Research Journal of Applied Sciences, Engineering and Technology 7(16): 3253-3265, 2014
DOI:10.19026/rjaset.7.669
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: July 22, 2013 Accepted: August 03, 2013 Published: April 25, 2014

Research Article

PM_{10} Forecasting Using Soft Computing Techniques

Mohammad F. Ababneh, Ala'a O. AL-Manaseer and Mohammad Hjouj Btoush
Al-Balqa Applied University, Salt, Jordan

Abstract: Air quality forecasting has acquired great significance in environmental sciences due to its adverse affects on humans and the environment. The artificial neural network is one of the most common soft computing techniques that can be applied for modeling such complex problem. This study designed air quality forecasting model using three-layer FFNN's and recurrent Elman network to forecast PM_{10} air pollutant concentrations 1 day advance in Yilan County, Taiwan. Then, the optimal model is selected based on testing performance measurements (RMSE, MAE, r, IA and VAF) and learning time. This study used an hourly historical data set from 1/1/2009-31/12/2011 collected by Dongshan station. The data was entirely pre-processed and cleared form missing and outlier values then transformed into daily average values. The final results showed that the three-layer FFNN with One Step Secant (OSS) training algorithm achieved better results than Elman network with Gradient Descent adaptive learning rate (GDX) training algorithm. Where, the FFNN required the less training time and achieved better performance in forecasting PM_{10} concentrations. Also, the testing performance measurements shown that the selected daily average input variables in previous day ($PM_{2.5}$), relative humidity, PM_{10}, temperature, wind direction and speed is critical to give better forecasting accuracy. Whereas, the testing measurements RMSE = 6.23 $\mu g/m_3$, MAE = 4.75 $\mu g/m_3$, r = 0.943, IA = 0.964 and VAF = 88.80 in PM_{10} FFNN forecasting model that used OSS training algorithm.

Keywords: Forecasting, neural networks, soft computing

INTRODUCTION

The accelerated growth in emission sources of air pollutants in residential megacities has caused urgent need to adopt specific policies in managing air pollution. During this growth, air-pollution forecasting is an important part of the air-pollution management policy (Nikov *et al*., 2005; Saffarinia and Odat, 2008). According to Kunzli *et al*. (2000), air pollution affects humans, animals and plants; it increases respiratory and heart illnesses. Recently, many agencies and organization such as the World Health Organization (WHO) have shown that approximately more than two million people die every year due to air pollution (WHO, 2005). As stated in Brunekreef and Holgate (2002), many environment agencies have set guidelines and limitations for air pollutants concentration levels such as: The U.S. Environment Protection Agency (USEPA), the World Health Organization (WHO) and the European Union (EU). Therefore, air quality forecasting systems are required to contribute in the development of good policies and warning bulletins when air pollutants exceed maximum limit values (Kurt *et al*., 2008).

According to USEPA (2003), early and accurate predictions of air pollutants offer useful information for the public in order to minimize human health hazards and alert people with special health conditions such as asthma patients; therefore, these forecasts must focus on pollutants that have significant impacts on both humans and the environment. In addition, air quality forecasts must be issued for particular times, locations and whenever air pollutants' concentration exceeds the permitted values. Nowadays, air quality forecasting systems provide awareness programs for educating people and to protecting them from air quality violations. Also, they encourage people to participate in voluntary works that reduce concentration levels of air pollutants. However, Suspended particles in the outdoor air with aerodynamic diameter <10 μm are called (PM_{10}). They reduce visibility and are formed by several sources such as: dust of deserts, burning of fossil fuels and chemical pollutants reactions. However, they consist of rigid and liquid molecules (Brunelli *et al*., 2007).

Air pollution, like any other environmental problem, is considered as a "chaotic and non-linear phenomenon" (Niska *et al*., 2004). Air pollutants and weather conditions are associated with each other in a complex relationship, thus the varying rate of air pollution in ambient air makes the modelling or prediction of air quality a more complex process (Kurt and Oktay, 2010). The daily variability in climate has a strong influence on the concentration levels of air pollutants. In the case of the Particulate Matter (PM), when wind speed is high and wind direction is varied,

Corresponding Author: Mohammad F. Ababneh, Al-Balqa Applied University, Salt, Jordan

this leads to increasing the concentrations and reducing visibility (Leung and Lam, 2008). According to Chan *et al*. (2001), low relative humidity leads to higher concentrations levels of Particulate Matter (PM). Undoubtedly, rainfall contributes in cleaning the atmosphere and reducing concentration levels of PM_{10}. Moreover, the variations in air temperature between daytime and nighttime contribute in radiological implications and increase the concentrations of air pollutants (Peirce *et al*., 1998).

Soft computing is techniques operate based on human mind approach. It solves problems approximately not exactly. The main objective for soft computing techniques is "to achieve tractability, robustness and low solution cost through the exploitation of the tolerance for imprecision and uncertainty" (Zadeh, 1994). Therefore, many researches have shown that soft computing techniques are mostly used to solve the complexity associated with the prediction and modeling of air pollution. Undoubtedly, good features of soft computing techniques have earned them an advantage over classical statistics approaches (Yildirim and Bayramoglu, 2006). Nowadays, an artificial neural network is one of the most popular components of the soft computing techniques that have a great capability to predict or model the atmosphere in general and air pollutants in particular (Yildirim and Bayramoglu, 2006). In spite of developing and enhancing several types and architectures of artificial neural networks models on air pollution area, the problem of prediction air pollutants concentration levels still needs to be addressed and a new approach needs to be proposed in this domain.

ANN's have other positive features such as: flexibility, no prior assumption, do not require complete knowledge, ability to learn or acquire experience with or without teacher and generalization (Jain *et al*., 1996). This study used Feed-Forward Neural Network with Back-Propagation algorithm (FFBPNN) and a simple recurrent neural network, called Elman network, with different architectures and training algorithms to forecast PM_{10} air pollutant one day advance. Finally, comparisons between these models are performed. In fact, building a model for forecasting air quality is considered to be quite a challenge. The process involved different stages such as: selecting variables, collecting data, analyzing data and developing ANN's architectures, applying several experiments on historical data sets and evaluating the models using success criteria's (Kaastra and Boyd, 1996).

Several studies were concerned in predicting and modeling PM_{10} concentrations. Chaloulakou *et al*. (2003) performed comparative study to predict the daily average of PM_{10} in Athens city, through two FFNN's models and two multi-linear regression models. In the first FFNN model researchers did not use previous value day of PM_{10} concentrations as input parameter, but in the second FFNN model, they used previous value day of PM_{10} as input variable and the same was applied on other multi-linear regression models. The four models used weather parameters inputs variables as follows: mean surface temperature, maximum moisture, minimum wind speed and wind direction index. Moreover, a comparison study between models was conducted. The conclusion of this comparison showed that the value of previous day of PM_{10} is necessary in forecasting PM_{10} concentrations and the ANN's models are reliable air prediction tools in current days. However, the statistical results for FFNN model with lagged PM_{10} value showed that the Root Mean Square Error (RMSE = 16.94) and coefficient of determination ($r^2 = 0.65$). Corani (2005) applied FFNN, pruned FFNN model and Lazy machine Learning (LL) model in forecasted PM_{10} and O_3 concentrations in Milan city. This study presented insights on enhancement between models, where the LL model achieved better Mean absolute error and pruned FFNN achieved better prediction for exceed concentrations values.

