J. ENVIRONMENTAL SYSTEMS, Vol. 30(3) 191-205, 2003-2004

AIR QUALITY PREDICTION: AN OPPORTUNISTIC NEURO-ENSEMBLE APPROACH

A. K. DIKSHIT
Indian Institute of Technology Bombay, India

S. V. BARAI
Indian Institute of Technology Kharagpur, India

SAMEER SHARMA
Tata Consulting Engineers Ltd., New Delhi, India

ABSTRACT

The present article discusses the development of neural-network-based air quality prediction models which can work with a limited number of data sets and are robust enough to handle data with noise. Five different variations of neural network models (partial recurrent network (PRNM), sequential network construction (SNCM), self-organizing feature maps (SOFM), moving window (MWM), and integrated normalized autoregressive moving average-self-organized feature maps models (NARMA-SOFM)), were implemented in a WINDOWS environment using MATLAB software. Developed models were run to simulate and forecast the daily average data for three parameters: RPM (respirable particulate matter), SO_2 (sulphur dioxide), and NO_2 (nitrogen dioxide) for the Ashram Chowk location in New Delhi, India. The implemented models were found to predict air quality patterns with modest accuracy. To improve the models' performance, an innovative approach using an opportunistic ensemble of the first four developed neural network models (OEM) was proposed for predicting the same short-term data. The ensemble approach indeed demonstrated an improvement on earlier models. However, the NARMA-SOFM model performed the best.

INTRODUCTION

The air-quality problems of a region may be characterized from two perspectives: the violations of air-quality standards and the public concern about air quality, effects on health, and degradation of visibility. For such purposes, local environmental protection agencies are required to make air pollution forecasts for public advisories, as well as provide input to decision makers on pollution abatement and air quality management issues [1].

Natural phenomena typically may be captured at a time series with some degree of randomness. Previous studies have demonstrated that the data of ambient air quality are also stochastic time series, thereby making it possible to make a short-term forecast on the basis of historical data [2]. Though models may be imperfect, they are the best tools for use in all aspects of air quality prediction.

Neural networks, which are massively parallel computational models, are based on the present understanding of the functioning of the biological brain. A neural network is an interconnected assembly of simple processing elements called nodes, which work similar to neurons. The processing ability of the network is stored in the inter-node connection strengths, also called weights, obtained by a process of adaptation to, or learning from, a set of training patterns. It is the learning capability of neural networks which have made them more popular than mathematically formulated models [3-5]. Particularly in environment-related areas, the neural networks have emerged as a more flexible, less assumption-dependent, and more adaptive methodology [6], and many researchers have illustrated their applications in rainfall runoff modeling, precipitation forecasting, stream flow forecasting, lake/reservoir/groundwater modeling, water management policy, reservoir operations, real-time control of water and wastewater treatment plants, water quality management, air quality management, adsorbent beds design, and hazardous waste management [7-13]. In the present study, air quality prediction models based on neural networks are developed to predict short-term air pollution.

Barai and Reich have shown that prediction models could be improved by the combination of multiple models into one through an approach called ensemble modeling [14]. There are two kinds of approaches in ensemble modeling: *opportunistic* and *principled.* The opportunistic approach emerges from the usual iterative data modeling process where, as various models are studied, the problem becomes better understood and the best model is selected while others are discarded. In opportunistic ensemble modeling, instead of discarding these models, they are combined into one. If the models are quite different in kind and reasonably good, the ensemble should improve upon the best of them. The principled ensemble approach seeks to generate systematically a set of models as accurate and diverse as possible from which a single model is composed. Typical models of this kind are described elsewhere [14].

The aim of the present study is to build the neural-networks-based air quality predictors, which can work with limited sized data sets and should be robust enough to handle data with noise. An attempt has been made to investigate the advantage of using an ensemble of the developed models for forecasting air pollution. The primary objectives of the study are as follows:

- To collect and analyze suitable data sets for multiple air quality parameters containing daily average pollutant concentrations at a specific location.
- To use various available variations of neural network models to predict short-term air quality.
- To conduct exhaustive neural network simulations using developed air quality prediction models and above-mentioned data.
- To compare air quality prediction model(s) with respect to daily (short-term) data.
- To develop and assess the performance of an ensemble model.

