SYSTEM IDENTIFICATION AND TIME SERIES PREDICTION BY USING EVOLUTIONARY COMPUTING TECHNIQUES

MUHAMMAD FAHAD MUNIR

Reg. No. 80-FET/MSEE/F07

Submitted in partial fulfillment of the requirement for the

Master of Philosophy degree in Electronic Engineering

with specialization in Signal and Image Processing

at the faculty of Engineering and Technology

International Islamic University,

Islamabad.

SYSTEM IDENTIFICATION AND TIME SERIES PREDICTION BY USING EVOLUTIONARY COMPUTING TECHNIQUES

SUBMITTED BY:

MUHAMMAD FAHAD MUNIR

Reg. No

80-FET/MSEE/F07

SUPERVISED BY:

DR. IHSAN UL HAQ

Assistant professor

Department of Electronic Engineering, Faculty of Engineering and Technology

International Islamic University, Islamabad.

2012

CERTIFICATE OF APPROVAL

Title of Thesis: System Identification and Time Series Prediction Using Evolutionary Computing techniques.

Name of Student: Muhammad Fahad Munir

Registration No: 80-FET/MSEE/F-07

Accepted by the Faculty of Engineering and Technology, INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD, in partial fulfillment of the requirements for the Master of Philosophy Degree in Electronic Engineering

Dean of Faculty
Dr. Ghulam Yasin Chohan
Professor & Dean

External Examiner
Dr. Abdul Jalil
Assistant Professor
PIEAS

Internal Examiner
Dr. Ijaz Mansoor Qureshi
Professor
DEE/FET-IIUI

Supervisor
Dr. Ihsan Ul Haq
Assistant Professor
DEE/FET-IIUI

Chairman of Faculty
Dr. Muhammad Zubair
Assistant Professor & Chairman

In the name of Allah (SWT) the most benificient and the most mercifiul.

DECLARATION

I certify that except where due acknowledgments has been made, the work is that of my alone: the work has not been submitted previously, in whole or in the part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

MUHAMMAD FAHAD MUNIR

80–FET/MSEE/F07

ABSTRACT

System identification is one of the challenging tasks in the current era due to rapid development of economy specially when we talk about Air Pollution prediction, researchers have put their emphasis in developing latest algorithm to predict the future response and made victories, the previously developed methods used on the datasets of Air pollutions use conventional methods to obtain the results like Artificial Neural Networks. In this thesis author exploits the potentials of Hybrid nature of Evolutionary computing techniques like Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) with Artificial Neural Networks (ANN), and presented the comparison. The proposed algorithm uses the less number of iterations, and uses less time to produce results as compared to Back propagation neural network.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

With the rapid development of economy, the size of the cities grow day by day, more and more people are migrating toward cities from countryside's in order to earn high, and spend luxurious life. This also cause increase in vehicles to move around, and factories to produce products. In order to facilitate all the communities in the country, we have to increase the number of industries and vehicles too. As the increase in economy benefits the humanity, on the other hand it has some harmful impact on life too, and we have to face series of severe issues regarding environmental pollution, like air pollution, shortage of land, wastes of the homes, disposal of the sewerage and noise of the traffic [1]. The proper care is needed to protect the communities from these issues caused by the vehicles and industries. It is seen from the research that there are many harmful ingredients in the vehicle and factories exhaust, which causes the Lungs Cancer, Asthma, cardio vascular and Skin diseases in human beings [2]. This is not the only issue, on the other hand more and more people get sick and they need hospitalization. Thus government have to make more and more hospitals and heir more staff, because to provide health facility, is the responsibility of government. Thus, among all the issues discussed above, the pollution in the air is directly related to human life

because we all inhale the natural air, if the concentration of prohibited pollutant level in air is low or fall within the scale which is not harmful for our life, we will spent happy life and need less hospitalization, on the other hand if the air of our community is not clean and fall into the category of objection, we will be directly victims of air embolism [3]. Air pollution control in long term will help us to put check on the entire situation and will prevent the situation to being worst. On the other hand, short term monitoring will help us to inform our community to take care of their everyday life, and properly scheduling the daily time table will protect us from being sick [3]. Thus by implementing the latest engineering research in the environmental sector will lead us to design such a power full tools which will put checks on concentration levels of air, and will help the humanity.

1.1 PROBLEM STATEMENT

In the last decade, work in the field of Air pollution concentration was done with the help of Auto Regressive Moving Average (ARMA) model, Support Vector Machine (SVM), and Artificial Neural Network (ANN). The ARMA model needs specific model parameters, and its shows good results, but when there is slightly change in the data set used, the results become poor and poor [4], similarly the SVM also uses the Statistical risk minimization concept which is also very difficult to implement and needs much computational time [5]. But the most prominent results were obtained from the ANN, by using its different flavors. However, there seems some problems related with the system Stability, Network Topology and proper selection of Transfer Function, because for some data sets the ANN performs better and some time it get stuck in the local minima, and we get diverted to some other points which were beyond our scope [6].

1.2 PROPOSED MECHANISM

As we have seen that the Evolutionary Computing (EC) performs better and now become the hottest topic in the field of research and development, and also it needs less mathematical knowledge as compared to other said research mechanisms. So I proposed two different techniques, i.e. Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) based Hybrid Neural networks. And will calculate its comparison in the standards of MSE. The results will show that PSO based Hybrid Neural Network technique (PSO-NN) performs much better than basic ANN, and also this performance was ranked higher than GA based Hybrid Neural Network methodology (GA-NN).

1.3 CONTRIBUTION OF DISSERTATION

This thesis is helpful in the following field of studies'

1.3.1 Environmental Sector

This thesis work is very helpful in the field of calculation of concentration levels of different air pollutants in air.

1.3.2 Medical Health Sector

This thesis is also helpful in the field of medical because the concentration of air pollutants in air can lead us towards the Lung Cancer, Skin diseases, Asthma infection to under age children's especially for infants', and also they can harm the cardiovascular system of human being. We can develop safety precautions like avoiding hiking or travelling and any other activities in those days when their concentration level exceeds their defined limits, and also we will be able to put checks on the root cause of these pollutants.

1.3.3 Industry and Traffic Maintenance Sector

We can also put checks on the flow of traffic causing pollution in the air because the main cause is due to the diesel exhaust from the vehicles, smoke coming out of brick kilns and fences of factories. New technology and latest equipment should be introduced to vehicles and on factories through concerned authorities.

1.4 ORGANIZATION OF THE THESIS

This thesis consists of three sections. First section consist of two chapters, having names chapter 1 **Introduction** and chapter 2, **System Identification for Time Series Prediction.** In this section we will discuss about the thesis overview, the technique upon which research will be done and the different aspects of data and its usage in order to obtain results. The second section contains the literature review in chapter 3 **System Identification Techniques**, and proposed mechanism in chapter 4 **Evolutionary Models For System Identification,** In the first part of this section we will discuss the adaptive behavior of the existing technology which was used in the last decade i.e. ANN, and then proposed methodology will be introduced in the next chapter 4. In this section two different approaches were used which are inspired from nature known as GA-ANN and PSO-ANN. In the last section, we also have two chapters, i.e. chapter 5, **Simulation and Results**; the simulation results will show us the behavior of different algorithms and their performances. While Conclusion of all the work and future contribution is discussed in chapter6 with the name **Conclusion and Future Work.**

CHAPTER 2

SYSTEM IDENTIFICATION FOR TIME SERIES PREDICTION

System identification and Time Series prediction (TSP) are the one of the most prominent topic of current ear's research. By using this terminology, we mean that, to formulate the linear or non-linear data into frame work of time. In real time data, we get corrupted data due to noise factor, due to which it is very difficult to detect our desired response, i.e. actual data; hence, we have to design such an efficient frame work which suits best for all types of similar data. This technique is also known as reverse modeling or reverse engineering because we do not know the behavior of our system and on the basis of output or system response, we can formulate our network and adjust the weights.

In the previous chapter, we have only explained the techniques that how to implement it on our selected data. This is called the Network training, and in results and discussion chapter, we will show our achievements. In this chapter, we will discuss how to find the future response of our trained algorithms. All the process will be done step by step, and the current value will tell us about the future value, once the value be predicted, the first value from our

filter or window will be discarded and that predicted value will be impended. This will lead us to predict another future value, and at that step we will discard the 2nd value from our filter's beginning, and insert the new value to it. Hence for the prediction of third value, we will have two values which we calculate and rest of the values will be from training section. The approach we followed in our thesis is explained below.

2.1 MULTI STEP DATA PREDICTION

In this technique, we define the size of filter or window which will be used to calculate the future response. The size of window is user dependent and may vary from problem to problem. All the observed data is saved in the form of vector and recorded at on some particular time 't'. Consider the following row vector which shows the past values being estimated from actual data sets by updating the weights of the system and by calculating the MSE through minimizing the error term.

$$X = [x_1, x_2, x_3, x_4, \dots, x_t]^T \tag{2.1}$$

In short, we can write sequence of past values as X_{t-p}^{t} as time series sequence and its predicted values will have form $[x_{t-p}, x_{t-p+1}, x_{t-p+2}, \dots, x_t]$, we will predict h step ahead and the form gets X_{t+1}^{t+h}, where p is the past values and h is the future values, the length of window will be p + h. each and every value in this window belongs to the values in the training set

2.2 PREDICTION METHODOLOGY

We will make the training set for future correspondence with the help of sliding window of length p + h from the given data set X. All the inputs in this are from previous data which was being used for training the proposed algorithms, while the output vector Y contains the remaining h values. The first window which will be made will look like this

$$X = [x_1, x_2, x_3, \dots, x_p] \quad (2.2)$$

$$Y = [x_{p+1}, x_{p+2}, x_{p+3}, \dots, x_{p+h}] \quad (2.3)$$

Similarly, for the second updated value, the input, output ratio will be as follow,

$$X = [x_2, x_3, \dots, x_p, \tilde{x}_{p+1}] \quad (2.4)$$

$$Y = [x_{p+2}, x_{p+3}, \dots, x_{p+h}, \tilde{x}_{p+h+1}] \quad (2.5)$$

Thus by increasing in this manner we will reach to the final stage and the values in our last window will be as fallow:

$$X = [x_{t-h-p+1}, x_{t-h-p+2}, \dots, x_{t-h}] \quad (2.6)$$

$$Y = [x_{t-h+1}, x_{t-h+2}, \dots, x_t] \quad (2.7)$$

Generally, we will use the term(Y_i), which will show all the instances in the i^{th} level of output vector Y. in short, we can write that we will predict the first value x_{t+1}, then x_{t+2}, and this procedure will be followed up to last value x_{t+h}. The values once predicted will be compared with the actual dataset and the MSE will be calculated which will be standard of the merit.

2.3 ACTUAL DATA SET

In this thesis we will first use the data set which contains the information about the air quality measured at Mong Kok station in Hong Kong in year 2010. This database contains seven major ingredients which are carbon monoxide (CO), nitric oxide (NO), nitrogen dioxide (NO_2) sulphur dioxide (SO_2), Nitrogen oxides (NO_x), ozone (O_3), and respirable suspended particulate (RSP) including particulate materials PM_{10} and $PM_{2.5}$. These ingredients values which are stored in the database are in numerical form, and contain the quantities of the particles present in the air, the information about 24 hours a day throughout the month is

stored properly without any missing or overwriting and are available at their official website for research and development purposes [11].

Among this available dataset, the months of January and June represents the two different seasons in Hong Kong. Therefore we will use these values in the system forecasting experiments. The weather in the month of January is dry and cold having strong winds from the north-east portion of the country, while the weather in the month of June is hot and wet, and under goes with the channel of winds from the south-east. The winds blown in the dry January month contains very large quantity of pollutants including the auto mobile vehicle exhaust and black smokes from the factories fences present in the mainland of china, which combines with the natural air and migrates to this area, Which is very dangerous for the health according to medical point of view. On the other hand, the winds in the June are hot due to which the pollutants pinched in the air are at their lowest level [12].

The main pollutants which are harm full for our life, health, infect our air are Particulate materials (PM_{10} & $PM_{2.5}$) these particles are also known as Respirable Suspended Particles (RSP) because their size fall in the category of size less than 10 nanometer to 2.5 micrometer , Carbon Monoxide (CO), Nitrogen dioxide (NO_2), Ammonia (NH3), Oxides of Sulphur (SO_x). The other factors which indirectly affect the air are Speed of wind (WS), direction of wind (WD), indoor temperature (IT), outdoor temperature (OT), and solar radiation (SR). The first five mentioned ingredients are the air pollutants while the last five are meteorological parameters [7]. Medical research shows that the particles having size greater than 10 micrometer are not hazards for human health, on the other hand, particles having size 10nm to 10 micrometer are much dangerous particle for health because they are inhaled legitimately into human respiratory system. In long term, the impact of these venomous materials on human life causes Cancer, Pulmonary diseases, Asthma, Cardiovascular diseases and Respiratory infection [8], these Particulate Materials which are composed of dust

particles due to cyclones, diesel vehicle exhaust, and smoke of the factories from Chemnies, Brick kilns, burning of municipal wastes and Smog.

2.4 INTERNATIONAL STANDARDS

Area's having more congestion, have high temperature as compared to those having less. Thus air in those areas rises directly towards sky, and the air in the surrounding came to that region, the speed of wind carry these particles with them and they come directly to our homes. That air pollutants also cause bad impact on our crops, soil, land, and made their nature acidify. In recent research, it is observed that ozone and oxides of Sulfur are main ingredients of polluted air which cause the decline of crops yield. [9] Report shows that every year three million people die due to air pollution. The following table made under [10], will tell us the RSP concentration in air, and also shows their impact on human life.

According to [10], the daily outdoor concentration for air pollutants are defined, if RSP or any other ingredients exceeds their limits, then that will be dangerous for health and especially for infants, asthma patients and those who are sufferings from respiratory diseases. The standard is made to protect human life [10].

The averaging of different particles is different for some cases. For example if we talk about the averaging of ozone, we have two main standards, one is 8 hour averaging and in some cases we will fallow 10 hour averaging, while for some other ingredients, we have 3 year averaging, and the actual concentration is in $\mu g/m^3$. From the above table, we conclude, that if the concentration level of RSP (PM10, PM2.5) is $150\mu g/m^3$, if this limit is crossed, then we have to face irritations and skin diseases.

2.5 METHODOLOGY

We will perform our initial experiments on RSP and then NO_x and SO_x data sets. System training will be done on the values from the start of the month to the first ten days in the month of January and June. Thus we will have 240 data points for training our model and on the basis of these results, we will built our new model which will predict the future behavior of that system, thus the error between actual values and calculated ones will be finding with the help of Mean Square Error (MSE).

CHAPTER 3

SYSTEM IDENTIFICATION TECHNIQUES

The idea for prediction of air quality by using time series prediction is one of the hottest topics of the current research, however, in the past some researchers have put their emphasis on this topic but the work done is very less due to non availability of proper modeling of data and algorithms. As the trend of using the statistical methods for prediction of Air quality along with traditional modeling is changing day by day [1]. New techniques are evolving and different comparisons have been made in between old statistical techniques and newly evolved one [13]. The comparison of NO2 is made with linear regressive models and NN [14][15] which conclude that NN can perform better. [16] Worked on flow of carbon pollutants in air, and made a NN based model. NN based GIS data base is developed for forest development project and hydro electric power plant on river Drava. [17][2] Presented case studies covering the pollutants concentration level in ambient air including RSP, CO, NO, SO, Ozone. Multilayer Back propagation neural network model was proposed by [18] for the prediction of RSP level. They used the data set having every day meditation of average values for 7 years. In order to have daily based prediction model for particulate material, NN were supposed to be good tool [19]. Calculation of rain fall at the different

locations of the country, NN performed well [15] in the comparison of linear model. The tracking and training was done with the help of Radial Basis Function (RBF). Another example from the field of accounting is that it was used to find the future response Bankruptcy due to credit flow [19]. Market value index over flow and under flow calculation and their daily assumptions was calculated through NN [20]. The results obtained through Moving Average model (MA) was compared with FFNN and Multilayer Preceptron (MLP)

3.1 THE NEURAL NETWORK

The idea behind using the neural network as a system identification tool is that a single neuron can act as an adaptive filter which updates its status by updating its previous values in a linear mode. We begin our discussion from the theoretical background and will finish it up to implementation phase. The idea was taken from the biology, and based on the structure on neurons present in the human mind. As neurons are used to store information and all the ideas take place here, so researchers have taken the idea and made some mathematical model based on the structure of those neurons. According to biologists, normal human brain contains ten billion neurons [21]. A rough estimate is made is that they have six thousands interconnections with each other, such that they can communicate with each other. The idea of mesh network was also taken from the neurons of the brain [22]. If we say that neuron is more likely to be simple silicon gate, then our mind or human brain is complex with non linear parallel computing device [23]. In the recent years research has been made on the quick performance of the neural networks and the computers and we concluded that processing through silicon gate is six times faster than human brain, i.e. the decision making in brain takes place milliseconds whereas in silicon gate chip, it takes nanosecond [24].

3.2 STRUCTURE OF NEURON

Almost all the systems are based on this working. In human brain when we perceive something, it is converted into electrical signals through receptors and all the electrical signals are then converted into the impulses, this impulses then activate the neurons present in the brain, and hence all the related neurons or network of neurons is activated [22][23]. When the necessary work has been done, impulses are then transmitted to Effectors to do the act. If we summarize all the above discussion into one sentence, then we can say that, input has been given to the brain, necessary work or processing has been done by neurons and output is produced. Now consider a system, in which we give input and each input is processed with some proper order, if we call the order as the weight of that input, and by representing it mathematically, then we have the following model.

$\{U_1, U_2, U_3, \ldots, U_n\}$, Are the inputs to the system, and the way we deal it, i.e. weights assigned to it are $\{W_1, W_2, W_3, \ldots, W_n\}$, and the instructions to be processed will be

$$Y = \sum_{k=0}^{n} U(k) * W(k) \tag{3.1}$$

Where, Y is the output of the system. If we change the wording, we will come to know that system is the brain, and the W is the impulses generated by the brain when it receives the signal of U. The graphical representation of that artificial system is

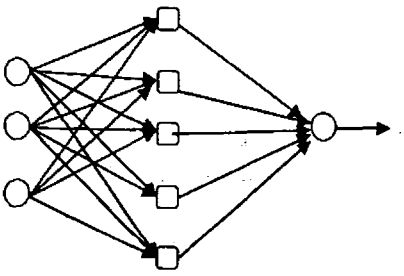

Figure 1: Architecture of Feed Forward Neural Network

All the work done in neurons takes place in two steps, in first phase, it is trained on the data set, i.e. it should give some self knowledge on the basis of which it can decide, and when it have enough memory, it is allowed to take necessary actions. In the second phase, appropriate decision is made, and in the mean while it updated its knowledge which is very helpful in decision making. There are two most famous types of neural networks which are Feed Forward neural network and Back Propagation Neural network.

3.3 TYPES OF NEURAL NETWORK

In the above sections we have discussed in detail about the history and basic structure of neuron. Now we have to explain its behavior and working methodology, its constraints formulation and implementing our daily life problems and find their solution. Many flavors have been developed so far, so we need to explain some of them because we need them in out thesis. The two main types of neural network are the feed forward neural network and back propagation neural network. The detail is stated below.

3.3.1 Feed Forward Neural Network

In this type of neural network we will only consider the flow of communication in the forward direction. It is further divided into two sub parts. One is called single layer and other is called as multi layer network.

In the single layer feed forward neural network (FFNN), there is only one input layer and only one output layer. This is also called as SISO, single input and single output network. Only one input is taken by neuron, it is processed and result is sent to the output layer. And there is no hidden layer present. The second type of FFNN is multilayer; in this type we have many hidden layers present among the input and output layers, these hidden layers works as memory of the system [22][21]. The system having more hidden layers, it is more robust and

performs well to stochastic systems [25]. The main function of the hidden layers is to dig out higher order information from the structure. Consider we have multilayer NN, the input from the source will become the output to the 1st hidden layer and the result generated by this hidden layer will be considered as the input to the 2nd hidden layer and so on, when all the hidden layer finishes, the result will be processed to the output layer.

3.3.2 Back Propagation Neural Network

The most famous type of neural network is the back propagation neural network (BPNN). This is used to calculate the weight vectors of multi layer NN [26]. The main idea behind this is to calculate the partial derivative of the function to be simplified with respect to the changing nature of weights. In the previous section, we have discussed the Feed forward NN in detail. Though FFNN has many advantages, but on the other hand, it has some limitations too. Those limitations can be overcome by using two layers FFNN [21]. But we got some problems in weights adjustment from input layer to hidden layer [26]. In order to overcome this problem, a new solution is proposed which is named as BPNN [27]. The main theme of this new proposed idea was that by determining the error of hidden layer with respect to the output layer. Therefore this idea is known as back propagation. We can also say that it is the implementation of delta rule in generalized way for non linear and multilayer NN. In recent years, we can apply the BPNN in multi layer other than two layers NN [22]. As we have mentioned that BPNN is the isolated form of delta rule, let's discuss the rule briefly. When we understand the learning pattern, the values of weights being transmitted to the output layer, and then the actual output value is being computed with the help of desired response. All the computations are done in same manner and at the end of the day, we have one vector of Errors which we obtain in the updating of all the weights and this error is the minimal error, and thus we will get the original statistics of the network. For any particular layer, let

us call this error e_o, and the output layer is o. In order to bring $e_o = 0$, we will use greedy method [28], and according to delta rule we can mathematically write it as fallow

$$\Delta w_{ho} = (d_o - y_o)y_h \tag{3.2}$$

By defining only in one step, we cannot reach optimistically, and we will have our input to hidden layer weights never changed, we cannot achieve the full power of NN. In order to update the weights of W_{ih}, we again need to apply the delta rule but with little amendments. Thus we can conclude that delta rule is applied on NN in two steps, in step one, we give input $U(n)$ to the network, and which is processed forward direction to the output layer and used to compute the error with the help of desired response, and in second step, that computed error is passed back to each layer of network in such a way, that necessary changes should be made in the currents weights. Thus, we concluded from this discussion, that back propagation NN is the way to update all the weights of a network in appropriate manner [28]. The multi layered NN i.e. two hidden layer BPNN is shown in Figure 3.1

3.4 THE NEURAL NETWORK ALGORITHM

In this section, we will define the basic steps of the BPNN algorithm, and will summarize all the discussion in mathematical form. The entire algorithm consists of three basic steps, which are input vector, output vector and black box. All the power of algorithm is enfolded in this black box. First of all we need to do the following basic steps which are as fallows

3.4.1 Initialization

We need to initialize all the weight vectors and assign them some random values. This assignment is problem dependent; in some cases we do assign zeros as initial value. Following points should be very clear in the mind while starting the algorithm.

The algorithm should start from the arbitrary setting of weights on neuron.