Ul-Saufie *et al*. (2011) presented a comparative study in the prediction of daily average PM_{10} concentrations one day advance in Pulau Pinang city, Malaysia. This study focused on meteorological parameters, chemical pollutants and daily concentrations of PM_{10} in previous day to build multiple linear regression and Multilayer Perceptron (MLP) neural network models. This work presented good statistical evaluation criteria for two models. The MLP model was superior to the multiple linear regression model and achieved prediction accuracy of ($PA = 0.95$) and coefficient of determination ($r^2 = 0.89$). The inputs of models were weather conditions and chemical pollutants. Where, these input variables were: humidity, wind speed, NO_2, temperature, CO, SO_2, O_3 and previous day value for PM_{10} concentration. However, the historical data that used in training and testing from January 2004-December 2007 had missing values for nearly one month (September 2007). Also, the authors trained MLP model using 1100 patterns and tested it using 330 patterns. This means that the study dealt with data as continuous historical data; although there was a gap of one month in the testing data. However, there are several other studies modeling or predicting PM_{10} using different algorithms and approaches as presented in these references (Perez and Reyes, 2006; Roy, 2012).

ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANN's) are inspired by understanding how the human brain operates in mathematical mode (Krenker *et al*., 2011). Artificial Neural Networks (ANNs) are not biological. They are implemented using electronic circuits or components in software (Csáji, 2001). According to Kröse and Van der Smagt (1996), artificial neurons are information process units which constitute major components of ANN's. Each of these units has connection weights transmit

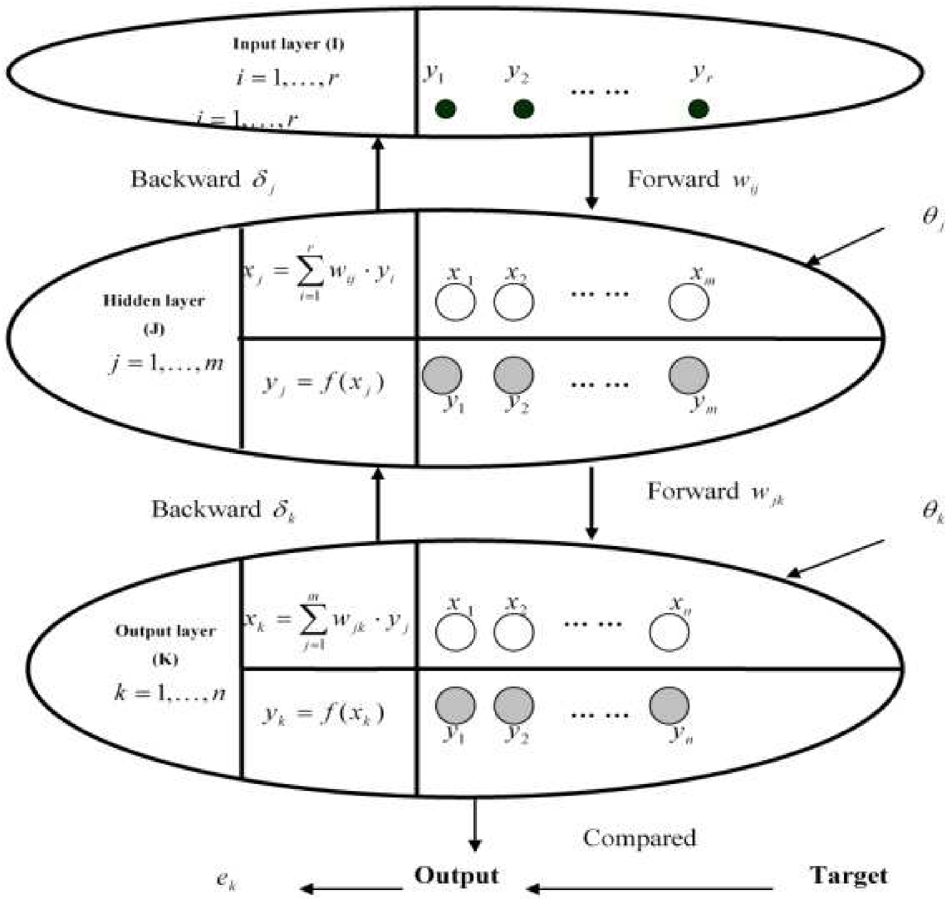

Fig. 1: Three-layer FFNN (Kröse and Van der Smagt, 1996)

input signals into neurons. Then the adder in neurons sum weights inputs. Afterwards, this sum enters to transfer function that limits the behavior of the outputs of neurons. The following equations illustrate the mathematical frame work of neurons (Kröse and Van der Smagt, 1996):

$$z = \sum_{i=0}^{n} w_i \cdot x_i = w_i^T \cdot x_i = w_0 \cdot x_0 + w_1 \cdot x_1 + \cdots + w_n \cdot x_n \quad (1)$$

$$y = f(z) \quad (2)$$

There are three types of transfer functions. In this study will be used hyperbolic tangent sigmoid function in hidden neurons and linear activation function in output neuron which demonstrate in Eq. (3) and (4) respectively:

$$f(z) = \begin{cases} 1, z \geq 0 \\ 0, z < 0 \end{cases} \quad (3)$$

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (4)$$

ANN's can be classified into Feed-Forward and Recurrent Networks based on the style of connections between neurons and layers. In the feed-forward type, data flows in a unidirectional mode from input layer into output layer without feedback connection between neurons in the same layer or other layer of neurons. But in the recurrent type there is feedback such as in Elman network. This research based on supervised learning approach and it used 3-layer FFBBNN and recurrent Elman network as mention previously. The FFNN's with only one hidden layer enough to approximate any function using non-linear transfer function in hidden neurons and called these networks "universal approximators" (Hornik *et al*., 1989). This study used single hidden layer in FFNN and Elman network. Rumelhart *et al*. (1986) proposed "generalization delta rule" to solve the problem of changing the free parameters (weights and biases) from hidden layer to input layer during back-propagations learning iterations.

The supervised learning approach using three-layer FFNN's, the learning patterns (inputs and desired outputs) are presented to network in sequential or batch mode, then propagated from input layer into hidden layer, thus from hidden layer into output layer. After that, the outputs of network were compared with the desired outputs to calculate errors for output unit (Kröse and Van der Smagt, 1996) (Fig. 1).

The aim of calculating the errors was to change the synaptic connection weights and biases in such manner that guaranteed converge into zero value after each iteration (epoch) during training process depending on Eq. (5) and (6):

$$w(t+1) = w(t) + \Delta w(t) \tag{5}$$

$$\theta(t+1) = \theta(t) + \Delta\theta(t) \tag{6}$$

where, w and θ represent weights and biases, respectively.

