ELSEVIER

Environmental Modelling & Software 17 (2002) 161–168

Environmental Modelling & Software

www.elsevier.com/locate/envsoft

Prediction of sulphur dioxide concentration using artificial neural networks

Asha B. Chelani *, C.V. Chalapati Rao, K.M. Phadke, M.Z. Hasan

Air Pollution Control Division, National Environmental Engineering Research Institute (NEERI), Nehru Marg, Nagpur, India

Received 4 January 2000; received in revised form 10 August 2000; accepted 16 June 2001

Abstract

A three-layer neural network model with a hidden recurrent layer is used to predict sulphur dioxide concentration and the predicted values are compared with the measured concentrations at three sites in Delhi. The Levenberg–Marquardt algorithm is used to train the network. The neural network is used to simulate the behaviour of the system. A multivariate regression model is also used for comparison with the results obtained by using the neural network model. The study results indicate that the neural network is able to give better predictions with less residual mean square error than those given by multivariate regression models.

Keywords: Neural network; Recurrent network; Multivariate regression model; Levenberg–Marquardt; SO_2 air pollution

1. Introduction

In most of the studies conducted on atmospheric dispersion of pollutants, wide-ranging prediction techniques, including gaussian models and numerical models are generally used. The primary inputs to dispersion models include emission inventory, meteorological data and receptor locations. The major output from these models is the predicted ground-level concentration at specified receptor locations. The models are mainly based on the mathematical formulation of the physics and chemistry of the atmosphere, which govern the dispersion of pollutants. Gaussian models can be applied to any situation where the distribution pattern of the pollutants in that direction can be represented by a gaussian or normal distribution over the selected averaged time. There are periods, especially under stable conditions, when the effect on horizontal dispersion due to gradual changes in the wind direction over a one-hour period could not be well represented by the gaussian distribution (Collet and Oluyemi, 1997). Even though dispersion models have some physical basis, detailed information about the sources of pollutants and other parameters is not generally known. To overcome this limitation, statistical models are also used to facilitate the prediction of pollutant concentrations (Ziomass et al., 1995; Finzi and Tebaldi, 1982). These models assume that the relationship between the variables is statistical in nature. However, such models require information about the distribution of the data which is generally not known a priori (Comrie, 1997). Recently, neural network-based models have also been applied to predict pollutant concentrations. These models provide a better alternative to statistical models because of their computational efficiency and generalization ability. They can handle data having high dimensionality. A neural network model was applied to predict SO_2 concentrations at Sostanj (Boznar et al., 1993) and the inputs used included historical data on observed SO_2 concentrations and meteorological parameters such as wind speed and direction. A feedforward network with a backpropagation learning algorithm was applied. The major finding of the study is that neural network models can be applied to predict the ground-level concentrations of pollutants in complex terrains. However, the results were not compared with those obtained using any other modelling techniques. Gardner and Dorling (1998) presented a review on the application of artificial neural networks in the atmospheric sciences. The applications of multilayer perceptrons in atmospheric sciences were discussed in

* Corresponding author.
E-mail address: apcneeri@nagpur.dot.net.in (A.B. Chelani).