The adjustment of the synaptic weights of neuron should be in continuous manner.

The adjustment of the weights updating should lie within the time interval which is equal to sampling period of the signal.

The initialization steps should contain the two nodes of neurons; one is for input to hidden layer and second is fro hidden to output layer. The order of both the layers may not be same because, greater the number of neurons at hidden layer will increase the effectiveness of the algorithm. Here we have three elements which should be initialized, which are

No. of Inputs $1 \leq i \leq I_1$

No. of Hidden Elements $1 \leq j \leq J_1$

No. of Output Elements $1 \leq k \leq K_1$

3.4.2 Feed Forward (Training Of Pattern)

In this case, we need the set of input values which are to be used as training the model and next value is used as desired response. And this is used to update the memory of model. Also we need another set of data which will be used to test the model whether it predicts the system properly. Here we need to explain that if we use same set of data which is used for training the model, the NN will not give us correct results. And it will be difficult for us that either the output is predicted one or he just gives us the values he memorized.

The net-output at hidden layer will be as fallowed

$$NET_j^h = \sum_{i=1}^{I_1} W_{ji}^h x_i + W_j^{h\emptyset} \qquad (3.3)$$

Where,

The $W_j^{h\emptyset}$ is the biased element, we will calculate the net output at hidden layer for each and every element at input layer.

The net output at the output layer will be

$$NET_k^o = \sum_{j=1}^{J_1} W_{kj}^o i_j + W_k^{o\emptyset} \tag{3.4}$$

Where, the $W_k^{o\emptyset}$ is the biased weights at the output layer.

3.4.3 The Back Propagation Step

In feed forward step, we will present the input to all the layers of model and thus we need to calculate the error vector, i.e. what we need and what we got. This error will be used to update the network and will lead us to the proper direction in which we need to go. The weight updating equations are based on the delta rule or steepest decent algorithm, as written below

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \Delta w_{kj}^o \tag{3.5}$$

$$\Delta w_{kj}^o = \eta \delta_k i_j \tag{3.6}$$

$$\delta_k = (y_k - O_k)\mathcal{F}(NET_k^o) \tag{3.7}$$

$$\mathcal{F}(NET_k^o) = (1 - O_k)O_k \tag{3.8}$$

So, by combining above equations, we have the weight updating equation.

Weight updating equation for output layer

$$W_{kj}^o(t+1) = W_{kj}^o(t) + \eta(y_k - O_k)(1 - O_k^o)i_j O_k \tag{3.9}$$

Similarly, input to hidden layer, the weight updating equation contains the following steps.

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \Delta w_{ji}^h \tag{3.10}$$

$$\Delta w_{ji}^{h} = \eta \delta_j x_i \tag{3.11}$$

$$\delta_j = \sum_k W_{kj}^{o} \delta_k \mathcal{F}(NET_j^h) \tag{3.12}$$

$$\delta_k = (y_k - O_k)(1 - O_k)O_k \tag{3.13}$$

$$\mathcal{F}(NET_j^h) = (1 - i_j)i_j \tag{3.14}$$

So by combining, we get

$$\delta_j = (\sum_k W_{kj}^{o}(y_k - O_k)(1 - O_k)O_k)(1 - i_j)i_j \tag{3.15}$$

Put this value of δ_j in equation

$$\Delta w_{ji}^{h} = \eta(\sum_k W_{kj}^{o}(y_k - O_k)(1 - O_k)O_k)(1 - i_j)i_j x_i \tag{3.16}$$

So, the overall equation for input to hidden layer weights updating is

$$W_{ji}^{h}(t+1) = W_{ji}^{h}(t) + \eta(\sum_{k=1}^{K_1} W_{kj}^{o}(t)\,(y_k - O_k)O_k(1 - O_k))(1 - i_j)i_j x_i \tag{3.17}$$

Weight updating equation for biased output layer:

$$W_k^{o\theta}(t+1) = W_k^{o}(t) + \eta(y_k - O_k)(1 - O_k)i_j O_k \tag{3.18}$$

Weight updating equation for biased Hidden layer:

$$W_j^{h\theta}(t+1) = W_j^{h\theta}(t) + \eta(\sum_{k=1}^{K_1} W_{kj}^{o}(t)\,(y_k - O_k)O_k(1 - O_k))(1 - i_j)i_j \tag{3.19}$$

3.4.4 Activation Function

The activation function is the objective function of any program, if we properly model the problem in the light of mathematics, we can choose the best suited activation function or objective function. This is also called as the constraint of the system. Generally we use three

different types of activation function in NN, which are threshold function, piece wise linear function and sigmoid function. The most widely used activation function in neural networks is sigmoid function, which is self increasing function and it also provides excellent balance in linear and non linear systems. Mathematically we can write it as

$$f(\mathrm{x}) = \frac{1}{1 + e^{-ax}} \tag{3.20}$$

Where α is the slope parameter of the equation, greater the value of α, greater will be the slope. It is also observed that when we put infinity at the place of slope parameter, we get the simple threshold function [11]. But the main difference between both the activation functions is that sigmoid function can be differentiated where as threshold function cannot. Here, we have two activation functions, one is used for the calculation of non linear output of the hidden layer and second is used for the calculation of non linear output at the output layer, the mathematical form of both the functions are as fallow.

Non linear output at hidden layer

$$i_j = \frac{1}{1 + e^{-NET_j^h}} \tag{3.21}$$

Non Linear output at Output layer.

$$o_k = \frac{1}{1 + e^{-NET_k^o}} \tag{3.22}$$

Another most prominent transfer function which is used is Tangent sigmoid function (Tansig), because we can also include all the values in negative domain also, while in log sigmoid (Logsig), we have only range from zero to one. The mathematical form is given below for clear understanding. This transfer function is also used in our work for getting optimum results and to check its validity on our data.

$$i_j = \left(\frac{1}{1 + e^{-2*NET_j^h}}\right) - 1 \tag{3.23}$$

$$O_k = \left(\frac{1}{1 + e^{-2*NET_k^o}}\right) - 1 \tag{3.24}$$

3.4.5 Stopping Criteria

Generally, no such stopping criteria are defined for BBNN. We have to put check on the error values, according to which, we can stop the weight updating process. According to literature, when the minimum value of the error's gradient vector is achieved is sufficiently very smaller than the threshold value [26]. But this is very time consuming because, some time we have to face the local minima problem. Thus we will run our program specified number of times which will be enough for best results.

The Pseudo code and Flow chart of the ANN is given below for clear understanding

STEP 1: Initialize all the weights of input, hidden output and bias layer

STEP 2: Calculate the non linear Transfer function of input to hidden layer.

STEP 3: Calculate the Transfer function for hidden to output layer

STEP 4: Find the estimation between actual data and estimated data

STEP 5: Update the weights for feed forward NN and bias unit

STEP 6: Update the weights for Back Propagation NN and bias unit check the stopping criteria

STEP 7: Else go to STEP 2

CHAPTER 4

EVOLUTIONARY METHODS FOR SYSTEM IDENTIFICATION

In the previous chapter we have discussed the use of neural network for the air pollution prediction and building of time series models. Literature review shows that NN performs well in the fields of System identification and time series prediction, however there are some limitations when we apply it in the field of environmental sciences, and these limitations are Local minima problem, over fitting problem, and high memory consumption, greater computational time, and for same set of data we have to readjust the parameters of the system. First work was done by using Least Mean Square (LMS) algorithm, in which they developed Auto Regressive (AR), and Moving Average (MA) model for prediction of data. they used all the seven air pollutant and five meteorological parameters In this approach, they first used the statistical values from the Autocorrelation (ACRF) and Partial Autocorrelation (PACRF) function, but the results achieved were not up to the mark, because, the developed model was in appropriate for all time series of data, if it was used on one set of data, we need to readjust its parameter manually for using it for another data set [29]. also, statistical

methodology was introduced, in which, only meteorological data i.e. air parameters were used only, as they were in linear combination with each other, this method uses steepest decent behavior and results obtained were good, but on the other hand, new technique, known as, Neural network was used prediction problems [30], because, it performed well in the field of waste management, water quality control [31], due to less assumption in modeling of system, the results [32] shows that the time series prediction of pollutant level is very difficult due to stochastic nature of data, and its randomness. As the statistics of the wind changes, thus causing the concentration of pollutants to vary with time.

4.1 EVOLUTIONARY COMPUTING

In the current era, researcher are trying to use the hybrid nature of neural network, this will increase the robustness of NN and will overcome all the drawbacks of the existing approaches, also gives us the results which will be up to the level of bench mark. Along with other techniques, Evolutionary Computing (EC) is one of the emerging domains, in the world of computing. Being inspired from nature, many clarifications have been made and on the basis of different algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant colony Optimization (ACO), Bee colony Optimization (BCO) along with NN. Among all the above mentioned flavors, GA and PSO performed well when we use the hybrid nature with NN, i.e. instead initializing the weights of the NN from random values, we get them trained through GA or PSO, the results shows extraordinary difference in simple NN and Hybrid NN. Previously Evolutionary computing techniques were used in the field of robotics, computer games, Pattern classification, but now new trend is following these algorithms in environmental sector for forecasting of air pollution, ozone level measurement flood forecasting, and electric load forecasting.

In order to build hybrid model, we have to first construct the basic model of GA and PSO, such that we can used the optimized parameters to be used in FFNN. But, first of all we have to understand the very basic concept of evolutionary computing that why they are used and how they performs well in the field of computing.

4.1.1 History of Evolutionary Computing

We believe that anything which changes with time has history. Same is the case with evolutionary computing. The term evolution means the change in pattern. Here we will discuss the basic theory of Darwin how proposed the idea of evolution. According to him, we should preserve the favored encounter in the struggle of life. Experienced gained in everyday life play an important role in Genetic influence, and most important one is the environment, all the changes occur in life is due to environment, and those spices will survive which had strong skills and they reproduce more as compared to those who have less opportunities to resources.

Evolution is the process in which gradual changes occur, these changes cannot be sudden and are according to the nature. Living things can evolve in a proper way and they can interact via mutating each other and also by reproducing new generation. Each and every individual is different from each other and this change is due to the structure of DNA, which is the basic unit of life. Different organisms, share their common idea and therefore they deviate with the passage of time. The study of DNA, changing behavior, or phylogenies, helped us in concluding the different aspects of biological evolution to solve the mysteries theories [33]. Here we will renovate our focus on the history of evolutionary computing and its most prominent flavor which is known as genetic algorithm and its basic steps of work.

Long before computers were used on large scales, the idea based on Darwin theory were used to solve the adaptive problems which is now known as Evolutionary Computing EC. After

sometime in late sixties, three variants are proposed in EC. Those variants are Evolutionary programming, Genetic algorithm and genetic programming [34]. In the world of computing, the EC is considered as a sub field or subset of Artificial Intelligence or baby sister of Neural Networks because both the techniques are evolved from the study of natural process. As we know that in natural process of evolution, the living beings exchange some of their basic information with another individual in the process of reproduction. The result of this reproduction will be strong and healthy only in those cases in which they will be provided the suitable environment. Same situation arises in the field of computers; we share some of our information with another source. The result will be most reliable if we provide them the best suitable resources. The results obtained depend upon the fitness criteria of the input to the system and also on the structure. If we study in depth of EC, we will come to know that it works on set of solutions and find optimum one by following the some instructions. The initial proposed solutions in EC are known as population and during evolving them, the fill value will be considered as solution. The performance evolution includes four basic steps. First of them is the selection on individual, then the reproduction, mutation, and at last the survival. From biological point of discussion, we can say that the study of EC will provide us the robust and powerful adaptive search algorithm [34].

4.2 THE GENETIC ALGORITHM

The most important variant of EC is Genetic Algorithm (GA). It is a tool of machine learning and its behavior is adaptive in nature, mostly this tool is used in fields of computing and its multidimensional optimization. The word Genetic is taken from Greek language word GENO which means to rebuild. Therefore in the science of biology it deals with heritage and variations occur in living beings. In other words we can say that it deals with the process of inheritance. The GA was created by a group of researches in the Michigan University in late

sixties as they were studding the natural behavior of inheritance; hence they make basic steps which later known as GA. As it deals with the evolution of population which is composed of Deoxyribonucleic Acid (DNA) that contain the basic unit of inheritance. Each DNA contains one attribute i.e. the color of eyes, or the color of skin. In the computer science we only deal with the zeros and ones. Therefore we can say that one string of zeros and ones in specific order as chromosomes and a group of chromosomes as population. It also works on same set of rules as discusses in EC section.

4.3 THE ALGORITHM

In order to implement the GA in computer, we need some instruction from programmer, which will be easy for computer to understand. These instructions are classified into six parts which are initiation, assessment, selection, reproduction, mutation and termination. The more detail of this algorithm will be presented in next section

4.3.1 Initiation

During initiation phase, we need to consider two main aspects which are population size and procedure. The population size should be more appropriate such that no extra emulation is needed. In some cases when we increase the input population size, it diverges from the solution and we may get stuck in minima problem and also time consumption. The best results are only obtained when we have average no of population, however there are no specific rule present which explains the limit of initial population, so, it is left up to user choice, and we can optimize our results by hit and trail method. The second main point in population initiation is the procedure. Here we have two different techniques, which are random approach and heuristic approach. As the name explains that random approach, we simply use the built-in function of random in any programming language and all the chromosomes will be generated randomly. In second approach, i.e. heuristic approach, we

need proper formulation which varies from application to application. How ever the best among two is random one because researchers experience says that results obtained are more reliable and optimized.

4.3.2 Assessment

The assessment is the main step in GA and all EC variants. Because all results depends upon them. This is the constraint of the problem, or the objective function. If we properly model our system in the mathematical form, we can achieve our goals easily. All the engineering design is enclosed in this basic step. The assessment includes the constraint or the objective function. In other words we can say that it is the fitness function of the problem. This fitness function varies from problem to problem. The chromosome having best fitted value will be awarder highest rank in the population and the chromosome having poor performance or ability to do work i.e. reproduce will be put in the bottom.

4.3.3 Selection

After the chromosomes are sorted in ascending or descending order, they are then selected to reproduce. This selection of chromosomes is done by specific set of rules. Some of most widely used are Elitism, generation replacement, roulette wheel method, and fight for survival. In elitism, we specify some limits, both in parents as well as in kids. In some cases we use sixty percent from the parents and forty percent from the kids and make new government which will be allowed to rule over. But these ratios are not fixed, and vary from problem to problem, and user to user. In the generation replacement, we replace all the chromosomes, like clean sweep. All the parents are discarded and killed and newly generated kids take over. This technique also shows good results. The most important and widely used is fight for survival of fittest. Once kids were produced, one single common merit is built, all the parents and kids are placed their, and then their fitness values are sorted, the best fitness

values go to top of the list and bad one goes to the bottom. Here we will discard the over flow and will keep the desired and required ones.

4.3.4 Reproduction

In the terminology of genetics, the reproduction is called cross over. In this we exchange the information in different chromosomes. The reproduction is classified into two parts, single point cross over and multi point cross over. In most of the problems, we use random function to generate the cross over point. In single point cross over, we have single value randomly generated, while in the multi point cross over, we have multiple points randomly generated. For example we have chromosome of length thirty genes, and we need a cross over point, then we will generate a random number with in the bound of one to thirty. And at this point we have exchange of information, as shown below.

Chromosome 1:

$$111111111111111111111111111111 \quad (4.1)$$

Chromosome 2:

$$000000000000000000000000000000 \quad (4.2)$$

Xover point = rand (1, length (chromosome1)), 18, for example

Kid 1:

$$111111111111111111000000000000 \quad (4.3)$$

Kid 2:

$$000000000000000000111111111111 \quad (4.4)$$

The above example is single point cross over, here one point should be remembered that the value before cross over point will be first chromosome and information after that point will be from second chromosome. Similarly, and for multiple point cross over we have 2 or 3 point of exchanges of information. For example in order to improve the performance of the system, we need three point cross over, consider the above example in which the length of each Chromosome is 21 genes, then the cross over points range will be

Xover 1 range= gene 1 to gene 7; limit1= chromosome1 (1:7)

Rand (1, length (limit1)) 5

Xover 2 range = gene 8 to gene 14; limit2= chromosome1 (8:14)

Rand (1, length (limit1)) 10

Xover 3 range = gene 15 to gene 21; limi3= chromosome1 (15:21)

Rand (1, length (limit1)) 18

Kid 1= 111110000011111111000

Kid 2= 000001111100000000111

4.3.5 Mutation

After reproduction step, each kid should be mutated, which means to change in the structure of the newly born kid. In mutation we use some probability function whose value will be determined by the algorithm. In this phase they will be altered with small change, and the size of mutation step is kept small. We have two techniques of mutation as like initiation step of algorithm. These types are random mutation and heuristic mutation. The size of this step remains constant throughout the program and the position of the mutation gene will be changed in random selection and will be fixed in heuristic mutation. One other point should

be considered is that the size of mutation step is inversely proportional to the length of chromosomes. This inverse proportion will gives us very low probability to change. In short, we can say that mutation can change only one gene or maximum of two genes in any chromosome.

4.3.6 Termination

The termination step in any GA program is the most important. We have three different criteria for stopping any algorithm. We put constraint on either the error value achieved or by considering the no of loops or generation performed.

After discussing all the steps of GA, we will write it in the form of computer algorithm and also label its block diagram.

STEP 1: Create Random Population.

STEP 2: Find Fitness of POP.

SETP 3: Sort the POP (Ascending, Descending)

SETP 4: Generate kids by Crossover technique.

STEP 5: Find Kids Fitness and sort them.

STEP 6: Replace the POP with Kids.

(Elitism, Generation replacement, Fight for survival).

STEP 7: Calculate MSE and Check stopping criteria.

STEP 8: If requirements fulfilled then stop, else go to step 2.

4.4 PARTICLE SWARM OPTIMIZATION

The newly emerged technology named Particle Swarm Optimization (PSO) is widely used to solve the complex problems in every day computing and it proved to give most prominent results, because it is very simple and easy to implement [35]. In PSO we only consider two basic behaviors, i.e. best performing individual and best moving forward and converging to one solution which will be the best optimum one. The algorithm or optimization technique which is based on Particles belongs to Evolutionary Computing branch of studies. Many biologists were studding the behavior of nature and specially birds that how they move and how they search food for themselves. And the end of the day, they wrote down their work, which later on used in the field of Computing and Engineering, and the mathematicians made the Algorithm from those information which are now a days known as PSO. The area which deals with the Computational Swarm intelligence is Particle swarm, whose origin is bird flocking. Each individual bird is known as Particle, as like we have discussed the G.A in the previous chapter that each solution is called as Chromosome and the length of each solution is known as Gene. Here we will discuss the PSO algorithm in detail and we will start with the History and then to the term optimization, after that we will extend our discussion to formulate the mathematical model with explaining all the necessary steps involved in this context.

4.4.1 History of PSO

After the Success of Genetic Algorithm, in the mid 90's instead of implementing the DNA structure based Evolutionary procedures in Algorithms, the researchers moved to model the populations in the algorithms and they have studied the many natural species like Ants, Bees and Fishes to solve the problems and obtain the optimum results, but they were helpful in very little scope [35]. Among these new races, the study of Birds gained much attraction, and new field of study has been build naming Particle Swarm Optimization. Hence two prominent

researchers namely Eberhart & Kennedy in 1995 published their research paper and formulated the basic steps of PSO algorithm [36]. Considering its importance and giving the efficient results in all the fields of computing, much more work has been done in the last Decade [37]. We can also say that it has plastic nature because it can easily adopt the parameters and operators to obtain him desired form for the given problem [35]. The proper work has been done by Beni and Wang in 1989 to implement Swarm Intelligence for controlling Robotic process [38]. after this research, three main fields have been developed known as ant colony optimization, stochastic diffusion search, and particle swarm optimization, there is somehow difference present in there but all of them can be categorized in Evolutionary computing .

4.4.2 What Is Optimization?

The word optimization deals with the selection of best suitable path or solution among many ideas. This best suitable solution is based on used dependent problem [35]. Here I can proudly give the example of Structural engineering problem whose solutions can be found through fundamental engineering theories which we study in our engineering design course. Some basic constraints will be imposed by the user or by problem upon it and we will have some best suitable solutions. So at the end, the solution we have will be called as feasible solution or globally optimal.

During the modeling phase, the actual problem is written down in mathematical form, noticing all its constraints, creating and developing the blocks for the candidate solution and hence defining the numerical variables for it. After doing this a mathematical formula will be built which will lead us to the globally optimal solution and from there we can have our desired result. This mathematical function is known as objective function. In some cases, the

objective function consists of single optimal solution and in some cases it has multiple solutions and the domain is known as optimal domain [39].

Working on optimization technique is the hot topic of current era, by applying this technique on the real world problems; we have to face the following difficulties.

- Finding the local best value and global best value.
- Removing noise from the solution set.
- Understanding the constraints

There are some basic properties which every optimization problem contains and those properties are continuity, differential, and nonlinearity. The optimization problems can be linear and non linear, also we can classify it into quadratic and stochastic, which deals with minimization of quadratic constraints and stochastic parameters like randomness and noise distribution [40].

In most of the cases, we have single objective function, but our goal is to locate the global minimize as it moves in current population, also it aims to robust solution which avoids the heavy computations

4.4.3 Swarm Intelligence

Particle swarm Intelligence is the field of Artificial Neural network, which studies the behavior of structure and its complex properties. Those systems which work in interactive environment and make small communities and find the optimal path in colonial form are known as Swarms. In this type of searching the best solution, each agent perform limited action and there is no central control like the communities we have, each agent have its own thinking and it shares its idea with all the members in the community, if this idea is appreciated then all the members will act upon that idea and thus the group proceeds.

The main idea of working together is taken directly from nature, like fishes, birds, ants, etc. all these species work in self organized manner and produce collective behavior, when someone find food, it informs all the group and they together enjoy. The group of Scientists and researchers have put their energies to understand their behavior and made mathematical mode and defined them in the form of simple instructions. In 1994 Millona's presented the five basic rules of the PSO, Which are Adaptive Behavior, i.e. the PSO can change its behavior when some change occur in the idea by any external media. Stability Factor, diverse nature, Quality in search method and ability to perform in space and time based computation [41], [45].

4.4.4 Early Editions of PSO

As we have discussed in the earlier paragraphs, that the 1st paper was published [36], as they were studding the birds flocking and wrote down their observations on the paper and they concluded like other Evolutionary techniques, PSO has some basic properties. In Genetic algorithm, we called the search space as population and the values inside the population are composed of chromosomes and each chromosome is made of gene, i.e. collection of genes makes chromosomes, and collection of chromosomes makes population. And the solution lies in this population or the second generation or in next generation. Similarly the solution is searched in the same generation and we do not have any other generation but we update our intelligence. It's like a human body which is baby, having small memory and when it grows up, and its mind gets matured and its intelligence sharpens. As we mentioned earlier, PSO is population based algorithm, which is called as swarms and its individuals are known as Particles.