General back-propagation learning algorithm: The major goal was to reduce performance error function by adjusting free parameters (weights and biases) between output layer and hidden layer. Also, this algorithm adjusts free parameters between hidden layer and input layer in backward direction by calculating the partial derivative of error function (Kröse and Van der Smagt, 1996). The performance error function in this study is mean square error function (Nygren, 2004):

$$E_{AVG} = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\sum_{k=1}^{n}(y(t) - d(t))^2 \tag{7}$$

where, N, n, y and d represent number of training patterns, output neurons, predicted output and desired output respectively. This method is called Gradient Descent (GD) algorithm where the changing of free parameter is based on negative of partial derivative of error function as shown in Eq. (8) and (9) (Hagan *et al*., 1996):

$$\Delta w = -\gamma \frac{\partial E_{AVG}}{\partial w} \tag{8}$$

$$\Delta\theta = -\gamma \frac{\partial E_{AVG}}{\partial \theta} \tag{9}$$

where, the γ notation showed that the learning rate which defined the step size and range was (0, 1). The momentum value (μ) is used as an optimization technique to solve the oscillation problem whenever the learning rate is larger. Therefore, this leads to a change in weights as shown in Eq. (10) (Hagan *et al*., 1996):

$$\Delta w(t) = -\gamma \cdot \frac{\partial E_{AVG}}{\partial w} + \mu \cdot \Delta w(t-1) \tag{10}$$

In addition, the FFNN with Gradient Descent (GD) and Gradient Descent Momentum (GDM) learning algorithms have other disadvantages associated with selecting an appropriate learning rate and momentum terms. Moreover, the speed of convergence at training iterations is slow. Therefore, several learning algorithms such as: Gradient Descent Adaptive learning rate (GDA), Gradient Descent adaptive learning rate with momentum (GDX), resilient back-propagation (Rprop), conjugate gradient back-propagation algorithms (CGb, CGF, CGP, SCG), Quasi Newton methods (BFGS, OSS) and Leven berg Marquardt (LM) algorithms are developed to solve these problems (Demuth *et al*., 2008).

Optimized learning algorithms: According to Demuth *et al*. (2008), the gradient descent adaptive learning rate with momentum algorithm uses variable learning rate during learning iterations instead of constant learning rate. As mentioned previously, the results of learning process end of iteration are new error and new free parameters. In this algorithm the new error after training iteration is compared with previous error. If the new error is larger with more than the predefined rate then the new free parameters will be ignored and the learning rate will decrease. Otherwise, the free parameters are kept. Moreover, if new errors are less than previous errors then the new free parameters are kept and the learning rate is increased. The main object from this algorithm is to increase the learning rate to the maximum (accelerate speed of training) without oscillation on surface error during convergence performance error function.

Riedmiller and Braun (1993) developed Rprop algorithm to overcome the slow convergence problem which occurred in FFNN's during training process due to the use of sigmoid activation function in hidden layer. Clearly, the sigmoid function has derivative closed to zero when inputs are larger. As a result, the standard gradient learning algorithm produces small gradient and small change in free parameters (slow convergence). From this perspective, the authors suggested update value (Δ_{ij}) control in weights change (Δw_{ij}) and biases change ($\Delta\theta_{ij}$).

In order to increase the speed of convergence at training iterations, several numerical optimization techniques were used to learn neural networks such as: Conjugate Gradient Back-propagation (GCB) algorithm, quasi Newton method and Levenberg Marquardt (LM) algorithm. However, there are many CGB algorithms. These algorithms use negatives of gradient as initial search direction similar with standard gradient descent algorithm. After that, its conjugate next search direction to previous search direction and using line search routine to determine the step size along search direction (Demuth *et al*., 2008). The Scaled Conjugate Gradient (SCG) algorithm one of these CGB algorithms. Where, the SCG algorithm used different approach instead of line search routine to reduce consumed time by line search routine (Møller, 1993).

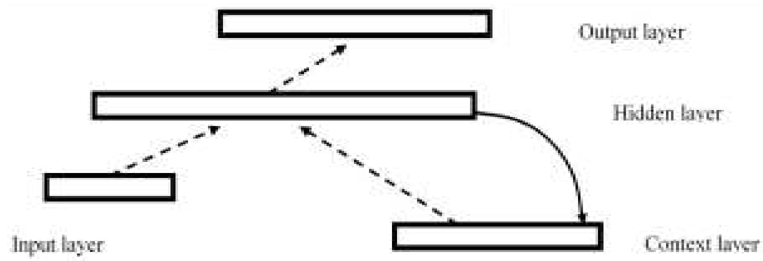

Fig. 2: Elman network structure (Elman, 1990)

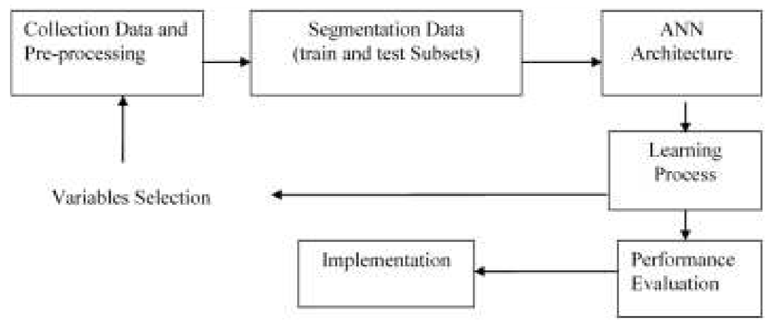

Fig. 3: ANN's forecasting framework (Kaastra and Boyd, 1996; Zhang *et al*., 1998)

Five learning algorithms are used in experiments because the speed of learning time compared with remainder algorithms is the lowest. These algorithms are: (GDX), Rprop and SCG, OSS and LM algorithms.

Recurrent Elman network: Elman (1990) proposed new recurrent network feedback outputs of hidden neurons layer into input layer. These feedbacks are achieved through context units which have constant connection weights with value one. Furthermore, each neuron in hidden layer corresponded with context unit in context layer. Figure 2 demonstrates the structure of Elman network.

Artificial neural network software: The software that will be used in this research is neural network toolbox version 6.0.2 by MATLAB 7.8.0.347 (R2009a) environment.), the MATLAB is an appropriate environment for neural network design because it deals in matrix and vector variables, high power computational potentials and provides graphical features (Hagan *et al*., 1996). These tools provide several fundamental facilities such as: automatic model construction, data splitting tools (train and test data subsets), normalization and demoralization methods, training processes and algorithms, testing (simulation) process, evaluating performance using some of success (statistical) measurements and visualization tools (Roy, 2012).

PM_{10} FORECASTING MODEL

Designing air quality forecasting model to predict PM_{10} one day advance in this research based on Kaastra and Boyd (1996) neural network forecasting framework with some amendment. The adopted framework involved seven recursive stages. Figure 3 illustrate adapted framework.