PII: S1364-8152(01)00061-5

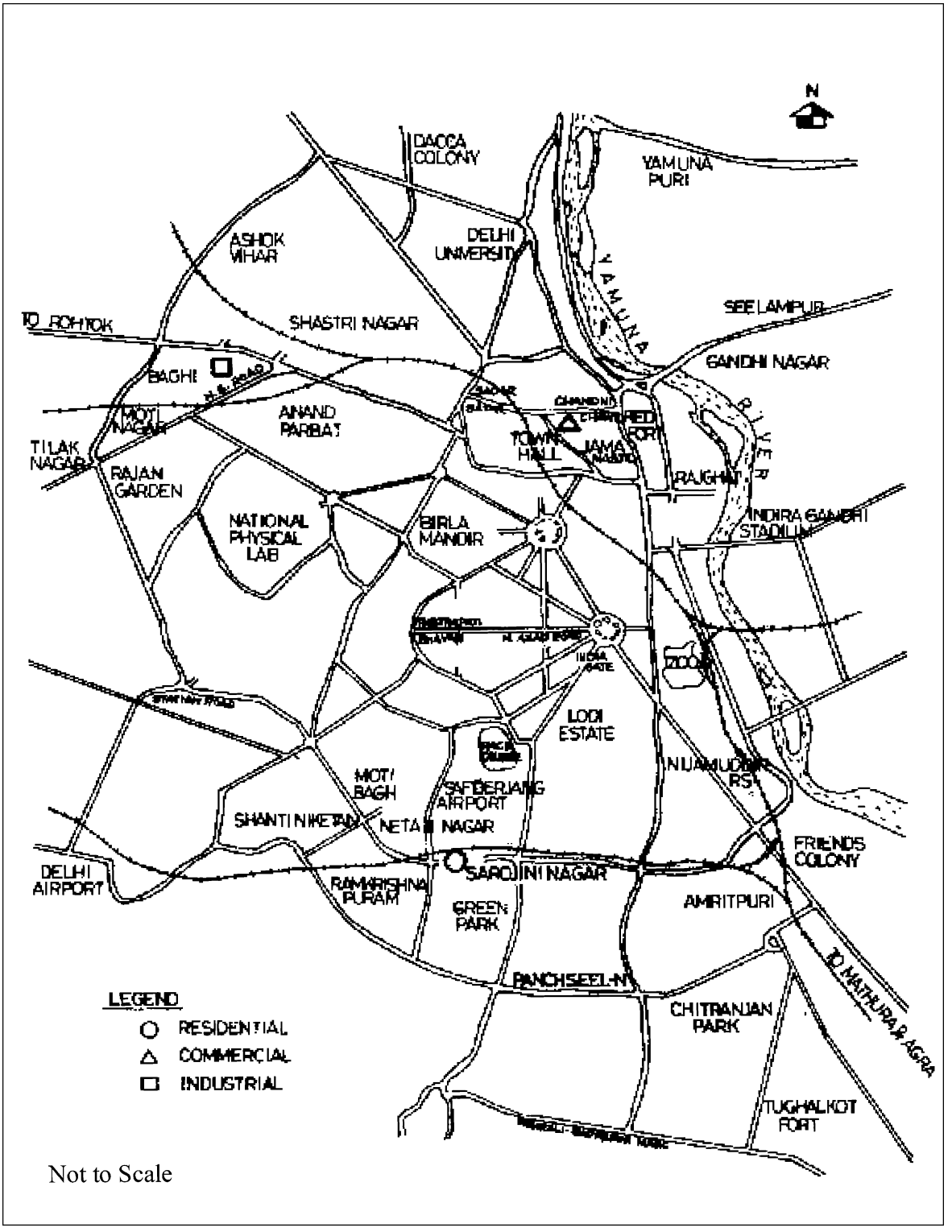

Fig. 1. Location of air-monitoring stations.

detail. Gardner and Dorling (1999) used artificial neural networks to predict NO_2 and NO_x concentrations with meteorological inputs including and excluding the related emissions factor. In all these studies, multilayer perceptrons were used to simulate the system. Generally, two types of multilayer perceptrons are available: feedforward networks and feedback networks. In all the above studies, feedforward networks were used. Feedforward networks can be used for function approximation and pattern recognition. Feedback networks can be useful in

predicting data series having temporal and spatial patterns (Beale, 1997) and these networks may improve the accuracy over feedforward nets. In this study, a recurrent network with feedback in the hidden layer is used to predict the SO_2 concentration observed at three air-monitoring sites in Delhi using meteorological data. Even though the data were collected from three sites, training on only a single network is performed instead of using three neural networks. A regression model is also used to compare the results.

2. Study area and data used

Delhi, as the capital city of India (latitude 28°35′N, longitude 77°15′E), experiences consistent massive influx of migrants from all over the country leading to an increase in population growth. In addition, it is also facing the problems of urban and vehicular growth especially during the past few decades, resulting in aggravation of the air pollution problem in the city. Mushrooming small-scale industries and use of coal by slum dwellers is also responsible for the rise in ambient air pollution levels. The climate of the Delhi region is semi-arid. Because of its proximity to hilly regions, temperatures in winter and summer vary between the extremities of 3°C and 47°C respectively. Thus the dispersion of pollutants is influenced by two extreme climatic conditions (too hot and too cold). The location of three sites is presented in Fig. 1. The air quality at the industrial site represents pollutants emitted mainly from elevated point sources, i.e. stacks of various industries. The quantity of emissions from these sources is more or less the same. Further release of pollutants is from mobile sources, i.e. vehicular emissions vary diurnally. The air quality of the commercial area is mainly influenced by vehicular emissions. The air quality in the residential areas is a function of emissions from household activities, such as cooking, as well as vehicular emissions (NEERI Report, 1991). The data used are a part of the National Ambient Air Quality Monitoring programme conducted by the National Environmental Engineering Research Institute, Nagpur, and sponsored by the Central Pollution Control Board, New Delhi, India. Sulphur dioxide concentration data (1997–1998) monitored 24 hourly during winter (November–February) at three sites; industrial, commercial and residential areas in Delhi were selected. The input parameters to the model include wind speed (in km h^{-1}), temperature (in °C), relative humidity (%) and the wind direction index (WDI). The predominant wind direction in winter was NW (19%) and W (13%) with 42% calm conditions. WDI is calculated using the formula