4.5 MATHEMATICAL MODEL OF PSO

Mathematically we use simple model in order to represent it in nomenclature used by mathematicians and other researchers. The swarm set is written as

$$P = \{x_1, x_2, x_3, \dots, x_N\} \tag{4.5}$$

N= Dimension of the Problem.

And each particle has its own indices like each chromosome is composed of genes; hence we can re present it as fallows

$$X_i = (X_{1,1}, X_{1,2}, X_{1,3}, \dots, X_{1,N}) \tag{4.6}$$

Where X_i shows the indices in i^{th}particle.

All the particles are analyzed by objective function say $F_i = f(P_i), P_i \in P$ those particles move with in the search space and have some velocities and they update their position by using some proper factor

$$V_i = (V_{i,1}, V_{i,2}, V_{i,3}, \dots, V_{i,n})^T \tag{4.7}$$

Where i = 1, 2... N

The velocity is made adaptive because in every move, it change its intelligence and some time it searches the food or solution by using its own intelligence and in some cases it fallows the results obtained by other fellow or neighbor particle. If all the values are time dependent and t denotes the counter used for counting, then the current position of i^{th} particle can be denoted as $X_i(t)$ *and* $V_i(t)$ respectively. As they move with in the search space, they also got some position in the swarm, which is to be stored and used in the next move or iteration, this information is very useful and all the movement is depends upon these values, here we make a table in which we can store the all the positions obtained by the particles, there are two types of the position which will be explained in detail later on, one is called the local position

and second is called as the global position and we make two columns in the table which stores both the entries. This table is also called as memory of the particles or the intelligence of the system. It can be written as

$$Pi = \{P_{i,1}, P_{i,2}, P_{i,3}, \ldots, P_{i,N}\} \quad (4.8)$$

Where i = 1, 2 ... N

And the best ever position obtained by any particle in the swarm is written as global intelligence and we can denote it as fallow:

$$P_G = \{P_{g1}, P_{g2}, P_{g3}, \ldots, P_{gN}\} \quad (4.9)$$

Where i = 1, 2 ... N

All the particles communicate with each other, and hence share the information, and this makes them social, and the algorithm will select that particle which has the highest position in the swarm or we can also say that algorithm will nominate as the globally optimized value on the basis of highest rant in the position matrix. Let k be the index of the all the possible moves, then the globally optimized value will be

$$P_k(t) = \max f\left(P_i(t)\right) \quad (4.10)$$

4.5.1 Position Update Rules of PSO

The complete mathematical model of PSO [5.2, 5.12, 5. 13] can be written as

$$V_{ij}(t+1) = V_{ij}(t) + C_1 R_1 \left(P_{ij}(t) - X_{ij}(t)\right) + C_2 R_2 \left(P_{gi}(t) - X_{ij}(t)\right) \quad (4.11)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad (4.12)$$

Where R1, R2 are the random variables which are uniformly distributed over the search space and the C1, C2 are the convergence weight factors which move the result towards the local best result or global best result. We can also call them as acceleration factor. In the above

equations, after the updating the position the new value will be stored in the variable having index value t+1 and the previous value will be stored in the index having variable t.

After achieving new position, the velocity is automatically updated and we use this simple formula to implement it in simulation model.

$$X_{ij} = X_{ij} + V_{ij} \tag{4.13}$$

Generally it takes few hundreds to few thousands iterations to updates its values to get convergence and all the convergence depended upon the variables which we select in PSO algorithm and proper formulation of constraint parameters [41] [42]. The above velocity update equation consists of three major components [46]. first of them is the inertia , it causes the particle to move in the same direction in which it is flowing, also, it is called as momentum or habitual nature of the particle, the 2nd item in the velocity equation is Local fitness which it ever achieved . This value is called as memory of the system and all the next possible move depends upon these values, the other names of this local performance is self-knowledge and remembrance [45]. The third component of velocity update equation is the global position ever achieved by all the particles i.e. the performance made by that particle within the group, we can also explain it with the help of following example. Consider we have 5 classes in one department, and each class contains 50 students, and after exam, ad result finalization, each student has its own marks, i.e. its own GPA, and when all the GPA of all the students are collected, one common merit is made and that student is listed on award list as $1^{st}, 2^{nd}, 3^{rd}$, and so on, so its own GPA will become its local performance and position it obtain at department wise merit lest will be considered as its Global result.

4.5.2 Objective Function of PSO

Likewise other techniques both in evolutionary and non evolutionary , we use some constraint this will solve the problem and converges to some specific result, this constraint In

PSO is called as objective function or fitness function. All the other parameters are totally dependent on this fitness function, if this is properly formulated then we can achieve our goal easily, and if we fail to model it precisely, we will trap in some local minima, and we can never achieve our destiny. The most famous objective function used in prediction of all the systems is log sigmoid and tan sigmoid function [43]. The performance of both the transfer functions will be shown in the result and discussed chapter.

4.6 PSO ALGORITHM

As we have discussed the PSO in detail, now time comes to explain the algorithm in the form of some simple steps and also draw its block diagram. According to [47], [44], [46] in his research, we have following structure

1. Initialize the population (swarm) randomly. And save it as Local_best
2. Find its fitness by using fitness function or objective function
3. Compare the new value with the Local_best value, if the new value is greater then the Local_best, then overwrite the Local_best value, if not, leave it unchanged.
4. Find the particle having maximum fitness value, make this value as Global_best, and also note down its position
5. Update the position and velocities of all the particles.
6. Repeat the steps 2 to 5, until stopping criteria is reached.

Also [44] [47], in 2004 proposed the idea that the initial population, or particles should be initialized in such a manner that they should be equally distributed, and hence we will have coverage on large area, and we will have improved performance as compared to random initialization. Alternatively, Campania et al, proposed the new idea to change the velocity update procedure and make all the variables orthogonal to each other, this will improve the convergence behaviors of the system [48].

4.7 HYBRID MECHANISM

After the detail study of both the evolutionary algorithms, i.e. PSO and GA, we came to know that they are the efficient methodologies to train any network and get the optimized results and better threshold value [49]. We only need to build such a dynamic relationship between the weights and dimensions of the threshold. And in our proposed algorithms, the optimization will be achieved by updating the weights of network. Thus in the case of PSO, the weights updating is done by updating the particle position and velocity, and in GA, the weight updating is done by means of achieving the best chromosome by selecting the next generation methodology and kids generation and their assessment.

The Fitness function in both the algorithms will be Minimum mean square, which will be calculated by using the Sigmoid as explained in Eq. 3.20 in chapter 3 respectively. Once the optimized weights have been obtained from both PSO and GA, these weights will be processed to Feed Forward NN, where we will predict the future value , the detailed procedure is discussed in the chapter 2, and the results are shown in chapter 5. The main theme behind the hybrid mechanism is that in order to update the weights by using Back Propagation NN, we use some different mechanism, and after all the weights have been updated, the are again set in the NN architecture. The detailed steps of Hybrid Algorithm are stated below.

4.7.1 Hybrid PSO-NN Algorithm

The steps for the modeling the hybrid mechanism for PSO-NN algorithms will be as follow:

1 Model the Neural Network Structure, Input Layers, Hidden layers, Output layers, values.

2 Make these Hidden layer and output layer weights as PSO Particles

3 Find the fitness of the said Swarm according to Eq. 3.20

4 Find the Local_best, and Global_best values.

5 Update the weights by using velocity update rule.

6 Update the velocity and positions of the Swarm

7 Repeat the algorithm until convergence or iterations are achieved.

8 Then pass these updated weights back to Feed-Forward Neural network

4.7.2 Hybrid GA-NN Algorithm

The main steps for obtaining the convergence in case of GA-NN hybrid mechanism, we should follow the following steps.

1 Model the Neural Network Structure, Input Layers, Hidden layers, Output layers, values.

2 The hidden layer and output layer Neuron will be used as Chromosomes for GA.

3 Find the fitness of the said Chromosomes according to Eq. 3.20

4 Sort the population in ascending or descending order.

5 Generate the crossover point randomly.

6 Generate the pair of kids by single point crossover.

7 Find the Fitness of Kids by using the Fitness Function

8 Update the newly generated chromosomes and old parent's chromosomes by using Elitism method.

9 Make newly selected chromosomes as the Neural network weights

10 Repeat the algorithm until convergence or iterations are achieved.

4.7.3 Future Values calculation Through FFNN

In the previous sections, we have just explained how to get the hybrid weights, and not yet performed any future values calculation, here, we will explain the detailed steps of future values which will be based upon the optimized weights.

The main steps are as fallow:

1. Take optimized weights from PSO, GA
2. Pass them back to Feed Forward NN
3. Provide Input to Neural Network
4. Calculate the Net output at hidden layer
5. Calculate the Transfer Function at Hidden Layer
6. Process step 5 result to weights present at hidden to output layer
7. Calculate the Net output at output layer
8. Calculate the transfer function at output layer, which will be predicted value

After calculating the future value from FFNN, we will calculate the MSE among actual value present and this predicted value, this will tell us about the correction and accuracy of system and efficiency of our proposed algorithm.

CHAPTER 5

SIMULATIONS AND RESULTS

In this chapter we will discuss how to implement the schemes studied in earlier chapters. First of all we will present the proposed model for Artificial Neural Network (ANN), in the second phase we will present our proposed algorithms i.e. GA based Hybrid NN (GA-NN), and PSO based hybrid NN (PSO-NN), then the results will show that which will perform better.

First of all, we will begin with the basic structure of ANN and will use the Back Propagation algorithm (BPNN), whose all necessary steps were explained in chapter 3. The data will be normalized by using simple scheme. All the received data, which was initially in the tabular form of dimensions (31 x 24), is converted into single array for the simplicity of programming tactics, the maximum value will be calculated by using the built in function of MATLAB and preserved for Demoralization. Thus, we will have all the values in the range from 0 to 1. The formula used is written below

$$X_{max} = \max(\mathrm{X_{data}}) \tag{5.1}$$

$$X_{norm} = \frac{X_{data}}{X_{max}} \tag{5.2}$$

$$X_{Denormalized} = X_{max}(X_{norm}) \quad (5.3)$$

At the Demoralization step, the X_{max}value will be multiplied to the results obtained during calculation thus the Normalized results will become in its actual form and can be used for further calculations.

Our experiments contain two steps. 1st the data training and 2nd is the validating future response to be Predicted. Thus the data of the first ten days will be used for testing purposes and the next seven days data will be used for validation. Each day contains 24 points. Thus we have total of 240 points in array. The variables like Mean Square Error (MSE), filter length, desired response, No of Iterations, and data set used, are listed in the table which is actually used in the program to achieve the results. The results achieved by using this algorithm are shown below. The initialization parameters used for neural network are shown in the table below.

Table 1: Parameters for Back Propagation NN

Name Of Parameter	Value
Converging Factor eta (η)	0.01
Total length of signal	240
Neurons at Input Layer	3
Neurons at Hidden Layer	5
Neurons at Output Layer	1
Input-Hidden Layer Neuron	3 x 5Matrix
Hidden–Output Layer Neurons	5 x 1 Matrix
Biased weight	1
No of Iterations	1000

In table 1, we use 3 input layer neurons and 5 neurons for hidden layers and 1 neuron at output layer. This combination suits well, and gives us best results, while the converging factor eta is set 0.01, because by putting greater value, we get stuck into local minima problem. Thus the size of input vector becomes 3 taps, and the matrix becomes 3 x 5 values, while the matrixes of hidden layer to output layer have dimensions of 5 x 1 as shown in Table 2. The value of biased weight is set to 1, and the numbers of iterations were set to 1000 for optimum results.

Table 2 weights Distribution for BP-NN

	Input to Hidden layer Neurons				
1st Input Neuron	W_{11}	W_{12}	W_{13}	W_{14}	W_{15}
2nd Input Neuron	W_{21}	W_{22}	W_{23}	W_{24}	W_{25}
3rd Input Neuron	W_{31}	W_{32}	W_{33}	W_{34}	W_{35}
	Hidden to Output Layer Neurons				
Output Layer neuron	W_1^o	W_2^o	W_3^o	W_4^o	W_5^o

The parameters set for GA are some how different from NN due change in structure of both algorithms. From table 3, we can see that the numbers of Chromosomes were set to 40, and each chromosome contains 20 genes. The composition of these 20 genes will be the sum of both input to hidden layer weights and hidden to output layer weights. We have used total of

three input neurons and each neuron is connected to 5 neurons at hidden layer, thus we have 15 weights for all three input neurons, and these hidden neurons are connected to output layer weights, thus we have total of 15 and 5 weights, thus one chromosome of genetic algorithm is made of 20 weights at all, as shown in figure 1, while we have proposed 40 random solutions which is our search space. In this search space, the cross over will be done and new kids will be produced in this these 40 chromosomes which we call population. The cross over point is made random, while each pair of chromosome produces 2 kids. The chromosomes used for searching the optimum solution were selected by Elitism methodology. The numbers of iterations were made 1000.

	Composition of Chromosome	
Genes of Chromosomes	Input to Hidden Layer Weights	Hidden to Output Layer Weights

	Input to Hidden layer Neurons				
1st Input Neuron	W_{11}	W_{12}	W_{13}	W_{14}	W_{15}
2nd Input Neuron	W_{21}	W_{22}	W_{23}	W_{24}	W_{25}
3rd Input Neuron	W_{31}	W_{32}	W_{33}	W_{34}	W_{35}

	Hidden to Output Layer Weights				
Output Layer neuron	W_1^o	W_2^o	W_3^o	W_4^o	W_5^o

W_{11}	W_{12}	W_{13}	W_{14}	W_{15}	W_{21}	W_{22}	W_{23}	W_{24}	W_{25}	W_{31}	W_{32}	W_{33}	W_{34}	W_{35}	W_1^o	W_2^o	W_3^o	W_4^o	W_5^o

Figure 1: Weights Distribution for GA-NN

Table 3: Parameters for GA-NN

Type Of Parameter	Value
No of Chromosomes	40
No of Genes	20
No of Kids	40
X over point value	random
Methodology	Elitism
Total iterations	1000

Similarly, the parameters used for PSO is listed below in the table 4, which will help us to understand the scenario. The width of the PSO will be same as GA, because we are going to make two hybrid approaches which will compare their results with BP-NN. Thus the swarm size will be 40, and each particle has 20 value points. The composition of particle is same as the GA chromosomes, i.e. 15 weights will b from input to hidden layer and remaining 5 will be from hidden to output layer. The distribution of weights has been shown in the figure 1. The desired response size is made 1 and the converging factors towards the best result Local best (C_1), and global best (C_2) were set to 0.02 and 0.98 respectively. The higher value is assigned to C_2 variable because, variable C_2 will converge the result towards the global optimum solution. R_1, and R_2 are two random variables, which are distributed on the particle. The V_{min} and V_{max} is the upper and lower bound on the optimization results and it keeps track of particle and limit the results with in the prescribed limits. The 1000 iterations were performed to get results.

Table 4: Parameters for PSO

Type Of Parameter	Value
Swarm members	40
Length of each Particle	20
R1	0.7
R2	0.8
V_{max}	4
V_{min}	-4
C1 (To Local_best)	0.02
C2 (To Global_best)	0.98
iterations	1000

After discussing all the parameters of both PSO-NN and GA-NN along with BP-NN, we will now model their flowcharts, which will be very helpful in clear understanding of the proposed concept. From Figure 2, and Figure 3, it becomes very easy to viewer to understand the concept of system identification and then its future values calculation. The future values calculation is explained in chapter with each and every detail. Here in both the flowcharts, we mentioned the proposed PSO model, we mean by this term is that, the weights will be according to the architecture of NN, and equal to the sum of input to hidden layer weights and hidden to output layer weights. Error will calculate the difference between the actual input and out based upon our proposed weights, in the start the difference will be higher, but as we proceed further, our difference gets minimized and we will have optimum weights

which are at some constant distance from actual input. This will tells us about unknown system statistics.

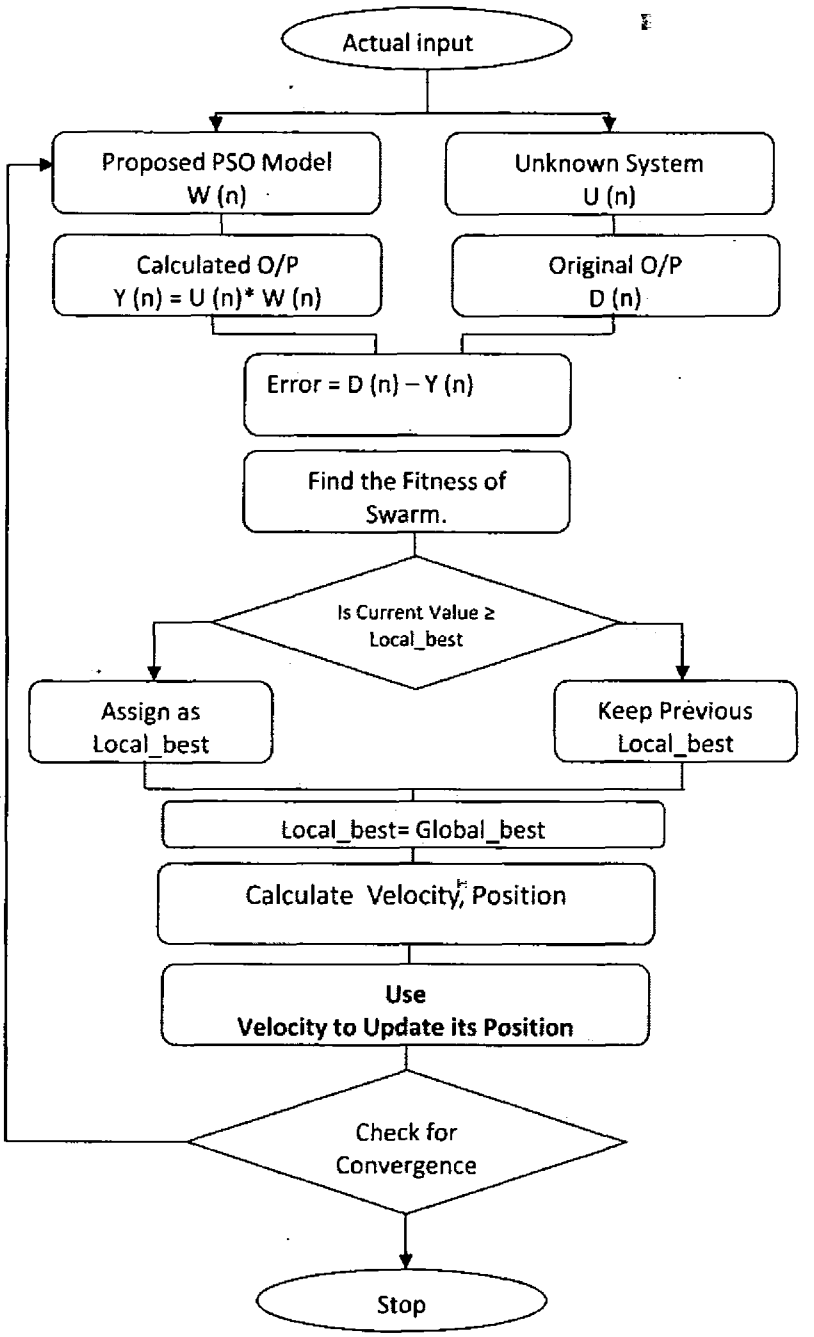

Figure 2: Flowchart of proposed PSO-NN

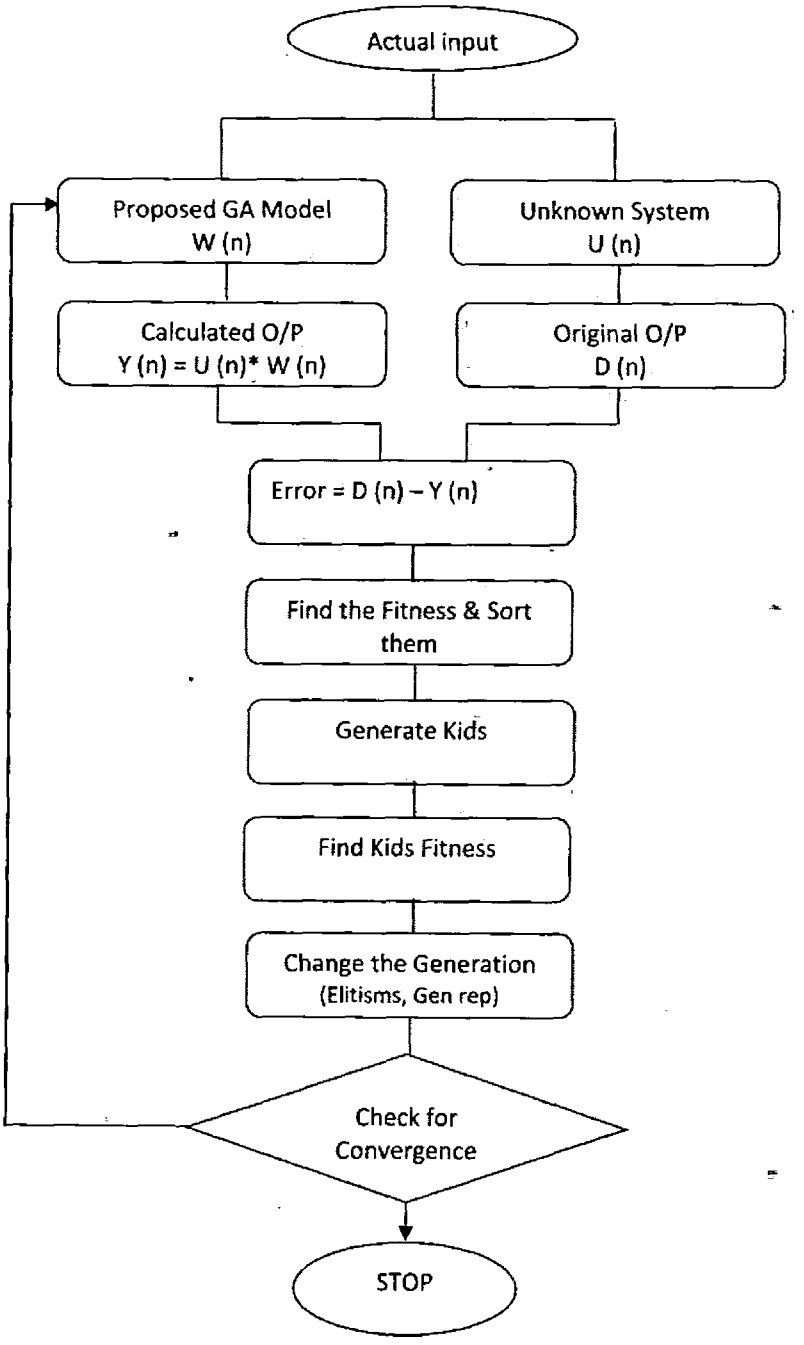

Figure 3: Flow chart of Proposed GA-NN

5.1 THE NORMALIZED DATA

First of all we plot the original data set in the normalized form, which is to be used for training purpose.

