Development of an ANN Interpolation Scheme for Estimating Missing Radon Concentrations in Ohio

Arjun Akkala[1], Vijay Devabhaktuni[1,*], Ashok Kumar[2] and Deepak Bhatt[1]

[1]*EECS Department, The University of Toledo, MS 308, 2801 W. Bancroft St., Toledo, OH 43606, USA*

[2]*Department of Civil Engineering, The University of Toledo, MS 307, 2801 W. Bancroft St., Toledo, OH 43606, USA*

Abstract: Radon (Rn) is a chemically inert, naturally occurring radioactive gas. It is one of the main causes of lung cancer second to smoking, and accounts for about 25,000 deaths every year in the US alone according to the National Cancer Institute. In order to initiate preventive measures to reduce the deaths caused by radon inhalation, it is helpful to have radon concentration data for each locality, *e.g.* zip code. However, such data are not available for each and every zip code in Ohio, owing to several reasons including inapproachability. In places where data is unavailable, radon concentrations must be estimated using interpolation techniques.

This paper presents a new interpolation technique based on Artificial Neural Networks (ANNs) for modeling and predicting radon concentrations in Ohio, US. Several ANNs were first trained and then validated using available data. From the resulting models, the model with lowest validation error was identified. Model accuracies using the proposed approach was found to be significantly better compared to conventional interpolation techniques such as Kriging and Radial Basis Functions.

Keywords: Artificial neural networks, Interpolation, Modeling, Ohio, Radon, Zip code.

1. INTRODUCTION

Radon, which is an invisible, colorless, odorless gas, is a daughter element in the radioactive decay series of uranium. Uranium is widespread in small quantities in rocks and sediments. Both radon and its decay products are radioactive. Radon can cause lung cancer in people exposed to high levels over a long period of time [1], a health issue that many homeowners unknowingly face. Radon is responsible for about 25,000 lung cancer deaths every year in the US. About 2,900 of these deaths occur among people who never smoked [2]. Radon is classified as a Class A carcinogen by the U.S. Environmental Protection Agency [3]. The USEPA and other organizations have launched research efforts to help assess risks and remedial options. In this context, some of the questions worth investigation include: (i) What is the statistical and spatial distribution of indoor radon; (ii) What methods can be used to reduce radon concentrations in homes; (iii) What is the risk as a function of exposure; etc. There have been ongoing efforts, including those at the University of Toledo, in terms of maintaining radon concentration databases for states with high radon levels, *e.g.* Ohio. Although Ohio's radon concentrations are not as high as those in some other states, they are well above the U.S. national average.

With an objective of providing a healthy living environment, the USEPA continues to support preventive actions for all homes with higher radon activity. For instance, Ohio Department of Health (ODH) runs a campaign aimed at measuring radon concentrations across Ohio. Health authorities, in conjunction with county health departments, commercial testing services, and university researchers have so far gathered information for more than 130,000 homes/schools across Ohio [4-8]. Data management has been carried out using different database management systems [9-13].

Radon data is unavailable for some locations or zip codes, owing to several reasons including inapproachability. Fig. (**1**) shows the geometric mean of concentrations across Ohio (based on the Ohio Radon Information System at the University of Toledo). The regions marked in white color correspond to the regions of Ohio, for which, no data are available. For zip codes where multiple data are available, a general practice has been to compute the Geometric Mean (GM) of all available data to account for random data collection by homeowners. The current database has radon concentrations available for 1261 zip codes out of 1492 zip codes in Ohio. For regions with no data availability, radon concentrations need to be estimated using interpolation techniques. In this work, we propose a new ANN based scheme for modeling and predicting radon concentrations. Neural networks employed in this work are 3-layer multi-layer perceptrons often referred to as 3-layer MLP or simply MLP3.

The distribution of radon concentrations depicted in Fig. (**1**) compares well with the general distribution of uranium across Ohio. As well, the results are in line with the general geology observed by Harrell *et al.* [4, 5].

*Address correspondence to this author at the EECS Department, The University of Toledo, MS 308, 2801 W. Bancroft St., Toledo, OH 43606, USA; Tel: (419) 530-8172; Fax: (419) 530-8146;
E-mail: Vijay.Devabhaktuni@utoledo.edu

1875-0400/11

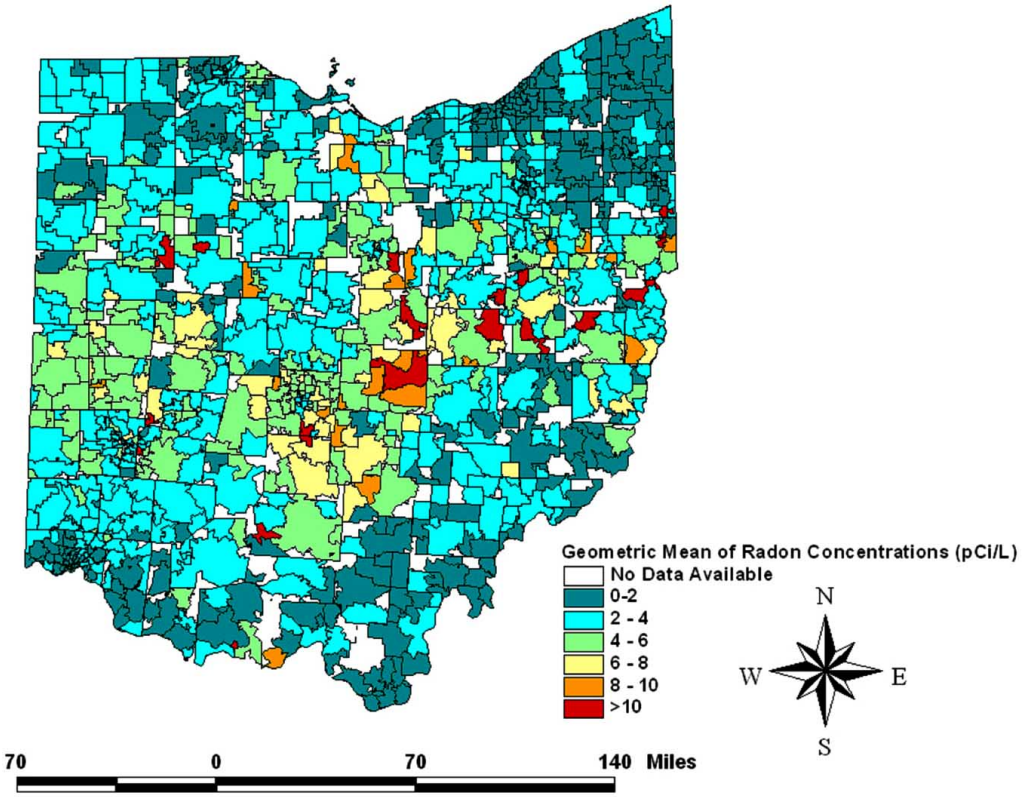

Fig. (1). Map of Ohio showing the geometric mean (GM) of radon concentrations (http://aprg.utoledo.edu/radon/).