$$\mathrm{WDI}=1+\sin(\mathrm{WD}+\pi/4) \tag{1}$$

where WD is the wind direction.

It was estimated that the maximum ground level SO_2 concentration corresponds to WDI=0.06, 0.03 and 0.05 at industrial, commercial and residential sites respectively. Similarly, the minimum concentration corresponds to WDI=1.9, 1.7 and 1.3 at industrial, commercial and residential sites respectively. The angle of zero degrees is taken to represent calm conditions. The input variables selected are based on the information available in the scientific literature (Boznar et al., 1993; Collet and Oluyemi, 1997). Data collection or the monitoring exercise was designed to meet the requirements of training and testing the neural network. The SO_2 concentration data were divided into two parts. The data observed during the period extending from November to January were used as the training set and the data for February were used as the testing set. To test the generalization ability of the neural network model, the data observed for the months of November and December during 1998 were also used.

2.1. Multivariate regression model

A multivariate nonlinear regression model was developed to compare the performance of the neural network. The data were first checked for the assumptions of regression analysis. For this, all the variables were examined for autocorrelations among themselves and noise in the data was removed using a log transformation. It was observed that the regression analysis can be used for the present data set. With the wind speed, wind direction index, temperature and relative humidity as input variables and historical patterns of the logarithm of the SO_2 series as the output variable, several models were tested as suggested by Box and Jenkins (1970). Ljung–Box statistics were used to examine the adequacy of the model and the most suitable model equation is

$$Y_{t+1}=a_1Y_t+a_2\times(W_{t+k}+a_3)^{-a_4}+a_5\times(T_{t+1}+a_6)^{-a_7}+a_8(R_{t+1}+a_9)^{-a_{10}}+e_t \tag{2}$$

where Y_t is the log of the SO_2 concentration, W_t is the wind speed (km h^{-1}), T_t is the temperature, R is the relative humidity, e_t is the white noise term, suffix t denotes the day, and a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10} are the regression coefficients.

The coefficients of the above equation can be determined by using the ordinary least squares technique.

2.2. Artificial neural networks

A neural network is a biologically motivated structure whose ith neuron has input value x_i, output value $y_i=f(x_i)$, and connections with the other neurons are described by weights w_{ij}. A three-layer network with one hidden layer is given in Fig. 2a. A brief description of neural networks is given in Gardner and Dorling (1998) and

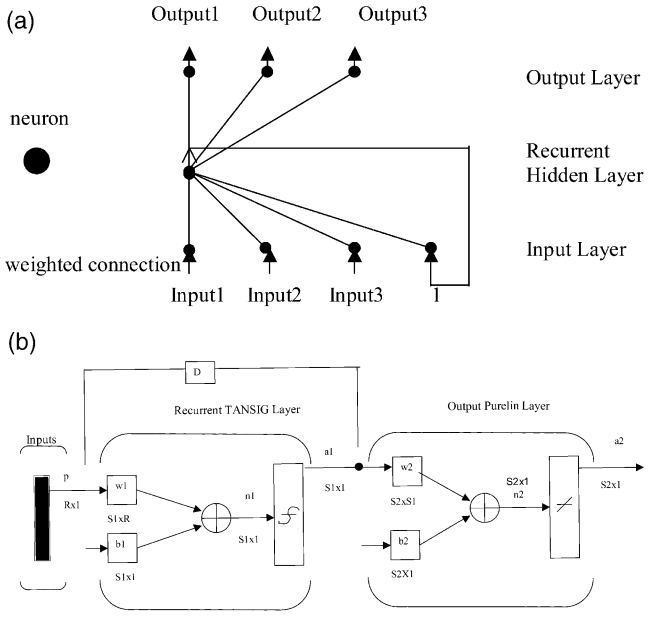

Fig. 2. (a) Structure of the three-layer recurrent network. (b) Recurrent network architecture: D, delay in the recurrent layer connection with input layer; w and b, weights and biases, respectively; S×R, the order of the matrix; p, the number of input vectors; n, the number of neurons; a, the output vector.