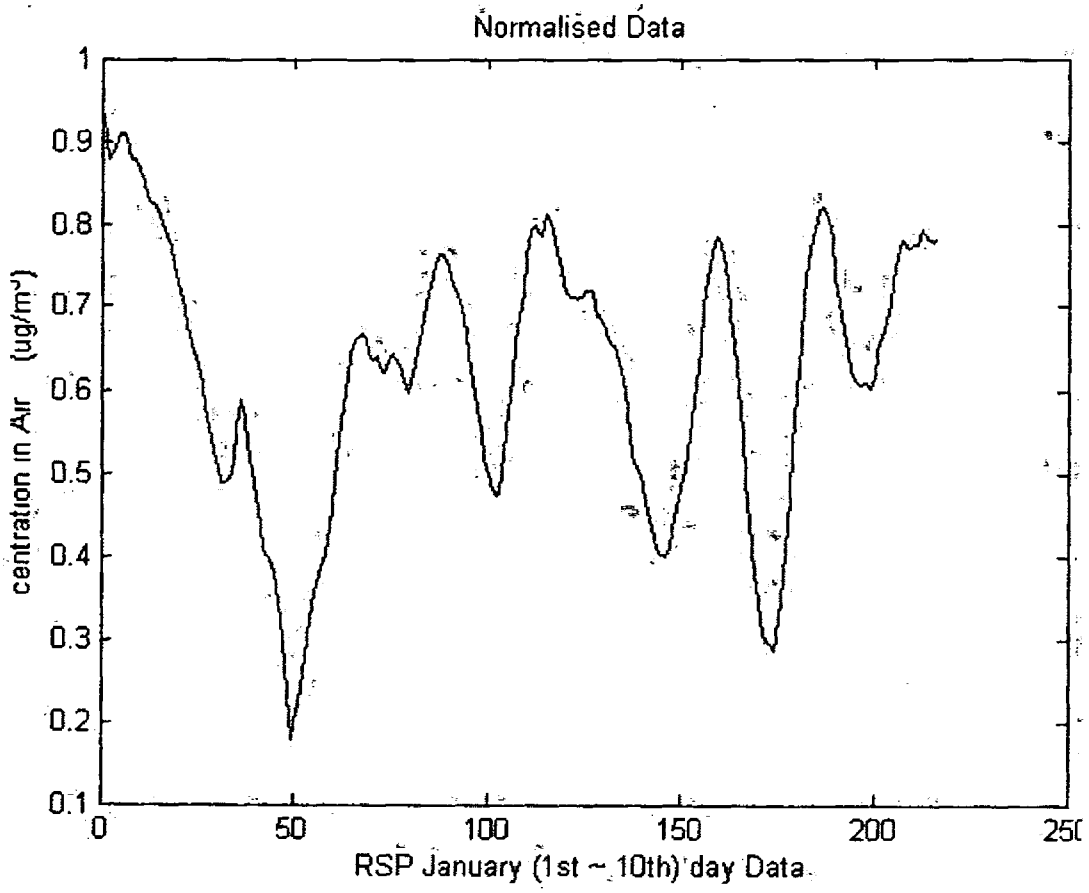

Figure 4: Normalized data for RSP January

Figure 2 shows us the 1st ten days normalized data of RSP January month for the year 2010, similarly, the Figure 3 shows us the normalized data of RSP June 2010 having values from 1st June to 10th of June. Figure 4 and figure 5 tells us about the normalized data of NO_2 June and January moth respectively. And both the figures contain the values from 1st of the moth to 10th of the month. Similarly the figure 6 and figure 7 contains the 1st ten days normalized values of SO_2 for the month of June and January 2010.

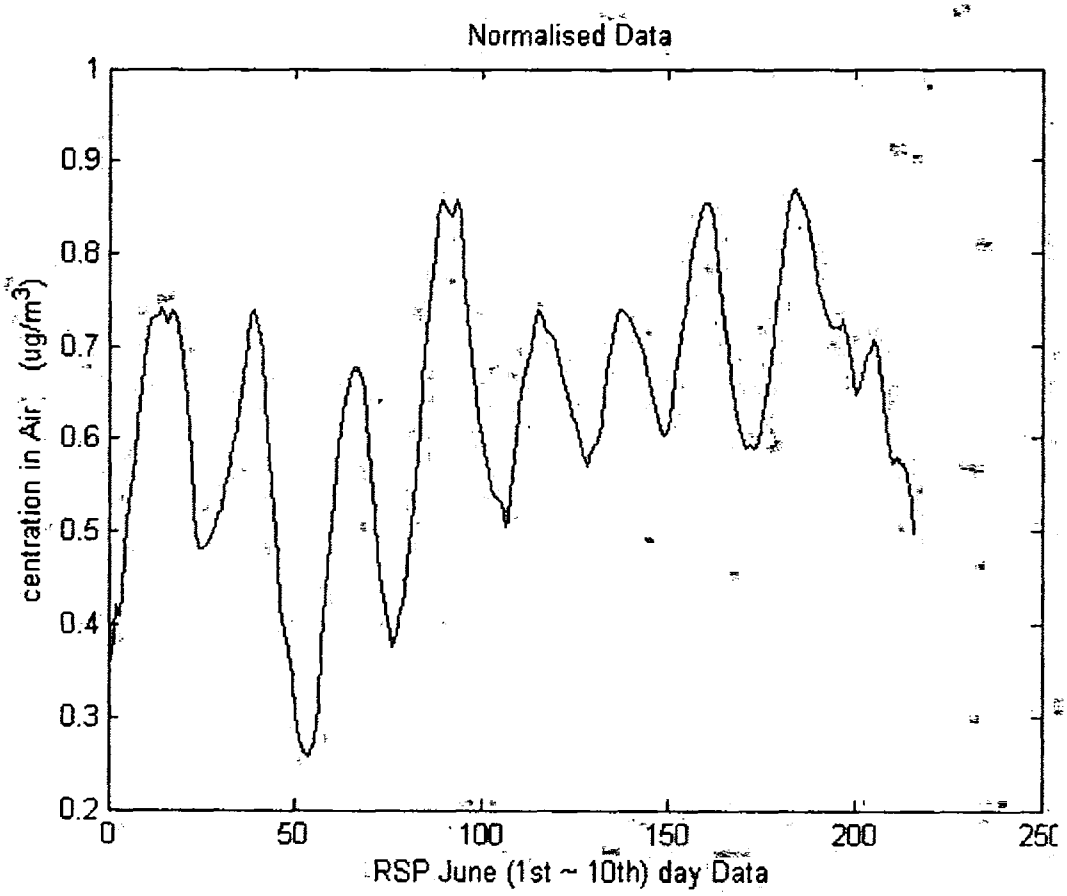

Figure 5: Normalized data for RSP June

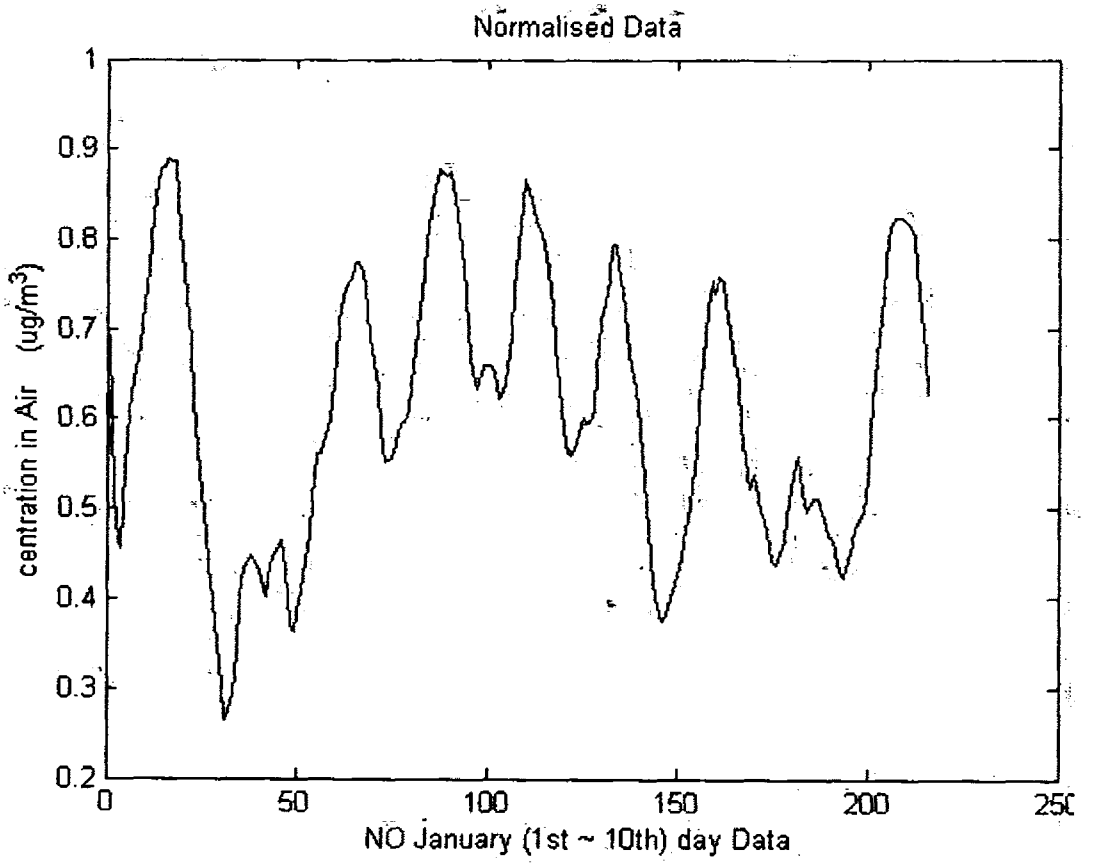

Figure 6: Normalized data for NO2 1st ~ 10th January

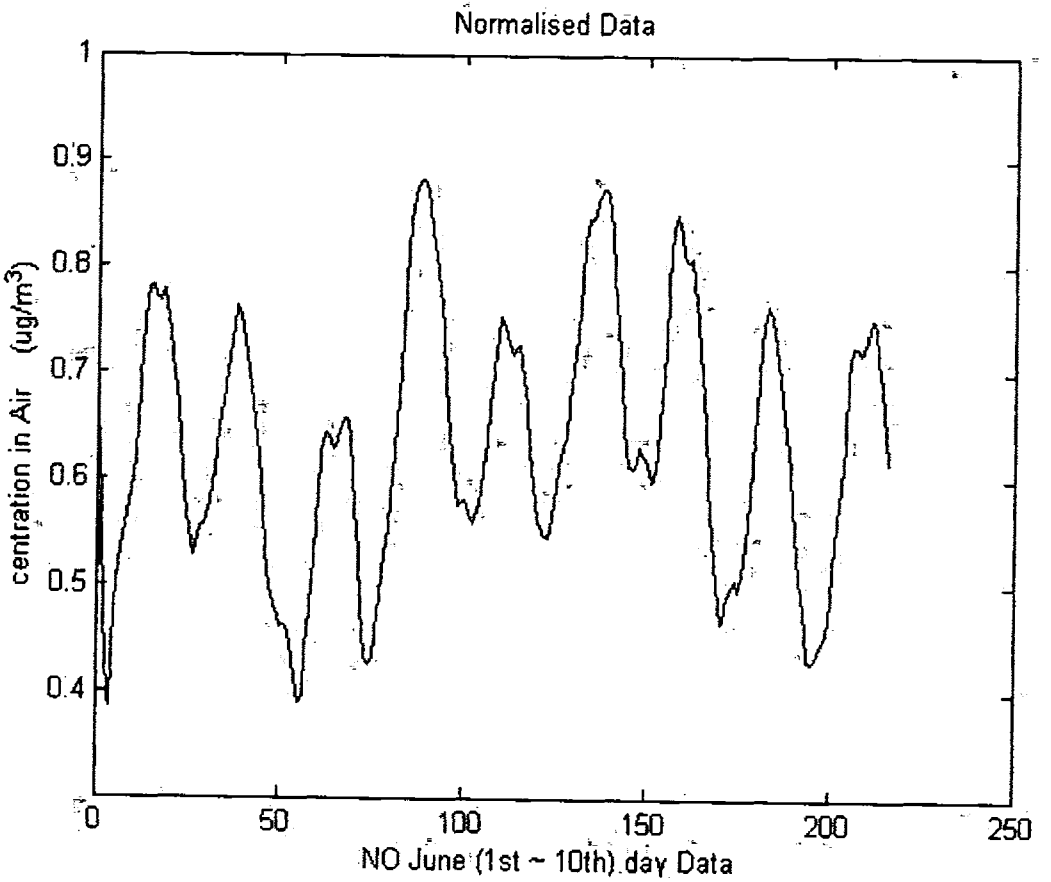

Figure 7: Normalized data for NO2 1st ~ 10th June

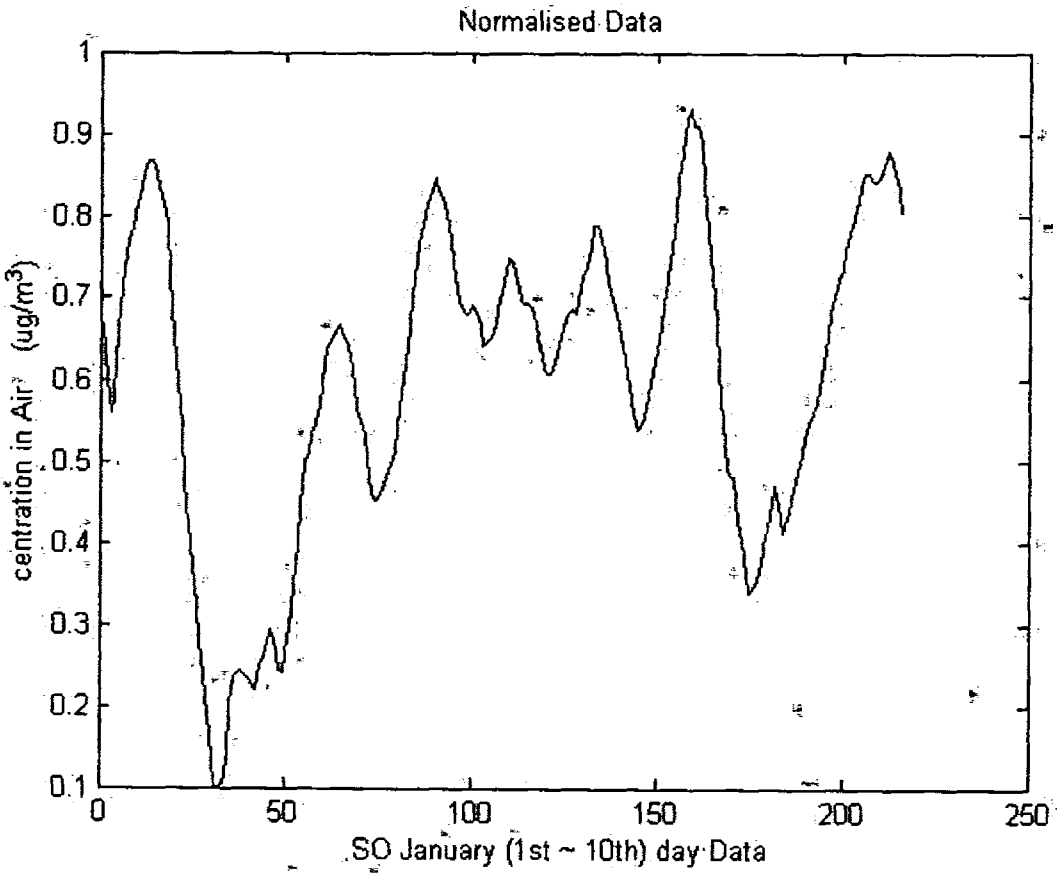

Figure 8: Normalized data for SO2 1st ~ 10th January

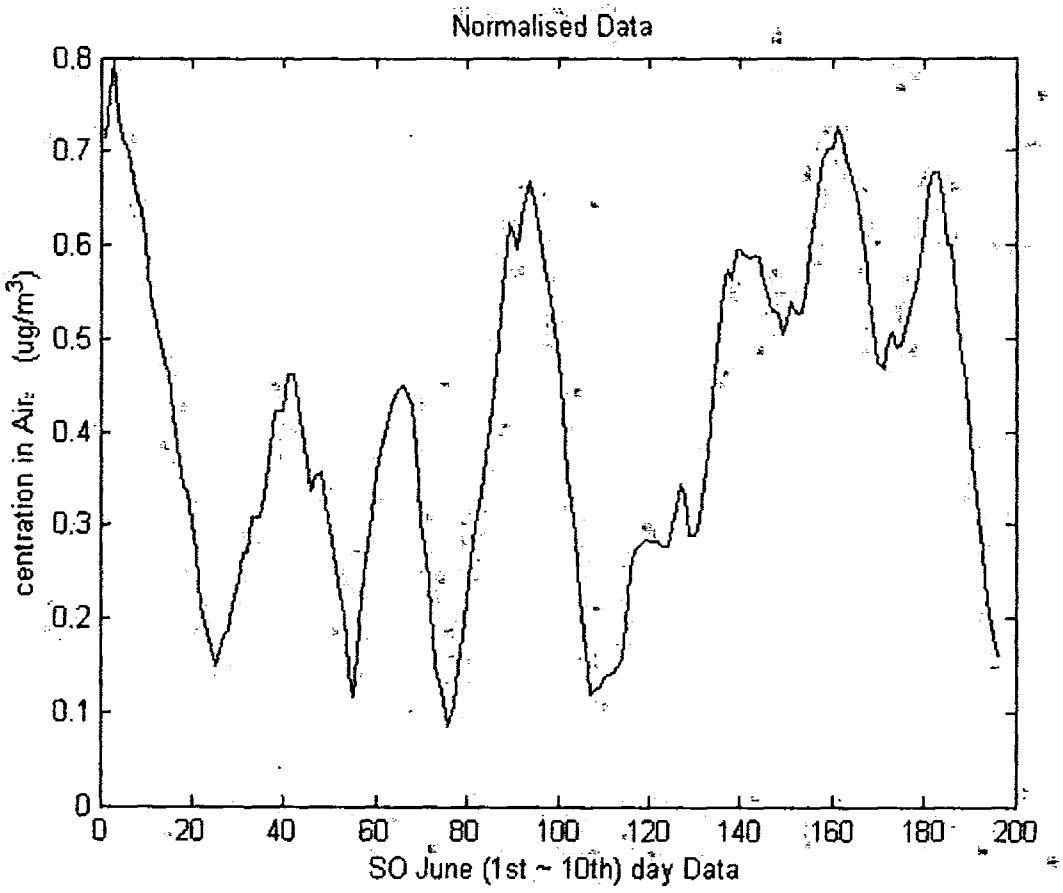

Figure 9: Normalized data for SO2 1st ~ 10th June

Three different algorithms were developed to get results and the performance is measured in the scale of MMSE. Different non linear transfer functions are used to check the validation of the algorithms. These Transfer functions were Log sigmoid (losing) and Tangent Hyperbolic (tanh) functions, whose mathematical form is given in the chapter 3.

In the NN, we used the Different combinations of these two transfer functions to check the validity of Algorithm such that we can use it in our problem. We have two portions in NN algorithm, which are Input to Hidden layer Transfer Function and Hidden to Output layer Transfer Function, at both the layers sigmoid performed very well and gives us the results up to the marks, while by using the tanh function, we get stuck in the local minima and the curve

of MSE get diverted and we get straight line on x-axis from the first iteration to last one. And when we used the tanh function at the input to hidden layer and sigmoid function at hidden to output layer, we also get stuck again in the local minima and the curve deviation results were obtained. About 10 trails were performed for each combination of Transfer functions. The results show that the transfer function Log sigmoid suits best for Training the network on Air Pollution data set 2010.

The same behavior was seen when we use the Hybrid nature of the NN with the Evolutionary Computing Algorithm i.e. GA-NN and PSO-NN. The Architecture of ANN having 3 inputs, 5 hidden and 1 output neurons, gives optimum results. This model took less convergence time and tracks the signal very well. About 10 trails were done to validate this model both in standalone NN, and with Hybrid nature of NN. The number of iterations used was 2000.

The Model used for GA is 40 chromosomes and 15 genes in each, which uses less time to track the desired signal, and converges faster than NN. The numbers of trails used at different schemes of GA were 10 with each trail has 1000 iterations. Similarly the Optimum Swarm generated for PSO also consist of 40 particles and each particle has 15 data point values. The numbers of iterations were 1000. In kid's production, we have two merits. One is Generation replacement and second is Elitism. We see the results that Elitism performs well as compared to generation replacement. Also, we choose the Cross over point selection as a random, because each iteration we will have different point, which provides us the dynamic nature of algorithm.

5.2 SYSTEM IDENTIFICATION

The following Graphs show us the Training behavior of the proposed algorithms at different data sets. The performance of all the algorithms was very good except for the month of June on NO_2 dataset. While the best performance of all the data sets was on RSP January month.

Similarly the result shows that the Future Prediction was good at SO_2 and NO_2 datasets for both January and June months. While poor performance is achieved at RSP data for both January and June month. The MSE results are shown in the graphs later in this chapter.

By training the Existing technique of NN having Back Propagation algorithm, and evolutionary computing based hybrid techniques, i.e. GA based NN (GA-NN) and PSO based NN (PSO-NN), we came to know that the performance of NN is poor as compared to Hybrid mechanism of NN. All the results are shown in the MSE graphs.

