercubes ($M = N_e \times N_b^2$). Each hypercube is identified by a variable like ijk, where k=1, 2, … M. If the memory locations of ijk were covered by one of the hypercubes of S then $a_{ijk}(s) = 1$, otherwise=0. At last, in time series prediction, predicted values depend more on recent data than on the old data. Therefore, factor c (s) is applied to reduce the effect of old data in the output values.

**Learning algorithm**
The learning algorithm in TD-CMAC is similar to the conventional CMAC. It has two steps: 1- forward step: applying inputs and obtaining outputs, 2- backward step: adjusting weights by comparing the obtained outputs with the desired outcomes:

$$W_{new} = W_{old} + \frac{\alpha}{N_e} c(s)a(s)(\hat{y}(s) - y(s)) \quad (6)$$

Where, $y(s)$ is the obtained output from equation (4). $\hat{y}(s)$ is the desired output, $N_e$ is the number of layers, $\hat{y}(s) - y(s)$ is the amount of error at the backward step. The amount of error is multiplied by $a(s)$ in order to adjust the weights that are covered by hypercubes. c(s) is the forgetting factor, which reduces the effect of old data in weighting time series prediction.

**Results**
As a reference for comparison and evaluation of the TD-CMAC model, the conventional CMAC (TW-CMAC) and a multi-layer perceptron (MLP) neural network are selected. It has been proved that a MLP, which has only one hidden layer with sufficient number of neurons, can act as universal approximator for non-linear mapping (Hornik, *et. al.,* 1989). The TD-CMAC model and TW-CMAC model are written in C++ language. A MLP with one hidden layer with hyperbolic tangent hidden units and linear output units based on Gradient descent with adaptive learning rate back-propagation algorithm is implemented in Matlab neural network toolbox. The simulations were conducted for 1-hour-ahead prediction and for 24-hour-ahead prediction (one-day-ahead) of air pollutants time series to compare the prediction capability of different models. Time series data used in this evaluation were taken from CO values recorded and averaged at Villa station in Tehran, Iran from October 3rd. 2001 to March 14th. 2002 at one-hour intervals (3912 samples). In these simulations, the models, learning algorithms, output mapping and inputs are the same, but the outputs are different. The output for 1-hour-ahead prediction is X (t+1) and for 24-hour-ahead prediction is X (t+24). First, the program was executed for CO pollutant by four-input TD-CMAC with three delays, 14 blocks, and 6 layers, for the 3912 observed and recorded data of CO. The first 3744 collected data will be used to train the model, and the next 168 accumulated data (one-week) will be used to evaluate and validate the proposed model.

Again, the program was executed by four-input TW-CMAC with three delays, 14 blocks, and 6 layers, for the same data. The observed data and the predicted data of 1-hour-ahead prediction of CO pollutant are shown in Figure 5 and Figure 6, more over the observed data and the predicted data of 24-hour-ahead prediction are shown in Figure 8 and Figure 9.In second attempt, the program was executed by a four-input MLP with three delays for the same collected data. It is acknowledged that selecting sufficient unit numbers (neurons) in the hidden layer is very critical and difficult in such networks. In practice, to achieve the desired degree of approximation, number of units in the hidden layer is determined by trial and error. In this evaluation, ten units were determined to be suitable for the time series data. The observed data and the predicted data of 1-hour-ahead prediction are shown in Figure 7 and the 24-hour-ahead predictions are shown in Figure 10.The following parameters were used to evaluate efficiency of different models. Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Correlation Coefficient (r) and Error Standard Deviation (s) are calculated as below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i(s) - y_i(s)\right| \qquad (7)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i(s) - y_i(s)\right)^2} \qquad (8)$$

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i(s) - \hat{\mu}\right)^2} \qquad (9)$$

$$r = \frac{Cov(\hat{y}_i(s), y_i(s))}{\sqrt{Cov(\hat{y}_i(s), y_i(s)) \cdot Cov(\hat{y}_i(s), y_i(s))}} \qquad (10)$$

Where $y(s)$ is the observed data, $\hat{y}(s)$ is the predicted data, $\hat{\mu}$ is the mean value of predicted data and n is the number of predicted data. $e_i(s) = \hat{y}_i(s) - y_i(s)$. $Cov(\cdot,\cdot)$ is covariance matrix. Table 1 shows the values for MAE, RMSE, s and r of TD-CMAC, conventional CMAC (TW-CMAC) and MLP models for 1-hour-ahead prediction. The same evaluation parameters and models for 24-hour-ahead prediction are shown in Table 2. As it is observed, the new TD-CMAC model has better prediction capability.