Zurada (1992). Generalization and error tolerance are the main features of neural networks (Warren and Cary, 1999). The general procedure for the network simulation includes:

1. Representation of input and output vectors;
2. Representation of the transfer function;
3. Selection of the network structure;
4. Selection of the random weights;
5. Selection of the learning procedure;
6. Presentation of the test pattern and prediction or validation set of data for generalization.

The learning of weights is done using the following procedure:

1. Selection of random numbers for all weights;
2. Calculation of output vectors (referred to here as the network output) and comparison with the destination output (referred to here as the desired output);
3. If the network output is approximately equal to the desired output, then continue with step 1, and if not, weights are corrected according to the correction rule and then continue with step 1.

The neural network was trained and tested using MATLAB 5.1 (Beale, 1997). A three-layer recurrent network (Elman network) that consisted of an input layer, output layer and one hidden layer with feedback was used. A typical Elman network is presented in Fig. 2b. Each layer has a number of nodes called neurons or elements. The nodes in the input layer distribute the input signals to the network and each node in the hidden and output layers receive as input a weighted sum of the outputs, also called activations, of the nodes in the layer below. The nodes in the hidden and output layers also have a bias term, which can be treated as a weight to a constant term with value unity. The weights and bias terms are parameters in the network training. Each node in the hidden and output layers has an activation function, which transfers the node input to an output signal. Feedback links (recurrent links) can be specified from hidden or output nodes to fictitious input nodes. For each feedback link specified, an additional fictitious input node is created. Each link has a weight of unity and is delayed with one unit. The first layer has recurrent connections, the delay in this connection stores values from the previous time step which can be used in the current time step (Beale, 1997). The output is a function of the inputs to the first layer. In the hidden layer, the neurons are sigmoidal and in the output layer, neurons are linear. The sigmoid transfer function from -1 to 1 is given as

$$y=-1+\frac{2}{1+e^{-x}}=\tanh\frac{x}{2} \tag{3}$$

where y is the node output and x is the total node input.

With these transfer functions, a three-layer network can approximate any function with arbitrary accuracy (Beale, 1997). Owing to the presence of an internal loop, Elman networks have the capability of detecting and generating spatio-temporal patterns. They have the ability to fit any input–output function with a finite number of discontinuities. This makes Elman networks useful in such areas as signal processing and prediction where temporal dimension plays an important role. They store the set of patterns and when similar patterns are presented, they link the input with the closest stored pattern.

A network training procedure is normally started from randomly assigned values for network weights and bias terms (and initial states for recurrent networks). To train the network, the Levenberg–Marquardt learning algorithm (Beale, 1997) based on Newton's method is used. The method minimizes the squares of the residuals (equal to the differences between the desired outputs and the network outputs) by modifying the network weights. This optimization technique is more powerful than the gradient descent technique (Beale, 1997) and prevents the network from falling into local optima. The only disadvantage with this algorithm is the need for more memory and a greater number of hidden neurons. The Levenberg–Marquardt weight update equation is given by

$$\Delta W=(J^TJ+\mu I)^{-1}J^Te \tag{4}$$

where J is the Jacobian matrix of derivatives of each error with respect to each weight, μ is a scalar, e is an error vector and I is the identity matrix. The μ parameter