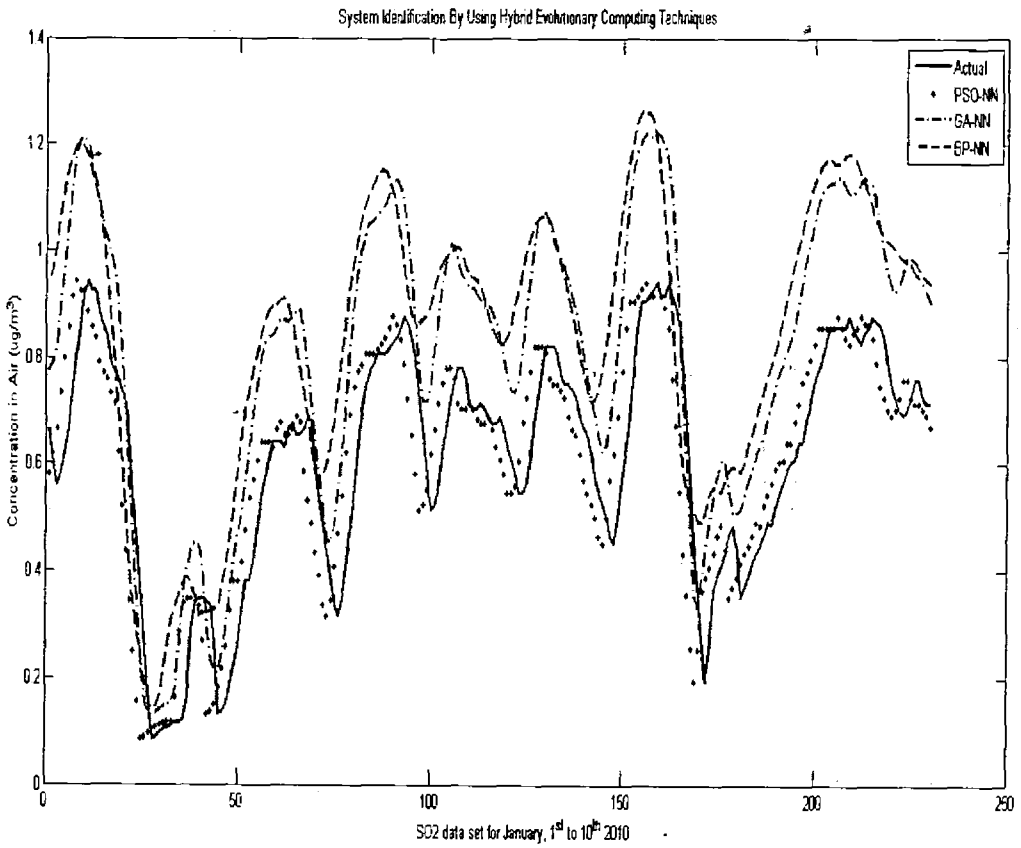

Figure 10: System Identification on RSP data for 1st ~ 10th day of January

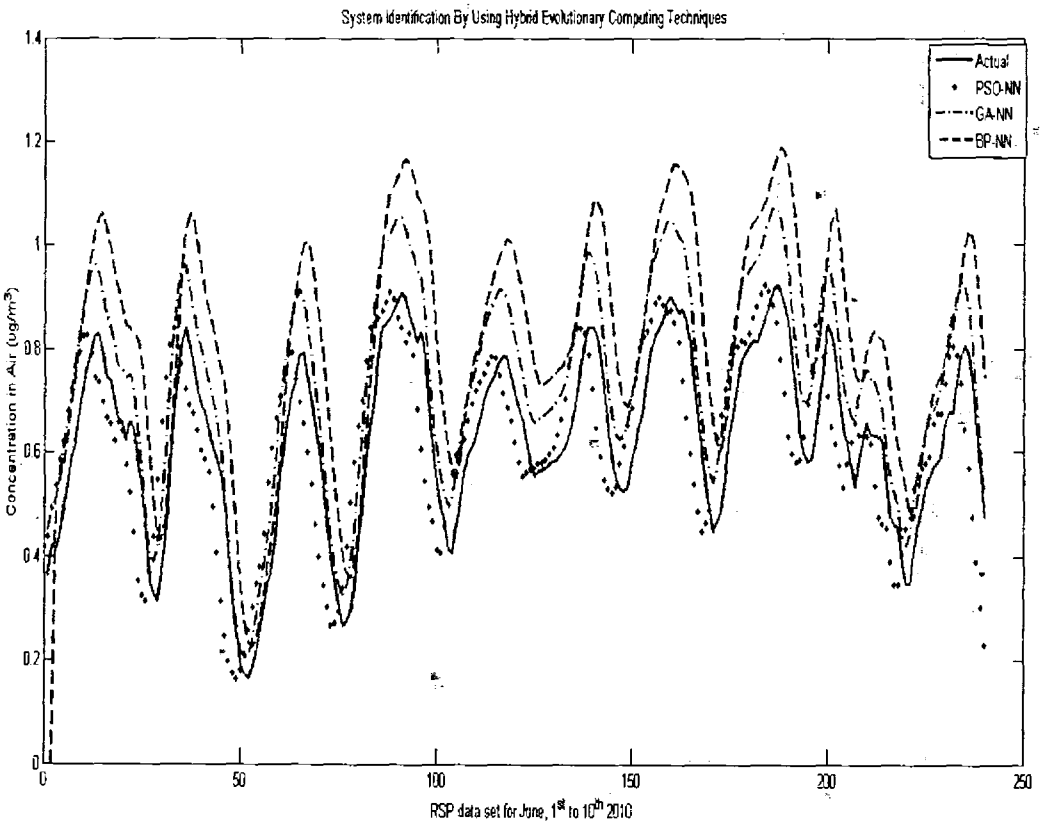

Figure 11: System identification on RSP data for 1st ~ 10th day of June

Figure 8 and figure 9 shows us the system identification results done by hybrid nature of PSO with NN (PSO-NN), and GA based NN (GA-NN), and the results of NN were also shown. We can conclude that the performance of PSO-NN is ranked best while the GA-NN performs well as compared to simple NN. These hybrids Nature of Algorithm with said parameters shows good results on the given data set.

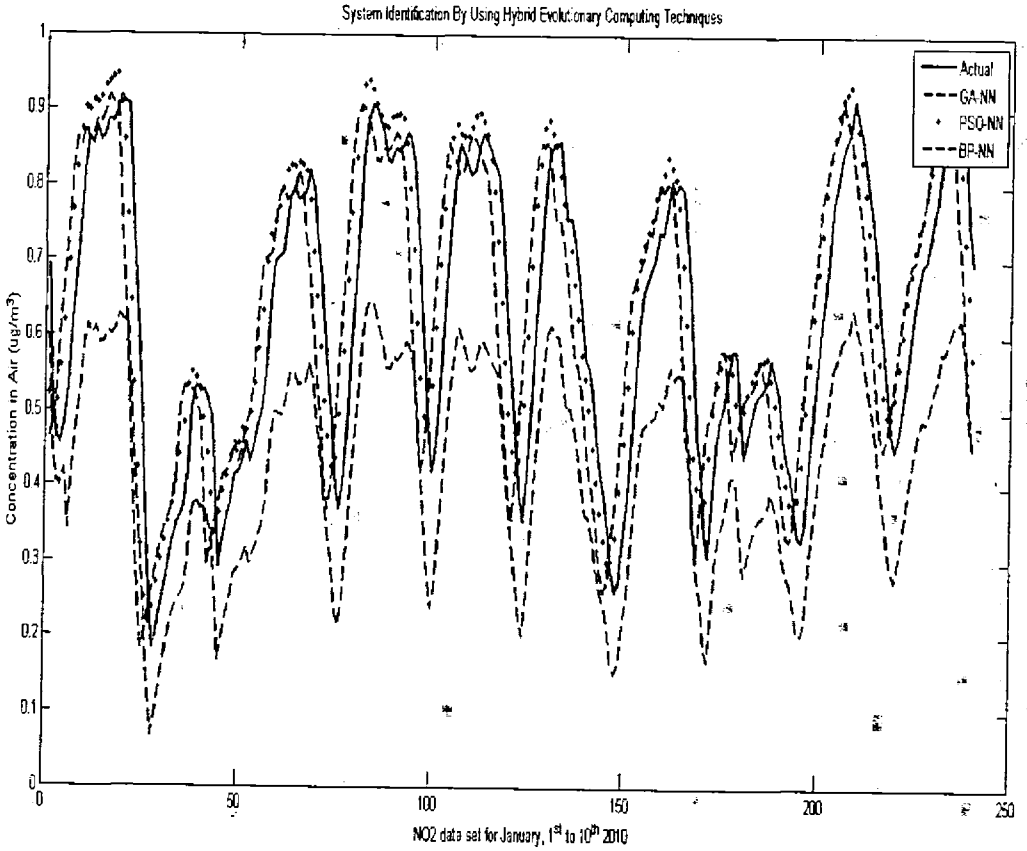

Figure 12: System identification on NO2 data for 1st ~ 10th day of January

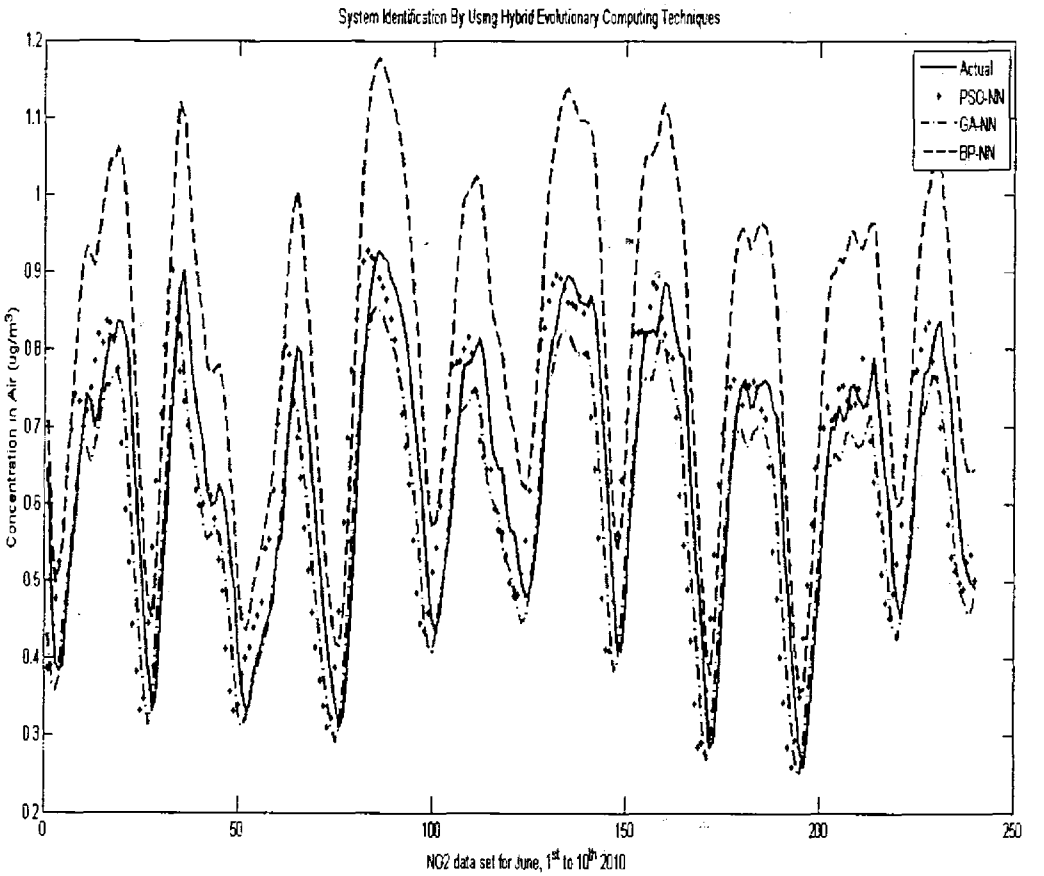

Figure 13: System identification on NO2 data for 1st ~ 10th day of June

The Figure 10 and figure 11 shows us the Tracking behavior or identification of system results for the dataset of NO_2 both for January and June month. We can see that from figures that when we face the minima or steeps in our data, our NN shows results which are not graded as best in comparison to PSO-NN and GA-NN. The PSO-NN is very close to actual data points while the GA-NN comes second.

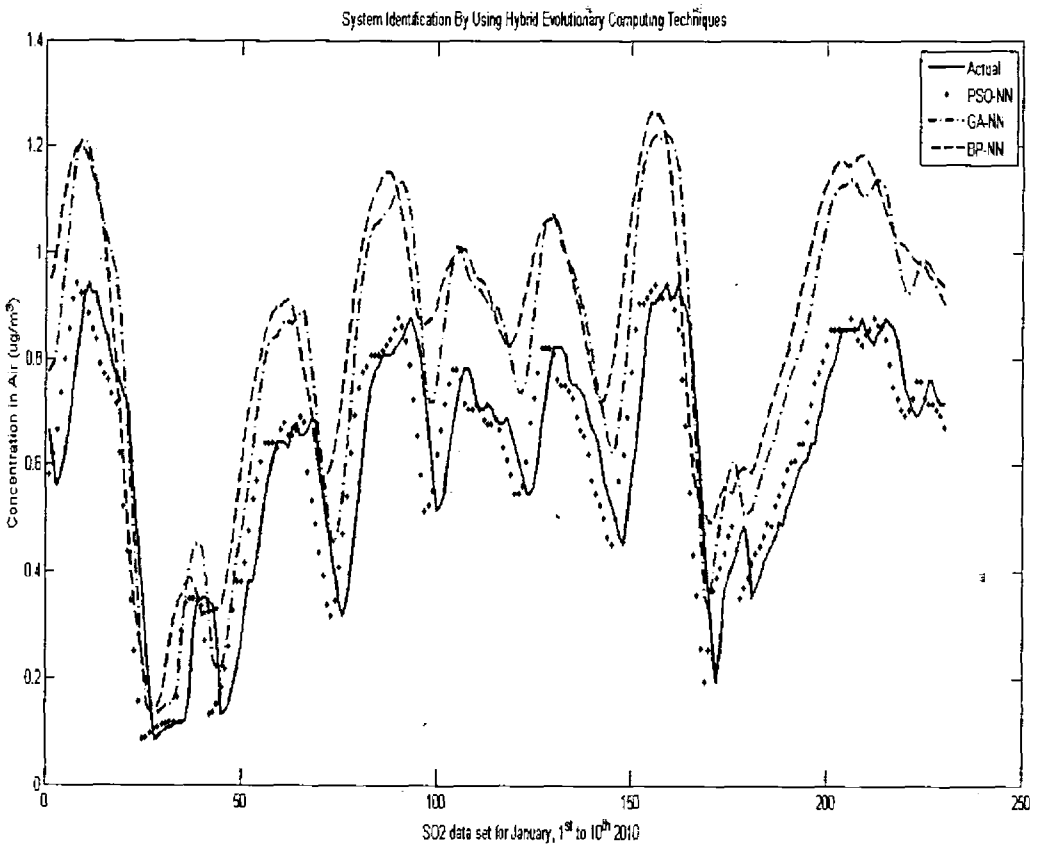

Figure 14: System identification on SO2 data for 1st ~ 10th day of January

The third ingredient of the polluted air is SO_2 as shown in figure 12 and figure 13. Here too the PSO-NN performs well as compared to GA-NN and NN, both for the month of January and June. Again the NN gets stuck in the local minima points and hybrids nature shows us the best results.

5.3 FUTURE VALUE PREDICTION

In the second phase of our thesis, we will calculate the Future value calculation in order to testify our algorithms for the unseen data and the obtained vector will then compared with the actual results. And the Difference will show us the performance achieved.

The Figure 14 and Figure 15 show us the future values calculated by PSO-NN, GA-NN, and NN for the data set of RSP both for the month of January and June. It is clear from graphs that the PSO-NN gives us optimum results.

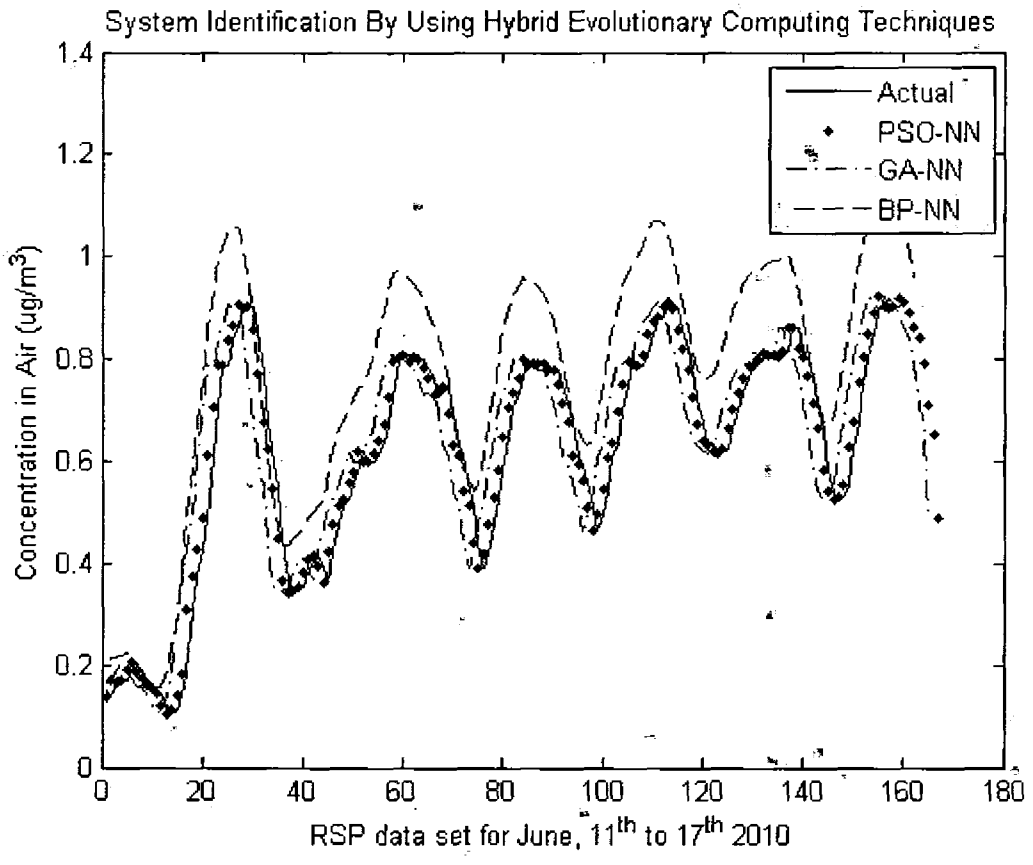

Figure 15: Future values on RSP data for 11th ~ 17th day of June

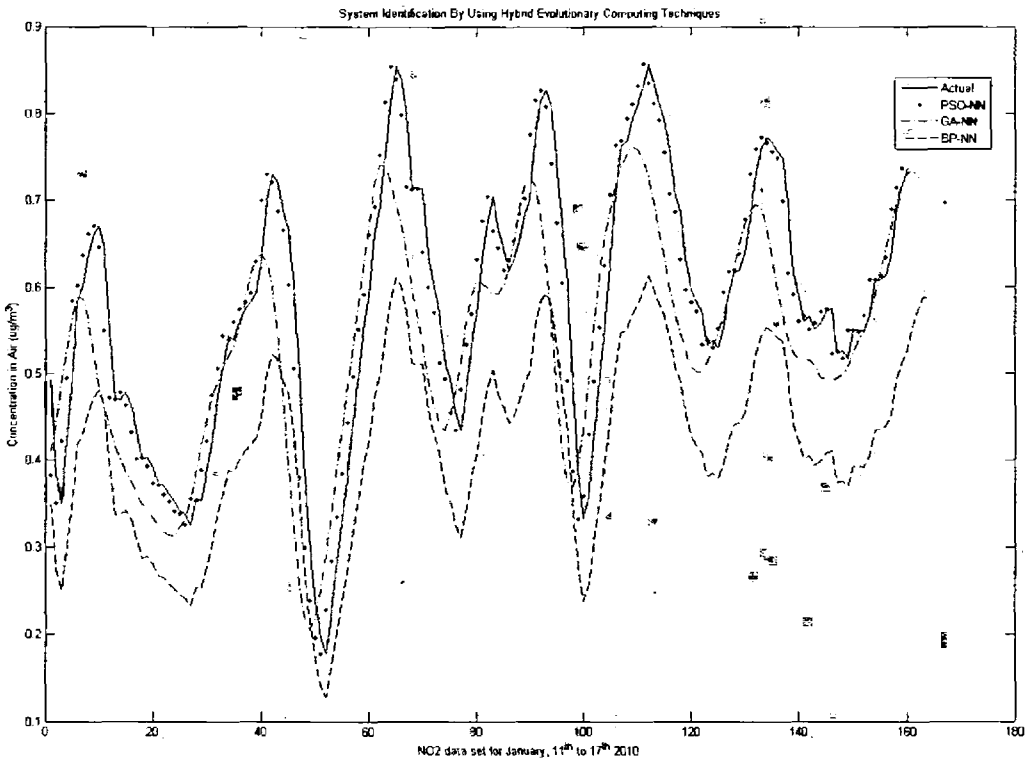

Figure 16: Future values on NO2 data for 11th ~ 17th day of January

Figure 16 shows us the future behavior for the data set of NO_2 and next seven days data is predicted, and then it is compared with the actual data, similarly the figure 18

