# Air Pollution Level Prediction System

**Rajeev Tiwari, Shuchi Upadhyay, Parv Singhal, Ujla Garg, Shefali Bisht**

***Abstract:*** ***Nowadays, the levels of air pollutants in the environment are increasing manifold. This has led to deterioration of human lifestyle. Various methods like 'Climatology' (based on the assumption that the past is a good predictor of the future) have been used for air quality forecasting. These approaches are usually used to predict exceeding limits from specific thresholds, not ambient concentrations. As a result, a lot of improvement is still required in this field for prediction analysis. With incomplete data parameters and their significance (priority), most of the methods fail to predict the pollution levels significantly. The advantage of artificial neural networks includes the problem-solving efficiency in the cases of unavailability of complete information, with no information about the analytical relationship among the input and processed output data. The aim is to develop an artificial neural network for air quality prediction that can perform with constrained dataset with highly robust feature in order to handle the data including noise and errors. Dataset used deals with pollution in the U.S. involving four major pollutants (Nitrogen Dioxide, Sulphur Dioxide, Carbon Monoxide and Ozone) on daily basis for the time period of year 2008 to 2017. We use prediction models like ARIMA etc. to validate our predicted AQI. This AQI analysis helps in telling the status of present air pollution and forecasted pollution levels in coming time. So, it plays a vital role for decision maker and for individual also to know about air pollution quality.***

***Index Terms:*** ***Artificial neural network; Environmental engineering; Air Quality index; ARIMA; Forecasting.***

## I. INTRODUCTION

### *A. General Introduction*

The alarming rise in the air pollution level can be defined as one of the most threatening problem in urban settlements in the current scenario. The rise in the permissible concentrations of different pollutants are observed in past few years, adding up to the increasing pollution level. As a result, affecting the weather conditions in unfavorable manner and the formation of smog phenomena. As a fact, the affected air quality level in the atmosphere impacts health of individual badly and may imbalance in its economy. One of the major issues faced by the urban settlements are the diseases caused due to the rising pollution level. One of the major aspects of air quality monitoring practiced in the urban settlements includes detection of crossing the permissible pollutant concentrations. Knowledge of reliable forecasts occurrences of high air pollution levels would allow taking preventive actions. The natural phenomena conclude a time series with some degree of randomness. The different pollutants present in the atmosphere can be observed dispersed or concentrated during varied time periods. As per the previous studies (Giorgio and Piero, 1996) the data of ambient air quality can be framed as stochastic time series, thereby helpful in estimating a short-term forecast on the basis of previously collected historical data. However, time-series models are preferred over the ambient air pollution methods of forecasting, the variations of pollutant level are generally observed as not as it is a simple autoregressive model, using moving average predictions. Time series is effective during consistent intervals of time, it may not be effective during varied intervals of data. Then it needs model parameters adjustment. These are the difficulties one may face by using time series prediction.

Forecasting Methods to address the problem

One of the most commonly used forecasting methodologies are numerical and statistical models. Numerical models generally work on some specific sections of data. It may require extra information about compositions of gases, and other processes in atmospheric layers. The major drawback over this requirement is that this critical information is often not available to full extent. Thus, approximations and simplifications are often used in the modelling process. In contrast, statistical models usually operate over a large measure of collected data under a large variation of atmospheric conditions. By using the regression and machine learning techniques, a collection of functions can be evaluated and utilized in order to fit the data of pollution as the selected predictors. The different pollutants present in the atmosphere can be observed dispersed or concentrated during varied time periods. As per (Giorgio and Piero, 1996), data can be considered as stochastic time series, thereby helpful in estimating a short-term forecast on the basis of previously collected historical data.

### *B. Artificial Neural Networks*

Artificial Neural Networks (ANNs) are a part of Statistical models. Apart from air prediction, they are also used in other branches of environmental sciences, business, medicine, industry etc. Artificial Neural Networks are famous for their ability to work with unknown relationships where we only have the input variables to get the output. They can be further divided into many types based on structure and principal of operation, one is multi-layer perceptron (MLP) which is fully connected feed-forward networks and other is radial basis function networks (RBF). A basic neural network consists of one Input Layer, one or more Hidden Layers and only one Output Layer.

**Revised Manuscript Received on December 22, 2018.**
**Rajeev Tiwari**, Dept. of Virtualization, SCS/ UPES, Dehradun, India.
**Shuchi Upadhyay**, Dept. of Virtualization, SCS/ UPES, Dehradun, India
**Parv Singhal**, Dept. of Virtualization, SCS/ UPES, Dehradun, India.
**Ujla Garg**, Dept. of Virtualization, SCS/ UPES, Dehradun, India.
**Shefali Bisht**, Dept. of Virtualization, SCS/ UPES, Dehradun, India.

These layers are made up of neurons (interconnected nodes). As the name suggests, input layer is used to insert data into the system, hidden layers(s) is(are) used to calculate the intermediate data required to calculate the final solution and the output layer which gives us the final predicted solution. It resembles the structure of the neural network inside our brain. The process of defining a multiple level neural networks consists of first defining the number of layers and their respective number of neurons. The input layer will contain as many neurons as the number of feature component vectors. Hidden layers depend on the complexity of the problem: the more the complexity more the hidden layers. However, most of the classification problems can be solved using one hidden layer. At the end, the number of neurons in the output layer is equal to the number of output data variables (in the prediction data).

### *C. ARIMA Prediction Model and Time Series*

ANNs can be used to predict future values but for that they need input values of the. These values can be monitored using sensors which take readings on a daily basis or we can predict future input values by applying time series to the dataset. There are many existing time models that are used for this purpose. One of them is ARIMA. It uses auto regressive moving average for its prediction. This model is fitted to time series data either to better understand the data or to predict future points in the series(forecasting).