DEVELOPING AND IMPLEMENTING VARIOUS NEURAL NETWORKS-BASED AIR QUALITY PREDICTION MODELS

For the present study, the following neural network model architectures have been used for developing air quality prediction models [1] and have been implemented on a microcomputer in MATLAB software [15] using the Neural Networks Toolbox [16].

Recurrent Network Model (RNM)

For a neural network to be dynamic, it must be given memory. Memory may be divided into "short-term" and "long-term" memory depending upon the retention time. Long-term memory is built into a neural network through supervised learning, whereby the information content of the training data set is stored in the synaptic weights of the networks [5]. However, if the task at hand has a temporal dimension, we need some form of short-term memory to make the network dynamic. The static network accounts for non-linearity and the memory accounts for time. Short-term memory can be implemented in continuous time or in discrete time. Such networks typically use a variant of back-propagation for training. There are three ways that a "memory" can be introduced into static neural networks [17, 18]. These are (in increasing order of complexity and capability).

Tapped Delay Lines Models

In these models, the network has past inputs explicitly available (through a tapped delay line) to determine its response at a given point in time. Thus, the temporal pattern is converted to a spatial one, which can then be learned through, say, classic back-propagation [5].

Context Model or Partial Recurrent Models

These models retain the past output of nodes instead of retaining the past raw inputs. For example, the output of the hidden layer neurons of a feed forward network can be used as inputs to the network along with true inputs. These "network derived" inputs are also called context inputs. When the interconnections carrying the context inputs are fixed, classical back-propagation can be used for training the network.

Fully Recurrent Models

These models employ full feedback and interconnection between all nodes [5]. Algorithms to train fully recurrent models are significantly more complex in terms of time and storage requirements.

In the present study, the partial recurrent network model (PRNM) has been used. The algorithm and details are discussed elsewhere [5].

Sequential Network Construction Model (SNCM)

This model introduces an application of the "sequential network construction" to select the size of several popular neural network predictor architectures for various benchmark-training sets. The specific architecture considered here consists of a finite impulse response (FIR) network and the partial recurrent Elman network for adding context units to the output layer [19]. This model considers an enhancement of an FIR network in which only the weights having relevant time delays are utilized. Bias-variance trade off in relation to the prediction risk estimation by means of nonlinear cross validation is discussed elsewhere [20].

Self-Organizing Feature Maps Model (SOFM)

These networks are based on competitive learning, i.e., the output neurons of the network compete among themselves to be activated or fired, with the result that only one output neuron or one neuron per group is on at any one time [5]. An output neuron that wins the competition is called a winning neuron. One way of inducing a winner-takes-all competition among the output neurons is to use inhibitory connections (negative feedback paths) between them.

In a "self-organizing feature map," the neurons are placed at the nodes of a lattice that is usually one- or two-dimensional. The neurons become selectively tuned to various input patterns or a class of input patterns in the course of a competitive learning. The locations of the neurons so tuned become ordered with respect to each other in such a way that a meaningful coordinate system for different input features is created over the lattice. Hence, neurons in the lattice are indicative of intrinsic statistical features contained in the input patterns. The spatial location of an output neuron in a topographic map corresponds to a particular domain or feature of data drawn from input space [21].

The principal goal of the self-organizing feature map (SOFM) is to transform an incoming signal pattern of arbitrary dimension into a one- or two-dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion. Each neuron in the lattice is fully connected to all the source nodes in the input layer. This network represents a feed forward structure with a single computational layer consisting of neurons arranged in rows and columns. The algorithm responsible for the formation of the SOFM proceeds first by initializing the synaptic weights in the network. This can be done by assigning them small values picked from a random number generator. Once the network has been properly initialized, there are three essential processes involved in the formation of the SOFM, explained elsewhere [5, 6].