In this section will be explained adopted framework stages except ANN architecture stage, since this stage is presented in below section.

Data collection and study area: The Republic of China (Taiwan) contains a large, accurate and reliable network of stations to monitor most air pollutants. This study applied the proposed models on Yilan County, Taiwan (R.O.C). In early 1990s, the Taiwan Environmental Protection Administration (TEPA) started monitoring ambient air pollutant concentrations in order to manage air quality and alleviate negative impacts on humans and the environment (Chang *et al*., 2007). TEPA established the Taiwan Air Quality Monitoring Network (TAQMN) (2012) which has 76 air pollution monitoring stations (TEPA, 2012). Moreover, Taiwan was divided in to seven Air Quality Regions (AQR's) based on topographical and climatic characteristics. These regions are: Northen, Chu-Miao, Central, Yun-Chia-Nan, Kao-Ping, Haw-Tung and Yilan (Chang *et al*., 2007).

TEPA uses five types of monitoring stations distributed on seven air quality regions according to the nature of locations that were to be monitored. These types are: general station, background station, national park station, traffic station and industrial station (Kao and Huang, 2000). Figure 4a illustrates the distribution of monitoring stations across Taiwan's AQRs. The collected data used in this thesis to develop the

Fig. 4: (a) Monitoring stations (TAQMN) (b) Dongshan station (TEPA, 2012)

forecasting model represents Yilan Air Quality Region through Dongshan station from 1/1/2009-31/12/2011 (3 years hourly observations) and is available on TEPA web page (TEPA, 2012). Figure 4b illustrates the location of Dongshan station.

Dongshan station is a general-type station whose operations began at the end of 1991. It is located in Yilan County, Dongshan village, Dongshan road, number 98 at Latitude 24° 38' 00.37" N and Longitude 121° 47' 37.94" E. Moreover, it is closed to platform 9, railway from north and Long De industrial area from southeast (TEPA, 2012). This station, like other stations in TAQMN, records hourly observations of ambient air pollutants and archived in MS Excel files on TEPA web page along with hourly meteorological observations provided by the Taiwan Central Weather Bureau (TCWB) (Yang, 2002). The available data in this study involved hourly observations for most air pollutants (SO_2, CO, O_3, PM_{10}, $PM_{2.5}$, NO, NO_2, NO_X) and meteorological conditions (temperature, relative humidity, wind speed and direction).

Variables selection: The collected data represents most meteorological conditions and chemical air pollutants. After the collection of such data is concluded, the variables selection is the next stage in designing forecasting model. According to May *et al*. (2011), the "Data-driven" approaches like ANN model have troubles in selecting input variables because the number of available variables and some variables may cause slight contribution in forecasting process. Moreover, the selected input variables determine the number of input nodes in ANN's forecasting models, where the larger number of inputs causes high computational cost such as increasing learning time (slowing training speed). Obviously, the selection of optimal input variables reduces redundant, over-fitting and noise variables. This study is based on previous studies (Chaloulakou *et al*., 2003; Maraziotis *et al*., 2008; Ul-Saufie *et al*., 2011) and experiments applied on combinations of available variables to select important input variables in the forecasting models.

These variables represent previous daily average observations, where the input variables: temperature, relative humidity, PM_{10}, $PM_{2.5}$, wind direction and speed represent final selected variables after performing many experiments. In experiment result section represent how selected these variables.

Data pre-processing: In order to prepare the collected data for developing models, it is better to perform analysis, filtration and transformation to reduce noise and highlight important relationships during training models. This study used data from Dongshan monitoring station which contains few missing and outlier observations values. In fact, the outlier values were the result of incorrect reading while monitor instruments were collecting observations. In addition, the missing values resulted from technical failures which prevented monitoring instruments from collecting observations.

Kurt and Oktay (2010) replaced missing values with the mean values between previous and next values. This thesis used same approach, where it took the average values between previous hourly observation and next hourly observation using Excel equation tool, then replaced the resulting values with missing and outlier ones. Furthermore, it took the daily average for each hourly parameter observations. Finally, the data after pre-processing represented the daily average value

Table 1: Statistical measurement for input variables

Variable	Unit	Min.	Max.	Mean
Temperature	°C	9.23	32.25	22.83
Relative humidity	Percent (%)	46.29	97.08	80.05
Wind direction	Degrees (°)	52.46	302.30	190.42
Wind speed	m/sec	0.78	8.61	1.70
PM_{10}	$\mu g/m^3$	3.83	119.58	37.66
$PM_{2.5}$	$\mu g/m^3$	0.92	101.67	18.99

Min.: Minimum; Max.: Maximum

for each parameter. Table 1 illustrates the statistical measurements for meteorological and air pollutant variables.

According to (Lapedes and Farber, 1988; Srinivasan *et al*., 1994), the scaling or normalization training data before training process increases the speed of training and prevents computational problems that occur due to differences in measurement units of variables. The normalization of data within range output of activation function of output layer is necessary when using non-linear activation function in output layer (Zhang *et al*., 1998). In this research the MATLAB neural network toolbox scaling (automatically) the input and actual output data for training data within the range (-1, 1) and rescaled the normalized output of neural network during training process.

Data segmentation: The segmentation of data sets after pre-processing stage is necessary for developing, selecting optimal architecture and evaluating performance of neural networks. Data is divided into training and testing subsets. Often, training data forms the largest subset and is used in learning the neural network during training process (developing models). The testing subset is smaller and independent from training subset; it is used to evaluate the performance and generalization potential of the neural network model (Zhang *et al*., 1998). This study divided data into training sample from 1/1/2009-31/12/2010 and test sample from 1/1/2011-31/1/2011 using MATLAB neural network toolbox. The size of data set was 1095 patterns (3-year data); the first two years were for training and the third one for testing. Since PM_{10} forecasting model used daily average values for the previous day as inputs into forecasting model. Therefore, the first pattern was excluded. In order to simplify the process of segmenting data without fractions, last two patterns were excluded. Eventually, the training data were 728 patterns (70%) and testing data were 364 (30%).