In addition, comparing the two Tables indicates that the 1-hour-ahead prediction provides more accurate and precise result than 24-hour-ahead prediction, because of high dynamic and nonlinear behavior of CO pollutant in atmosphere. Although, long-term prediction provides acceptable results, the short-term prediction delivers better outcomes. On the other hand, as the number of delays increases, the size of memory decreases more in TD-CMAC model in comparison to TW-CMAC. If it is assumed that the number of blocks in each layer is $N_b$, the number of layers or elements as $N_e$ and the number of delays as d, then the used memory size of (d+1)-input- TW-CMAC model will be equal to $N_e \times N_b^{d+1}$ and the used memory size of (d+1)-input TD-CMAC model will be $\frac{d \times (d+1)}{2} \times N_e \times N_b^2$. Therefore, hardware implementation will be simpler and more feasible (Miller, *et al.,* 1990). Finally, the advantage of using the TD-CMAC model in comparison to MLP is its simplicity and ability to conduct online implementation due to existence of fast learning algorithms in confronting new data.

**Discussion and Conclusion**

So far, numerous researches and studies on indentification, modeling and prediction of air pollutants have been carried out and different methods and algorithms have been designed and implemented. One method is the analytical method, which is not user friendly and needs extensive experience and expertise. Another method is intelligent neural networks, which simulates the predictive model based on the observed data. The CMAC model is a kind of neural network, which shows superior ability to memorize and predict time series. In this paper, a new neural network structure known as TD-CMAC was introduced. It was used as a model for 1-hour-ahead prediction and 24-hour-ahead prediction of carbon monoxide. The efficiency evaluation revealed that under equal conditions TD-CMAC has better prediction capabilities than the conventional CMAC (TW-CMAC) and MLP neural network meanwhile the short-term prediction provides better result than the long-term prediction. On the other hand, the new model uses a decreased memory size with lower hardware implementation costs comparing to TW-CMAC. Finally the last advantage of using the TD-CMAC model is its simplicity and ability to conduct online implementation in comparison to MLP.

Table 1: MAE, RMSE, r and s values of different models for 1-hour-ahead prediction

| Model | MAE | RMSE | s | r |
|---|---|---|---|---|
| Four -input TD-CMAC | 1.717 | 2.398 | 3.318 | 0.783 |
| Four -input TW-CMAC | 2.750 | 3.982 | 2.056 | 0.393 |
| Four -input MLP | 2.128 | 3.293 | 5.341 | 0.781 |

Table 2: MAE, RMSE, r and s values of different models for 24-hour-ahead prediction

| Model | MAE | RMSE | s | r |
|---|---|---|---|---|
| Four -input TD-CMAC | 2.201 | 2.997 | 3.552 | 0.672 |
| Four -input TW-CMAC | 3.625 | 4.808 | 2.152 | 0.105 |
| Four -input MLP | 2.525 | 4.102 | 5.677 | 0.662 |
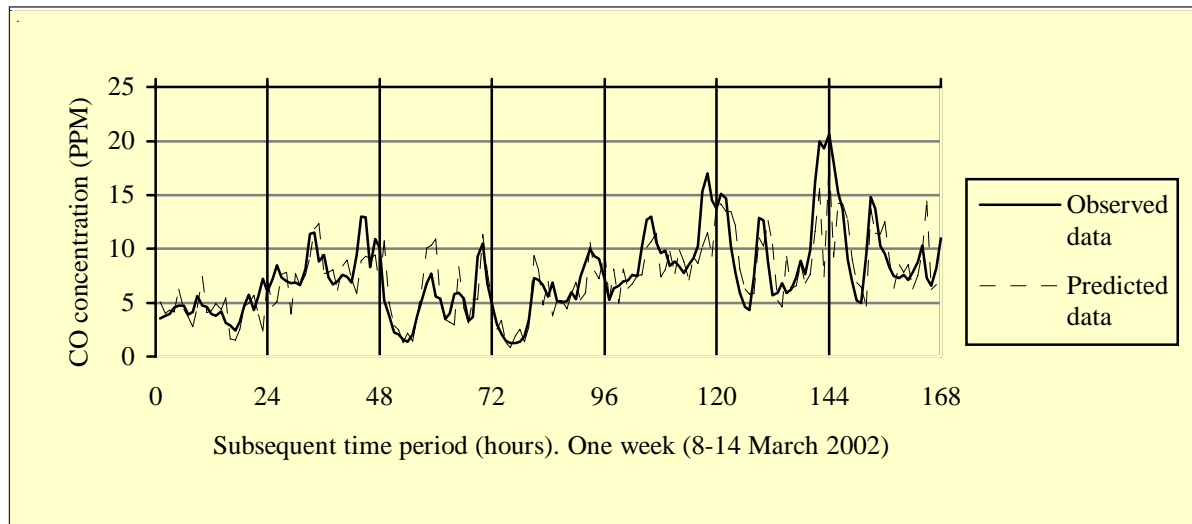