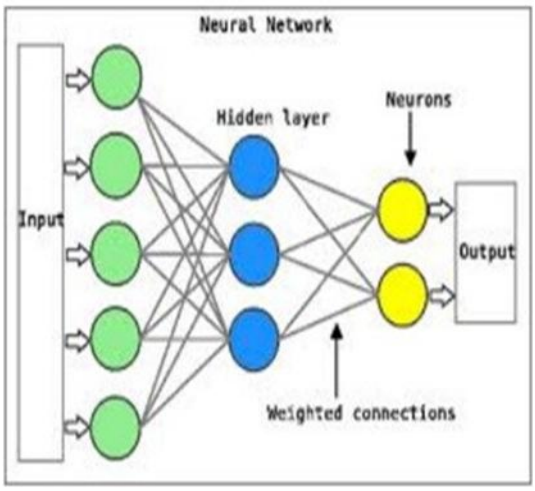


Figure 1 A three-layer neural network perceptron [1]

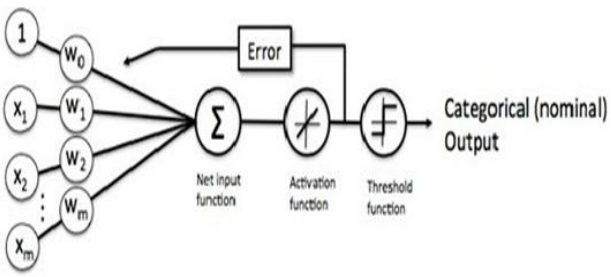


Figure 2 Adaptive Linear Neuron

## II. PROBLEM STATEMENT

To estimate the air pollution level using past data-set and to predict the increasing pollution levels after a specific time period.

## III. CONTRIBUTION

- Designing ANN Model
- Training and validation of Model using 70-30Method.
- Prediction of four components using prediction models like ARIMA etc. and pollution levels.

## IV. PROPOSED WORK

In this work, MADALINE (Multiple Adaptive Linear Neuron) model to build the neural network. MADALINE is a multiple layered neural network with a combination of multiple nodes where multiple inputs are forwarded via input and hidden layers to generate one output. Considering the following variables: x as input vector, w as weight vector, n as number of inputs and y as output of the model. The final output as per the ANN model is defined as the given equation:

$$y = \sum_{i,j=0}^{n} x_i y_j$$

*Considering some assumptions:*
l as learning rate (some positive constant), y as output of the defined ANN model and o as target (desired) output
The equation to update the weights after each iteration is define as:

$$w \leftarrow w + n(o - y)x$$

The MADALINE model converges the result to the least squares error:

$$E = (o - y)^2$$

The step-wise procedure for training the designed neural network is:

**Algorithm:**

Step 1: Extraction of historical dataset
Step 2: Data preprocessing and normalization
Step 3: Define Neural Network Structure and objectives
Step 4: Divide dataset in 70:30 ratio
Step 5: Select input and output parameters
Step 6: Present data to the environment to the neural network
Step 7: Train the neural net with 70% data
Step 8: Calculate the neural network response and the error
Step 9: If (Response within acceptance criteria): Neural Network has learnt from the environment. Go to step 10
Else:
Reconfigure the neural weights Go to step 7.
Step 10: Test Neural Network to verify it
Step 11: Input rest 30% data for testing


*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

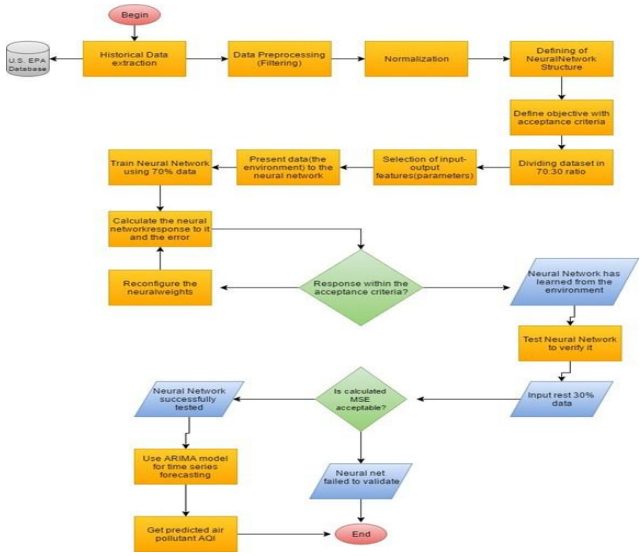


**Figure 3** Complete work flow of our proposed work

In the initial step, designing was the extraction of a historical dataset according to our requirements that would be used to train the neural network. The dataset selected was sourced from the United States Weather Department "https://aqs.epa.gov/aqsweb/airdata/download files.html" which in- cluded data from 2000-2017 of various counties of the US. Every record of the data consisted of 3 measured values of each pollutant (SO2, O3, CO2, NO2) that were Mean value, 1st Max Hourly Value and 1st Max Mean value. Then data pre-processing and normalization is done in step 2nd.The data was cut down to 2008-17 having a frequency of 1 reading per day for the Arizona county. The dataset was further processed in order to eliminate noise and resolve inconsistencies. This was done in order to reduce the complexity of the data. Then Neural Network Structure is defined in step 3. After the data was ready to be used the neural network was designed and modeled. The Neural Network is modeled including 3 layers: An Input layer, 1 hidden layer and an output layer. The input layer is defined over the inputs as the parameters for the evaluation of result. The hidden layer includes the intermediate nodes where artificial neurons take in a set of weighted inputs and produce an output through an activation function.