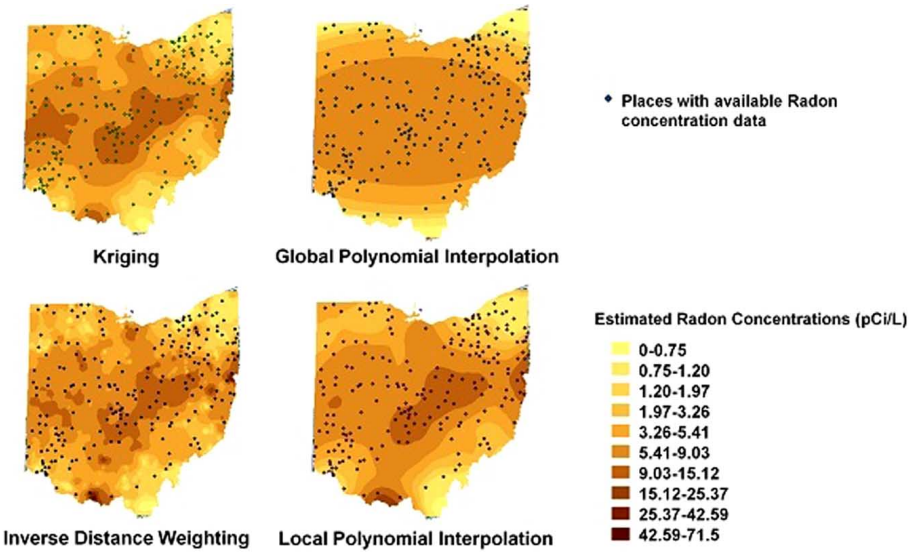

Fig. (2). Estimated radon concentrations obtained for Ohio by using four conventional interpolation techniques.

2. PRIOR ART

Conventional interpolation techniques such as Inverse Distance Weighting (IDW), Kriging, Radial Basis Functions (RBF), Global Polynomial Interpolation (GPI), and Local Polynomial Interpolation (LPI) have already been employed for estimating the missing radon concentrations [14, 15]. Each of these techniques has some advantages and disadvantages [16-18]. For the purpose of illustration, estimations of radon concentrations using Kriging, IDW, GPI, and LPI are shown in Fig. (**2**).

ANNs are being regarded as accurate and fast vehicles to computer aided modeling in comparison to empirical and polynomial models [19]. With their remarkable ability to abstract or derive inherent functional relationships from complicated or imprecise data samples, ANNs are gaining attention for modeling complicated patterns/trends that fail other conventional interpolation techniques.

Akkala *et al*. [18] has shown that ANNs have previously been employed on a limited basis for the purpose of interpolating concentration data. Some such applications are given in Table **1**. Typically, ANN models provide a better alterna-

Table 1. Applications of ANNs for Interpolation of Environmental Data

Author	Purpose of Interpolation	ANNs Employed
Ruiz-Suarez *et al.* [20]	Short-term forecasting of Ozone in Mexico City	Heteroassociative neural networks
Yi and Prybutok [21]	Prediction of ambient ozone	MLP3
Boznar *et al.* [22]	Predictions of ambient SO_2 Concentrations in complex topography	MLP3
Rigol, *et al.* [23]	Prediction of daily minimum temperature	MLP3
Chelani *et al.* [24]	Prediction of SO_2 concentrations in Delhi, India	MLP3
Chelani *et al.* [25]	Prediction of ambient PM10 and toxic metals in Jaipur, India	MLP3
Bryan *et al.* [26]	Annual mean precipitation and temperature	Rprop feed-forward, back-propagation neural network
Gardner and Dorling [27]	Prediction of hourly NO_x concentrations in London	MLP3-MLP10
Hernandez *et al.* [28]	Forecasting of daily air particulate iron and lead concentrations in Madrid, Spain	MLP3
Hooyberghs *et al.* [29]	Forecasting daily average PM10 concentrations in Belgium	MLP3
Siqueira *et al.* [30]	Spatial interpolation of daily solar irradiation in northeastern region of Brazil	MLP3
Snell *et al.* [31]	Interpolation of surface air temperatures	MLP3
Chowdhury *et al.* [32]	Estimation of arsenic contamination in ground water in Bangladesh	Kriging, MLP4
Coutinho *et al.* [33]	Prediction of maximum daily PM10 concentration near Cincinnati	MLP3, RBF

tive to statistical models owing to their computational efficiency and generalization ability. It is observed that the number of studies based on the use of ANN techniques for predicting atmospheric pollutant concentrations are constantly increasing.

3. BRIEF INTRODUCTION TO ANNS

ANNs have been applied to an increasing number of real-world problems of varying complexities. They are known for their ability to model highly complicated input-output relationships that are difficult for conventional techniques [34]. After learning and abstracting from either measured or simulated data, referred to as training data, through a process called training, neural models provide instant answers to the task learnt [35, 36]. Theoretically, neural models can be considered as black-box models, whose accuracy depends on the training data presented. A good collection of training data, *i.e.*, data that is well-distributed, sufficient, and accurately measured/simulated, is suggested for obtaining an accurate neural model [35-37].

An ANN operates by creating connections between many different processing elements called neurons. Each neuron takes many input signals and produces a single output signal that is typically sent as input to other neurons. The neurons are tightly interconnected and organized into different layers. While the input layer receives the input, the output layer produces the final output. One or more hidden layers are sandwiched in between the input and output layers.

Let $\boldsymbol{x}$ be an n-vector containing the model inputs. In the case of radon modeling, the inputs are latitude and longitude. Let $\boldsymbol{y}$ be an m-vector containing the model outputs, *e.g.* radon concentration corresponding to $\boldsymbol{x}$. The relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ is multi-dimensional and nonlinear, and is given by

$$y = f(x). \tag{1}$$

In (1), $\boldsymbol{f}$ represents the functional relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$. In this work, $\boldsymbol{f}$ is a neural network (see Fig. **3**), which is derived or modeled through a training process using a set of sample pairs given by

$$\{(x_p, d_p), p = 1, ..., N\}. \tag{2}$$

In (2), $\boldsymbol{d}_p$ represents the desired output corresponding to p^{th} training input vector $\boldsymbol{x}_p$, N is the number of data samples available for training, and p is simply a sample index. Since our work involves modeling of radon data alone, ANN has only one output, *i.e.* $m = 1$, and $\boldsymbol{y}$ and $\boldsymbol{d}$ are vectors of size one (or scalars).