Figure 5: Four-input TD-CMAC. The observed data and the predicted data of 1-hour-ahead-prediction of CO
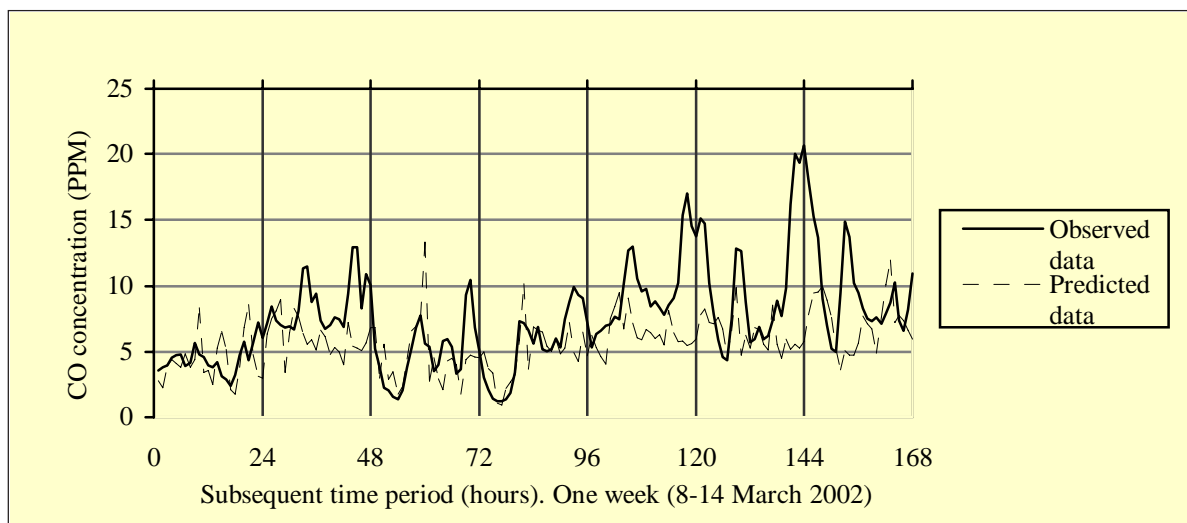