Performance evaluation: There are many performance measurements to evaluate the ability of neural networks in the forecasting process. In this research, the forecasting performance and accuracy for PM_{10} forecasting model are evaluated using Root Mean Square error (*RMSA*), Mean Absolute Error (*MAE*), correlation coefficient (*r*), Index of Agreement (*IA*), Variance Account for (*VAF*) and learning time. The following equations represent these measurements (Zhang *et al*., 1998; Yilmaz *et al*., 2010; Ul-Saufie *et al*., 2011):

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(d_t - y_t)^2} \tag{11}$$

$$MAE = \frac{1}{N}\cdot\sum_{t=1}^{N}\left|d_t - y_t\right| \tag{12}$$

$$r = \sqrt{\left(\frac{\sum_{t=1}^{N}(d_t - \bar{d})\cdot(y_t - \bar{y})}{N\cdot S_d\cdot S_y}\right)^2} \tag{13}$$

$$IA = 1 - \left[\frac{\sum_{t=1}^{N}(y_t - d_t)^2}{\sum_{t=1}^{N}\left(\left|d_t - \bar{d}\right| + \left|y_t - \bar{d}\right|\right)^2}\right] \tag{14}$$

$$VAF = \left[1 - \frac{\operatorname{var}(d-y)}{\operatorname{var}(y)}\right]\cdot 100 \tag{15}$$

where, N = number of output observations, d = actual output observations, y = predicted values (outputs of neural network), var = variance, $\bar{d}$ and $\bar{y}$ are means of actual output observations and predicted values respectively, S_d and S_y are standard deviations for actual outputs and predicted outputs respectively. The optimal values that indicate better performance and highest accuracy between actual and predicted values are 0 for *RMSE* and *MAE*. Also, 1 for *r*, *IA* and *VAF*. According to Ma (2005) there is another measurement, namely the linear regression analysis between outputs of neural network (predicted output) and observed outputs (desired output) according to following equation:

$$y = s\cdot d + i \tag{16}$$

where, *s*, *i* are: slop, intercept respectively. If the values of slop and intercept are 1 and 0 respectively, then the model achieves perfect prediction. The difference measures (*RMSE*, *MAE*) and correlation coefficient (*r*) provide adequate information when evaluating performance air quality models. Nonetheless, *LA* and *VAF* are necessary to perform comparison between prediction models (Willmott, 1982). The performance ability of neural network models evaluated based on testing set. This stage used MATLAB neural network toolbox.

Implementation model: Implementation is final stage after the design forecasting model for PM_{10} air pollutant. As mentioned previously, the previous stages were repeated until reached the final forecasting model with good accuracy criteria based on testing set and satisfied the major goals. This stage used neural network toolbox by MATLAB environment to implement PM_{10} forecasting models.

EXPERIMENTIAL RESULTS AND DISCUSSION

This section introduces experimental results and analysis accuracy for PM_{10} air quality forecasting model. This model used FFNN's and recurrent Elman networks, whereas each of these models went through many experiments to select optimal architecture parameters and input variables. The number of input neurons (variable selection), hidden neurons, interconnections between neurons and training algorithms is specified by simulation (trial and error analysis). However, optimal architecture decisions depend on the results of performance evaluation measures by testing data sets.

PM_{10} FFNN result: This section presents experimental results that have been obtained by selecting a number of hidden neurons, input variables and training algorithm for the PM_{10} air quality forecasting model. The selected optimal number of hidden neurons and training algorithm was based on the results of performance measures, namely: Root Mean Square Error (*RMSE*), Mean Absolute Error (*MAE*), correlation coefficient (r) for training and testing data sets. Moreover, the selection of an optimal number of hidden neurons was based on evaluating *RMSE* for training and testing data sets by experiments. When *RMSE* decreases for training data and increases for testing data, this means that an over-fitting has occurred and thus the number of hidden neurons should be decreased. On the other hand, if the training *RMSE* increases and testing *RMSE* decreases, then the number of hidden neurons should be increased (Petchsuwan, 2000).

This study used four training algorithm: SCG, Rprop, OSS, LM for training FFNN's and the training iterations were 500 in FFNN's. Moreover, 20 experiments were performed for each training algorithm. However, the numbers of hidden neurons during experiments were: H_n = 4, 6, 8, 10 for FFNN and H_n = 6 for Elman network. Table 2 illustrates the effects of selecting training algorithms and the number of hidden neurons on performance of PM_{10} forecasting models using FFNN's, where *m*, *s* represent minutes and seconds respectively. Five experiments were conducted upon selection H_n. Further, Table 2 illustrate the optimal H_n when using SCG, Prop, LM and OSS training algorithm were 6, 4, 4 and 4 respectively based on the lowest *RMSE* or *MAE* for testing data. Also, Table 2 illustrates the effects training algorithms and H_n based on testing correlation coefficients (*r*). Where, the

Table 2: Effects learn algorithms and H_n in FFNN model

Training algorithm	H_n	Testing RMSE (μg/m^3)	Testing MAE (μg/m^3)	Testing r	Learn time m: s
SCG	4	6.95	5.25	0.924	1:10
SCG	6	6.36	4.81	0.937	1:12
SCG	8	6.85	5.15	0.926	1:13
SCG	10	7.21	5.26	0.917	1:15
Rprop	4	6.61	4.98	0.936	0:39
Rprop	6	7.31	5.28	0.917	0:39
Rprop	8	7.61	5.85	0.911	0:39
Rprop	10	8.79	5.73	0.900	0:40
LM	4	6.60	5.05	0.934	1:40
LM	6	6.64	5.27	0.933	1:58
LM	8	6.77	5.31	0.930	2:08
LM	10	7.10	5.49	0.926	2:20
OSS	4	6.23	4.75	0.943	1:19
OSS	6	6.58	5.11	0.933	1:20
OSS	8	6.93	5.22	0.929	1:23
OSS	10	7.40	5.41	0.917	1:24