Moving Window Model (MWM)

This model is used for time series analysis of the given data sets. It utilizes the past values of the parameter which is to be modeled. The input values may vary from one to any desirable number, which gives lower error. This model can utilize both back-propagation and recurrent networks for training and simulation [5]. This model predicts one step ahead values of the parameter that is being modeled. For multi-step ahead prediction, predicted values are used as the feedback input again. The following are possible ways the data modeling was carried out:

- MWM2: with two values of the past input data
- MWM3: with three values of the past input data
- MWM4: with four values of the past input data
- MWM5: with five values of the past input data

In the present study, back-propagation has been used.

Opportunistic Ensemble Model (OEM)

To improve the performance, an opportunistic ensemble model has been developed which essentially combines the above mentioned models with equal weights as shown in Figure 1.

Integrated Nonlinear Auto Regressive Moving Average and Self-Organizing Feature Maps Model (NARMA-SOFM)

This model is applied, keeping in view the erratic nature of the data sets. A polynomial of suitable degree is fit into the data sets. This model uses the Statistical Toolbox of MATLAB [15]. After getting a polynomial of suitable

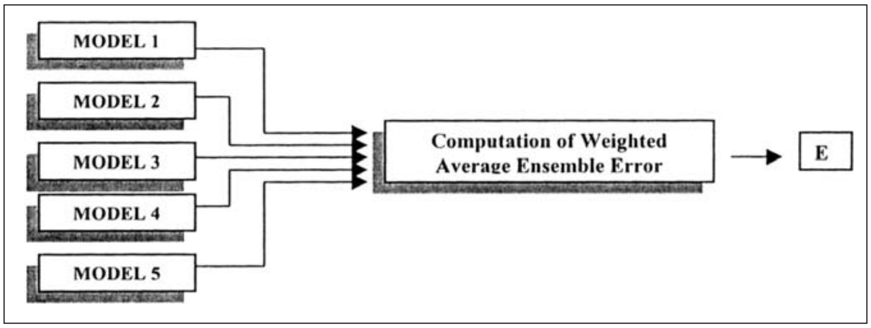

Figure 1. Opportunistic ensemble model.

degree, artificial data sets are generated. One set of this data is used for training and another set is used for testing. This model uses self-organizing feature maps. For multi-step prediction, the last predicted values are used as the feedback to the network again. This model uses the Polyfit, Polytool, and Plyval functions of Statistical Toolbox of MATLAB.

The algorithmic details of all above models and their performance capabilities has been discussed in Sharma [22]. All the models have been implemented in WINDOWS environment using MATLAB.

INPUT DATA AND MODELS APPLICATION

The data for three parameters (RPM (respirable particulate matter), SO_2 (sulphur dioxide), and NO_2 (nitrogen dioxide)) has been collected for nine locations in New Delhi, India from Tata Energy Research Institute Web site (www.teri.in). This dataset consists of daily average pollutant concentrations for the two years from July 2000 to August 2001. However, for the present study only the data for Ashram Chowk has been used for carrying out simulation studies [22]. The statistical properties of the data are given in Table 1.

The input data (w_t) to the network is the normalized as follows such that normalized values (I_t) fall between 0.1 and 0.9:

$$I_t = 0.5 - (W_{mean} - W_t) / (W_{max} - W_{min})*0.8$$

where W_{max}, W_{mean}, W_{min} are the maximum, mean, and minimum values in the dataset.

After modeling, the normalized output predictions (W_{pr}) produced by model are reverse-normalized using:

$$O_t = W_{mean} - (0.5 - W_{pr}) / 0.8*(W_{max} - W_{min})$$

where O_t is predicted value produced by the model.

Table 1. Statistical Properties of Input Dataset

Statistical property	Air quality parameter RPM	SO_2	NO_2
Mean	172.83	9.587	77.18
Std. deviation	119.88	3.7	31.59
Median	141	9	72

The parameters used for applying various models, (PRNM, CPDM, SNCM, SOFM, MWM, and NARMA-SOFM models) are presented in Tables 2 to 6, respectively. Out of 110 data points of the time series, initial 80 points of time series data were used for training the models. Remaining 30 points of time series data were used to predict air quality parameters.