determines the weighting of the steepest descent and Gauss–Newton of the two search methods. Steepest descent is found to be dominating when μ is high, whereas Gauss–Newton is dominating when μ is low (near zero). As the aim is to shift towards Newton's method (for more detailed information, refer to the MATLAB user's guide; Beale, 1997), the value of μ in this study is taken as 0.03. The number of neurons in the input and output layers are taken as the number of input and output variables (Zurada, 1992). The number of neurons in the hidden layer is selected on the basis of convergence criteria (the error minimization criterion is used, for which the minimum error is specified) and also on the basis of conditions such as $2^k \geq N$ (where k is the number of hidden neurons and N is the number of output variables) (Kinnebrock, 1995). The initial values of the weights are assigned randomly based on an input random number seed. The initial values for this study are set between 0.1 and 0.9. Analysis of the convergence of the network to the actual output enables the selection of the neural network configuration. A detailed exercise is conducted by varying the number of hidden neurons. With the variation in the number of neurons in the hidden layer, a decrease in the residual mean square (RMS) error is observed as the iterations progress. The number of neurons has been varied from 4 to 20. Complete stability is observed with 13 hidden neurons. The number of hidden neurons also satisfies the condition $2^k \geq N$ (k=13 and N=3). The training brings the network close to input–output patterns. The training was continued until the RMS error reaches the value 0.1. It is found that 12 000 iterations are required to reach the error goal.

3. Results and discussions

The coefficients determined using the least squares technique for the multivariate regression model are presented in Table 1. For the regression model the meteorological variables at one site (near the airport) are used as an input variable to all the sites. For validation of the models, the RMS error is used, calculated as $\text{RMS}=(\text{SSQ}/n)^{1/2}$, where n is the number of residuals and SSQ the sum of squared residuals. The RMS error for the values fitted at industrial, commercial and residential sites is 0.40, 0.25 and 0.34, respectively, and for the predicted values it is 0.62, 0.47, 0.35, respectively. Fig. 3 illustrates the predicted results obtained using the multivariate regression models. The correlation between the observed and fitted data for industrial, commercial and residential sites is 0.64, 0.59 and 0.61, respectively, and for the predicted data, it is 0.57, 0.52 and 0.48, respectively.

A single neural network is selected to train the data

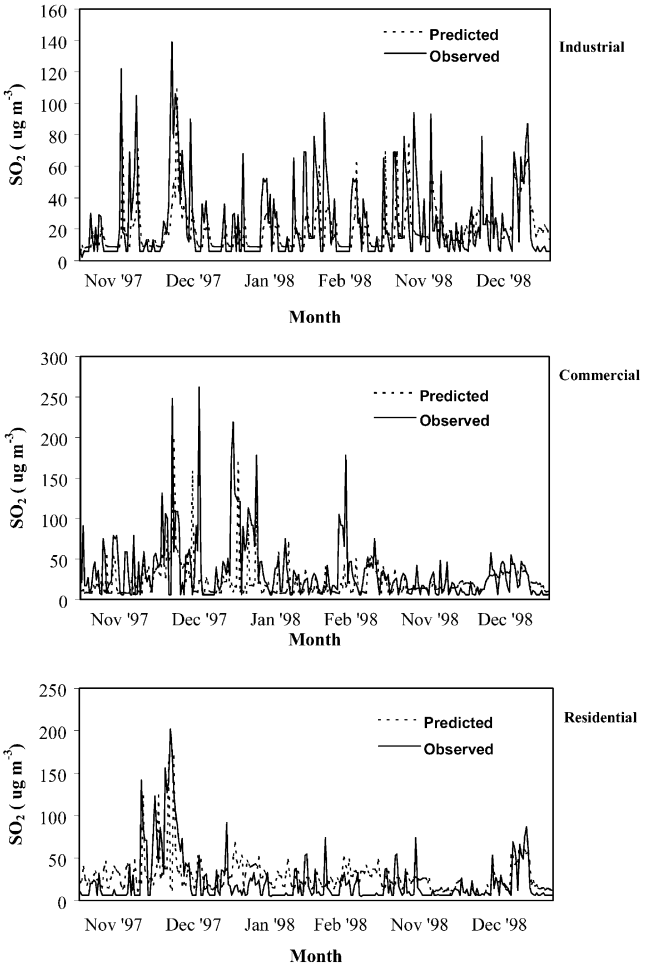

Fig. 3. Prediction using the multivariate model.