In reality, the neural network also contains model parameters $\boldsymbol{w}$, referred to as weights, which are first initialized and then adjusted during the training process. As such, (1) can be re-written as

$$y = f(x, w). \tag{3}$$

In the case of a 3-layer MLP, weight vector $\boldsymbol{w}$ contains two sets of weights $\boldsymbol{u}$ (weights between input layer and hidden layer) and $\boldsymbol{v}$ (weights between hidden layer and output layer). Size of $\boldsymbol{w}$ depends upon the size of the neural network, *e.g.* number of hidden neurons. From a theoretical perspective, the definition of $\boldsymbol{w}$, and how $\boldsymbol{y}$ is computed through $\boldsymbol{x}$ and $\boldsymbol{w}$, determine the structure of the neural network. It is important to note that the neural network in (3) does not represent the original problem (*i.e.* radon concentration in this particular case), unless the ANN is trained using the available data.

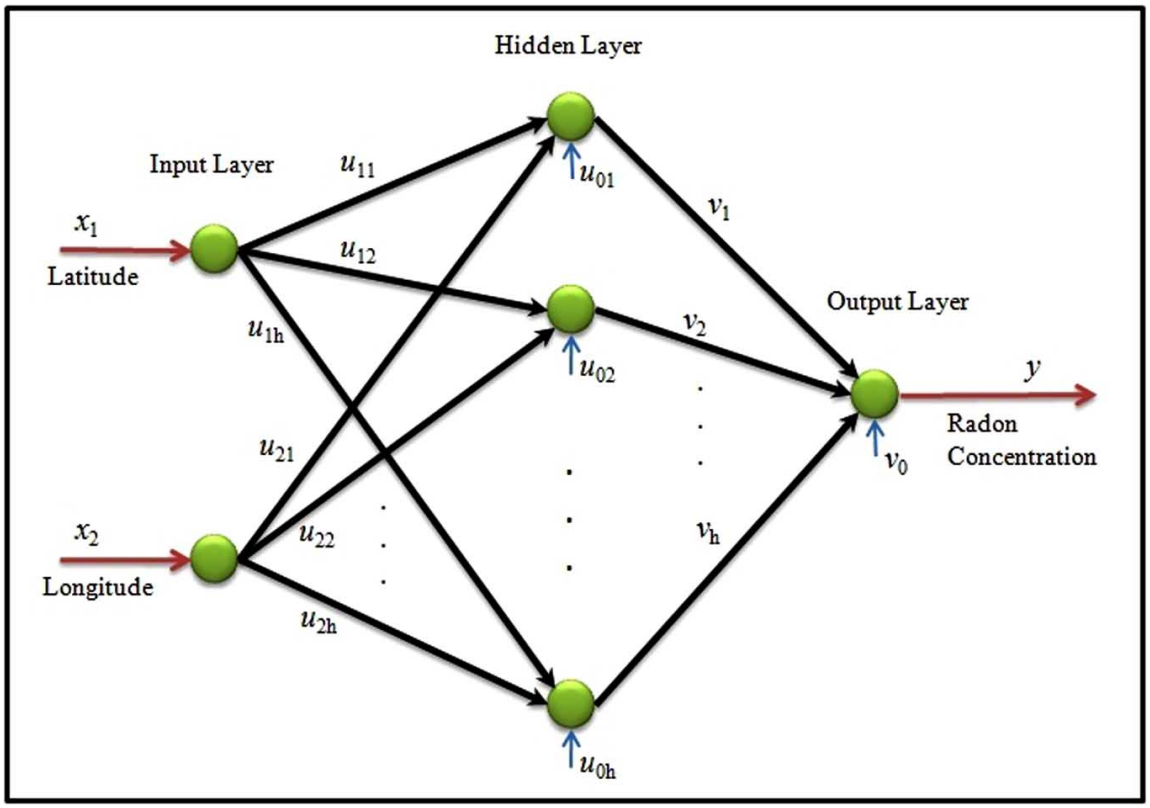

Fig. (3). Proposed 3-layer MLP architecture for modeling radon concentrations. The network has two inputs (latitude and longitude), one output (radon concentration), and *h* hidden neurons.

The objective of training is to determine $\boldsymbol{w}^*$ that minimizes the difference between neural model output $\boldsymbol{y}$ and desired output $\boldsymbol{d}$, given by

$$E(\mathrm{w}) = \frac{1}{2}\sum_{p=1}^{N}\sum_{q=1}^{m}(y_{\mathrm{pq}}(\mathrm{x_p}, \mathrm{w}) - d_{\mathrm{pq}})^2 . \quad (4)$$

In (4), $y_{\mathrm{pq}}(\boldsymbol{x}_{\mathrm{p}}, \boldsymbol{w})$ is the q^{th} output of the neural network when presented with input $\boldsymbol{x}_{\mathrm{p}}$. In our case, where $m = 1$, equation (4) can be simplified as

$$E(\mathrm{w}) = \frac{1}{2}\sum_{p=1}^{N}(y_{\mathrm{p}}(\mathrm{x_p}, \mathrm{w}) - d_{\mathrm{p}})^2 . \quad (5)$$

Owing to the complexity of $E(\boldsymbol{w})$, iterative methods are typically used to determine $\boldsymbol{w}^*$. In such methods, we begin with an initial assumption $\boldsymbol{w}_{\mathrm{initial}}$, referred to as initial weights, and then iteratively update $\boldsymbol{w}$ as

$$\mathrm{w}_{\mathrm{next}} = \mathrm{w}_{\mathrm{now}} + \eta \mathrm{g} . \quad (6)$$

In (6), η is a positive step-size and $\boldsymbol{g}$ is the update direction. In other words, $\boldsymbol{w}_{\mathrm{next}}$ is determined by adjusting the current weights $\boldsymbol{w}_{\mathrm{now}}$ along an update direction $\boldsymbol{g}$. Different training algorithms use different update directions $\boldsymbol{g}$. Experience helps while choosing the neural network, number of hidden layers, number of hidden neurons, and training method. Too small a network could lead to under-learning; while too large a network could lead to over-learning [35].

Weights $\boldsymbol{w}^*$ of a trained neural network help calculate/estimate the output of the neural network. We define z_{j} as the output of the j^{th} hidden sigmoid neuron given by

$$z_{\mathrm{j}} = \frac{1}{1 + \exp(-(\sum_{i=1}^{n} u_{\mathrm{ij}} * x_{\mathrm{i}} + u_{0\mathrm{j}}))} , \quad (7)$$

where x_{i} is the i^{th} input (which is either latitude or longitude in our case), u_{ij} represents weight of the link between i^{th} input neuron and the j^{th} hidden neuron, and $u_{0\mathrm{j}}$ is the bias parameter of the j^{th} hidden neuron. In this work, y, which is the model output or radon concentration, is calculated as

$$y = \sum_{j=1}^{h}(z_{\mathrm{j}} * v_{\mathrm{j}}) + v_0 . \quad (8)$$

In (8), v_0 is the bias parameter of the output neuron, v_{j} represents the weight of link between j^{th} hidden neuron and the output neuron, and h is the total number of hidden neurons.