Figure 6: Four-input TW-CMAC. The observed data and the predicted data of 1-hour-ahead-prediction of CO
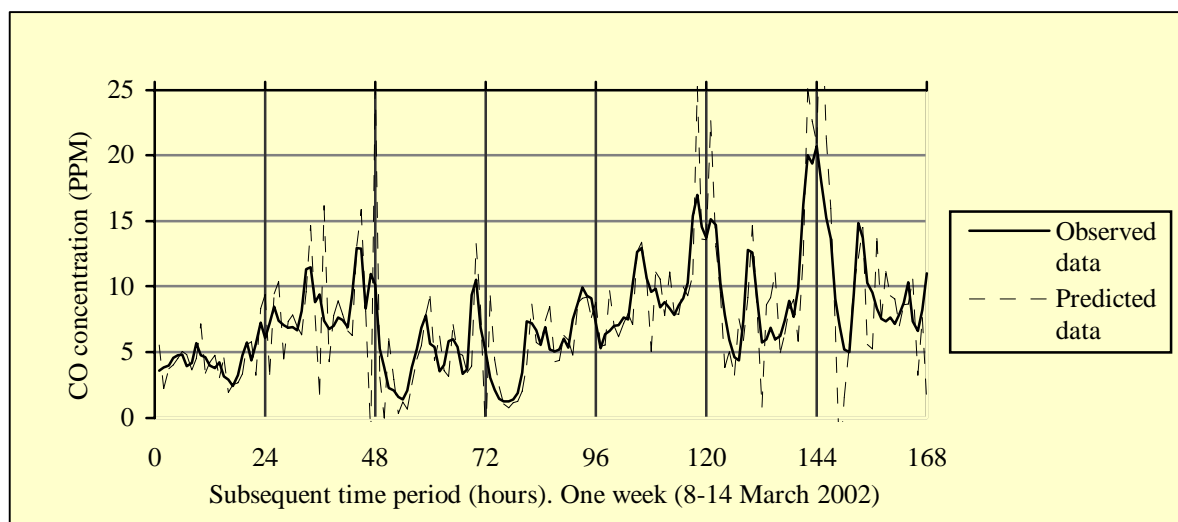
Figure 7: Four-input MLP. The observed and the predicted data of 1-hour-ahead-prediction of CO pollutant
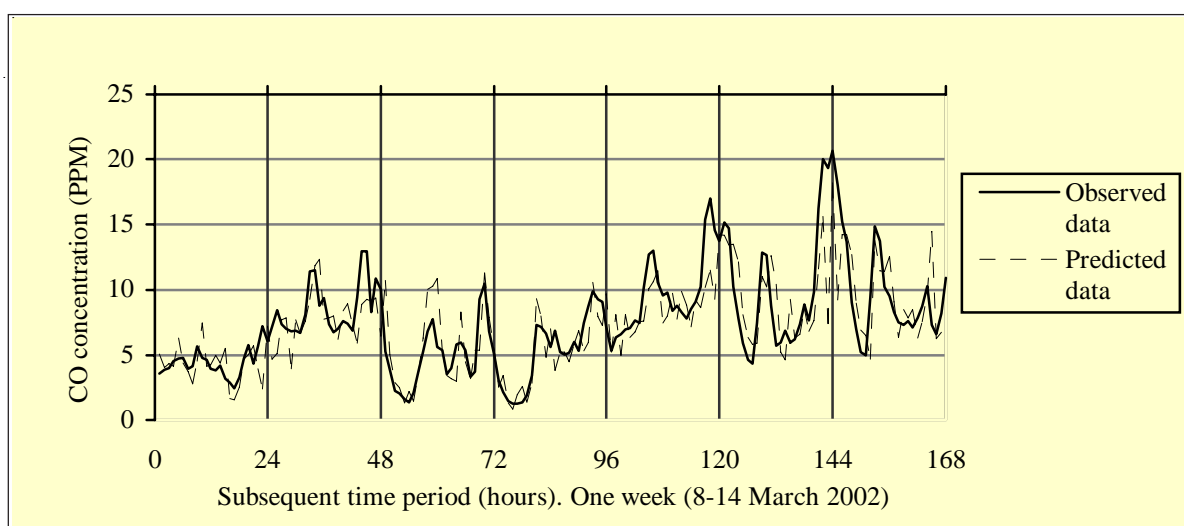


Figure 8: Four-input TD-CMAC. The observed data and the predicted data of 24-hour-ahead-prediction of CO
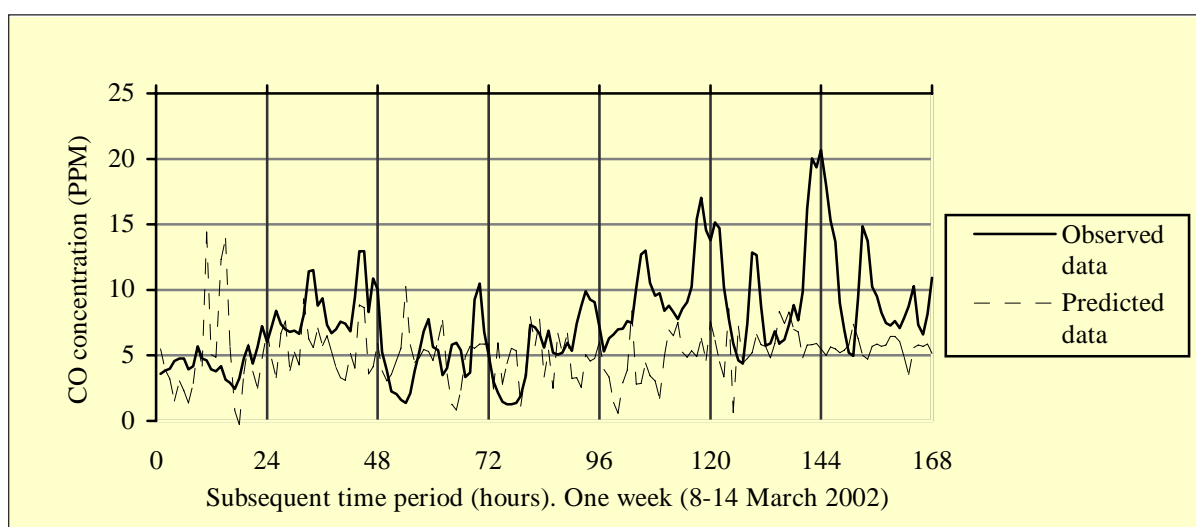