Table 1
Coefficients a_1–a_{10} of the multivariate model

Station	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
Industrial	0.68	120	0.90	1.00	148	0.30	1.00	123	1.00	0.58
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.001)	(0.00)	(0.00)	(0.00)	(0.003)
Commercial	0.35	149	1.2	0.22	166	0.48	2.00	65	1.00	1.00
	(0.001)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Residential	0.71	178	0.9	0.67	96	0.44	1.00	54	0.89	0.92
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)

The values in parentheses indicate the significance level.

collected at the three sites. Owing to variations in micrometeorological data, it is preferable to use more than one neural network; however, it was not possible to use separate networks because meteorological data were not available. With only a single network, if the network architecture is selected carefully, it may be possible to predict the concentrations at the three stations with the meteorological data available for one station. The analysis of convergence for the model enables the selection of the neural network structure and also the number of neurons in the hidden layer. The network prediction results are shown in Fig. 4a and b. At the industrial site, the correlation between observed and fitted values for the training data is 0.89 and for the testing data, it is 0.71. At the commercial site, the correlation is 0.80 and 0.79 for the training and testing data, respectively, while for the residential site, correlation is 0.95 and 0.91, respectively. The correlation between observed and predicted data during November–December 1998 is 0.68, 0.72 and 0.63 at industrial, commercial and residential sites, respectively. The RMS error for testing data is 0.10 at all the three stations. However, the RMS error for prediction data is 0.59, 0.53 and 0.46 at industrial, commercial and residential sites, respectively (Table 2). The results show that the network is able to memorize the data set and give the predictions accurately. The distributions of residuals are plotted in Fig. 5a and b. A significance test was carried out to test for the normality of the residuals. For the neural network model, the residuals are distributed normally compared to the multivariate models. The peak forecasting ability of the network shows that the network has learned the training patterns correctly.

4. Conclusion

In this study, SO_2 concentrations were predicted using statistical models and neural networks. The neural net-