4. METHODOLOGY

4.1. Data Preparation

Data collected during the radon project as well as that being collected from homes across Ohio on a regular basis is organized into a database (see Table **2**). Each row contains radon concentration for the corresponding zip code, along with other relevant information (*e.g.* county name, population, etc).

For the zip codes, where data is available, it is to be noted that the number of data available varies largely. This may affect the final outcome of modeling studies. However, for modeling purposes, researchers continue to employ geo-

Table 2. An illustrative Portion of the University of Toledo Radon Database Containing Radon Concentrations in Addition to Other Relevant Information. GM Denotes Geometric Mean. NA Indicates Non-Availability

Zip Code	PO Name	State	Population (1999)	County ID	County Name	GM
43019	Fredericktown	OH	9180	69	Richland	NA
43021	Galena	OH	5685	20	Delaware	4.01
43022	Gambier	OH	3649	41	Knox	7.01
43023	Granville	OH	10586	44	Licking	8.85
43025	Hebron	OH	7889	22	Fairfield	0.00

metric mean of radon concentrations available at each zip code. In other words, the geometric mean concentrations for each zip code were derived using the observed radon concentrations in houses located in that zip code. In the conventional GM calculation, a reading of zero present in multiple readings would result in GM being zero (although all other readings for the zip code are non-zero). To overcome this problem, the zero readings are replaced with 0.1, which is the minimum detectable radon concentration [9]. The limit is still true because the homeowners continue to use the same type of instruments that were used in the past and they continue to follow the USEPA guidelines during testing.

The indoor radon database presented through the web site titled *Radon Concentrations Across Ohio (www.radon.utoledo.edu)* was developed in the following manner. Government agencies, university researchers, and commercial testing companies that made and continue to make indoor radon measurements in Ohio are contacted and copies of their data are requested. The data sets received are combined to produce a unified database that includes the zip codes used in this study. Most of the data (*i.e.* greater than 90 percent) were provided by companies engaged in radon testing, and the rest came from surveys conducted by university researchers and various government agencies. The database includes results from several kinds of detectors, which were set out by both owners and professional technicians in a variety of buildings and room types during all seasons and with test periods of different durations. From yet another perspective, a vast majority of radon data (*i.e.* over 95 percent or so) comes from houses, where the tests were done by the homeowners using either charcoal canisters or alpha-track detectors. Despite the eclectic nature of radon data, the measurements for a given county or zip code area seem to provide a good estimate of the year-round average radon concentration. More than 90 percent of the data are received through computer disks/files and transferred without transcription errors. The remaining data are provided as printed tables. Most of these are read directly into ASCII computer files using a text scanner and so transcription errors are avoided. Only less than 2 percent of radon data had to be typed into the database manually. Finally, the data is always triple-checked for accuracy and is hence believed to be free of transcription errors.

In this work, zip codes and the associated GMs of radon concentrations are used as the training data. Note that the school data are not used in this study. Out of a total of 1492 zip codes across Ohio, radon concentrations are available only for 1261 zip codes. For the remaining 231 zip codes, the neural network models developed would predict the radon concentrations.

It is not meaningful to use zip code information directly as the ANN input during training. Appreciating this, we obtained the latitude and longitude for each of the zip codes, since such a mapping is one-to-one. In doing so, the locations of radon concentrations remain the same irrespective of the interpolation technique employed. The neural network to be trained has two input neurons corresponding to latitude and longitude, as shown in Fig. (**3**). An advantage of this approach is that the input space, *i.e.* spatial regions of Ohio, can be viewed as a 2D grid.

4.2. Typical ANN Error Measures

The universal approximation theory states that standard multilayer feed-forward networks with a single hidden layer that contains finite number of hidden neurons are universal approximators [38]. Multiple hidden layers have a greater likelihood for over-learning, especially in such situations, where data patterns are complex [31]. In light of this observation, neural networks employed in this work are 3-layer multi-layer perceptrons or MLP3 networks. This being the first attempt to develop ANN models for the radon problem, there is no existing knowledge in terms of number of hidden layer neurons etc. As such, it became essential to train several neural networks with different number of hidden layer neurons.

The available radon data is first randomized and then split into two independent sets of data, namely, training data and validation data. While the former is used to train the neural network, the latter is used to validate the trained model. An appropriate ANN structure may fail to yield an accurate model, unless trained by a suitable training algorithm [35]. In this work, we used two training algorithms, namely, Backpropagation and quasi-Newton. In both the cases, the number of hidden neurons is varied to obtain a neural model that yields minimal error. A CAD tool [39] is used for training and validation.

The quality of a trained neural model is tested with an independent set of data (validation data) and the resulting error is called validation error. For the case of ANN modeling with 1 output, which is the case of this paper, we define a relative error δ_k for the k^{th} validation data as

$$\delta_k = \frac{y_k(\boldsymbol{x}_k, \boldsymbol{w}^*) - d_k}{d_{max,k} - d_{min,k}}, k = 1,...,N_v . \quad (9)$$

In (9), N_v is the number of validation data, $y_k(\boldsymbol{x}_k, \boldsymbol{w})$ is the output of the trained ANN model when presented with $\boldsymbol{x}_k$ as input, and $d_{max,k}$ and $d_{min,k}$ are the maximum and minimum values of d respectively. In this work, the minimum value of radon concentration is 0.1 for zip code 44648, and the maximum is 39 for zip code 43930. A quality measure based on n^{th}-norm is defined as

$$M_n = \left[\sum_{k=1}^{N_v} |\delta_k|^n \right]^{1/n} . \quad (10)$$

When n = 1, average test error can be directly calculated from M_1 as

$$E_{avg} = \frac{M_1}{N_v} . \quad (11)$$

When n = 2, the n^{th}-norm measure is the Euclidean distance between the neural model prediction and the test data. When $n = \infty$, the n^{th}-norm measure is the maximum test error, which is often referred to as worst-case error among the entire validation data, *i.e.*,

$$E_{worst} = M_\infty = \max |\delta_k|, k = 1,...,N_v \quad (12)$$

Based on the two parameters E_{avg} and E_{worst}, the best ANN model can be identified. When the training data itself is used to validate the model, the resulting E_{avg} becomes the training error.