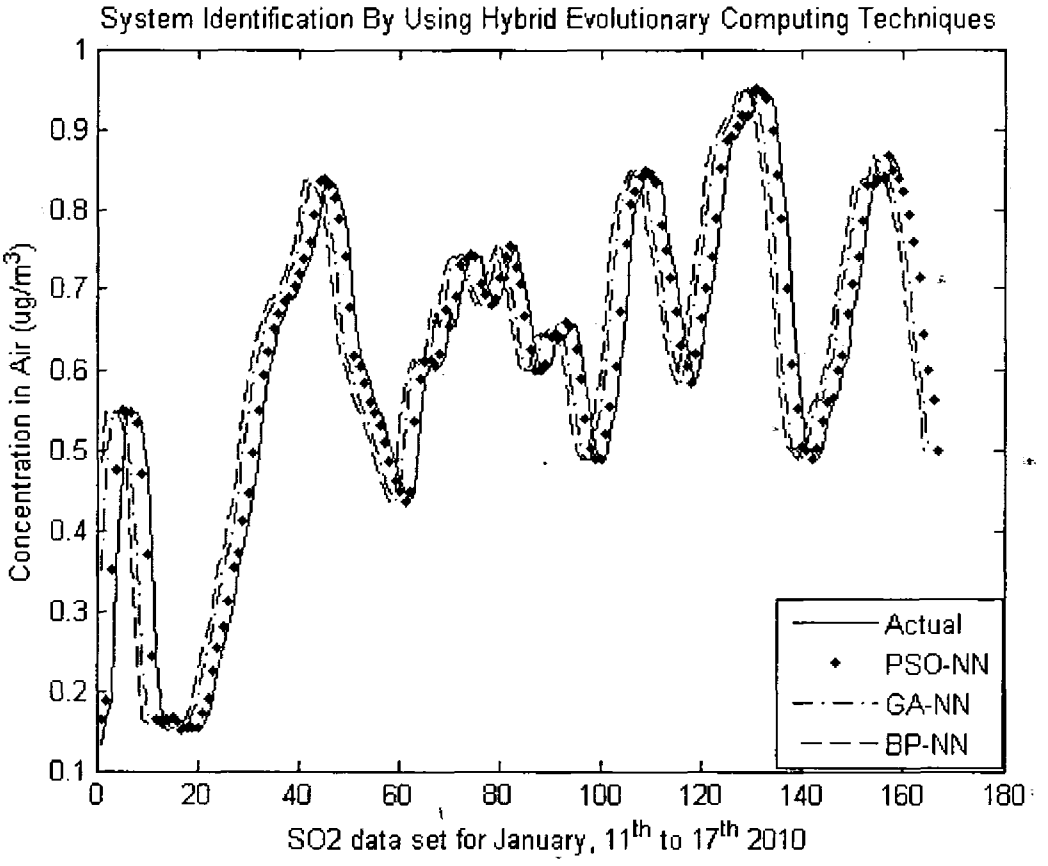

Figure 17: Future values on SO2 data for 11th ~ 17th day of January

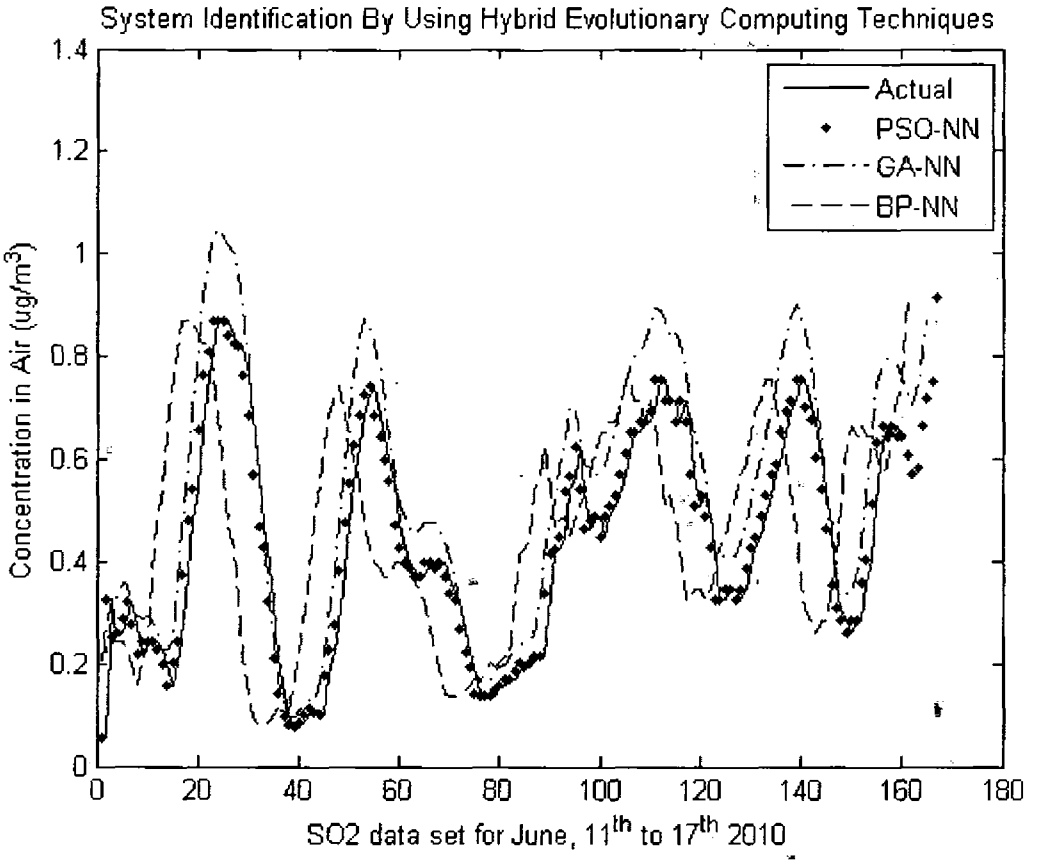

Figure 18: Future values on SO2 data for 11th ~ 17th day of June

And figure 19 shows the results of data set SO_2 both for the month of January and June, and next seven days data is predicted which is then compared with the actual data set and hence results were obtained. The MSE and time consumed by all the algorithms were shown in the coming figures.

5.4 MSE TRACKING

In the above graphs, we concluded that that the proposed algorithms performs well and produces results up to the mark. Their MSE results are shown below and we can see easily that the achievement of PSO-NN is on the top of the list and the GA-NN makes its position

runner up, while the standalone NN comes at the bottom in achieving the minimum value of MSE. The number of iterations was set to

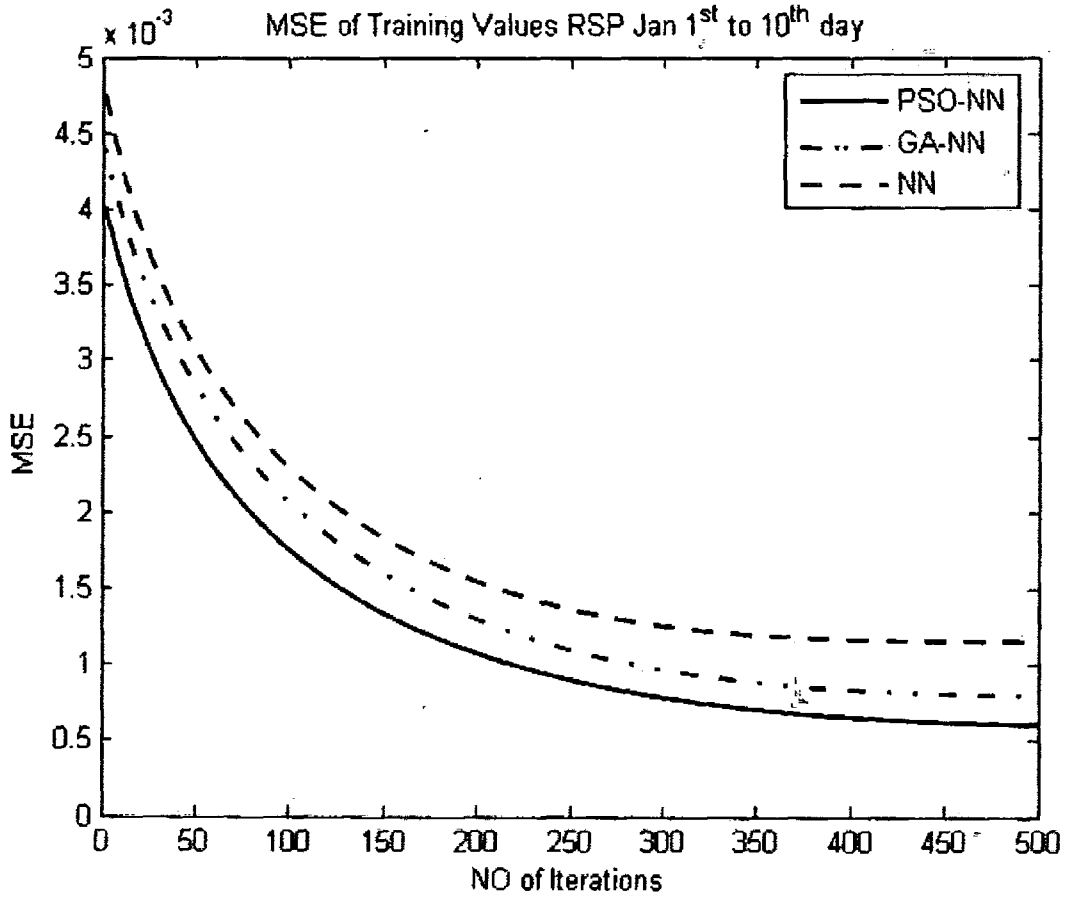

Figure 19: MSE of RSP dataset for the Month of January 1 to 10, 2010

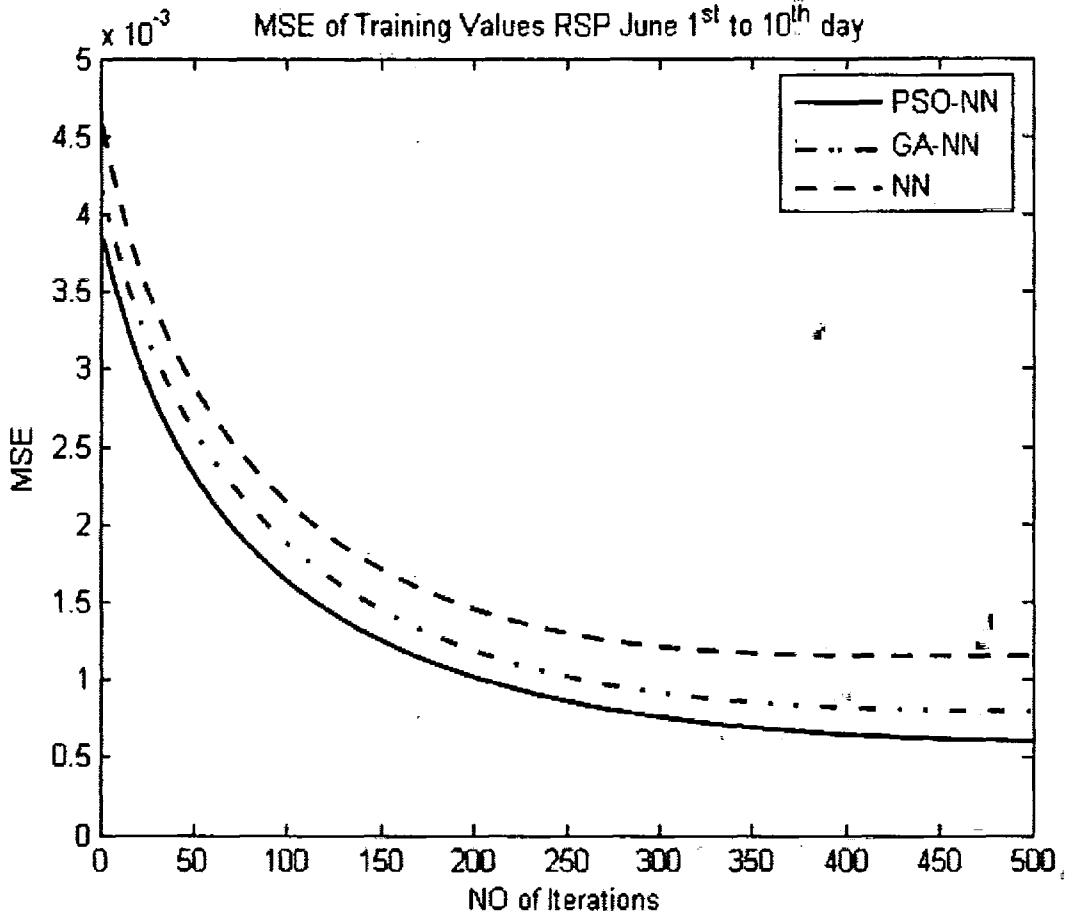

Figure 20: MSE of RSP dataset for the Month of June 1 to 10, 2010

1000 in order to achieve the results, but here only 500 iterations were shown because the results remain constant after 500 iterations for all the results and 1000 iterations both for RSP January and June datasets.

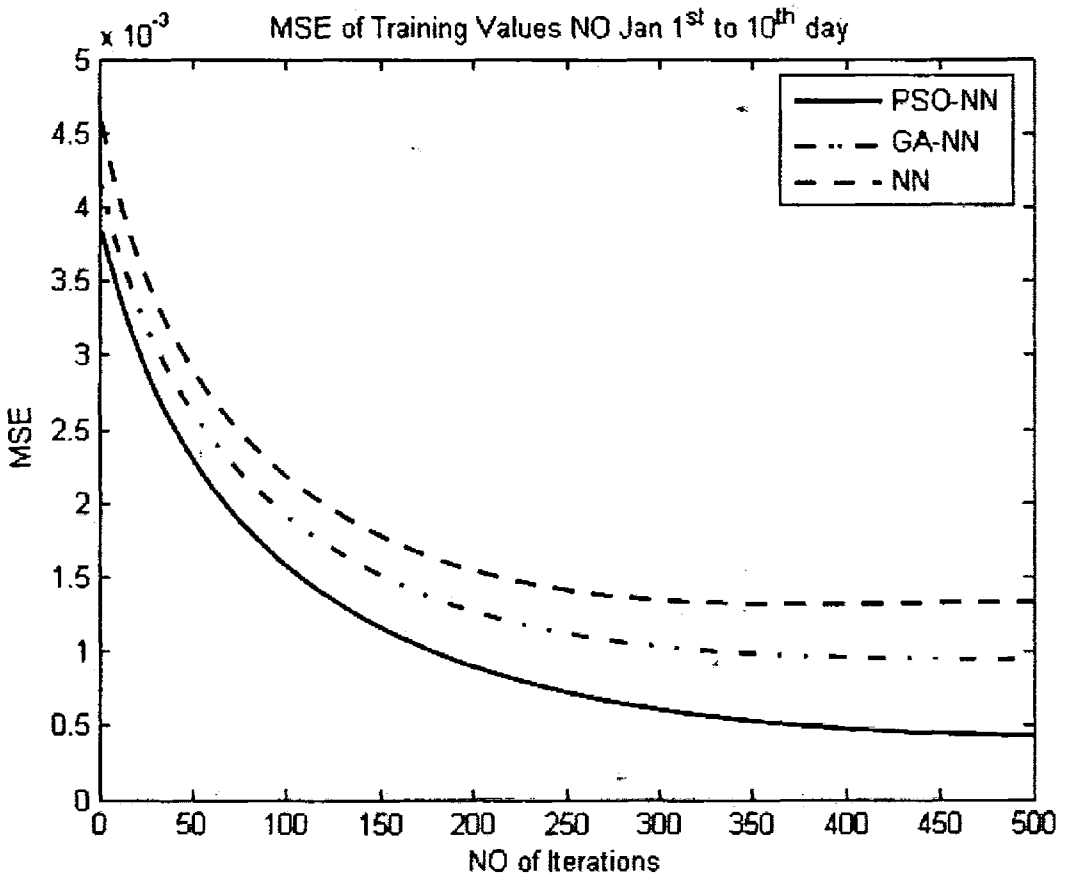

Figure 21: MSE of Training data for NO2 data for 1st ~ 10th day of January

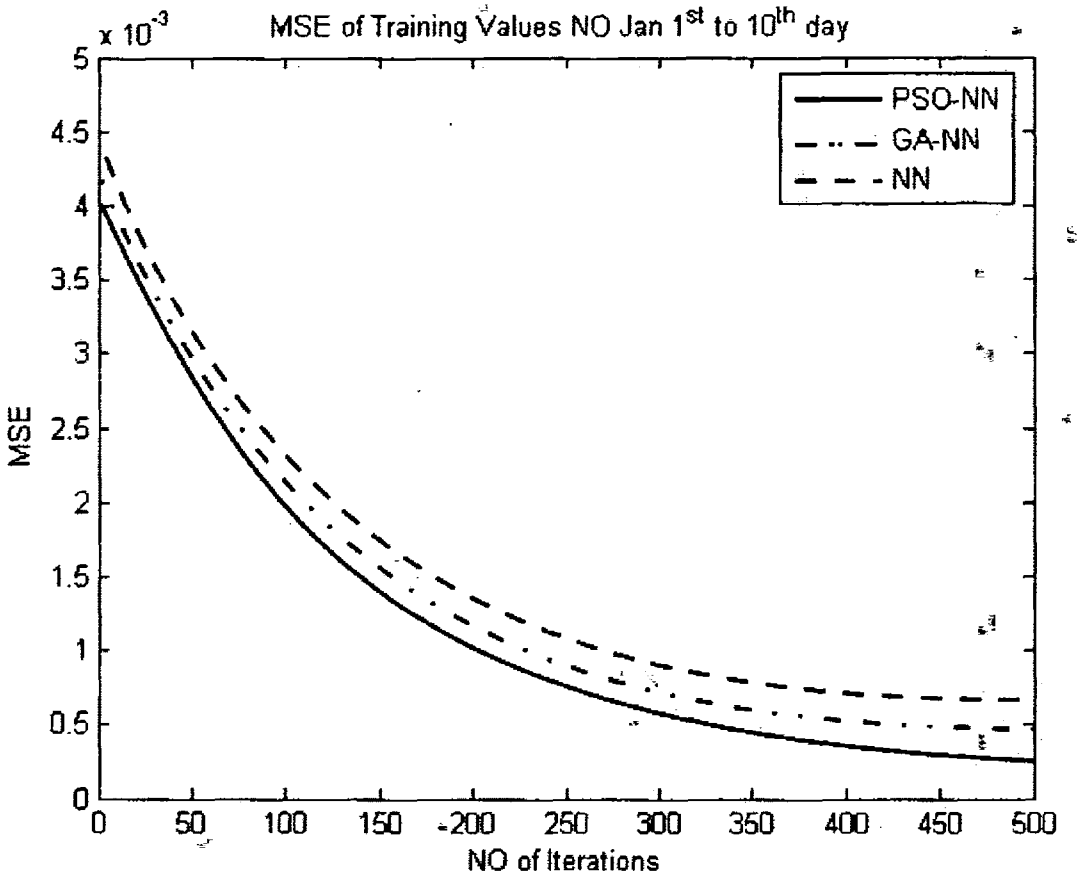

Figure 22: MSE of Training data for NO2 data for 1st ~ 10th day of June

The figure 20 and Figure 21, having the MSE graphs for the training data for RSP month of January and June shows that after 100 to 150 iterations, we obtain the Minimum values of our proposed algorithm. However, 500 iterations were shown in the graph in order to analyze the performance achievement. We can see from the figures that the minimum value achieved by PSO-NN and GA-NN is at the big distance from NN

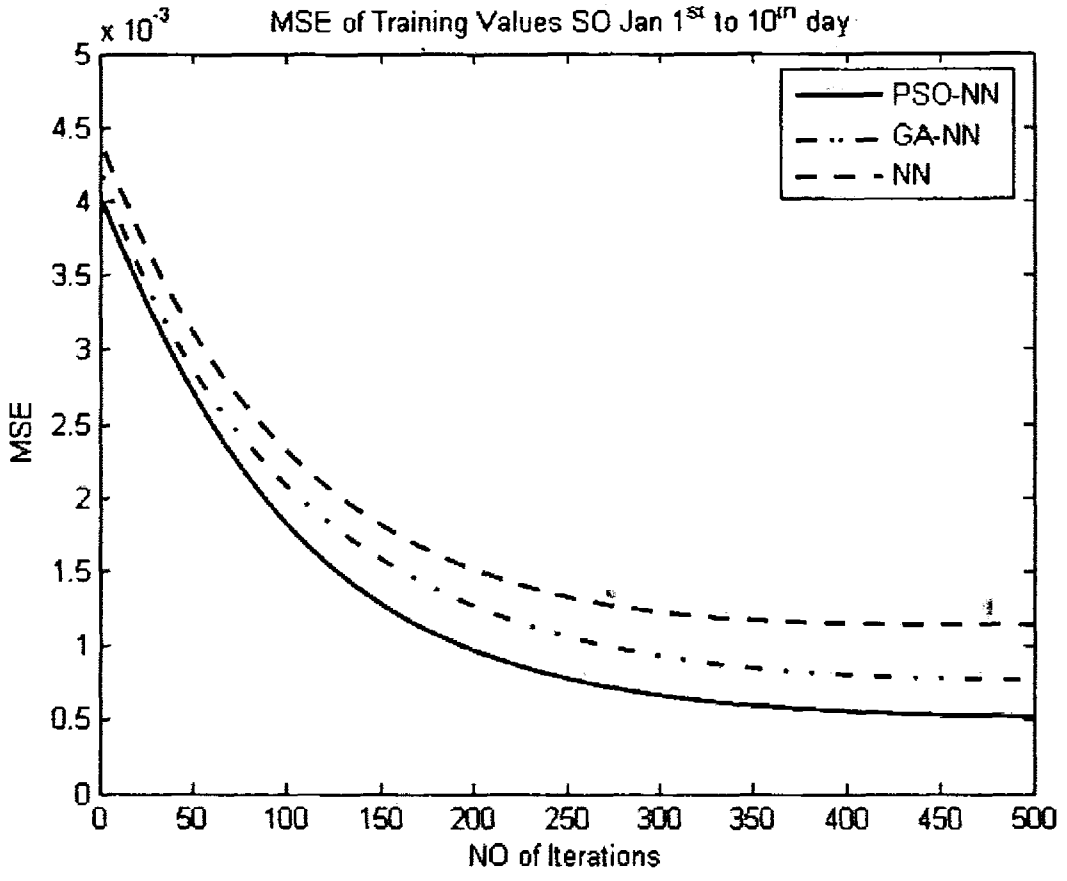

Figure 23: MSE of Training data for SO2 data for 1st ~ 10th day of June

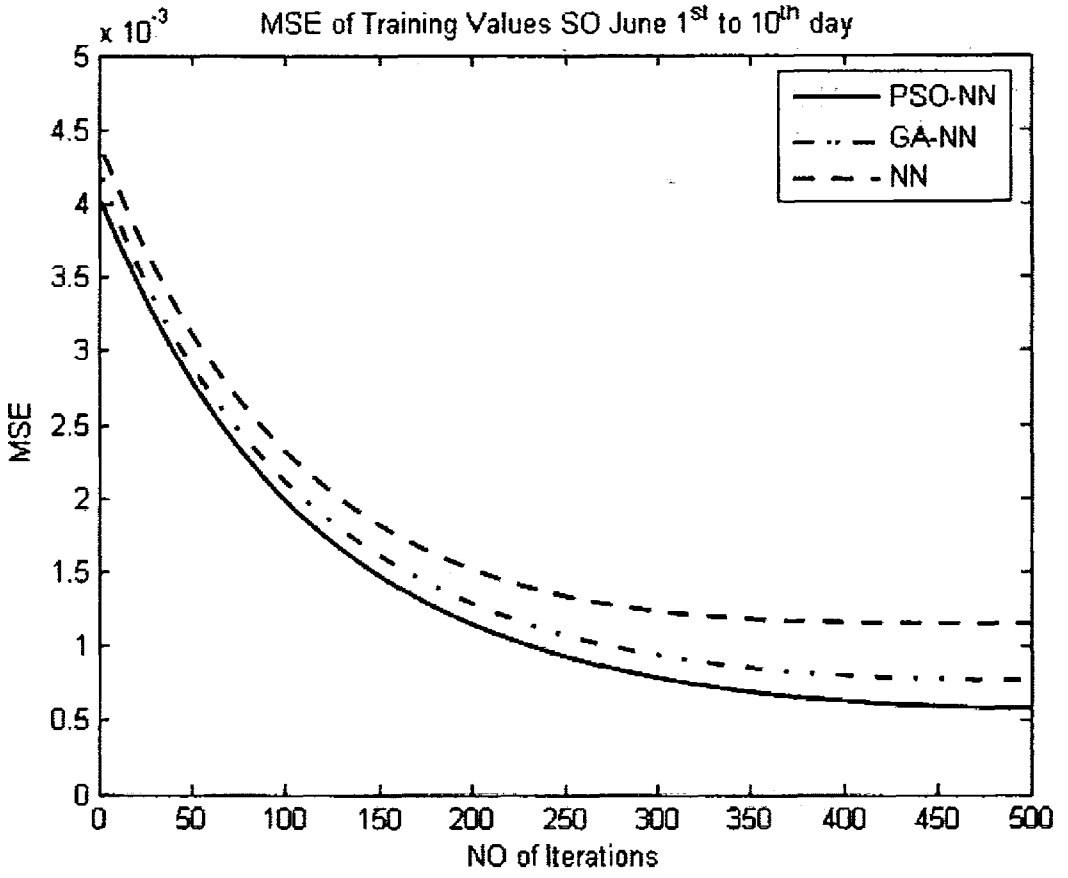

Figure 24: MSE of Training data for SO2 data for 1st ~ 10th day of June

Similarly, we can see in the figure 22, and figure 23, that the difference between the MSE graph of NN, GA-NN and PSO-NN lies close to each other and there is not much difference as compared to RSP dataset. The graphs having data of January month becomes constant after 80 to 120 iterations, while for the June month, after 300 iterations it becomes straight horizontal line.