The performance of each model is measured in terms of the mean percentage error (MPE) defined as follows for 5,000 typical iterations or for achieving specified sum squared error (SSE) by the network, whichever comes first:

$$\text{MPE} = (\text{target} - \text{output}) / \text{target}*100$$

RESULTS AND DISCUSSION

The performance of various models for different air quality parameters has been estimated in terms of mean percentage error (MPE) as shown in Table 7. The model having minimum MPE is expected to be the best model for forecasting. The present case study demonstrates an example of a daily average emission data prediction using various neural networks model for a reasonable size dataset (80 data points). Models, in general, have performed reasonably well even though data was chaotic by nature (Table 7).

To improve the model performance, the weighted average opportunistic ensemble model OEM (Figure 1) combining all the first four models has been developed and tested. The ensemble of model was developed using PRNM, SNCM, SOFM, MWM2, MWM3, MWM4, and MWM5 models. The results given in Table 7 show that ensemble model indeed improved the performance of neural networks over the other independently developed models. The typical results from the ensemble model for the predictions of RPM, SO_2, and NO_2 are shown in Figures 2 through 4. From the figures, it is clear that ensemble model predicts the trend reasonably close to the actual air quality data. A striking observation has been that though the ensemble model, in general, could predict with modest accuracy, NARMA-SOFM has performed extremely well in

Table 2. The Parameters Selected for PRNM Model

PRNM model parameter	RPM	SO_2	NO_2
NN architecture	1-10-10-1	1-10-10-1	1-10-10-1
Activation function	Tansig/purelin	Tansig/purelin	Tansig/purelin
Learning rate	0.04	0.04	0.04
No. of epochs	5000	10000	10000
SSE	0.1e-4	0.1e-4	0.1e-4
Input parameters	Day No.	Day No.	Day No.
Output parameters	RPM	SO_2	NO_2

Table 3. The Parameters Selected for SNCM Model

SNCM model parameter	RPM	SO_2	NO_2
NN architecture	1-3-3-1 to 1-8-8-1	1-3-3-1 to 1-8-8-1	1-3-3-1 to 1-8-8-1
Activation function	Tansig/purelin	Tansig/purelin	Tansig/purelin
Learning rate	0.03	0.03	0.03
No. of epochs	1000	1000	1000
SSE	0.1e-7	0.1e-7	0.1e-7
Input parameters	Day No.	Day No.	Day No.
Output parameters	RPM	SO_2	NO_2

comparison to all other models (Table 7). This performance could be attributed to an integrated approach of statistical model with neural networks model.

FUTURE PROJECTIONS

The studied models predict based on the limited history of air quality. However, model prediction can be improved by incorporating the following features:

- Models can have as inputs data from multiple sources, such as historical air quality measurements and meteorological data
- Models can have, along with emissions data, episode levels definition and historical measurements of surface and upper air meteorological data.
- Models should have an easy-to-use interface and should be able to present the results in an understandable way to non-computer experts.

Table 4. The Parameters Selected for SOFM Model

SOFM model parameter	RPM	SO_2	NO_2
NN architecture	1-5-1	1-5-1	1-5-1
Learning rate	1	1	1
No. of epochs	5000	5000	5000
Input parameters	RPM	SO_2	NO_2
Output parameters	RPM	SO_2	NO_2

Table 5. The Parameters Selected for MWM Model

MWM model parameter	RPM	SO_2	NO_2
NN architecture	1-12-12-1	1-12-12-1	1-12-12-1
Activation function	Logsig/Logsig	Logsig/Logsig	Logsig/Logsig
Learning rate	0.03	0.03	0.03
No. of epochs	20000	20000	20000
SSE	0.1e-6	0.1e-6	0.1e-6
Input parameters	RPM	SO_2	NO_2
Output parameters	RPM	SO_2	NO_2

Table 6. The Parameters Selected for NARMA-SOFM Model

NARMA-SOFM model parameter	RPM	SO_2	NO_2
NN architecture	1-5-1	1-5-1	1-5-1
Learning rate	1	1	1
No. of epochs	10000	10000	10000
Input parameters	RPM	SO_2	NO_2
Output parameters	RPM	SO_2	NO_2