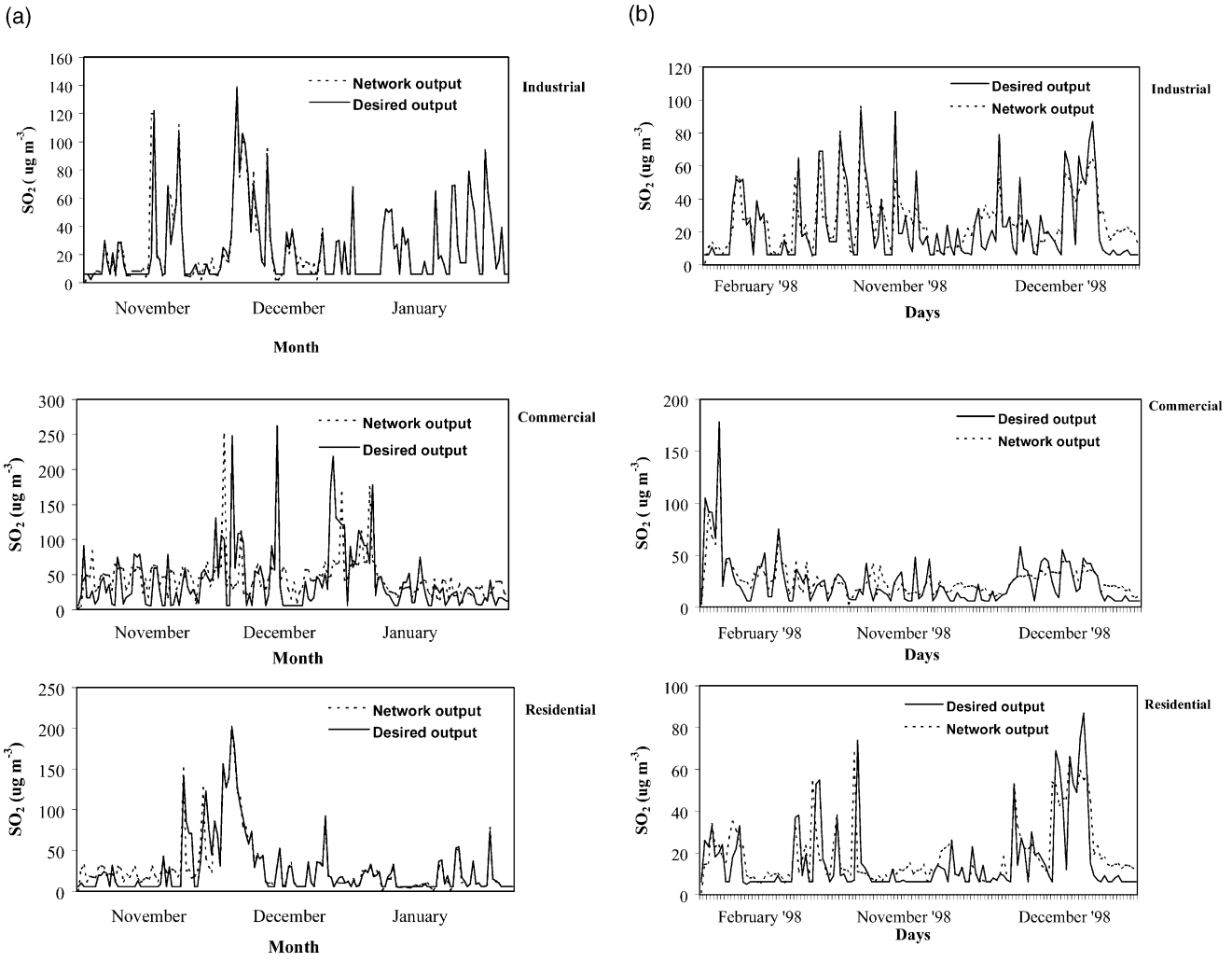

Fig. 4. Behaviour of neural network with (a) training set; (b) testing set (February 1998) and prediction set (November–December 1998).

Table 2
RMS error and correlation between observed and estimated concentration for regression and neural network models

Site	Neural network model						Regression model			
	Training set		Testing set		Prediction set		Observed and fitted values		Observed and predicted values	
	RMS error	Correlation	RMS error	Correlation	RMS error	Correlation	RMS error	Correlation	RMS error	Correlation
Industrial	0.10	0.89	0.10	0.71	0.59	0.68	0.40	0.64	0.62	0.57
Commercial	0.10	0.80	0.10	0.79	0.53	0.72	0.25	0.59	0.47	0.52
Residential	0.10	0.95	0.10	0.91	0.46	0.63	0.34	0.61	0.35	0.48