5.5 MSE FUTURE VALUES

The figure 24, and figure 25, shows us the converging behavior of our proposed algorithms with NN on the SO_2 dataset. We can see that for the month of January, the NN converges and get stuck in some local minima, and other two algorithms did not face that problem, how

ever, NN becomes Successful in handling the case , but it results the slow convergence of algorithm. How ever, the MSE graphs of all the algorithms become same shape after 350 iterations, while the graph is shown up to 500 iterations. On the other hand, the dataset for the month of June, the algorithms converge quickly, and the graphs become straight after 200 iterations. In the start of the graph, PSO-NN and GA-NN get stuck in local minima due to complex nature of data, however, the quickly overcome the hurdle and converges smoothly.

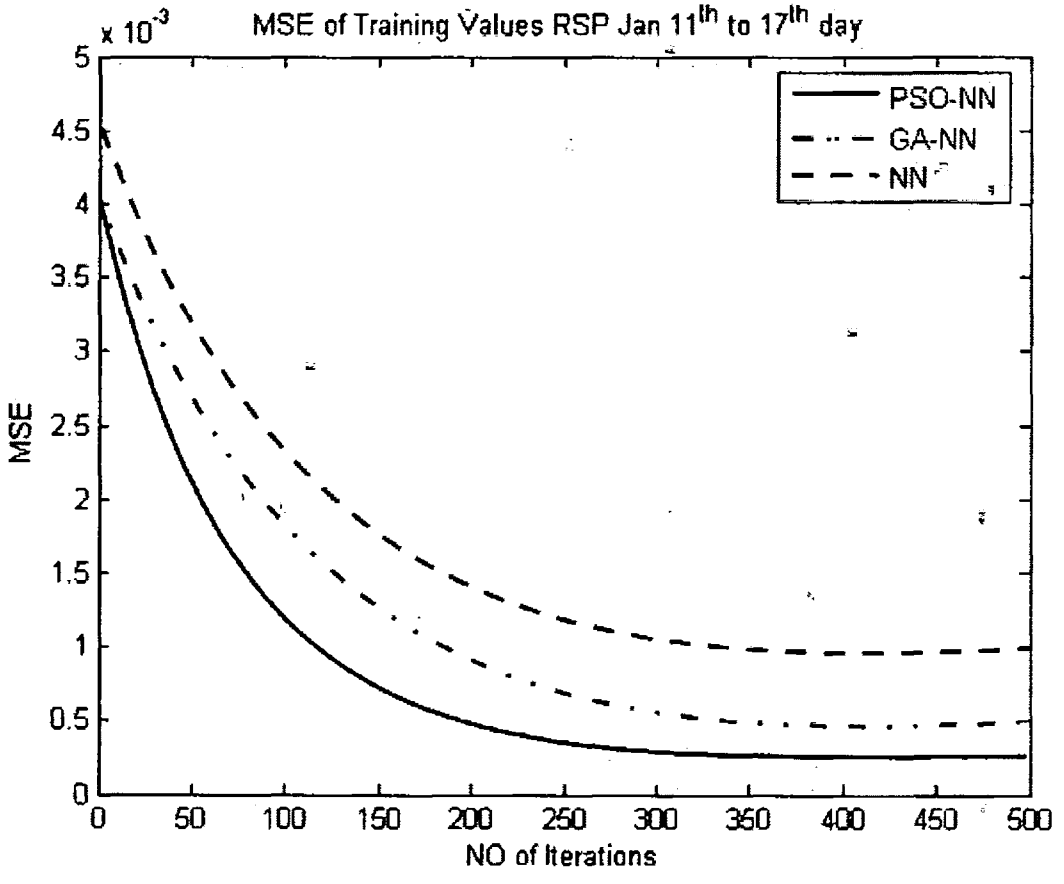

Figure 25: MSE of Future values on RSP data for 11th ~ 17th day of January.

From figure 26, we conclude that for the future value calculation and convergence, the GA-NN converges faster than PSO-NN, but after 50 iterations; the PSO-NN improves its self and gives us the minimum error. The minimum error achieved by PSO-NN is 1.109 x 10-3, while

the minimum error achieved by GA-NN is 1.22 x 10-3 for the month of January, similarly, we can see the same behavior for the month of June, in which PSO-NN get 5.26 x 10^{-3}, while GA-NN have 6.21 x 10^{-3}. The graph shows us the 1000 iterations. However, all the curves are close to each other. But for the dataset of June month, there is much distance and the poor performance is given by standalone BP-NN and PSO-NN becomes straight line after 600 iterations.

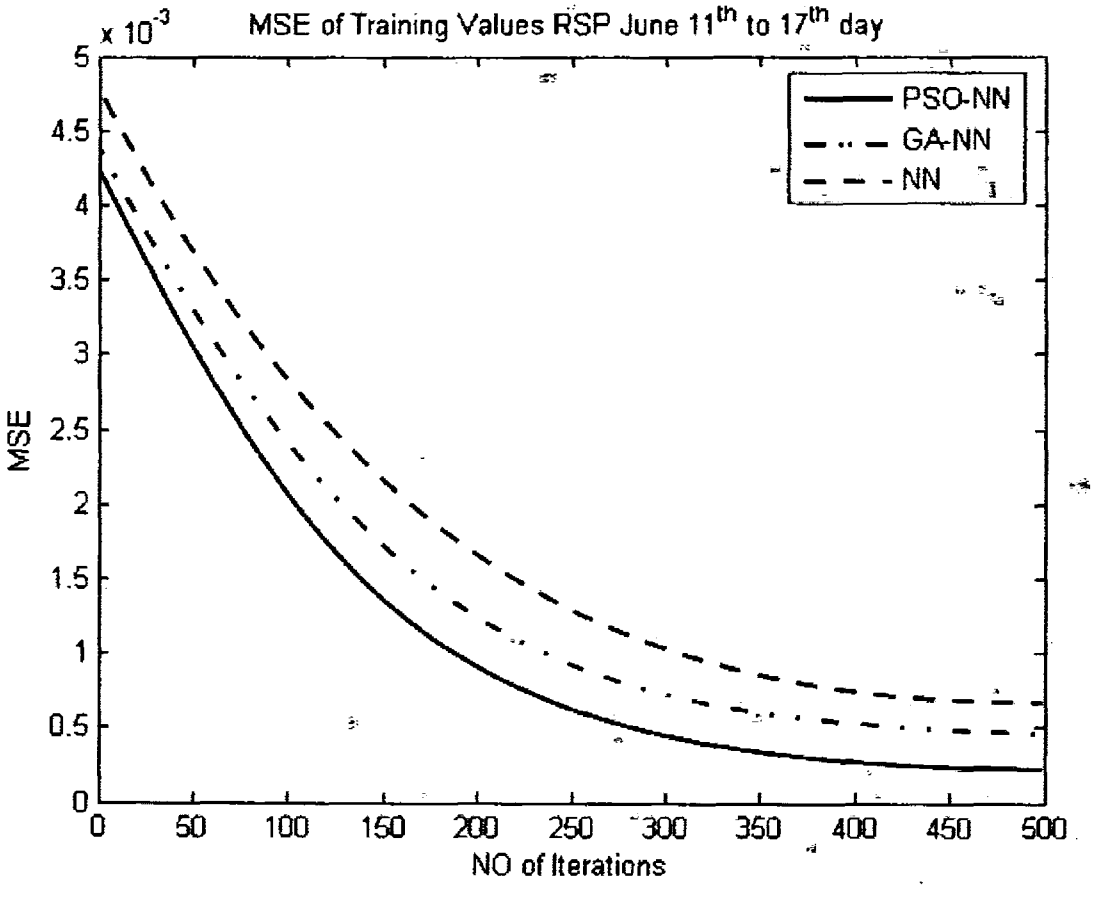

Figure 26: MSE of Future values on RSP data for 11th ~ 17th day of June

Similarly, from figure 28, and figure 29, we conclude that PSO-NN converges very fast as compared to other proposed algorithm i.e. GA-NN and with comparison with NN. The minimum error between actual data and the predicted by PSO-NN is 1.98×10^{-3} for the

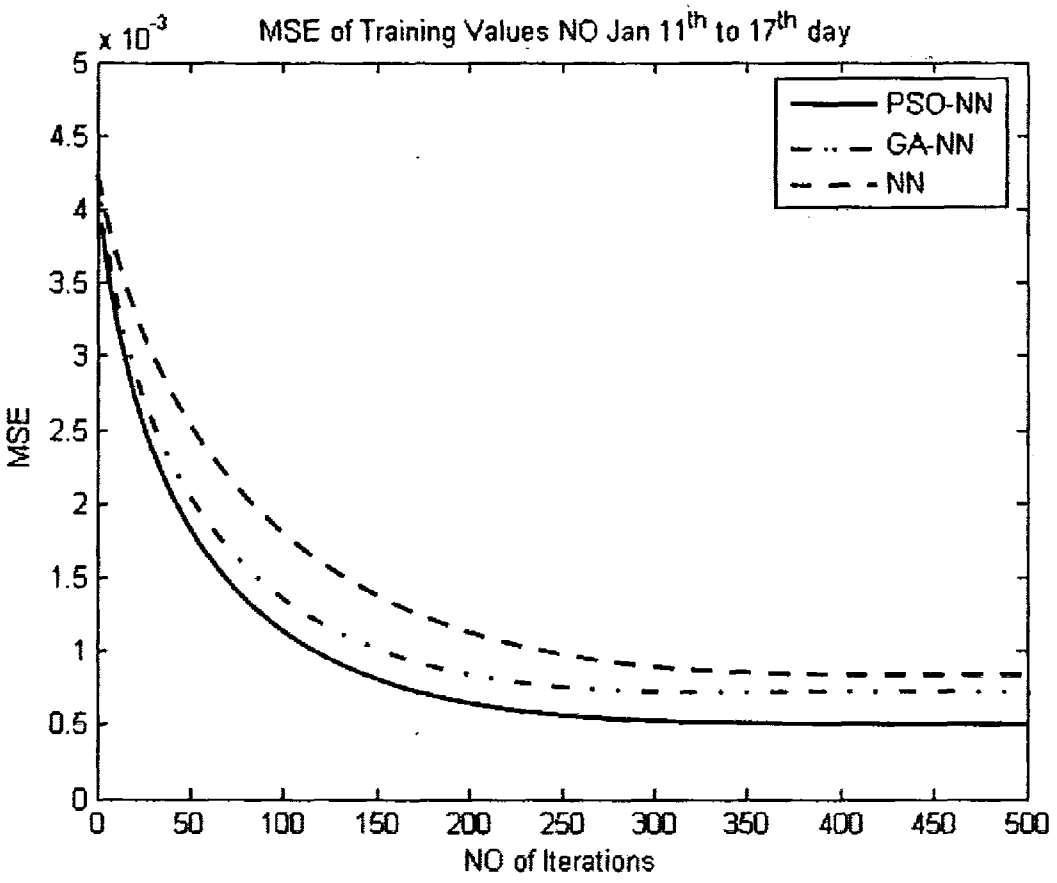

Figure 27: MSE of Future values on NO2 data for 11th ~ 17th day of January

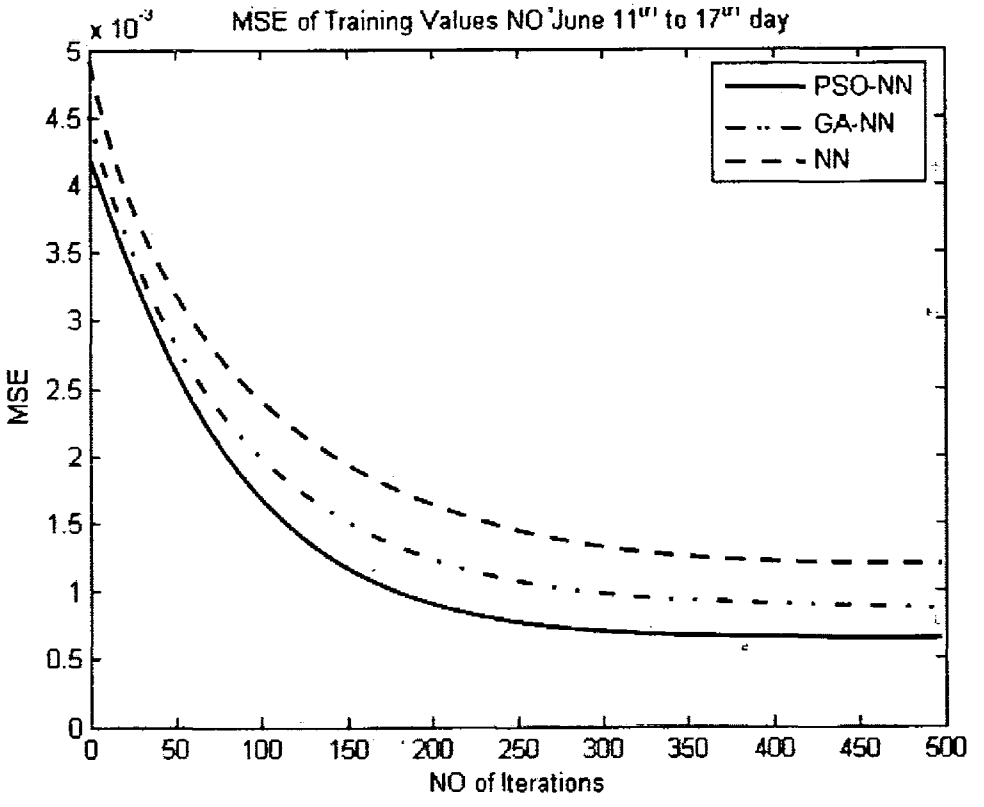

Figure 28: MSE of Future values on NO2 data for 11th ~ 17th day of June

Month of January, while for the month of June the minimum error achieved is 2.36×10^{-3}. The details of other algorithms are given in the tables 6 and table 7 respectively. If we look at the data set of air pollution ingredient SO2, in the figures 30, and 31, we will conclude that

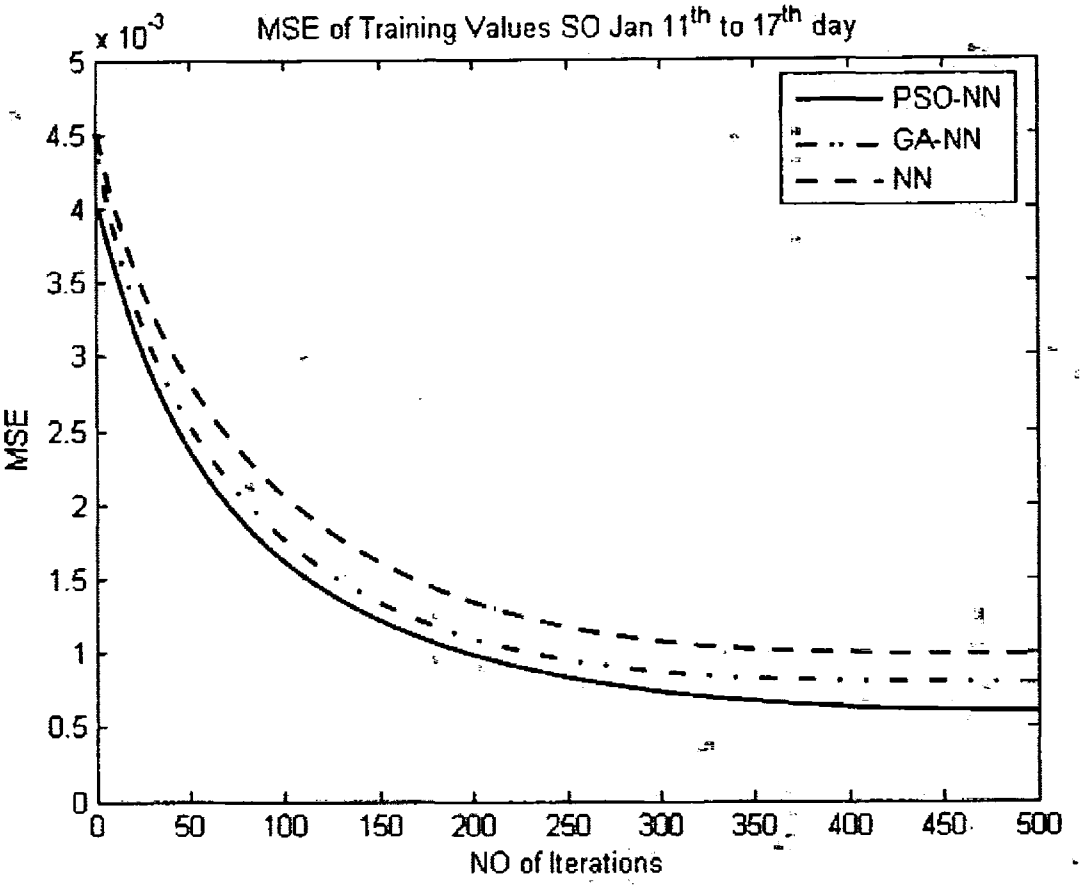

Figure 29: MSE of Future values on SO2 data for 11th ~ 17th day of January

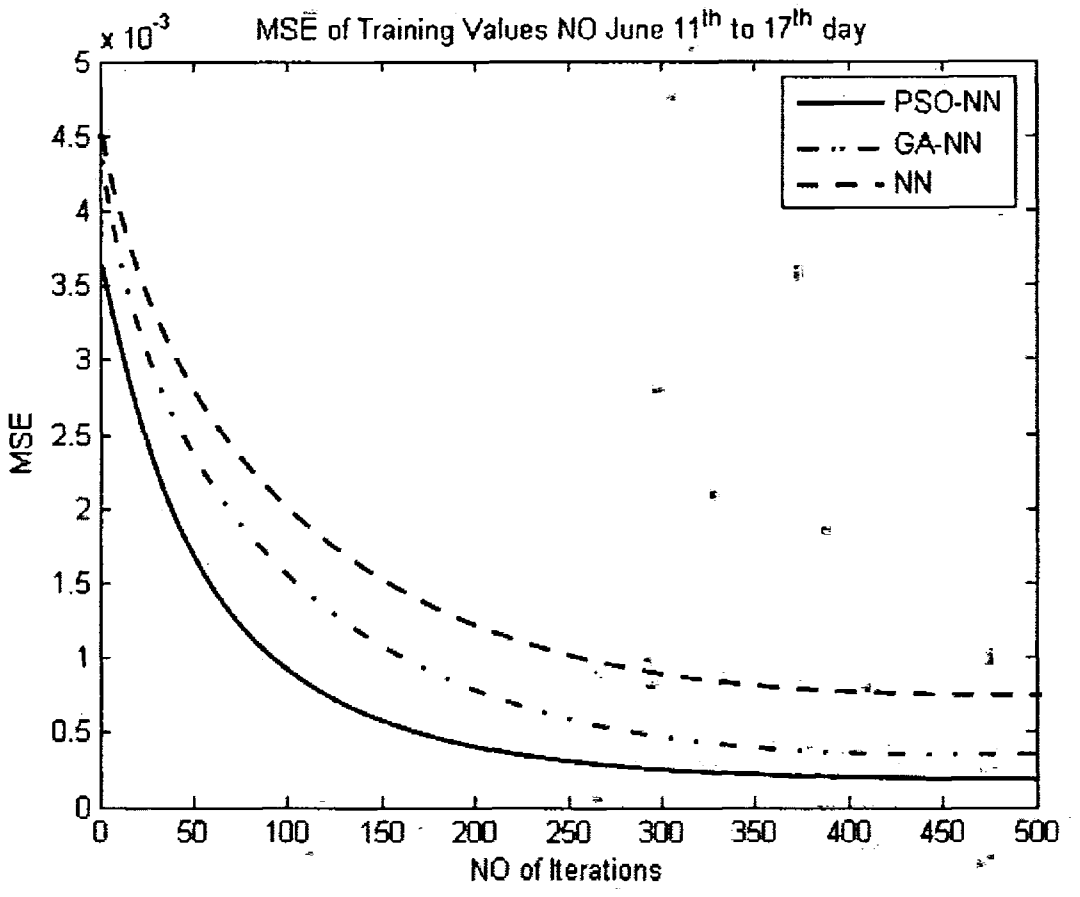

Figure 30: MSE of Future values on SO2 data for 11th ~ 17th day of June

The minimum error achieved is 1.265x 10^{-3}, while the mean square error for the month of June is 1.022 x 10^{-3} respectively.

5.6 MSE COMPARISION

Below is the tabular form of the minimum error achieved during our experiments and comparison will prove that PSO-NN performance is much better than other two algorithms,

Table 5: MSE of Training Data for the Month of January

MSE Training for the Month of January (10^{-3})			
	PSO-NN	GA-NN	NN
RSP	7.28×10^{-3}	8.27×10^{-3}	9.42×10^{-3}
NO2	1.386×10^{-3}	1.461×10^{-3}	1.527×10^{-3}
SO2	4.45×10^{-3}	5.57×10^{-3}	7.31×10^{-3}

Table 6: MSE of Training Data for the Month of June

MSE Training for the Month of June			
	PSO-NN	GA-NN	NN
RSP	7.49×10^{-3}	8.11×10^{-3}	1.082×10^{-3}
NO2	1.373×10^{-3}	1.424×10^{-3}	1.476×10^{-3}
SO2	1.244×10^{-3}	1.319×10^{-3}	1.415×10^{-3}

In the training phase, we can see from the tabular form, the data set of SO_2 achieved the minimum error of all the values while the data of RSP, the values can be seen from the table easily. While the data of NO_2 shows less MMSE, this is due to more complexities in the data. Same is the case for the month of June too. This also put effect on the tracking behavior of the algorithm.