Table 7. Neural Network Models Performance

S.N.	Air quality prediction mode	Mean percentage error		
		RPM	SO_2	NO_2
1	PRNM	56.76	48.63	43.5
2	SNCM	33.45	37.79	35.87
3	SOFM	25.6	30.73	28.94
4	MWM2	25.71	25.51	22.93
	MWM3	32.38	28.87	28.61
	MWM4	36.51	21.97	30.94
	MWM5	43.52	31.59	32.57
5	Opportunistic ensemble of PRNM+SNCM+SOFM+MWM2+MWM3+MWM4+MWM5	29.11	19.87	15.79
6	NARMA-SOFM	21.66	8.82	17.98

- The model parameters and architecture of models in this research work were arrived at through trial and error. One can arrive at optimal and better performance model after carrying out systematic studies on network models and their parameters using optimization techniques such as genetic algorithms.

CONCLUSIONS

In this study, air quality forecasts were made using several neural network models, concentrating on single variable(s)-based time series prediction from short-term daily average air quality data sets. The models were easily implemented by commercial software and could deliver fast predictions, unlike some other modeling techniques, and accommodated input noise well. An opportunistic ensemble modeling approach improved model performance further. The results indicate that neural networks are a practical tool for real-time air quality management and prediction.

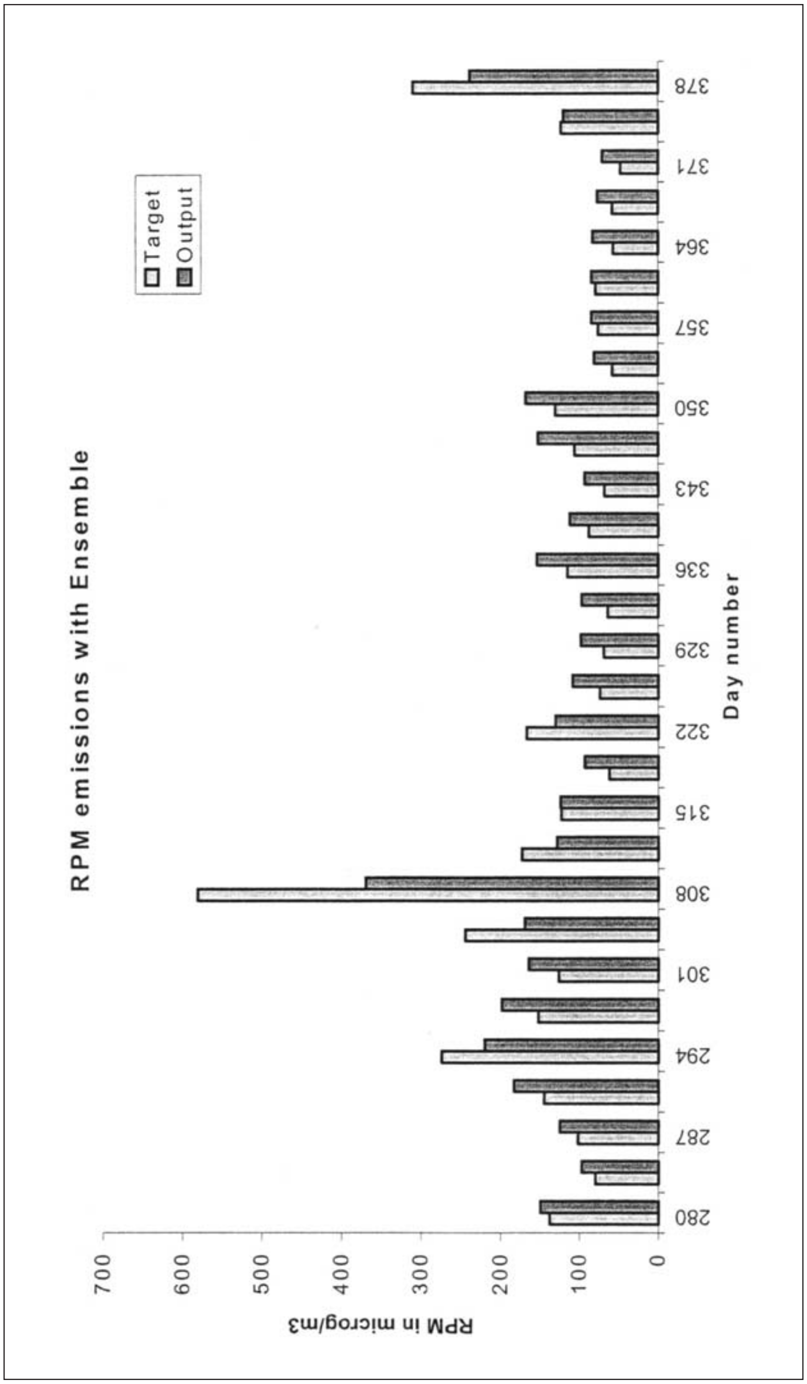

Figure 2. Ensemble model performance of RPM prediction.