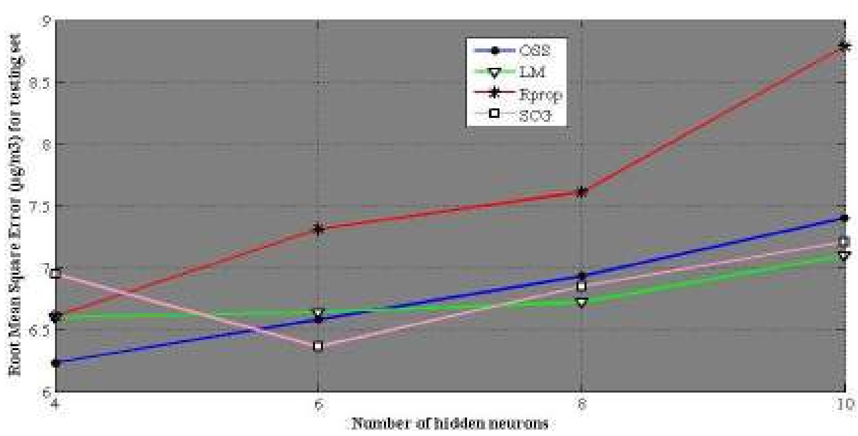

Fig. 5: Correlation between testing *RMSE* and H_n

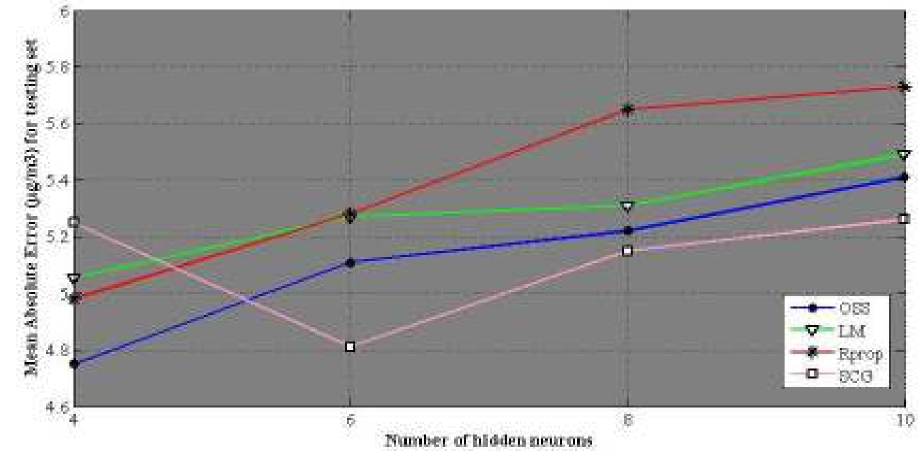

Fig. 6: Correlation between testing *MAE* and H_n

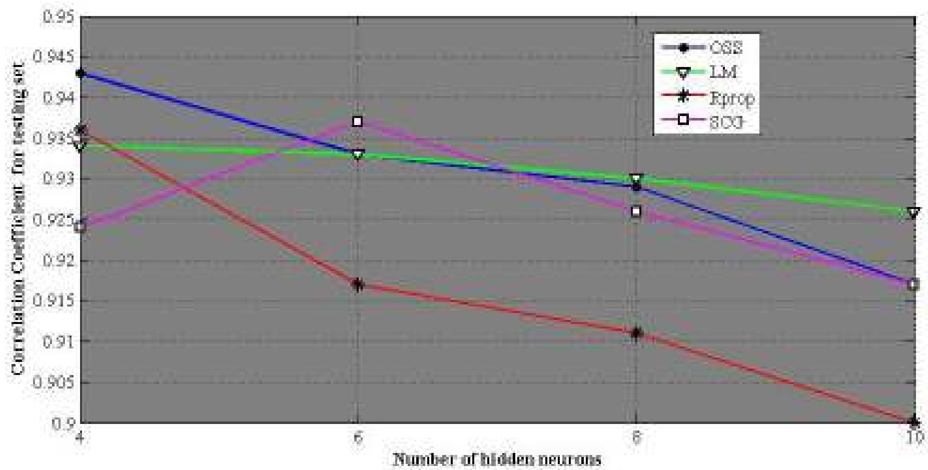

Fig. 7: Correlation between testing r and H_n

Table 3: IA and VAF for 3-layer FFNN'S models

Training algorithms	H_n	Testing VAF (%)	Testing IA
SCG	6	87.76	0.963
Rprop	4	87.13	0.961
LM	4	87.07	0.960
OSS	4	88.80	0.964

FFNN with OSS training algorithm and 4 hidden neurons achieved the lowest *RMSE*, *MAE* and the highest correlation coefficient (*r*) for testing data to forecast PM_{10} air pollutants, where: *RMSE* = 6.23 µg/m^3, *MAE* = 4.75 µg/m^3 and *r* = 0.943 for testing data. Figure 5 to 7 confirm same result.

Another measure to evaluate performance of neural network is training time. As demonstrated in Table 2, most training algorithms require training time less than 2 min. In addition, Rprop satisfy lowest training time (39 sec). Therefore, Rprop is better for online implementation. But, the urgent need for achieving the highest accuracy prediction, this study adapted FFNN's with OSS training algorithm. Although FFNN's with OSS train algorithm and 4 hidden neurons achieved the highest correlation (*r* = 0.943) on testing data, there are other measurements (*IA*, *VAF*) needed for prediction accuracy analysis. Table 3 illustrates the Index of Agreement (*IA*) and Variance Account for (*VAF*) results for previous experiments.

Table 3 have presented the FFNN's with OSS train algorithms and 4 hidden neurons achieved better results in forecasting PM_{10} pollutants. Where, the *VAF* = 88.80% and *IA* = 0.964 for testing data. Figure 8 demonstrate the slop and intercept of linear regression between observed and predicted values of PM_{10} for testing data when using FFNN forecasting model with OSS algorithm were 0.923 and 5.24, respectively. Where, the perfect forecast satisfies when slop and intercept are 1 and 0, respectively.

Figure 9 compare between observed and predicted concentrations of PM_{10} using FFNN's with OSS train algorithm for testing data set. This figure demonstrates error between observed and predicted concentrations.

The results obtained above, have been reached after performing different experiments. These experiments were based on selecting different inputs combinations. Table 4 illustrates experimental results