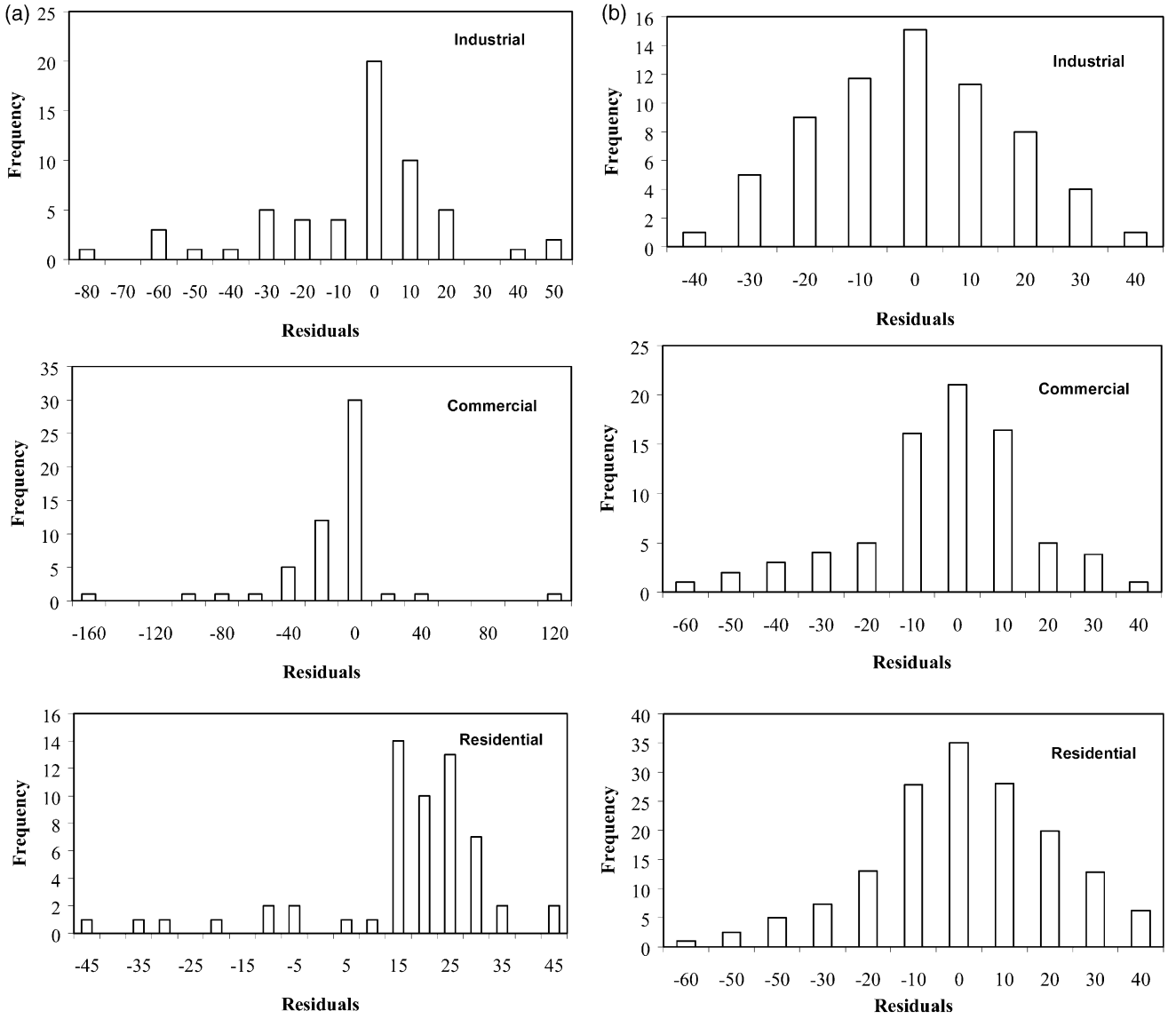

Fig. 5. Distribution of residuals of (a) the multivariate model and (b) the neural network model.

work is trained with feedback in the hidden layer. The neural network with feedback in the hidden layer predicts the values, making use of the measured values. While comparing the results obtained by using the neural network model with those obtained by using multivariate models, it is observed that the response of the neural network to the prediction data indicates that the neural network model provides a better alternative than regression models. It is also possible to use a single neural network model that is able to give accurate predictions for the observed concentrations at three sites using the meteorological data collected from another nearby site instead of using three multivariate models. The approach proposed here can be extended for other applications in which nonlinear relationships are observed. The developed model can be used with or without slight variation for the prediction of pollutant concentration data which is a part of the National Ambient Air Quality Monitoring (NAAQM) programme, the project conducted by the National Environmental Engineering Research Institute, Nagpur, all over India.

Acknowledgements

The authors thank the Director of the National Environmental Engineering Research Institute, Nagpur, for permission to publish this paper, and their colleagues who conducted field studies for generating data on ambient air quality.

References

Beale, M., 1997. MATLAB User's Guide (Version 5.1). Neural-network Toolbox. The Math Work Inc.

Box, G.E.P., Jenkins, G.M., 1970. Time Series Analysis, Forecasting and Control. Holden-Day, San Francisco, CA.

Boznar, M., Lesjak, M., Mlakar, P., 1993. A neural network-based method for short-term predictions of ambient SO_2 concentrations in complex terrain. Atmospheric Environment 27B (2), 221–230.

Collet, R.S., Oluyemi, K., 1997. Air quality modeling: a technical review of mathematical approaches. Meteorological Applications 4, 235–246.

Comrie, A.C., 1997. Comparing neural network and regression models for ozone forecasting. Journal of Air and Waste Management Association 47, 653–663.

Finzi, G., Tebaldi, G., 1982. A mathematical model for air pollution forecast and alarm in an urban area. Atmospheric Environment 16 (9), 2055–2059.

Gardner, M.W., Dorling, S.R., 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric Environment 32 (14/15), 2627–2636.

Gardner, M.W., Dorling, S.R., 1999. Neural network modeling and prediction of hourly NO_x and NO_2 concentrations in urban air in London. Atmospheric Environment 33, 709–719.

Kinnebrock, W., 1995. Neural Networks—Fundamental, Applications, Examples, 2nd revised edn. Rolnberbourg Publishing House, Munich–Vienna.

NEERI Report, 1991. Air pollution in ten cities in India. National Environmental Engineering Research Institute (NEERI), Nagpur, India.

Warren, S.S., Cary, N.C., 1999. Frequently asked questions, ftp://ftp.sas.com/pub/neural, USA.

Ziomass, I.C., Dimitrios, M., Christos, S.Z., Alkiviadis, F.B., 1995. Forecasting peak pollutant levels from meteorological variables. Atmospheric Environment 29 (24), 3703–3711.

Zurada, J.M., 1992. Introduction to Artificial Neural Systems. Jaico Publishing House, Mumbai.