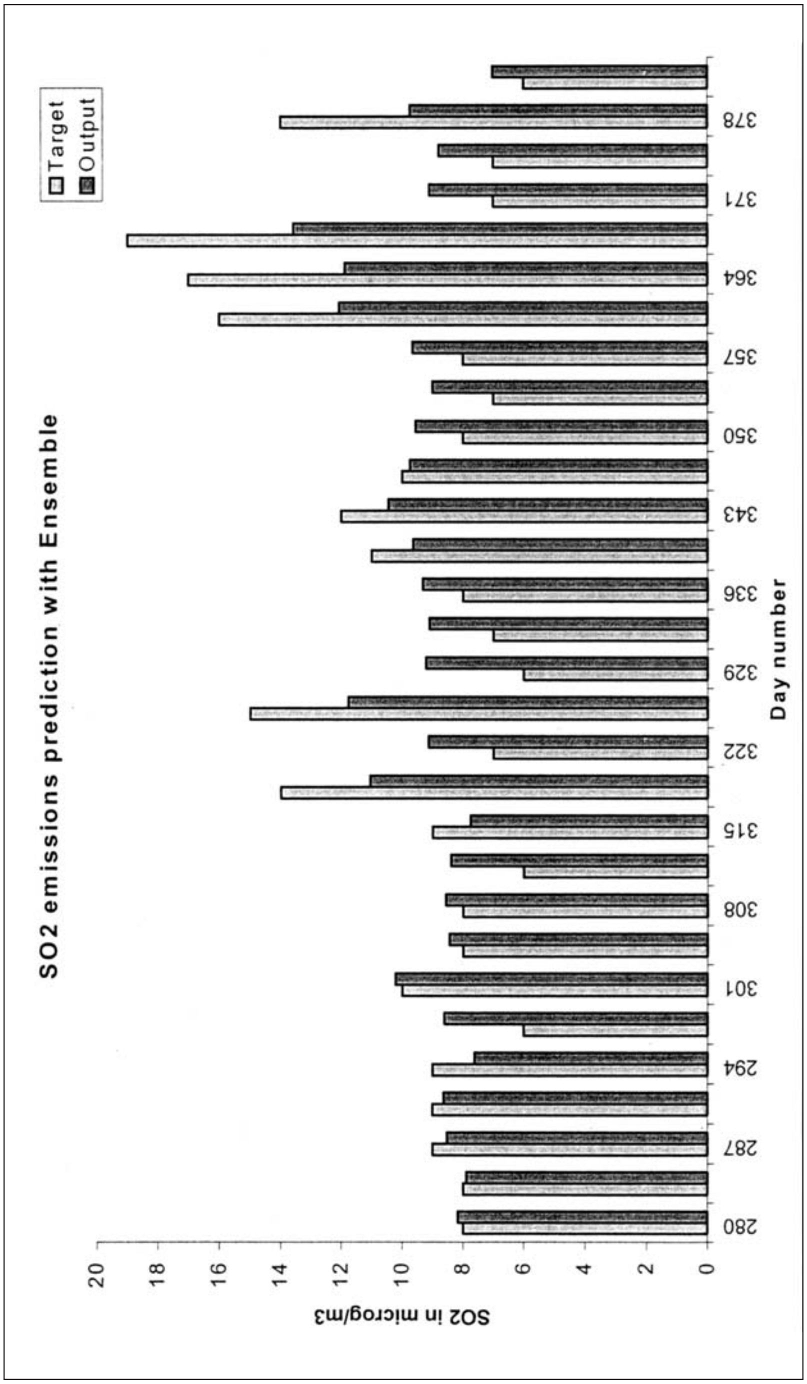

Figure 3. Ensemble model performance of SO_2 prediction.