4.3. Other Comparative Error Measures

Research work done during 80's and 90's led to the development of many performance measures to evaluate the air-quality models. EPA has laid some guidelines in order to validate and calibrate models in a comprehensive manner. Kumar and Gudivaka [40] have discussed in detail the statistics relevant to model evaluation and has applied it to heavy gas models. Similarly, Kumar *et al.* [41] has used statistical tools to evaluate the prediction of lower flammability distances. Patel and Kumar [42], Kumar *et al.* [43], and Kumar *et al.* [44] have indicated that Mean Absolute Error (*MAE*), Factor of Two (*Fa2*), Root Mean Square Error (*RMSE*), Fractional Bias (*FB*), and Normalized Mean Square Error (*NMSE*) are important parameters to assess the performance of air-quality models. Here, we re-visit these parameters for the purpose of comparison with typically used ANN measures (section 4.2). To keep the various formulas simple and relevant to this work, we assume the number of model outputs as 1, which is radon concentration.

1. Mean Absolute Error

The *MAE* expressed as

$$MAE = \frac{\sum_{k=1}^{N_v} |y_k(x_k, w^*) - d_k|}{N_v} \quad (13)$$

measures the average magnitude of the errors in a set of estimations. Ideal value of *MAE* is zero, indicating perfect estimation.

2. Factor of Two

Fa2 is defined as the percentage of the predictions within a factor of two of the observed values. The ideal value for *Fa2* should be 1 (or 100%).

Fa2 = Fraction of data satisfying

$$0.5 \le \frac{y_k(x_k, w^*)}{d_k} \le 2.0$$

3. Root Mean Square Error

The *RMSE* given by

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N_v} (y_k(x_k, w^*) - d_k)^2}{N_v}} \quad (14)$$

is a quadratic scoring rule that measures the average magnitude of the error. It is an indicator of sensitivity of outliers (*i.e.* it indicates the magnitude of extreme errors). The ideal value for *RMSE* is zero.

4. Fractional Bias

The *FB* expressed as

$$FB = \frac{\sum_{k=1}^{N_v} (y_k(x_k, w^*) - d_k)}{\frac{1}{2}\sum_{k=1}^{N_v} (y_k(x_k, w^*) + d_k)} \quad (15)$$

is a normalized bias. The *FB* varies between +2 and -2 and has an ideal value of zero.

5. Normalized Mean Square Error

The *NMSE* given by

$$NMSE = \frac{\sum_{k=1}^{N_v} (y_k(x_k, w^*) - d_k)^2}{\frac{1}{N_v}\sum_{k=1}^{N_v} y_k(x_k, w^*) * \sum_{k=1}^{N_v} d_k} \quad (16)$$

emphasizes the scatter in the entire data set. Normalization by the denominator term assures that the *NMSE* will not be biased towards models that over predict or under predict. Smaller values of *NMSE* denote better model performance, the ideal value being zero.

5. RESULTS

Neural networks with 1 hidden layer having different number of hidden neurons have been trained using two well-known training methods, namely, Backpropagation and quasi-Newton. Resulting ANN model errors are presented in Tables **3** through 6. For instance, Table **3** presents training and validation errors for neural models developed using the Backpropagation algorithm.

Table 3. Training and Validation Errors of ANN Models Trained Using the Backpropagation Algorithm. Available Data is Divided into two Sets, *i.e.* 90% Training Data and 10% Validation Data

No. of Hidden Neurons	Training Error (%)	Validation Error	
		E_{avg} (%)	E_{worst} (%)
10	4.26	3.57	26.32
20	4.27	3.55	26.60
30	4.49	3.81	25.61
40	4.34	3.62	26.35
50	4.62	4.02	27.41
60	5.38	4.85	25.39
70	5.06	4.39	26.54
80	4.76	4.14	26.41
90	4.63	4.00	27.04

Table 4. Training and Validation Errors of ANN Models Trained Using the Backpropagation Algorithm. Based on Table 3, the Number of Hidden Neurons is Fixed to be 20

% Training Data	% Validation Data	Training Error (%)	Validation Error	
			E_{avg} (%)	E_{worst} (%)
90	10	4.27	3.55	26.60
80	20	4.30	4.47	67.84
70	30	4.44	4.73	68.57
60	40	5.02	4.99	65.72
50	50	5.44	5.43	63.89

Table 5. Training and Validation Errors of ANN Models Trained Using the Quasi-Newton Algorithm. Available Data is Divided into two Sets, *i.e.* 90% Training Data and 10% Validation Data

No. of Hidden Neurons	Training Error (%)	Validation Error	
		E_{avg} (%)	E_{worst} (%)
10	4.30	3.59	25.76
20	4.26	3.43	25.81
30	4.27	3.53	25.95
40	4.24	3.42	26.88
50	4.21	3.66	26.30
60	4.23	**3.37**	26.79
70	4.23	3.78	23.96
80	4.19	3.77	24.86
90	4.19	3.69	25.68
100	4.20	3.85	25.78

Tables **4** and **6** confirm that more the training data, lesser the validation error, which is natural. From Tables **3** and **5**, it can be concluded that training by quasi-Newton leads to relatively more accurate models as compared to that by Backpropagation. Fig. (**4**) shows output of the ANN model (trained by quasi-Newton algorithm) when presented with validation data.

The best ANN model is evaluated against the conventional models/techniques using the evaluation parameters discussed in section 4.3. These comparison details have been shown in Table **7**. Training and validation sets used to train the ANNs have been used to develop the conventional models as well (*i.e.* 90% of available data is used for training and 10% is used for validation). As discussed earlier, the ideal

Table 6. Training and Validation Errors of ANN models trained using the Quasi-Newton Algorithm. Based on Table 5, the Number of Hidden Neurons is Fixed to be 60

% Training Data	% Validation Data	Training Error (%)	Validation Error	
			E_{avg} (%)	E_{worst} (%)
90	10	4.23	3.37	26.79
80	20	4.12	4.31	64.16
70	30	4.16	4.40	64.17
60	40	4.80	4.76	64.27
50	50	4.25	4.78	62.88