Now data is divided into dataset in 70:30 ratio

The dataset was divided into a 70-30 model where 70% data would be used for training the neural network and the remaining 30% would be used for validation.

Now select input and output parameters

The input layer is defined with 4 nodes: Bias, Mean value, 1st Max Mean and 1st Max Hourly value. The output layer consisted of a single output node that gave the predicted output in form of Air Quality Index (AQI) value, describing the pollution level of the specific pollutant as per the given parameters.

Data is offered to the environment to the neural network in step 6. The data set is presented to the neural network in a tabulated format with attributes as Mean value, 1st Max Mean and 1st Max Hourly value and each row defining values for different single days.

Train the neural net with 70% data as shown in step 7.

After the designing of the neural network and data preprocessing 70% of the dataset was fed to the neural network in order to train it. In next step, calculate the neural network response and the error After each iteration, neural network response and minimum squared error (MSE) was collected. The AQI value as the output calculated for every pollutant separately. The AQI value calculated is then compared to the available AQI value from dataset and processed to evaluate the minimum squared error (MSE).

Response within acceptance criteria is selected in step 9. Neural Network has learnt from the environment and now can move on step

10. Otherwise reconfigure the neural weights and go to step 7. After the complete iterations the MSE is compared to the standard de- fined as the minimum value, if the condition is fulfilled, the weights are stored and the neural network is trained. Then proceed to the test phase of neural network as per the stored weights of nodes. In order if the MSE error achieved not meet the minimum condition, the neural weights are reconfigured and trained.

Now Input rest 30% data for testing in step 10. After the training of the neural network and evaluating the weights for the neural net, it was tested and validated using the remaining 30% dataset by exposing the dataset corresponding to the evaluated weights.

In Step 11, If (MSE acceptable): Neural Network successfully test- -ed then move to step 12, otherwise Neural Network failed to validate Terminate

For testing of the neural net If MSE achieved is acceptable then the neural network has been successfully validated and can be used for future prediction else it has failed validation and the process is terminated. ARIMA model for time series forecasting is used in step 12th. If the model is successfully validated then the next step was to implement time series forecasting using ARIMA model over the entire data and get input parameter values for the desired time period. This was done using Python scripts which used libraries like pandas, NumPy, scikit etc.

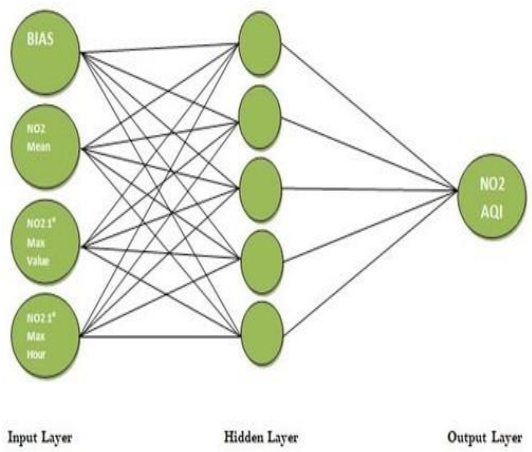


**Figure 4** Used Neural Network with hidden layer


Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Now get predicted air pollutant AQI.

The predicted input values were fed to the trained and validated neural network which gave the final predicted output.

## V. IMPLEMENTATION

This research work is divided into five different sections of implementations for getting results and accuracy:

### *A. Data Acquisition and Filtering*

During this phase, data retrieval for the training of neural network was practiced. The dataset being used is US Pollution Data (2000- 2016). It shows the daily readings (Mean, First Max, Max Hourly, AQI) of 4 major pollutants: NO2, CO, SO2, O3. The source of the data is US EPA. The data was filtered and brought down to 2010-16 as 16 years' data wasn't possible to be processed with basic computation power.

### *B. Build Neural Network*

Here a feed-forward artificial neural network (ANN) model is created to predict the Air Quality Index (AQI) for a specific region. We designed and implemented a multi-layer perceptron (MLP) with a single input layer, one output layer and one hidden layer. The single hidden layer architecture was designed for the purpose of reducing the complexity of the neural network, and increasing the required computational efficiency. The three-pollutant metrics i.e. Mean, First Max Hour and First Max Value were selected as the input parameters and AQI as the output parameter. As artificial neural network works by using inputs and output data, then it compares input and take feedback from outputs, so that correct selection of the data set is very important in building a model of neural network.

### *C. Train Neural Network*

In this module, the model proposed above was implemented using java on Netbeans IDE. 70% of the data was used to train the network.

```
run:
---------ADALINE INIT NET---------
### INPUT LAYER ###
Neuron #1:
Input Weights:
[0.09408532564009309]
Neuron #2:
Input Weights:
[0.7690597145032418]
Neuron #3:
Input Weights:
[0.10255227489671104]
Neuron #4:
Input Weights:
[0.8643927092607541]
```

Figure 5 Initial weights for Input layer of Neural Network

### *A. Validate Neural Network*

In this module, validation of the model took place. 30% of the data i.e. remaining data was used to do the final validation of the network.