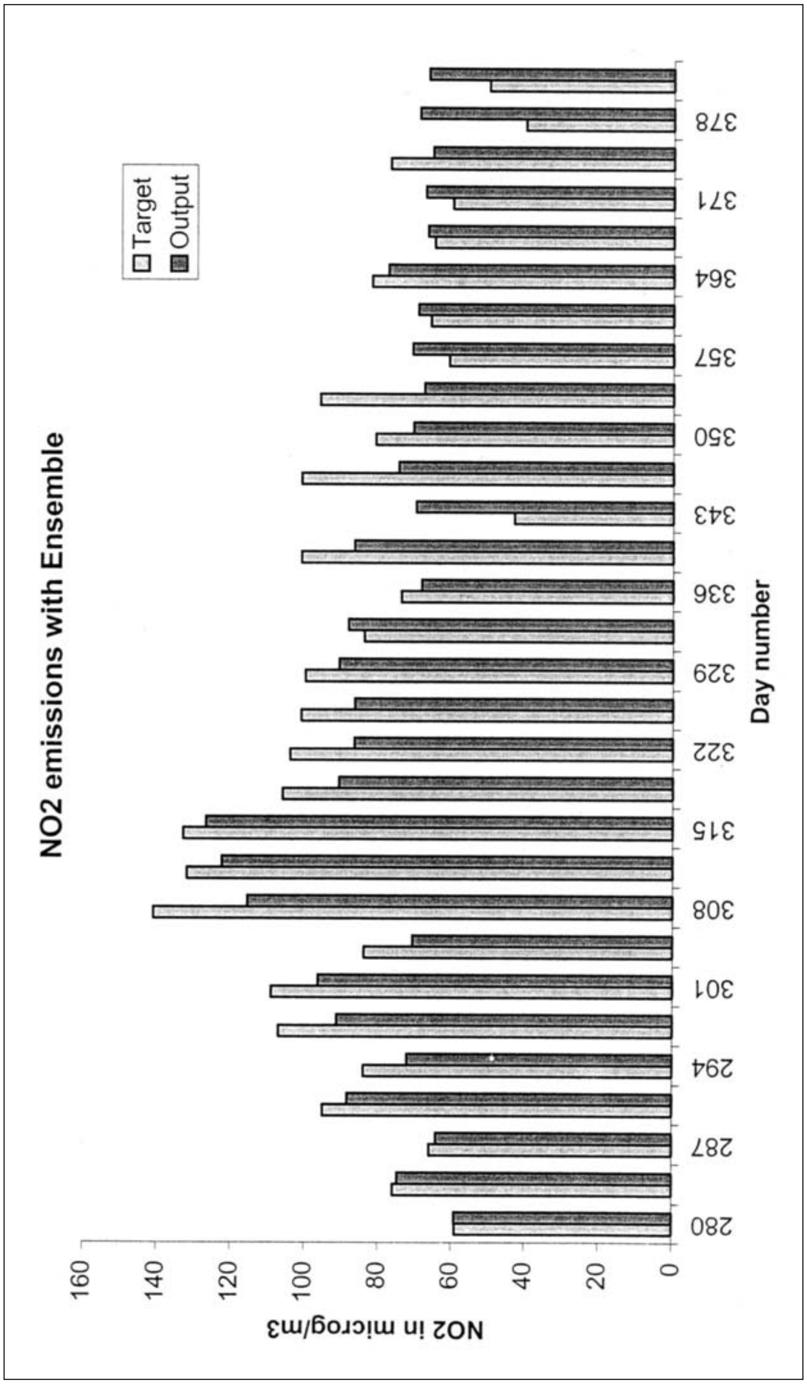

Figure 4. Ensemble model performance of NO_2 prediction.