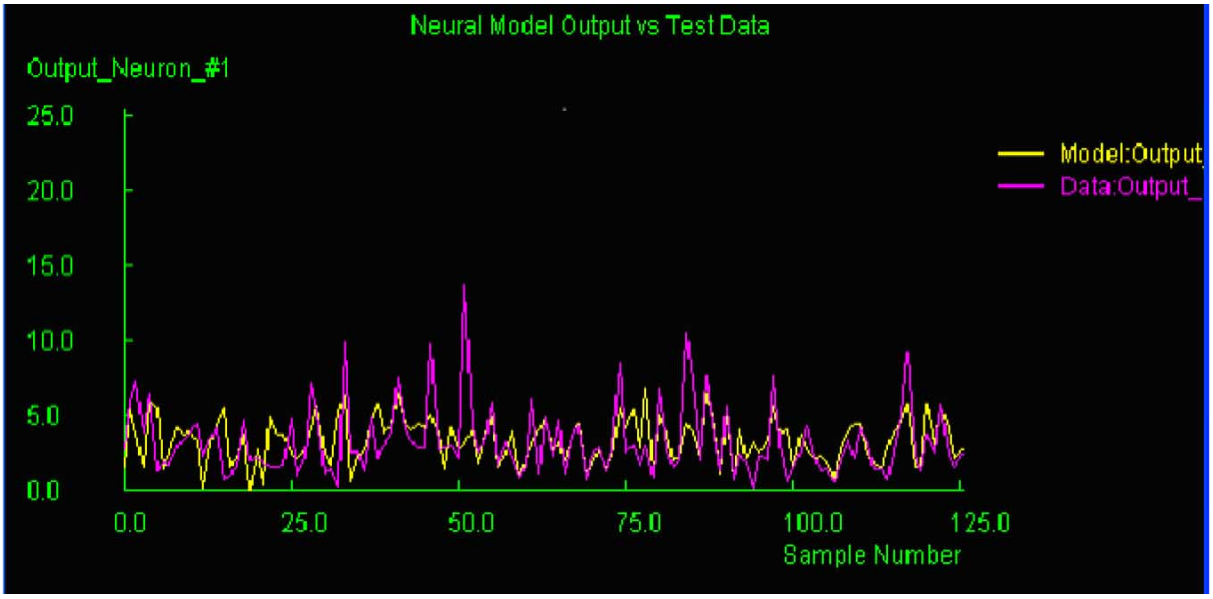

Fig. (4). A screen-shot of the Neuromodeler software showing the validation data (magenta line) and the radon concentration values as predicted by the neural model (yellow line).

Table 7. Performance of ANN Models in Comparison with Several Other Interpolation Techniques, Based on Statistical Measures. Here, 90% of Available Data is Used for Training while 10% is Used for Validation. Ideal Value of E_{avg} is 0

Interpolation Technique	$E_{avg.}$	*MAE*	*Fa2*	*RMSE*	*FB*	*NMSE*
Kriging	.0370	1.441	**.8333**	2.2625	0.089	.4318
Radial Basis Function	.0376	1.463	.8165	2.3752	0.094	.4736
Inverse Distance Weighting	.0385	1.498	.8095	2.4148	0.089	.4920
Local Polynomial Interpolation	.0343	1.335	.8254	**1.9526**	0.030	**.3412**
Global Polynomial Interpolation	.0376	1.465	.7859	2.0226	0.065	.3535
Artificial Neural Networks	**.0337**	**1.312**	.8175	1.9556	**0.017**	.3499

value of *Fa2* is 1, and the ideal value of *MAE*, *RMSE*, *FB* and *NMSE* is 0. Compared to the conventional interpolation schemes, ANNs perform better, as evidenced by Table **7**.

One of the advantages of the proposed ANN technique is the ease of using the neural models, which are in the form simple input-output equations. From the results obtained using the quasi-Newton training algorithm (see Table **5**), it can be observed that the neural model with 60 hidden neurons gives the least average validation error. For this neural model, the outputs of the hidden neurons are given by

$$z(1) = 1.0 / (1.0 + \exp(-1.0 * (0.993393+x(1)*(4.81768)+x(2)*(-10.8887)))) \quad (17)$$

,

$$z(2) = 1.0 / (1.0 + \exp(-1.0 * (1.31538+x(1)*(1.42118)+x(2)*(-2.02847)))) , \quad (18)$$

$$z(3) = 1.0 / (1.0 + \exp(-1.0 * (-2.28203+x(1)*(-11.249)+x(2)*(-5.96931)))) , \quad (19)$$

and

$$z(60) = 1.0 / (1.0 + \exp(-1.0 * (-0.186061+x(1)*(1.96284)+x(2)*(4.00457)))) , \quad (20)$$

where x_1 and x_2 are latitude and longitude respectively. In the above equations, ANN weights after training can also be seen. For this neural model, the output can be calculated using

$$y(1) = 0.618806+z(1)*(1.96112)+z(2)*(2.59364)\textbf{.....}z(60)*(1.71955) \quad (21)$$

6. DISCUSSION AND CONCLUSIONS

For the first time, this paper demonstrates the use of ANNs for modeling radon concentrations in Ohio. The available radon concentration data was reformatted to suit ANN training, by translating the zipcode information into latitude and longitude data. A comparison of the ANN model with conventional interpolation techniques shows that the proposed technique results in relatively better accuracies. In a worst-case scenario, where fewer training data were used (*e.g.* the case of 50% training data of Table **6**), validation error E_{avg} is 4.78%, which shows the efficiency of the proposed technique in scenarios representing sparsely available radon data (or any other environmental data being modeled).

The best neural model, *i.e.* ANN model with 60 hidden neurons, trained using quasi-Newton was chosen for the purpose of estimating radon concentrations in zipcodes with no available data. From Tables **4** and **6**, it can be seen that more the training data, lesser the validation error. Based on this observation, it is recommended that all available data be used for training for being able to accurately predict the radon concentrations in the 231 zip codes with no available data.

In this paper, the fact that radon gas is associated with the geologic occurrence of uranium and other radioactive elements/products has not been considered. Radon is formed from uranium by the decay chain shown in Fig. (**5**). As such, houses and other structures built above uranium-bearing rocks or sediments may have higher indoor radon levels [5]. This fact is corroborated by the uranium concentration map of Ohio (Fig. **6**) with the radon concentration map (Fig. **2**). The observation represents knowledge, which in future, could be used for further improving ANN model accuracies. Future work could also include the use of different hidden neuron activation functions, different training algorithms, and so forth.

7. ACKNOWLEDGEMENTS

Arjun Akkala and Deepak Bhatt acknowledges financial support from the EECS Department and the Department of Civil Engineering in the form of graduate/research assistantships. The authors thank their colleague Kranthi Mogireddy for help with repetitive *NeuroModeler* execution for training and validating the ANN models. Assistance from Akhil Kadiyala and Dilip Manthena, in terms of providing the radon data from the Ohio Radon Information System maintained for the ODH, is gratefully acknowledged. The data collection has been supported by the ODH/USEPA and Ohio Air Quality Development Authority for the past 22 years.