### *B. Component Prediction using ARIMA*

This module consists of the use of ARIMA prediction model and Time Series to forecast the future input values. These values were inputted into the network to get the final output values of AQI.

```
### HIDDEN LAYER ###
Hidden Layer #1
Neuron #1
Input Weights:
[0.5690359650947352, 0.915254083788764, 0.07070747872990657, 0.2850155876220358]
Output Weights:
[0.9525707840366743, 0.4375435465899753, 0.7710204198067878, 0.1837233139020894, 0.7243196807146286]
Neuron #2
Input Weights:
[0.8067234318202648, 0.8753370052128812, 0.03680380251458359 5, 0.6235846977274467]
Output Weights:
[0.9093349463055436, 0.5667846393791 08, 0.8637255576163688, 0.7019360893175356, 0.3575532896745782]
Neuron #3
Input Weights:
[0.5930660547761069, 0.5253736653962158, 0.46582091466819475, 0.4468373639705435]
Output Weights:
[0.9247682508273056, 0.6156895295835738, 0.9093096396017215, 0.4309544727766120 3, 0.7000716774764661]
Neuron #4
Input Weights:
[0.4898018954581368, 0.9981855852653218, 0.7521642536472154, 0.4379898475576445]
Output Weights:
[0.9150266095047721, 0.6833979938273661, 0.4408296584776155 7, 0.01743837047756 89, 0.09015945935973413]
Neuron #5
Input Weights:
[0.4193119680506713, 0.7084977601252991, 0.22416172225404674, 0.9242239987609868]
Output Weights:
[0.2511432320119541, 0.3088941542918830 5, 0.3858657906952394, 0.17565944420604396, 0.8565827392923695]
```

Figure 6 Initial weights for Hidden layer of Neural Network

```
### OUTPUT LAYER ###
Neuron #1:
Output Weights:
[0.6781224870852657]
```

Figure 7 Weights for Output layer of Neural Network

```
---------ADALINE TRAINED NET---------
### INPUT LAYER ###
Neuron #1:
Input Weights:
[-0.03233717340457168]
Neuron #2:
Input Weights:
[0.44146632933229335]
Neuron #3:
Input Weights:
[0.7749958210003725]
Neuron #4:
Input Weights:
[0.050944873922845385]
```

Figure 8 Adaline Trained Weights of Input Layer



*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

```
### HIDDEN LAYER ###
Hidden Layer #1
Neuron #1
Input Weights:
[0.5690359650947352, 0.915254083788764, 0.07070747872990657, 0.2850155876220358]
Output Weights:
[0.9525707840366743, 0.4375435465899753, 0.7710204198067878, 0.1837233139020894, 0.7243196807146286]
Neuron #2
Input Weights:
[0.8067234318202648, 0.8753370052128812, 0.036803802514583595, 0.6235846977274467]
Output Weights:
[0.9093349463055436, 0.566784639379108, 0.8637255576163688, 0.7019360893175356, 0.3575532896745782]
Neuron #3
Input Weights:
[0.5930660547761069, 0.5253736653962158, 0.465820914668194 75, 0.4468373639705435]
Output Weights:
[0.9247682508273056, 0.6156895295835738, 0.9093096396017215, 0.4309544727766120 3, 0.7000716774764661]
Neuron #4
Input Weights:
[0.4898018954581368, 0.9981855852653218, 0.7521642536472154, 0.4379898475576445]
Output Weights:
[0.9150266095047721, 0.6833979938273661, 0.4408296584776155 7, 0.01743873704775689, 0.09015945935973413]
Neuron #5
Input Weights:
[0.4193119680506713, 0.7084977601252991, 0.2241617222540467 4, 0.9242239987609868]
Output Weights:
[0.2511432320119541, 0.3088941542918830 5, 0.3858657906952394, 0.17565944420604396, 0.8565827392923695]
```

Figure 9 Trained hidden layer Weights of Input Layer

```
### HIDDEN LAYER ###
Hidden Layer #1
Neuron #1
Input Weights:
[0.5690359650947352, 0.915254083788764, 0.07070747872990657, 0.2850155876220358]
Output Weights:
[0.9525707840366743, 0.4375435465899753, 0.7710204198067878, 0.1837233139020894, 0.7243196807146286]
Neuron #2
Input Weights:
[0.8067234318202648, 0.8753370052128812, 0.036803802514583595, 0.6235846977274467]
Output Weights:
[0.9093349463055436, 0.566784639379108, 0.8637255576163688, 0.7019360893175356, 0.3575532896745782]
Neuron #3
Input Weights:
[0.5930660547761069, 0.5253736653962158, 0.465820914668194 75, 0.4468373639705435]
Output Weights:
[0.9247682508273056, 0.6156895295835738, 0.9093096396017215, 0.4309544727766120 3, 0.7000716774764661]
Neuron #4
Input Weights:
[0.4898018954581368, 0.9981855852653218, 0.7521642536472154, 0.4379898475576445]
Output Weights:
[0.9150266095047721, 0.6833979938273661, 0.4408296584776155 7, 0.01743873704775689, 0.09015945935973413]
Neuron #5
Input Weights:
[0.4193119680506713, 0.7084977601252991, 0.2241617222540467 4, 0.9242239987609868]
Output Weights:
[0.2511432320119541, 0.3088941542918830 5, 0.3858657906952394, 0.17565944420604396, 0.8565827392923695]
```