Table 7: MSE of Tracking Data for the Month of January

MSE Tracking for the Month of January			
	PSO-NN	GA-NN	NN
RSP	1.109×10^{-3}	1.122×10^{-3}	1.157×10^{-3}
NO2	1.98×10^{-3}	3.02×10^{-3}	4.86×10^{-3}
SO2	1.26548×10^{-3}	1.59665×10^{-3}	1.74493×10^{-3}

Table 8: MSE of Tracking Data for the Month of June

MSE Tracking for the Month of June			
	PSO-NN	GA-NN	NN
RSP	5.26×10^{-3}	6.21×10^{-3}	5.94×10^{-3}
NO2	2.36×10^{-3}	3.18×10^{-3}	4.12×10^{-3}
SO2	1.02283×10^{-3}	1.37857×10^{-3}	1.71703×10^{-3}

The table 6 and table 7 shows us the tracking results of the proposed algorithms and their comparison with the NN, the minimum MSE achieved is for the data set of SO2, and after that NO2 comes , while the bad performance can be observed for the data of RSP for the month of January. If we look at the data set of June, we will observe that the MMSE achieved is for the SO2, while the second number comes NO2, and the third number comes is of RSP.

5.7 TIME COMPSRISION

From the time chart, we can conclude that the time taken by PSO-NN is less than the GA-NN, and the time taken by NN is highest of all. For the month of January also, we can see that the minimum time consumed is for SO2 data set because it is less complex as compared to RSP data set. While the NO2 data set took greater time comparatively to RSP due to its tough nature. Thus RSP is ranked normal as compared to both NO2 and SO2 data values. Thus the NN took more time as compared to other two proposed techniques for both month of January and June.

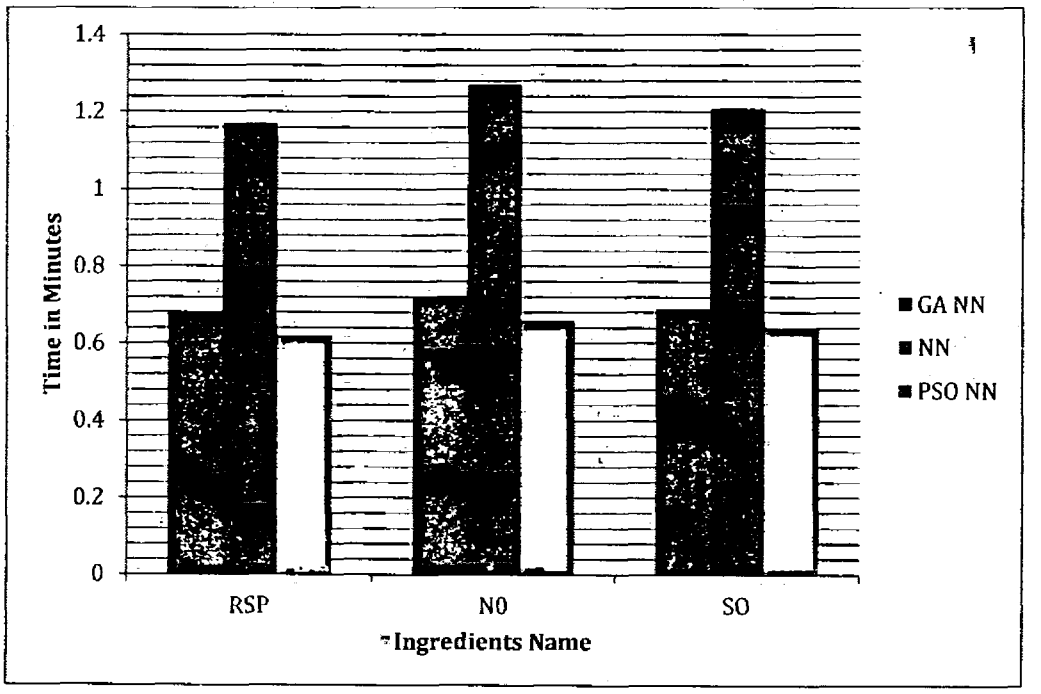

Figure 31: System Identification Time Consumption for January Month data

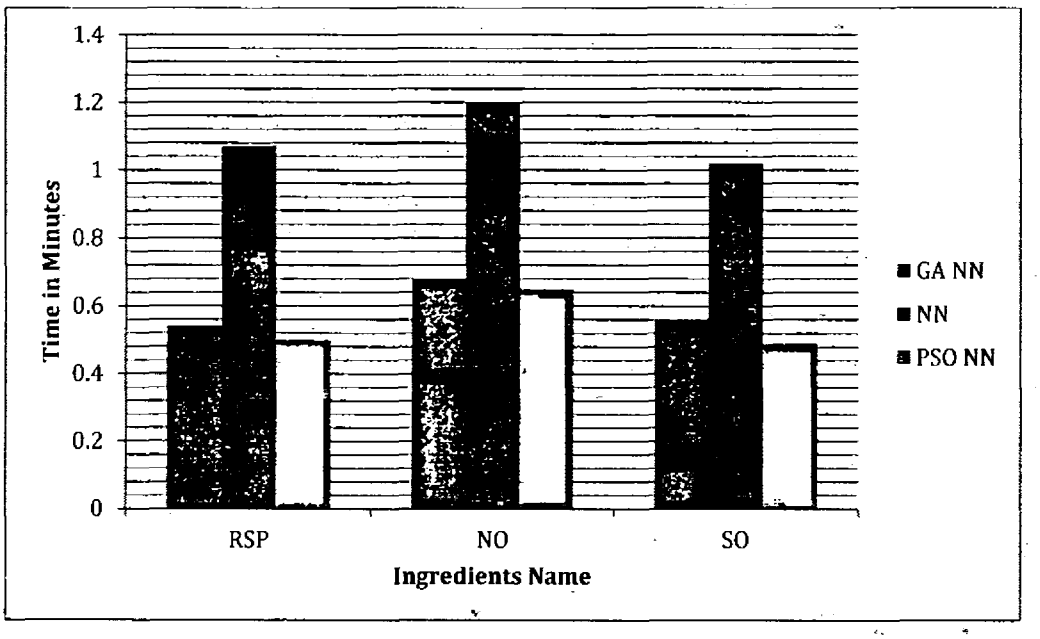

Figure 32: System Identification Time Consumption for June Month data

And the Future value calculation time can be seen in the table 34 for the month of January and Figure 35 for the month of June. We can see that the RSP took little bit greater time as compared to NO2 data set and SO2 data took least of all. Similarly, for the month of June, the minimum time consumed is also SO2, while the maximum time to predict the future response is RSP.

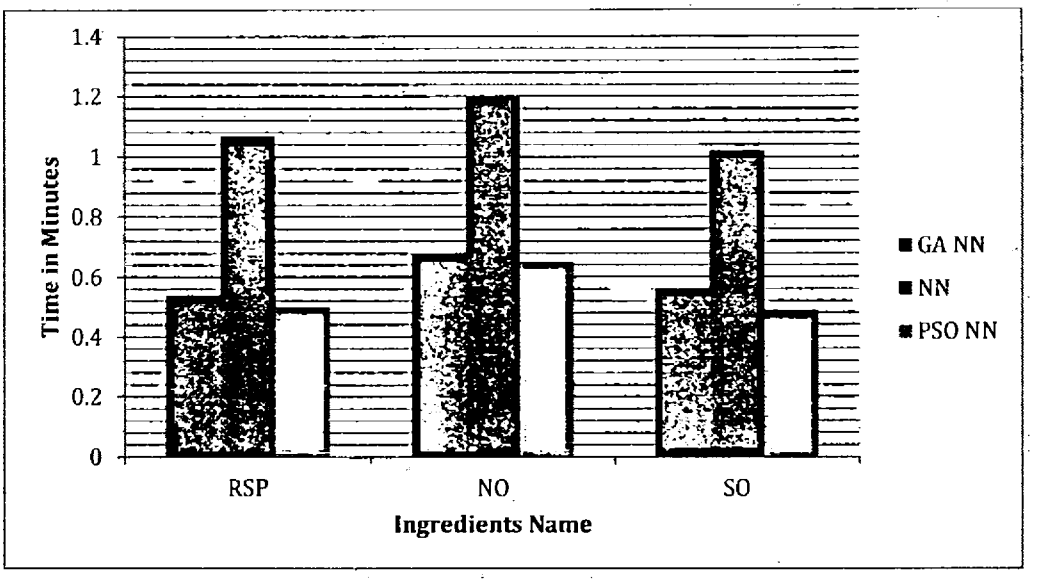

Figure 33: Future values time calculation for the month of January data

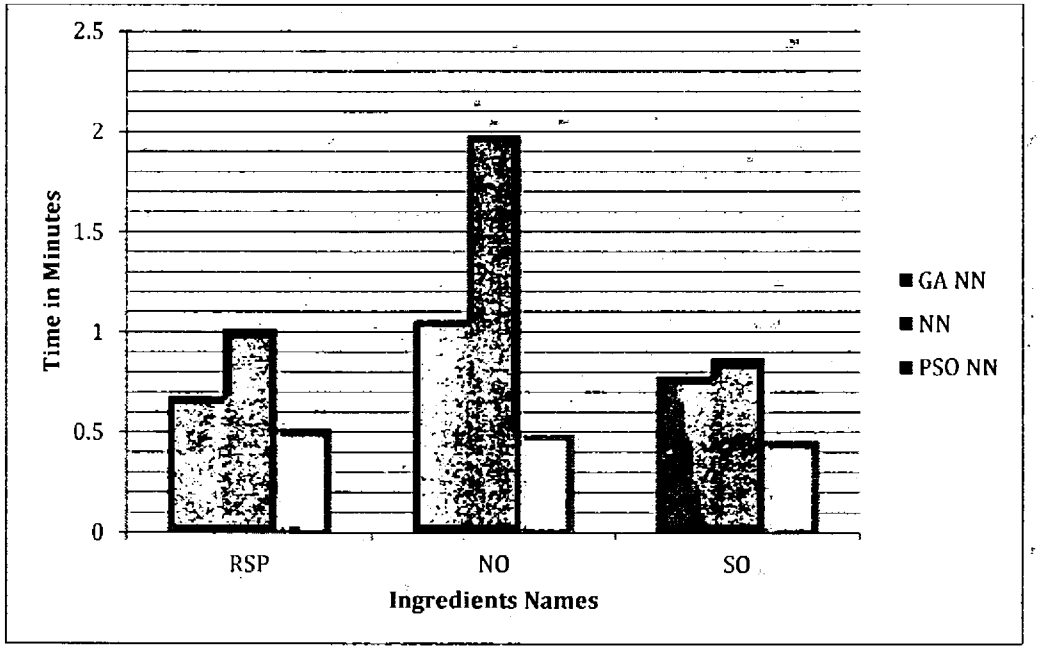

Figure 34: Future values time calculation for the month of June data

From the above analysis we noted that the results of NN achieved are not good for this specific data set, and it varies from data to data. When we run the program, we generate the weights at the start randomly. And it gives us the different result, and when we run the program second time with all the parameters unchanged. We get different results. This proves the instability of NN on this particular data set.

5.8 PERFORMANCE COMAPRISION

Up till now have shown all the results individually, now its time to summarize the whole work into single table from where it will be easy to analyze the performance of proposed algorithm and previously used algorithms. We will discuss about the convergence time of algorithms, time to complete the specified number of iterations, and the minimum error achieved. And at the end we will put our focus on the efficiency of algorithms.

Table 9: performance comparison between Proposed and Existing Techniques

Parameters	PSO-NN	GA-NN	BP-NN
Convergence	Very Fast	Above Average	Slow
Time Consumption (/1000 iter)	Less than min	Average	Higher than PSONN & GANN
Efficiency	Global Optimum	More randomness	May get into Local Minima
Implementation	Very Easy	Easy	Very Complex

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Evolutionary computing based Artificial Intelligent system identification and calculation of its future response for chaotic data is not an easy field. This study enables us to predict the future response of pollutants present in air for the next seven days. In this research we didn't start from environmental engineering point of view, but followed the natural inspiration method to generate results.

Proposed system shows better results for predicting the desired results in the domain of future.

6.1 CONCLUSION

Five different ingredients were obtained from the environmental protection agency measured at Mong-kok road side station at Hong Kong for the data set of year 2010, among which three were used in this research according to their dangerousness to the human health, which are then normalized for preprocessing, then Evolutionary computing techniques were applied on it to get best optimized weights which were further used in neural network to calculate future

response for one day and one week respectively. This hybrid technique provides us results better that standalone PSO, GA and NN in the standards of MSE. First of all basic neural network was implemented and results were obtained, but it does not suits best for given data, however, GA and PSO performs better that NN. The data of first ten days were used to train the NN and also GA and PSO, and the results will be testified by the data of remaining days. Here we predict the results for week ahead prediction. Performance of all the algorithms were compared and we see that the results obtained from PSO based Hybrid NN were better than GA based hybrid NN. From the numerical results shown in simulation and results chapter we conclude that the convergence time of PSO-NN is faster than other schemes used. Following main points were concluded in this research work

1 The model of PSO –NN is proposed by virtue of the dynamic characteristics of Air pollutants level, which increased the forecast precision. Therefore, the method is reliable and effective.

2 This method can be used to deal with the nonlinear and periodical issues to improve the precision of Air pollutants prediction.

3 This method of prediction is also suitable in the other fields of Environmental sector i.e. underground water prediction, surface water quality prediction.

6.2 FUTURE WORK

The model designed in this research is used on the data of air pollution and it can be used on other datasets by doing little amendments. The interested researchers may also use the Fuzzy logic for prediction of air pollution, and hybrid mechanism with NN.

REFRENCES

[1]. Yi and V. R. Prybutok, "A Neural Network model forecasting for Prediction of daily maximum Ozone concentration in an industrialized urban area", Journal of Environmental Pollution, Vol. 92, pp. 349-357, 1996.

[2]. K. P. Melas, and I. Kioutsioukis, "Development and Assesment of Neural Network and Multiple Regression Models in Order to Predict PM_{10} levels in a Medium sized Mediterranean City", Water, Air and Soil Pollution, Vol. 182, pp. 325-334, 2007.

[3]. National Ambient Air Quality Standards (NAAQS), available online http://www.epa.gov/air/criteria.html [accessed on 20 Nov 2011]

[4]. J. V. Hansen, J.V. McDonald, and R. D. Nelson, "Time Series Prediction with Genetic-Algorithm designed Neural Networks: an Empirical comparison with modern Statistical models". Journal of Computational Intelligence, Volume 15, Number 3, 1999.

[5]. W. J. Wang, C. Men, and W. Z. Lu, "Online Prediction model based on Support Vector Machine". Elseiver Journel of Neurocomputing, Vol. 71, pp. 550–558, 2008.

[6]. W. Z. Lu, and W. J. Wang, "Potential assessment of the support vector Machine method in forecasting ambient air pollutant trends". Elseiver, journel of Chemosphere 59 (5), pp 693–701, 2005.

[7]. W. Z. Lu, H. Y. Fan, A.Y. T. Leung and J. C. K.wong, "analysis of pollutant levels in central hong kong applying neural networkmethod with particle swarm optimization", Environmental Monitoring and Assessment 79 pp 217–230, 2002.

[8]. J. L. Repace," Respirable Particles and Carcinogens in the Air of Delaware hospitalityvenues Before and After a Smoking Ban", Journal of Occupational and Environmental Medicine, Vol. 46, pp 887-905,2004.

[9]. World Health Organization (WHO) on air pollution concentration levels. Available online at http://en.wikipedia.org/wiki/Air_pollution [Accessed on 20 NOV 2011]

[10]. Air quality standards. Available online http://en.wikipedia.org/wiki/National_Ambient_Air_Quality_Standards [Accessed on 20 NOV 2011]

[11]. Air Quality Data, Environmental protection department, Available online http://epic.epd.gov.hk/ca/uid/airdata/p/1.htm [Accessed on 20 march 2011]

[12]. W. Z. Lu, W. J. Wang, H. Y. Fan, A. Y. T. Leung, Z. B. Xu, and S. M. Lo, "Air pollutant parameter forecasting using support vector machines", in IJCNN, vol. 1–3, IEEE, pp 630–635, 2002

[13]. Yi. J and V. R. prybutok, "A neural network model forecasting for prediction of daily maximum ozone concentration in an industrialized urban area". Journal of Environmental pollution, vol 92. pp 349-357. 1996

[14]. M. W. Garder, And S. R. Dorling, "Artificial Neural Network, The Multilayer Preceptron A Review To Applications In Atmospheric Sciences", Journel On Atmospheric Environment. Vol 32 pp 2627-2636. 1998

[15]. M. W. Garder, and S.R. Dorling, 'Neural network modeling and prediction of of hourly NO2 concentration in urban air in London .'Atmospheric Environment . vol 32. pp 709-719. 1998

[16]. J. J. Heymans, and A.A Barid.,'Carbon flow model and network analysis of the northern bergoela up welling system, Namibia. Journal of Ecological Modeling. Vol 126. pp 9-32. 2000

[17]. Melas, D. I. Kioutsoukis and I. Ziomas, "Neural network model for predicting peak photochemical pollutants level". Journal of air and waste management Association. Vol 50. Pp 495-501. 2000

[18]. T. Silini, A. Kaprara, K. Karatzas and M. Moussiopoulos," $PM_{2.5}$ Forecasting for Thesaloniki, Greecee'. Environmental Modeling and software. Vol 21. pp 559-565. 2006

[19]. A. F. Atiya, "Bankruptcy Prediction for credit risk using Neural network". IEEE Transactions on Neural Network. Vol 12(4). pp 929-935

[20]. B. Egeli, M. Ozturan and B. Badur, 'Stock market Prediction Ising Artificial Neural Networks'. In proceedings of 3rd Hawaii International Conference n Business, Hawaii, USA. pp 1-8. 2003

[21]. S. Hykins, "Neural Networks. A Comprehensive Foundation", Macmillan College Publishing, New York, 1994.

[22]. K. Nygren,"Stock prediction – A Neural Network Approach" Royal institute of technology, KTH. 2004

[23]. R. Grothmann,"Multi-Agent Market Modeling Based On Neural Networks", PhD. Thesis, Faculty Of Economics, University Of Bremen, Germany, 2002.

[24]. J. R. Rabuñal and J. Dorado. "Artificial neural Network in real Life Applications". Idea group publisher, London 2006

[25]. P. S churchland and T.J Sejnowski. "The Computational Brain". MIT press. cambridge, pp.130-148, 1994

[26]. A. I. Hanna, and D. P. Mandic,"An improved approach for nonlinear system identification using neural networks" Journal of the Franklin Institute, London, 2003.

[27]. W. L. Bundine and A. S. Wiegend, "Computing Second Derivatives in Feed-Forward Networks: A Review", IEEE Trans on Neural Networks, 5(3), pp 1-9. KTH Stockholm, Sweden 1997

[28]. P. Pivoňka and V. Veleba," Adaptive Controllers by Using Neural Network Based Identification for Short Sampling Period ". International journal of circuits, systems and signal processing. Issue 1, Vol. 1, 2007.

[29]. K. Chakraborty, C. K. Mohan and S. Ranka, "Forecasting the behaviour of multivariate time series using neural networks", Neural Net-works, vol 2, pp 53-58. 1992

[30]. J. Principe, And J. M. Kuo, "Dynamic Modeling Of Chaotic Time Series With Neural Networks", Proc. Of Neural Infor. Proc. Sys., 7. 1995.

[31]. M. Boznar, M. Lesjak, and P. Mlakar, "A neural network-based method for short term prediction of ambient SO_2 concentrations in highly polluted in-dustrial areas of complex terrain", Atmos. Environ., 27B, pp 221-230. 1993

[32]. F. Giorgio, and M. Piero, "Mathematical models for planning and control-ling air quality", Proceedings of IIASA Workshop, 17. 1996

[33]. J. R. Rabuñal and J. Dorado. "Artificial neural Network in real Life Applications". Idea group publisher, London 2006

[34]. S. Sumathi, T. Hamsapriya and P. Surekha. "Evolutionary Intelligence an Introduction to Theory and Applications with Matlab". Springer 2006

[35]. K. E. Parsopoulos, and M. N. Vrahatis, "Particle Swarm Optimization and Intelligence: Advances and Applications". By Information Science Reference, December 2009.

[36]. J. Kennedy and R. C. Eberhart, "Particle swarm optimization". In Proceedings of the IEEE International Conference on Neural Networks, Vol. IV, pp. 1942-1948. Piscataway, NJ: IEEE Service Center. Perth, Australia, 1995.

[37]. M. R. AlRashidi, and, M. E. El-Hawary, "A survey of particle swarm optimization applications in electric power systems". IEEE Transactions on Evolutionary Computation, Vol. 13(4), pp 913-918. 2009.

[38]. G. Beni and J. Wang. "Swarm intelligence in cellular robotic systems". Computer and System Science Vol. 102 (pp. 703–712). IGI Global Publisher 1989.

[39]. J. Branke, "Evolutionary optimization in dynamic environments". Kluwer Academic Publishers, Netherlands 2002

[40]. D. V. Arnold, "Noisy optimization with evolution strategies". Springer Berlin 2002.

[41]. M. M. Millonas, "Swarms, phase transitions, and collective intelligence". In C.G. Langton (Ed.), Artificial Life III (pp. 417-445). Reading, MA: Addison-Welsey. 1994.

[42]. M. S. Voss and X. Feng," Emergent System Identification using Particle Swarm Optimization", Proceedings Complex Adaptive Structures Conference, SPIE, Bellingham, 2001 [35] M.S. Voss, and X. Feng," ARMA model selection by using Particle swarm Optimization and AIC criteria", 15th Triennial World Congress, Barcelona, Spain, 2002.

[43]. M. R. AlRashidi and K.M. EL-Naggar, "Long term electric load forecasting based on particle swarm optimization", Applied Energy 87 pp 320–326 Elsevier Ltd, 2010.

[44]. R. Eberhart, Y. Shi, and J. Kennedy, "Swarm Intelligence".Morgan Kaufmann, San Francisco, 2001.

[45]. Y. D. Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley," Particle Swarm Optimization: Basic Concepts, Variants and Applications in

Power Systems", IEEE transactions on evolutionary computation, vol. 12, no. 2, April 2008

[46]. Y. D. Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and Ronald G. Harley," Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", IEEE transactions on evolutionary computation, vol. 12, no. 2, April 2008

[47]. Y. Shi, "Feature article on particle swarm optimization," IEEE Neural Network Society, Feature Article, pp. 8–13, Feb. 2004.

[48]. E. F. Campana, G. Fasano, and A. Pinto, "Dynamic system analysis and initial particles position in particle swarm optimization," in Proc. IEEE Swarm Intell. Symp. pp. 202–209. May 2006