LIST OF SYMBOLS

x	=	A vector of size n representing ANN input
y	=	A vector of size m representing ANN output
f	=	Relationship between $\boldsymbol{y}$ and $\boldsymbol{x}$ to be modeled employing ANNs
d_p	=	Desired output of the ANN model when presented with input $\boldsymbol{x}_p$
p	=	Sample index
N	=	Number of sample pairs (or training data) used to train a given ANN

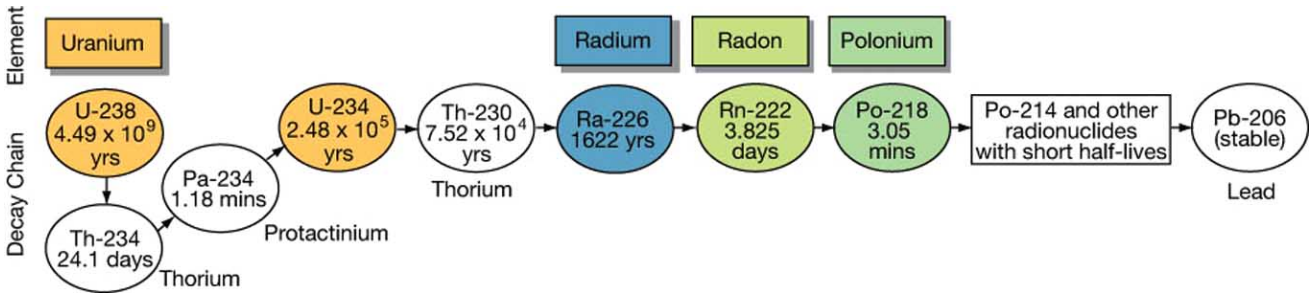

Fig. (5). Decay series of uranium with daughter products and half-lives, showing the formation of radon from uranium. This chain helps us in understanding dependency of radon concentrations on those of uranium.

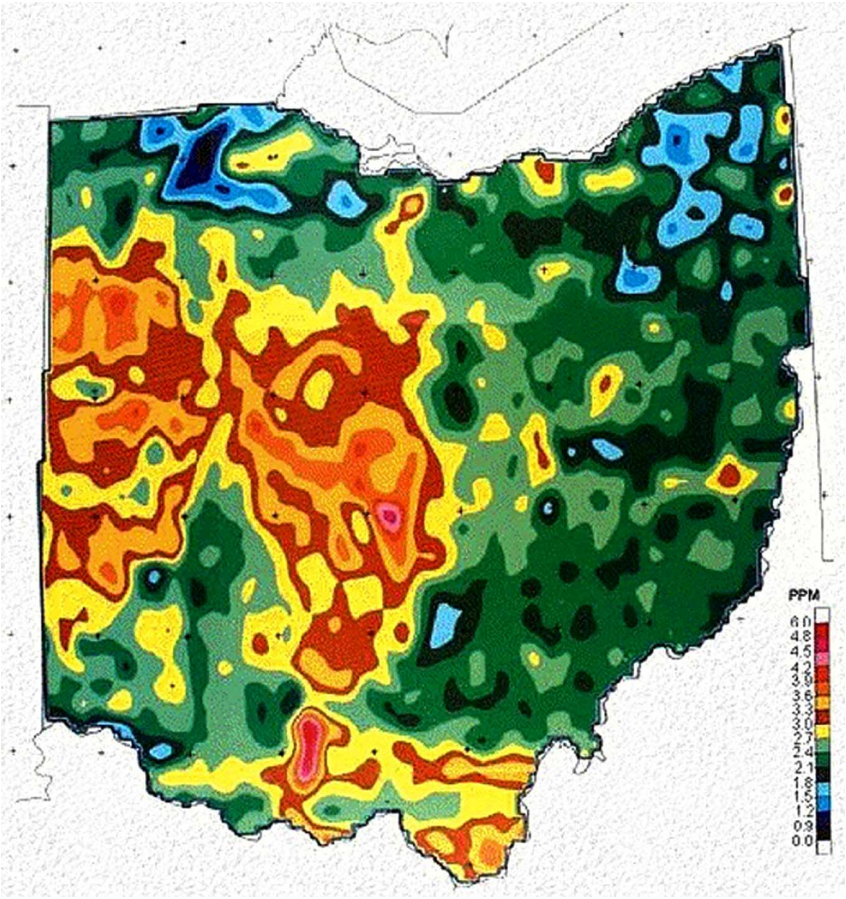

Fig. (6). Radiometric map of Ohio showing the uranium concentration in soil (Source: http://radon.utoledo.edu).

w = Weight vector of a neural network

u_{0i} = Weight parameter representing the bias of the i^{th} hidden neuron

u_{ij} = Weight of the link between i^{th} input neuron and j^{th} hidden neuron

v_0 = Weight parameter representing the bias of the output neuron

v_j = Weight of link between j^{th} hidden neuron and output neuron

$E(w)$ = Error between ANN model outputs $\boldsymbol{y}$ and desired outputs $\boldsymbol{d}$

w^* = Optimal weight vector determined through a process called ANN training

w_{now} = Current weights of the neural network

w_{next} = Updated weights of the neural network

η = Step-size during ANN training

g = Update direction used to iteratively update weights $\boldsymbol{w}$

z_j = Output of the j^{th} hidden neuron

δ_k = Error between ANN output and desired output for k^{th} validation input $\boldsymbol{x}_k$

N_v = Number of sample pairs (or validation data) used to test a trained ANN

M_n = A quality measure based on the n^{th} norm

$d_{max,k}$ = Maximum value of all available d_k

$d_{min,k}$ = Minimum value of all available d_k

E_{avg} = Average test error of a trained ANN model

E_{worst} = Worst-case error of a trained ANN model

REFERENCES

[1] Price PN, Gelman A. Should you measure the Radon concentration in your home? In: Statistics: A Guide to the Unknown, Cengage Learning, Florence, KY 2006.

[2] U.S. Geological Survey. The Geology of Radon: What is Radon? Accessed 23 March 2010. Available from: http://energy.cr.usgs.gov/radon/georadon/2.html

[3] Kansas Geological Survey. The Geology of Radon in Kansas. Public Information Circular 25. 2007.

[4] Harrell JA, Belsito ME, Kumar A. Radon hazards associated with outcrops of Ohio shale in Ohio. Environ Geol Water Sci 1991; 8(1): 17-26.

[5] Harrell J, McKenna JP, Kumar A. Geological controls on indoor Radon in Ohio. Investigation no. 144. State of Ohio: Div Geo Survey, Dept. of Nat. Resources, 1993; p. 36.

[6] Heydinger A, Kumar A, Harrell J. An indoor Radon information system. Environ Soft 1991; 6(4): 194-201.

[7] Kumar A, Harrell J, Heydinger A. Ohio goes online to combat indoor Radon. Environ Manag 2001; 2: p. 10-12.

[8] Kumar A, Varadarajan C. Development of Ohio radon information system. Proc. Environ. Data Analysis - Assessing Health and Environ Impacts, Dev Policy, and Achieving Regul Complian Conf, Air & Waste Management Association, Oak Brook, IL 2005.

[9] Kumar A, Heydinger AG, Harrell JA. Development of an indoor information system for Ohio and its application in the study of the geology of radon in Ohio. Prepared for Ohio Air quality Dev Auth 1990; p. 192.