Figure 10 Trained output layer Weights of Input Layer

```
---------ADALINE PRINT RESULT---------
1.0     0.19041667      0.49    0.19      NET OUTPUT: 0.44115285327953     REAL OUTPUT: 0.46        ERROR: -0.01884714672047
1.0     0.22958333      0.36    0.19      NET OUTPUT: 0.35769415817188 76  REAL OUTPUT: 0.34        ERROR: 0.01769415817188 75
1.0     0.38125 0.51    0.08     NET OUTPUT: 0.53529532377382 7  REAL OUTPUT: 0.48        ERROR: 0.05529532377382735
1.0     0.4026087       0.74    0.08      NET OUTPUT: 0.72297350899577 81  REAL OUTPUT: 0.72        ERROR: 0.00297350899577880447
1.0     0.4845000000000004      0.61    0.22      NET OUTPUT: 0.66558058623017 76  REAL OUTPUT: 0.58        ERROR: 0.08558058623017765
1.0     0.3995  0.73    0.08     NET OUTPUT: 0.71385116440777 91  REAL OUTPUT: 0.71        ERROR: 0.00385116440777910 63
1.0     0.29625 0.43    0.09     NET OUTPUT: 0.43628046834333565  REAL OUTPUT: 0.41        ERROR: 0.02628046834333527
1.0     0.29666667      0.41    0.0      NET OUTPUT: 0.41637945904571 58  REAL OUTPUT: 0.39        ERROR: 0.02637945904571581 4
1.0     0.25083333      0.37    0.2      NET OUTPUT: 0.37533472461943 1  REAL OUTPUT: 0.35        ERROR: 0.02533472461943103 7
1.0     0.37666667      0.7     0.2      NET OUTPUT: 0.68663452826697 62  REAL OUTPUT: 0.68        ERROR: 0.00663452826697619 2
1.0     0.585   0.81    0.22     NET OUTPUT: 0.82955781018156 42  REAL OUTPUT: 0.8         ERROR: 0.02955781018156411 5
1.0     0.49125 0.81    0.21     NET OUTPUT: 0.82297819414016 7  REAL OUTPUT: 0.8         ERROR: 0.02297819414016667
1.0     0.73285714      1.17    0.22      NET OUTPUT: 1.20914756094965 27  REAL OUTPUT: 1.04        ERROR: 0.16914756094965266
1.0     0.66541667      1.24    0.1      NET OUTPUT: 1.22751118680959 25  REAL OUTPUT: 1.05        ERROR: 0.17751118680959244
1.0     0.53166667      0.87    0.22      NET OUTPUT: 0.88782999636280 21  REAL OUTPUT: 0.86        ERROR: 0.02782999636280212
1.0     0.4575  0.73    0.19     NET OUTPUT: 0.74506014764056 5  REAL OUTPUT: 0.71        ERROR: 0.03506014764056509
1.0     0.5925  1.01    0.23     NET OUTPUT: 1.02369472693744 28  REAL OUTPUT: 1.01        ERROR: 0.01369472693744278
1.0     0.66791667      1.01    0.22      NET OUTPUT: 1.05647919867357 93  REAL OUTPUT: 1.01        ERROR: 0.04647919867357793
1.0     0.59041667      0.87    0.2      NET OUTPUT: 0.91274724573181 73  REAL OUTPUT: 0.86        ERROR: 0.05274724573181733
1.0     0.4835714 3     0.65    0.0      NET OUTPUT: 0.66489061441773 85  REAL OUTPUT: 0.63        ERROR: 0.05489061441773846
1.0     0.545   0.61    0.21     NET OUTPUT: 0.69170785041555 29  REAL OUTPUT: 0.58        ERROR: 0.11170785041555298
1.0     0.38083333      0.64    0.21     NET OUTPUT: 0.64248366784195 82  REAL OUTPUT: 0.62        ERROR: 0.02248366784195821
1.0     0.3795833300000005      0.66    0.2      NET OUTPUT: 0.65692230261107 18  REAL OUTPUT: 0.64        ERROR: 0.01692230261107175
1.0     0.53333333      0.77    0.21     NET OUTPUT: 0.81055673979518 14  REAL OUTPUT: 0.76        ERROR: 0.05055673979518138
1.0     0.42583333      0.59    0.0      NET OUTPUT: 0.61290143808809 52  REAL OUTPUT: 0.56        ERROR: 0.05290143808809511
1.0     0.27217391      0.42    0.0      NET OUTPUT: 0.41331668840330 27  REAL OUTPUT: 0.4         ERROR: 0.01331668840330269
1.0     0.33375 0.51    0.08     NET OUTPUT: 0.51432567263409 88  REAL OUTPUT: 0.48        ERROR: 0.03432567263409888
1.0     0.36875 0.54    0.21     NET OUTPUT: 0.55964970240071 02  REAL OUTPUT: 0.51        ERROR: 0.04964970240071205
1.0     0.39625 0.66    0.22     NET OUTPUT: 0.66529897371662 14  REAL OUTPUT: 0.64        ERROR: 0.02529897371662137 6
1.0     0.35208333      0.57    0.2      NET OUTPUT: 0.57503235466440 02  REAL OUTPUT: 0.54        ERROR: 0.03503235466440013 4
1.0     0.37913043      0.53    0.0      NET OUTPUT: 0.54578393099589 97  REAL OUTPUT: 0.5         ERROR: 0.04578393099589974
1.0     0.3458333300000005      0.55    0.08     NET OUTPUT: 0.55065988881532 45  REAL OUTPUT: 0.52        ERROR: 0.03065988881532444 3
1.0     0.41043478      0.64    0.08     NET OUTPUT: 0.64892887770640 17  REAL OUTPUT: 0.62        ERROR: 0.02892887770640173 4
1.0     0.4583333300000004      0.71    0.23     NET OUTPUT: 0.73196591333369 39  REAL OUTPUT: 0.69        ERROR: 0.04196591331369392 6
1.0     0.53333333      0.92    0.23     NET OUTPUT: 0.92782501042369 41  REAL OUTPUT: 0.91        ERROR: 0.01782501042369411
```

Figure 11 Results Output through Adaline

```
---------ADALINE MSE BY EPOCH---------
[2.6975512775357404E-4,
6.8037555116571824E-6,
1.1083858754622052E-4,
4.565764400319434E-4,
8.647371610334851E-4,
0.0012143630152517306,
0.001448155607115645,
0.0015625996257058094,
0.0015779867283753797,
0.0015197404319735566]
BUILD SUCCESSFUL (total time: 1 second)
```

Figure 12 Mean Square error Values

## VI. RESULTS

### A. Result Graphs

The result graphs were plotted using the scikit library of Python. The graphs show the trends in AQI over the years and the forecasted values for each input parameter.