REFERENCES

1. S. Sharma, A. K. Dikshit, and S. V. Barai, Studies on Neural Networks Based Air Quality Predictors, *International Journal of Environment and Pollution, 19*:5, pp. 1-12, 2003.
2. F. Giorgio and M. Piero, Mathematical Models for Planning and Controlling Air Quality, *Proceedings of an October 1979 IIASA Workshop,* Pergamon Press, 1982.
3. R. P. Lippman, An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine,* pp. 6-22, April 1987.
4. J. E. Dayoff, *Neural Networks Architecture: An Introduction,* Van Nostrand Reinhold, New York, 1990.
5. S. Haykin, *Neural Networks: A Comprehensive Foundation* (2nd Edition), Pearson Education Asia, Addison Wesely Longman, Delhi, 2000.
6. J. Principe and J. M. Kuo, Dynamic Modeling of Chaotic Time Series with Neural Networks, *Proceedings of Neural Information Processing Systems,* NIPS' 1994, Volume 7, pp. 311-318, MIT Press, Massachusetts, 1995.
7. M. Boznar, M. Lesjak, and P. Mlakar, A Neural Network-Based Method for Short Term Prediction of Ambient SO_2 Concentrations in Highly Polluted Industrial Areas of Complex Terrain, *Atmosphere and Environment, 27B,* pp. 221-230, 1993.
8. A. Basheer and M. Najjar, Predicting Dynamic Response of Adsorption Columns with Neural Nets, *Journal of Computing in Civil Engineering, ASCE, 10*:1, pp. 211-220, 1996.
9. ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, Artificial Neural Networks in Hydrology. I: Preliminary Concepts, *Journal of Hydrologic Engineering, ASCE, 5*:2, pp. 115-123, 2000.
10. ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, Artificial Neural Networks in Hydrology. II: Preliminary Concepts, *Journal of Hydrologic Engineering, ASCE, 5*:2, pp. 124-137, 2000.
11. L. Hadjiiski and P. Hopke, Application of Artificial Neural Networks to Modeling and Prediction of Ambient Ozone Concentrations, *Journal of Air and Waste Management Association, 50,* pp. 894-901, 2000.
12. K. Thirumalaiah and M. C. Deo, River Stage Forecasting Using Artificial Neural Networks, *Journal of Hydrologic Engineering, ASCE, 3*:1, pp. 26-32, 1998.
13. K. Thirumalaiah and M. C. Deo, Hydrological Forecasting Using Neural Networks, *Journal of Hydrologic Engineering, ASCE, 5*:2, pp. 180-189, 2000.
14. S. V. Barai and Y. Reich, Ensemble Modeling or Selecting the Best Model: Many Could be Better than One, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 13,* pp. 377-386, 1999.
15. Math Works, *MATLAB User's Guide,* Math Works Inc., Massachusetts, 2000.
16. H. Demuth and M. Beale, *Neural Networks Toolbox Manual,* Math Works Inc., Massachusetts, 1992.
17. J. T. Connor, R. D. Martin, and L. E. Atlas, Recurrent Neural Networks and Robust Time Series Prediction, *IEEE Trans. Neural Networks, 5*:2, pp. 240-253, 1994.
18. A. G. Parlos, O. T. Rais, and A. F. Atiya, Multi-Step-Ahead Prediction Using Dynamic Recurrent Neural Networks, *Neural Networks, 13*, pp. 765-786, 2000.
19. A. D. Back and A. C. Tsoi, FIR and IIR Synapses, a New Neural Network Architecture for Time Series Modeling, *Neural Computation, 3,* pp. 375-385, 1991.

20. J. C. Tomasz and J. M. Zurada, Sequential Network Construction for Time Series Predictions, in *Proceedings of the IEEE International Joint Conference on Neural Networks,* Houston, Texas, pp. 2034-2039, June 9-12, 1977.
21. T. Kohonen, The Self-Organizing Map, *Proceedings of the IEEE, 78,* pp. 1464-1480, 1990.
22. S. Sharma, *Neuro Air Quality Predictors,* M.Tech. thesis, Department of Civil Engineering, IIT Kharagpur, 2002.

Direct reprint request to:

Dr. A. K. Dikshit
Associate Professor
Center for Environmental Science and Engineering
Indian Institute of Technology Bombay
Powai, Mumbai 400 076, India
e-mail: dikshit@cese.iitb.ac.in