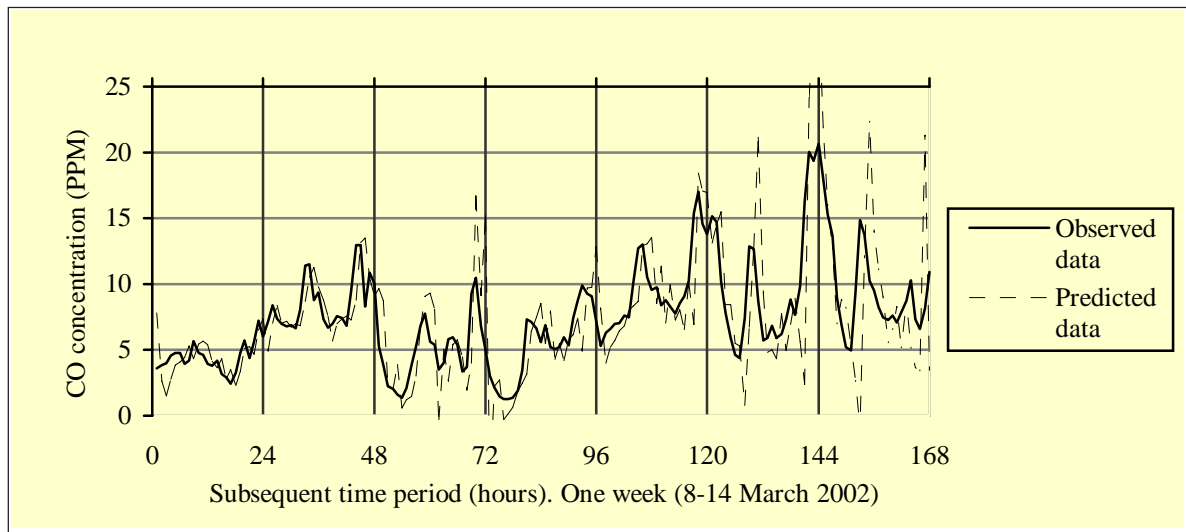Figure 9: Four-input TW-CMAC. The observed data and the predicted data of 24-hour-ahead-prediction of CO

Figure 10: Four-input MLP. The observed and the predicted data of 24-hour-ahead-prediction of CO pollutant

## References

Albus, J. S., *A new approach to manipulator control: The cerebellar model articulation controller (CMAC). Trans. of the ASME. Journal of dynamic systems, measurement and control: 220-227, 1975*

Albus, J. S., *Theoretical and experimental aspects of a cerebellar model. University of Maryland, 1972*

Boznar, M., *Pattern selection strategies for a neural network-based short term air pollution prediction model. Proceedings of intelligent information systems, IIS'97, IEEE Computer society: 340-345, 1997*

Cotter, N. E., T. J. Guillerm, *The CMAC and a theorem of Kolmogorov, Neural networks, 1(5): 221-228, 1992*

Ferna´ndez de Castro, B. M., J. M. Prada-Sa´nchez, W. Gonza´lez-Manteiga and M. F. Bande, *Prediction of SO₂ levels using neural networks. Journal of the air and waste management association, 53(5): 532-539, 2003*

Gonzalez-Serrano, F. J., A. R. Figueiras-Vidal and A. Artes-Rodriguez, *Fourier analysis of the generalized CMAC neural network. neural Networks, 1(11): 391-396, 1998*

Haase, P. and U. Schlink, *Non-parametric short-term prediction of ozone concentration in Berlin. Paper presented at the air pollution modeling and simulation conference, Paris, 2001*

Hornik, K., M. Stinchcombe and H. White, *Multilayer feedforward networks are universal approximators. Neural networks, 2(5): 359-66, 1989*

Miller, W. T., B. A. Box and J. M. Glynn, *Design and implementation of high speed CMAC neural network using programmable CMOS logic cell arrays. IEEE conference on neural information processing systems-natural and synthetic, 1990*

Pollack, A. K. and T. S. Stocking, *General linear models approach to estimating national air quality trends. Systems applications. Inc., San Rafael, California (SYSAPP-89/098), 1989*

Sarmadi, N. and M. Teshnehlab, *Short-term weather forecasting using CMAC Structure, 20ᵗʰ IASTED International conference applied informatics ,Innsbruck, Austria: 284-351, 2002*

Shumway, R. H. and D. S. Stoffer, *Times series analysis and its applications. Springer-Verlag, 2000*