### B. AQI Plots

The AQI plots show the trend in the Air Quality Index of each pollutant over the course of 10 years (2008-2017). These trends were used to forecast future values of AQI using the trained neural network. From the NO2 AQI graph, it can be seen that the AQI first increases and reaches a maximum and again increases before reaching a minimum. At the end of the year it dips down after achieving another maximum. In the CO AQI graph it can be seen that the AQI starts at high point in the year which gradually goes down as the year goes on. At midyear it again starts climbing and ends at a maximum. In the O3 graph it can be seen that the year starts at a low point which gradually increases as the year goes on. At mid-year a maximum is achieved following which AQI again starts going down till the end of the year. In the SO2 AQI plot the AQI from a local maximum and decreases till midyear. It then again increases till the end of the year. From these graphs it can be inferred that AQI for each pollutant followed a yearly trend with a slight variation in SO2 in the year 2009 (AQI achieves a global -maxima which is almost 3.5 times that of local maximums achieved each year).

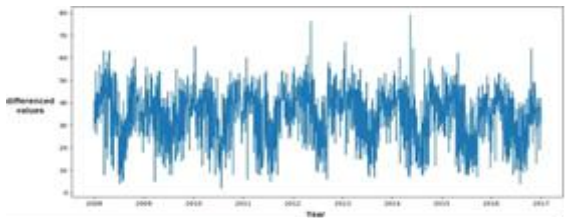


Figure 13 AQI Index of NO2

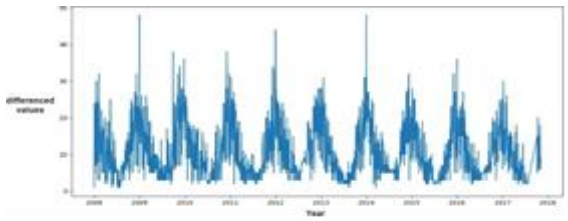


Figure 14 AQI Index of CO

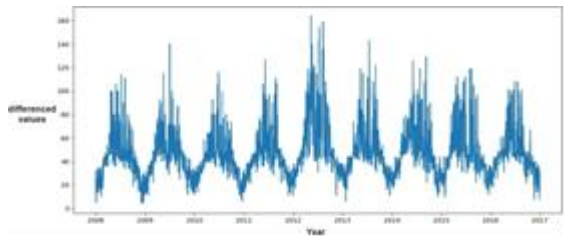


Figure 15: AQI Index of O3


Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

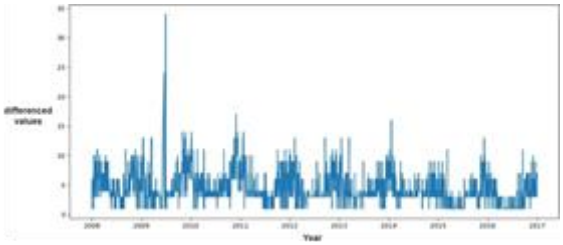

Figure 16 AQI Index of SO2

### C. Forecast Graphs

Each graph shows the forecast of each input parameter for all the pollutants (NO2, CO, O3 and SO2) calculated using ARIMA prediction model for the first week of April 2018 (01-04-2018 to 07-04-2018). These predicted values were used to forecast the final end AQI for each pollutant. The parameter values are stored as the parts per million form.

### D. Forecast Graphs for NO2

The following three graphs show forecasted values of three different parameters i.e. Mean, 1st Max and 1st Hourly Max for a week (01-04-2018 to 07-04-2018). The first graph shows 1st Max value of NO2 while second one shows 1st Hourly Max and the third one shows the mean value in parts per million (ppm) corresponding to each day of the week (fig. 17, 18 and 19).