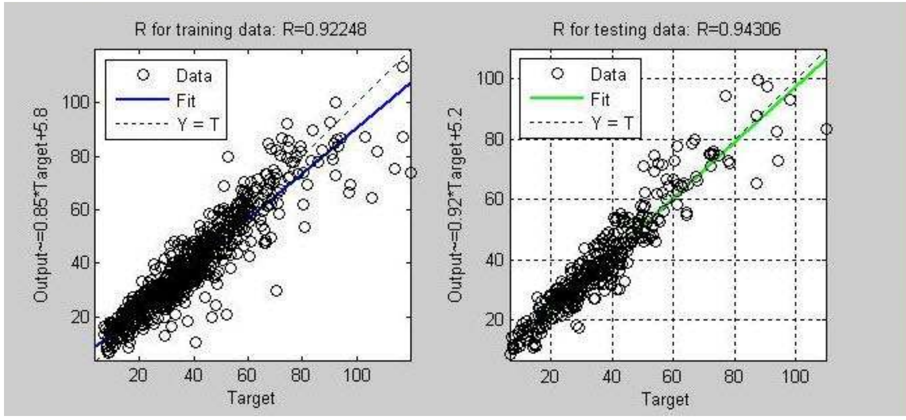

Fig. 8: Linear regression between observed and predicted concentrations

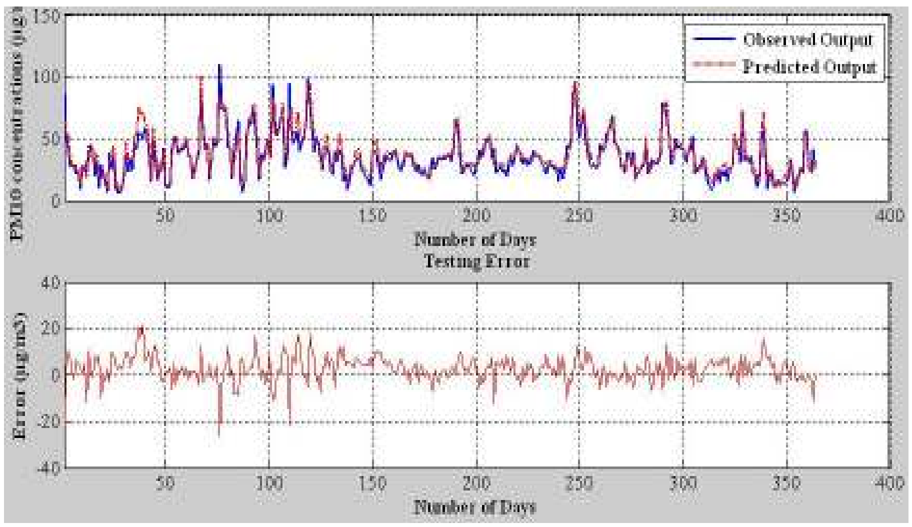

Fig. 9: Observed and predicted values of PM_{10} and error

Table 4: Selection inputs variables for PM_{10} model

Input variables (t-1)	H_n	RMSE	r	IA	VAF (%)
PM_{10}	2	12.90	0.650	0.750	43.00
Temperature and PM_{10}	2	13.00	0.650	0.750	42.25
Temperature, wind direction and PM_{10}	2	13.00	0.648	0.747	42.00
Temperature, wind direction, wind speed and PM_{10}	4	13.20	0.648	0.750	41.60
Temperature, wind direction and speed, relative humidity and PM_{10}	4	10.90	0.768	0.850	59.00
Temperature, wind direction and speed, relative humidity, $PM_{2.5}$ and PM_{10}	4	6.23	0.943	0.964	88.80
Relative humidity and PM_{10}	4	11.90	0.710	0.810	51.00
Relative humidity, $PM_{2.5}$ and PM_{10}	4	7.00	0.920	0.953	85.75

based on testing performance measurements (*RMSA*, *r IA*, *VAF*) during selecting input variables using same previous FFNN with OSS training algorithm and optimal H_n.

According to test performance measurements shown in Table 4, the important input variables which contribute in increasing forecasting accuracy for PM_{10} are PM_{10} (t-1), relative humidity (t-1) and $PM_{2.5}$ (t-1). Also, the combination of input variables: temperature (t-1), wind direction (t-1), wind speed (t-1), relative humidity (t-1), $PM_{2.5}$ (t-1) and PM_{10} (t-1) is critical to give optimal accuracy to forecast PM_{10} one day advance. However, this model different in previous studies in using $PM_{2.5}$ (t-1) variables as input variable. In spite of any subsets of these input variables were used in previous studies, the combination from 6 input variables were not used previously.

Table 5: Comparison between Elman and FFNN models

ANN type	ALG	H_n	RMSE	r	IA	Learn time
Elman	GDX	6	6.16	0.944	0.966	6:02
FFNN	OSS	4	6.23	0.943	0.940	1:19

PM_{10} Elman result: The Elman network is trained using GDX algorithm and the number of iterations is

5000. In addition, the learning rate and momentum are 0.01 and 0.9, respectively. Table 5 illustrates a comparison between FFNN with OSS algorithm and Elman network with GDX algorithm to forecast PM_{10} based on testing performance measures: *RMSE*, *r*, *IA* and learning time.

Table 5 demonstrated more training time (6 min and 2 sec) when forecasting PM_{10} using Elman network compared with FFNN model. The performance measurements for testing data were: *RMSE* = 6.16 $\mu g/m^3$, *MAE* = 4.72 $\mu g/m^3$ and $r = 0.944$. Nevertheless, the FFNN model described in previous section is better, since it required fewer training time (1 min and 19 sec).

CONCLUSION AND RECOMMENDATIONS

The main objective of this study was to develop accurate and efficient air pollution forecasting model for PM_{10} air pollutants in Yilan County, Taiwan (R.O.C). In order to achieve reliable forecasting model, various algorithms and neural network types have been used in this study. However, the development process passed through seven recursive stages which included variables selection, data collection and pre-processing, segmentation data, neural network architecture selection, training process, performance evaluation and the implementation of models using MATLAB neural network toolbox.

The outcome of this research is air quality forecasting model using neural network techniques to predict PM_{10} air pollutants one day advance. These models used three-layer FFNN. Also, the models used hyperbolic tangent and linear activation functions for hidden and output layers. However, selecting the optimal number of hidden neurons and training algorithms was based on performance evaluation measurements: *RMSE*, *MAE* and *r* for testing data set. The results of experiments presented the OSS training algorithm which achieved better performance and the optimal numbers of hidden neurons was: 4. whereas, the Rprop learning algorithm is better for online implementation, since it require less training time. In addition, this study performed a comparison between three-layer FFNN and Elman network forecasting models. The result of this comparison showed that FFNN requires the lowest training time and produces better performance based on testing measurements: *RMSE*, *MAE*, *r*, *IA* and *VAF*. However, this research different than the previous studies in selected input variables and training algorithm (OSS). Any previous studies may be used subsets of these variables as inputs but not all. Moreover, the PM_{10} model used $PM_{2.5}$ as new variable was not used in previous studies.

Further studies can use this research to forecast other air pollutants. Moreover, this research can be extended using other soft computing techniques (fuzzy logic and genetic algorithm) in comparison with this study. Also, further researches that focus on the forecasting field using neural network can use the adapted development phases in this study.

REFERENCES

Brunekreef, B. and S.T. Holgate, 2002. Air pollution and health. Lancet, 360: 1233-1242.

Brunelli, U., V. Piazza, L. Pignato, F. Sorbello and S. Vitabile, 2007. Two days ahead prediction of daily maximum concentrations of SO_2, O_3, PM_{10}, NO_2, CO in the urban area of Palermo, Italy. Atmos. Environ., 41(14): 2967-2995.

Chaloulakou, A., G. Grivas and N. Spyrellis, 2003. Neural network and multiple regression models for PM_{10} prediction in Athens: A comparative assessment. J. Air Waste Manage. Assoc., 53(10): 1183-1190.

Chan, L.Y., W.S. Kwok, S.C. Lee and C.Y. Chan, 2001. Spatial variation of mass concentration of roadside suspended particulate matter in metropolitan Hong Kong. Atmos. Environ., 35(18): 3167-3176.

Chang, S.C., T.Y. Pai, H.H. Ho, H.G. Leu and Y.R. Shieh, 2007. Evaluating Taiwan's air quality variation trends using grey system theory. J. Chinese Inst. Eng., 30(2): 361-367.

Corani, G., 2005. Air quality prediction in Milan: Feed-forward neural networks, pruned neural networks and lazy learning. Ecol. Model., 185(2): 513-529.

Csáji, B.C., 2001. Approximation with artificial neural networks. M.S. Thesis, Faculty of Sciences, Eötvös Loránd University, Budapest, Hungary, pp: 45.

Demuth, H., M. Beale and M. Hagan, 2008. Neural Network Toolbox 6™ User's Guide. Version 6, The Math Works Inc. Retrieved from: http://filer.case.edu/pjt9/b378s10/nnet.pdf (Accessed on: 25th December, 2012).

Elman, J.L., 1990. Finding structure in time. Cognitive Sci., 14(2): 179-211.

Hagan, M.T., H.B. Demuth and M. Beale, 1996. Neural Network Design. PWS Publishing Co., a Division of Thomson Learning. Boston, U.S.A.

Hornik, K., M. Stinchcombe and H. White, 1989. Multilayer feed forward networks are universal approximates. Neural Networks, 2(5): 359-366.

Jain, A.K., J. Mao and K.M. Mohiuddin, 1996. Artificial neural networks: A tutorial. IEEE Comput., 29(3): 31-44.