[10] Kumar A, Sud A, Heydinger A. Application of ORACLE 7.3 database management system for the development of an environmental database. Environ Prog 1998; 17(3): F11-F14.

[11] Ojha S, Thomas SJ, Kumar A. Experience in integrating Geographical Information Systems (GIS) to an indoor Radon database. Environ Prog 2001; 20(3): O7-O10.

[12] Joshi A, Manne GK, Kumar A. Management of Ohio's Radon data with MS access/SQL server 7.0. Environ Prog 2002; 21(Pt 4): D8-D12.

[13] Kumar A, Tandale A, Kalapati RS, Ghose S. Management of radon mitigation data in the state of Ohio. Environ Prog 2003; 22(Pt 3): O19-O24.

[14] Kumar A, Maroju S, Bhat A. Application of ArcGIS geostatistical analyst for interpolating environmental data from observations. Environ Prog 2007; 26(3): 220-25.

[15] Manthena DV, Kadiyala A, Kumar A. Interpolation of radon concentrations using GIS based kriging and cokriging techniques. Environ Prog Sustain Energy 2009; 28(4): 487-92.

[16] Muller SV. An assessment of areal interpolation methods and their use in estimating population at the block level affected by flooding. MA Thesis, University of Missouri-Rolla, Rolla, MO 1995.

[17] Largueche FB. Estimating soil contamination with Kriging interpolation method. Am J Appl Sci 2006; 3(6): 1894-8.

[18] Akkala A, Devabhaktuni VK, Kumar A. Interpolation techniques and associated software for Environmental Data. Environ Prog Sustain Energy 2010; 29(2): 134-41.

[19] Meijer PBL. Fast and smooth highly nonlinear multidimensional table models for device modeling. IEEE Trans Circuits Syst I 1990; 37: 335-46.

[20] Ruiz-Suarez JC, Mayora-Ibarra OA, Torres-Jimenez J, Ruiz-Suarez LG. Short-term ozone forecasting by artificial neural network. J Adv Eng Softw 1995; 23: 143-49.

[21] Yi J, Prybutok VR. A neural network model for the prediction of daily maximum ozone concentration in an industrialized urban area. Environ Pollut 1996; 92: 349-57.

[22] Boznar M, Lesjak M, Mlakar P. A neural network based method for short-term predictions of ambient SO_2 concentrations in highly polluted industrial areas of complex terrain. J Atmos Environ 1993; 27B: 221-30.

[23] Rigol J, Jarvis C, Stuart N. Artificial neural networks as a tool for spatial interpolation. Inter J Geogr Inf Sci 2001; 15(4): 323-43.

[24] Chelani AB, Rao, CVC, Phadke, KM, Hasan MZ. Prediction of sulphur dioxide concentration using artificial neural networks. J Environ Modell Softw 2002a; 17: 161-8.

[25] Chelani AB, Gajghate DG, Hasan MZ. Prediction of ambient PM_{10} and toxic metals using artificial neural networks. J Air Waste Manag Assoc 2002b; 52: 805-10.

[26] Bryan B, Adams J. Three-Dimensional Neurointerpolation of annual mean precipitation and temperature surfaces for China. Geograph Anal 2002; 34(2): 93-111.

[27] Gardner MW, Dorling SR. Neural network modelling and prediction of hourly NO_x and NO_2 concentrations in urban air in London. J Atmos Environ 1999; 33: 709-19.

[28] Hernandez E, Martin F, Valero F. Statistical forecast models for daily air particulate iron and lead concentrations for Madrid, Spain. J Atmos Environ 1992; 26B: 107-16.

[29] Hooyberghs J, Mensink C, Dumont G, Fierens F, Brasseur O. A neural network forecast for daily average PM_{10} concentrations in Belgium. J Atmos Environ 2005; 39: 3279-89.

[30] Siqueira AN, Tiba C, Fraidenraich N. Spatial interpolation of daily solar irradiation through artificial neural networks. Proc ISES Solar World Congress Beijing, China 2007.

[31] Snell SE, Gopal S, Kaufmann RK. Spatial interpolation of surface air temperatures using artificial neural networks: evaluating their use for downscaling GCMs. J Clim 2000; 13(5): 886-95.

[32] Chowdhury M, Ali A, Faisal H. Comparison of ordinary kriging and artificial neural network for spatial mapping of arsenic contamination of groundwater. Stoch Environ Res Risk A 2010; 24(1): 1-7.

[33] Coutinho J, Kumar A, Kadiyala A. Development of artificial neural network models for predicting maximum daily PM10 concentration near Cincinnati, Ohio. In: Veress B, Szigethy J, Eds. Horizons in Earth Science Research Nova Science Publishers Inc., 2010, vol. 2, pp. 201-224.

[34] Kangas LJ, Keller PE, Hashem S, Kouzes RT, Allen PA. Adaptive life simulator: A novel approach to modeling the cardiovascular system. P Am Control Conf, Evanston, IL 1995; pp. 796-800.

[35] Wang F, Devabhaktuni V, Xi C, Zhang Q. Neural network structures and training algorithms for RF and microwave applications. Int J RF Microw C E 1999; 9(3): 216-40.

[36] Devabhaktuni VK, Xi C, Wang F, Zhang QJ. Robust training of microwave neural models. Int J RF Microw C E 2002; 12: 109-24.

[37] Zhang QJ, Gupta KC, Devabhaktuni VK. Artificial neural networks for RF and microwave design - From theory to practice. IEEE T Micro Theory 2003; 51(4): 1339-50.

[38] Pinkus A. Approximation theory of the MLP model in neural networks. Acta Numerica 1999; 8: 143-95.

[39] Zhang QJ. NeuroModeler, Dep Electron. Carleton University, 1125 Colonel By Drive, Ottawa, K1S5B6, Ontario, Canada 1999.

[40] Kumar A, Gudivaka V. An evaluation of four box models for instantaneous dense-gas releases. J Hazard Mater 1990; 25: 237-55.

[41] Kumar A, Luo J, Bennett G. Statistical evaluation of lower flammability distance (LFD) using four hazardous release models. J Process Saf Prog 1993; 12: 1-11.

[42] Patel VC, Kumar A. Evaluation of three air dispersion models: ISCST2, ISCLT2, and SCREEN2 for mercury emissions in an urban area. J Environ Monit Assess 1998; 53: 259-77.

[43] Kumar A, Bellam N, Sud A. Performance of industrial source complex model in predicting long-term concentrations in an urban area. J Environ Prog 1999; 18: 93-100.

[44] Kumar A, Dixit S, Varadarajan S, Vijayan A, Masuraha A. Evaluation of the AERMOD dispersion model as a function of atmospheric stability for an urban area. Environ Prog 2006; 25(2); 141-51.

Received: January 20, 2011 Revised: February 28, 2011 Accepted: March 04, 2011