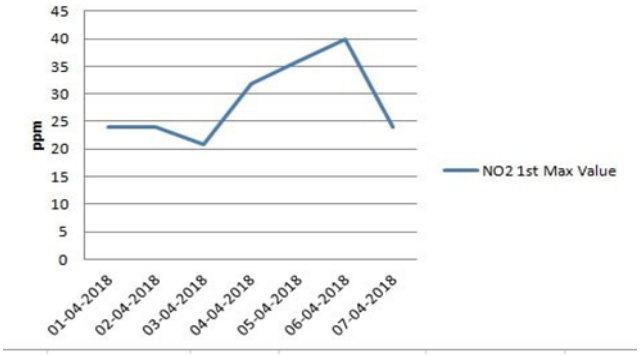


Figure 17 Forecast week NO2 max1

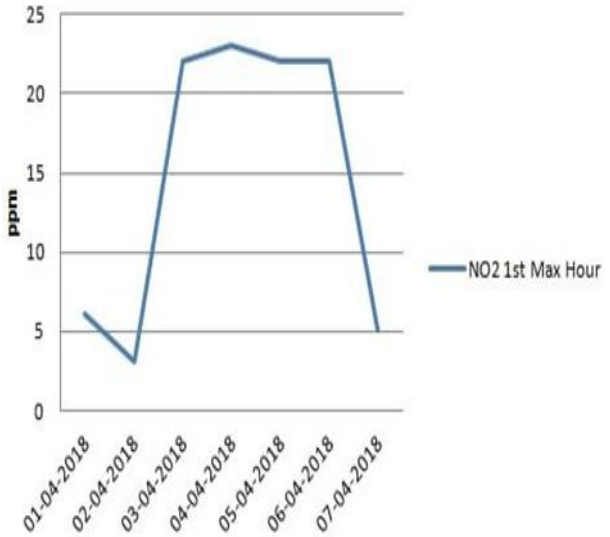


Figure 18 Forecast week NO2 max2

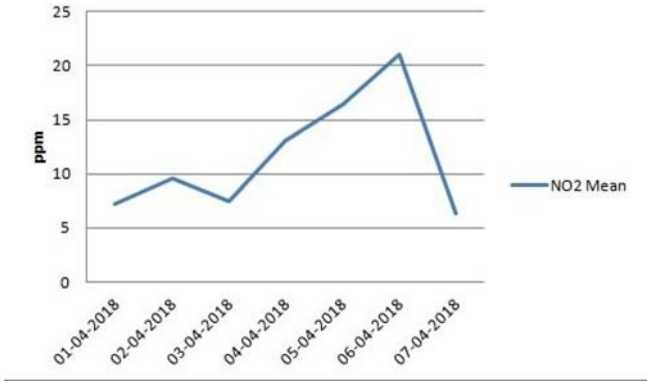


Figure 19 Forecast week NO2 Mean

### E. Forecast Graphs for CO

The following three graphs show forecasted values of three different parameters i.e. Mean, 1st Max and 1st Hourly Max for a week (01-04-2018 to 07-04-2018). The first graph shows 1st Max value of CO while second one shows 1st Hourly Max and the third one shows the mean value in parts per million (ppm) corresponding to each day of the week (fig. 1, 22 and 23).

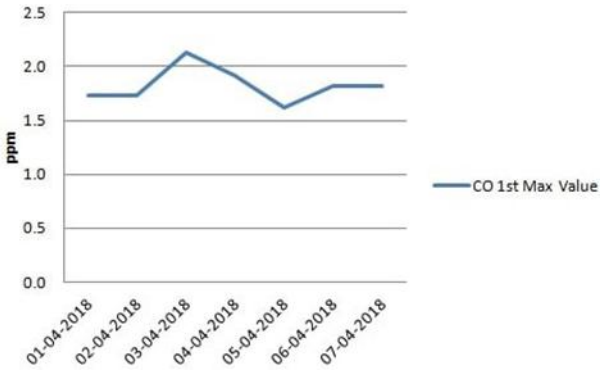


Figure 20 Forecast week CO max1

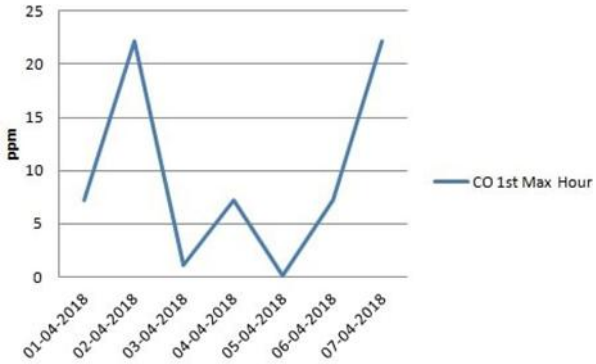


Figure 21 Forecast week CO max2


Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

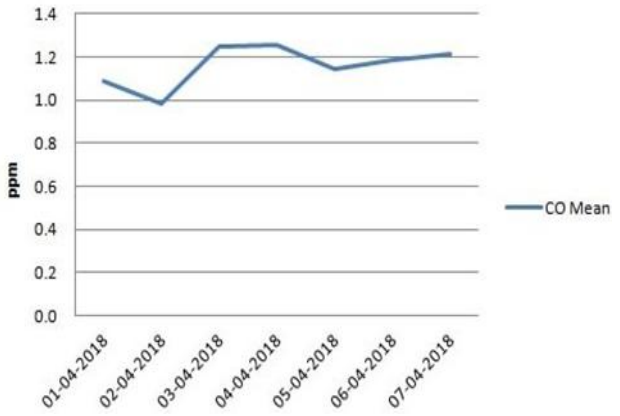


Figure 22 Forecast week CO Mean

*F. Forecast Graphs for O3*

The following three graphs show forecasted values of three different parameters i.e. Mean, 1st Max and 1st Hourly Max for a week (01-04-2018 to 07-04-2018). The first graph shows 1st Max value of O3 while second one shows 1st Hourly Max and the third one shows the mean value in parts per million (ppm) corresponding to each day of the week (fig. 23, 24 and 25).