Kaastra, I. and M. Boyd, 1996. Designing a neural network for forecasting financial and economic time series. Neuro-Comput., 10(3): 215-236.

Kao, J.J. and S.S. Huang, 2000. Forecasts using neural network versus Box-Jenkins methodology for ambient air quality monitoring data. J. Air Waste Manage. Assoc., 50(2): 219-226.

Krenker, A., J. Bešter and A. Kos, 2011. Introduction to the Artificial Neural Net Works. In: Suzuki, K. (Ed.), Artificial Neural Networks-methodological Advances and Biomedical Applications. In Tech, India, pp: 3-18.

Kröse, B. and P. Van der Smagt, 1996. An Introduction to Neural Networks. 8th Edn., University of Amsterdam, Netherlands.

Kunzli, N., R. Kaiser, S. Medina, M. Studnicka, O. Chanel, P. Filliger, M. Herry, F.H. Jr, V. Puybonnieux-Texier, P. Quénel, J. Schneider, R. Seethaler, J.C. Vergnaud and H. Sommer, 2000. Public-health impact of outdoor and traffic-related air pollution: A European assessment. Lancet, 356(9232): 795-801.

Kurt, A. and A.B. Oktay, 2010. Forecasting air pollutant indicator levels with geographic models 3 days in advance using neural networks. Expert Syst. Appl., 37(12): 7986-7992.

Kurt, A., B. Gulbagci, F. Karaca and O. Alagha, 2008. An online air pollution forecasting system using neural networks. Environ. Int., 34(5): 592-598.

Lapedes, A. and R. Farber, 1988. How Neural Nets Work. In: Anderson, D.Z. (Ed.), Neural Information Processing Systems. American Institute of Physics, New York, pp: 442-456.

Leung, Y.K. and C.Y. Lam, 2008. Visibility impairment in Hong Kong-a wind attribution analysis. Bull. Hong Kong Meteorol. Soc., 18: 33-48.

Ma, J., 2005. Application of neural network approach to acid deposition prediction. M.S. Thesis, Environmental System Engineering, University of Regina, Regina, pp: 112.

Maraziotis, E., L. Sarotis, C. Marazioti and P. Marazioti, 2008. Statistical analysis of inhalable (PM_{10}) and fine particles ($PM_{2.5}$) concentrations in urban region of Patras Greece. Global NEST J., 10(2): 123-131.

May, R., G. Dandy and H. Maier, 2011. Review of Input Variable Selection Methods for Artificial Neural Networks. In: Suzuki, K. (Ed.), Artificial Neural Networks-methodological Advances and Biomedical Applications. In Tech, India, pp: 3-18.

Møller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks, 6(4): 525-533.

Nikov, A., F. Karaca, O. Alagha, A. Kurt and H. Hakkoymaz, 2005. Air poltool: A web based tool for Istanbul air pollution forecasting and control. Proceeding of 3rd International Symposium on Air Quality Management at Urban Regional and Global Scales. Istanbul, 1: 247-255l.

Niska, H., T. Hiltunen, A. Karppinen, J. Ruuskanen and M. Kolehmaxinen, 2004. Evolving the neural network model for forecasting air pollution time series. Eng. Appl. Artif. Intell., 17(2): 159-167.

Nygren, K., 2004. Stock prediction-a neural network approach. M.S. Thesis, Royal Institute of Technology, KTH, Stockholm.

Peirce, J.J., R.F. Weiner and P.A. Vesilind, 1998. Meteorology and Air Pollution. In: Environmental Pollution and Control, 4th Edn., Butterworth-Heinemann, Boston, pp: 271-286.

Perez, P. and J. Reyes, 2006. An integrated neural network model for PM_{10} forecasting. Atmos. Environ., 40(16), 2845-2851.

Petchsuwan, T., 2000. A neural network approach for the prediction of ambient SO_2 concentrations. M.S. Thesis, Environmental System Engineering, University of Regina, Canada, pp: 188.

Riedmiller, M. and H. Braun, 1993. A direct adaptive method for faster back propagation learning: The RPRO Palgorithm. Proceedings of IEEE International Conference on Neural Networks (ICNN 93), pp: 586-591.

Roy, S., 2012. Prediction of particulate matter concentrations using artificial neural network. Resour. Environ., 2(2): 30-36.

Rumelhart, D.E., G.E. Hintont and R.J. Williams, 1986. Learning representations by back-propagating errors. Nature, 323: 533-536.

Saffarinia, G. and S. Odat, 2008. Time series analysis of air pollution in Al-Hashimeya town Zarqa, Jordan. Jordan J. Earth Environ. Sci. (JJEES), 1(2): 63-72.

Srinivasan, D., A.C. Liew and C.S. Chang, 1994. A neural network short-term load forecaster. Elect. Power Syst. Res., 28: 227-234.

Taiwan Air Quality Monitoring Network (TAQMN), 2012. Taiwan Environmental Protection Administration (TEPA). Retrieved form: http://taqm.epa.gov.tw/taqm/en/ (Accessed on: 23th December 2012).

Ul-Saufie, A.Z., A.S. Yahya, N.A. Ramli and H.A. Hamid, 2011. Comparison between multiple linear regression and feed forward back propagation neural network models for predicting PM_{10} concentration level based on gaseous and meteorological parameters. Int. J. Appl., 1(4): 42-49.

USEPA, 2003. Guidelines for Developing an Air Quality (Ozone and $PM_{2.5}$).Forecasting Program, U.S. Environmental Protection Agency Office of Air Quality Planning and Standards Information Transfer and Program Integration Division AIRNow Program Research Triangle Park. North Carolina, EPA-456/R-03-002.2.

WHO, 2005. Air Quality Guidelines for Particulate Matter, Ozone, Nitrogen Dioxide and Sulfur Dioxide. Global Update 2005, Summary of Risk Assessment (2005). World Health Organization (WHO), Geneva.

Willmott, C.J., 1982. Some comments on the evaluation of model performance. Bull. Am. Meteorol. Soc., 63: 1309-1369.

Yang, K.L., 2002. Spatial and seasonal variation of PM_{10} mass concentrations in Taiwan. Atmos. Environ., 36(21): 3403-3411.

Yildirim, Y. and M. Bayramoglu, 2006. Adaptive neuro-fuzzy based modeling for prediction of air pollution daily levels in city of Zonguldak. Chemosphere, 63(9): 1575-1582.

Yilmaz, I., N.Y. Erik and O. Kaynar, 2010. Different types of learning algorithms of Artificial Neural Network (ANN) models for prediction of gross calorific value (GCV) of coals. Acad. J., 5(16): 2242-2249.

Zadeh, L.A., 1994. Fuzzy logic, neural networks and soft computing. Commun. ACM, 37(3): 77-84.

Zhang, G., B.E. Patuwo and M.Y. Hu, 1998. Forecasting with artificial neural networks: The state of the art. Int. J. Forecasting, 14(1): 35-62.