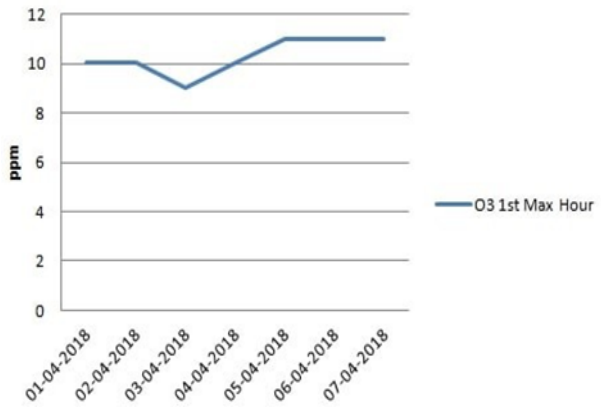


Figure 23 Forecast week O3 max1

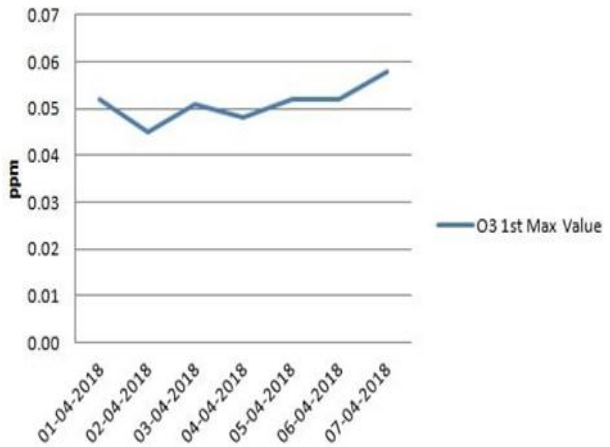


Figure 24 Forecast week O3 max2

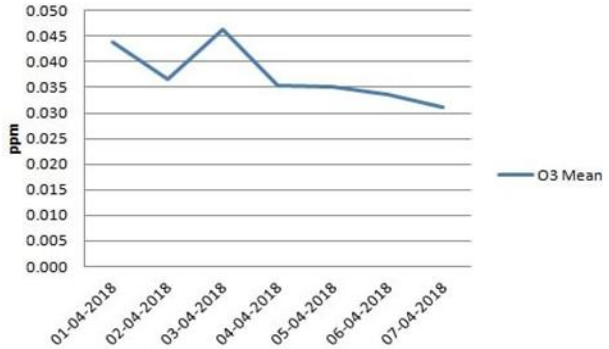


Figure 25 Forecast week O3 Mean

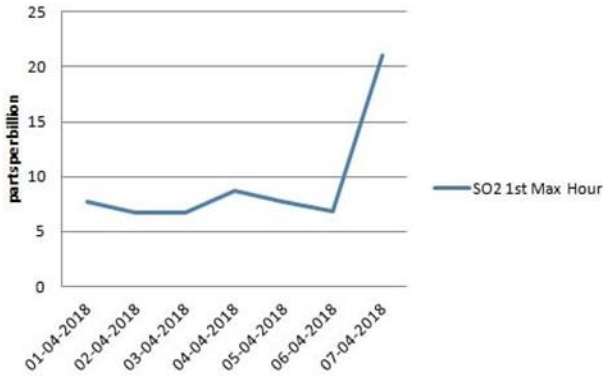


Figure 26 Forecast week SO2 max1

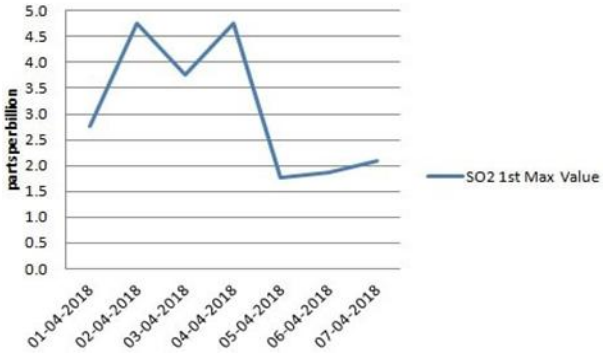


Figure 27 Forecast week SO2 max2

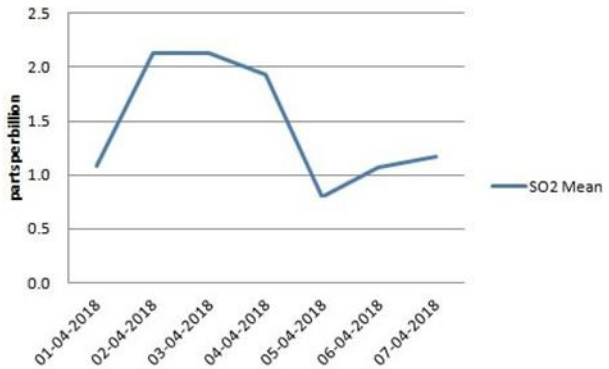


Figure 28 Forecast week SO2 Mean


*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

## VII. CONCLUSION

The neural network model for the forecasting of AQI is sufficiently effective. The values of Mean Square Error (MSE) (in order of 10-8) prove the adequacy of the offered model. There is also a possibility of further refinement in the results if additional parameters are also included in the daily experimental data.

## REFERENCES


1. Air Pollution – Monitoring, Modelling and Health, Edited by Mukesh Khare p. cm. ISBN 978-953-51-0424-7, InTech Janeza Trdine 9, 51000 Rijeka, Croatia
2. Finardi, S.; de Maria, R.; D'Allura, A.; Cascone, C.; Calori, G.; Lollobrigida, F. A deterministic air quality forecasting system for Torino urban area, Italy. Environ. Model. Softw. 2008, 23, 344–355.
3. Pai, T.Y.; Ho, C.L.; Chen, S.W.; Lo, H.M.; Sung, P.J.; Lin, S.W.; Lai, W.J.; Tseng, S.C.; Ciou, S.P.; Kuo, J.L.; Kao, J.T. Using seven types of GM (1, 1) model to forecast hourly particulate matter concentration in Banciao City of Taiwan. Water Air Soil Pollut. 2011, 217, 25–33.
4. Comrie, A.C. Comparing neural networks and regression models for ozone forecasting. J. Air Waste Manag. Assoc. 1997, 47, 653–663.
5. Schlink, U.; Dorling, S.; Pelikan, E.; Nunnari, G.; Cawley, G.; Junnine, H.; Greig, A.; Foxall, R.; Eben, K.; Chatterton, T.; et al. A rigorous inter-comparison of ground-level ozone predictions. Atmos. Environ. 2003, 37, 3237–3253.
6. Yi, J.; Prybutok, V.R. A neural network model forecasting for prediction of daily maximum ozone concentration in an industrialized urban area. Environ. Pollut. 1996, 92, 349–357. 14.
7. Grivas, G.; Chaloulakou, A. Artificial neural network models for prediction of PM10 hourly concentrations, in the Greater Area of Athens, Greece. Atmos. Environ. 2006, 40, 1216–1229.
8. Małgorzata Pawul1 , Małgorzata S´liwka, 2016, 'Application of artificial neural networks for prediction of air pollution levels in environmental monitoring' AGH University of Science and Technology, Journal of Ecological Engineering.




*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*