

# Estimation of Solar Radiation at a Particular Place: Comparative study between Soft Computing and Statistical Approach

Bimal Datta<sup>1</sup>, Srimanta Pal<sup>2</sup>, Rohit Roy Chowdhury<sup>3</sup>, Swagata Sarkar<sup>3</sup>, Abhiroop Datta<sup>3</sup>

<sup>2z</sup>Electronics & Communication Sciences Unit  
Indian Statistical Institute, 203 B.T. Road,  
Kolkata 700108  
E-mail: [srimanta@isical.ac.in](mailto:srimanta@isical.ac.in)

<sup>1</sup>Department of Computer Science and Engineering  
Hooghly Engineering and Technology College  
E-mail: [bkd\\_hetc@iyahoo.co.in](mailto:bkd_hetc@iyahoo.co.in)

<sup>3</sup>AKChoudhary School of IT  
University College of Science & Technology, Kolkata  
E-mail: [rohit.roy6288@gmail.com](mailto:rohit.roy6288@gmail.com); Email: [abhiroopdatta7@gmail.com](mailto:abhiroopdatta7@gmail.com)  
E-mail: [swagata.sarkar88@gmail.com](mailto:swagata.sarkar88@gmail.com)

**Abstract**—This study focuses on the development of connectionist model such as neural network based method to efficiently predict solar radiation of a particular place. Here a comparative study is given between a conventional approach, that is Multiple Linear Regression (MLR) with the soft computing approach, that is Multi Layer Perceptron Neural Network (MLP). In each approach the input parameters are taken as several meteorological parameters that influences solar irradiance. The estimated values of solar radiation are compared with measured values in terms of percentage error and Root Mean Square Error (RMSE). To improve prediction accuracy Feature Selection technique is used which can select good features on-line while learning the prediction task. The paper also gives a comparative study of results of using both MLP and FSMLP. (*Abstract*)

**Keywords:** multi layer perceptron neural network, feature selection, meteorological parameter, multiple linear regression.

## I. INTRODUCTION

Estimation of solar radiation is very useful for various meteorological applications such as predicting weather pattern and water cycles, generating electricity etc. In order to design a solar energy system and to study its thermal performance evaluations, information on the availability of solar radiation is required. Solar radiation also has a direct effect on crop production [1]. This parameter is frequently required in orientation and design of greenhouses, heating, cooling and crop drying systems. Solar radiation is an important parameter necessary for most ecological models and serves as input for different photovoltaic conversion system; hence, it is of economic importance to renewable energy.

Several models have been proposed to estimate global solar radiation. Many Researchers use Statistical approaches [2], Empirical modelling [3], Fuzzy approaches [4-5], Geostatistical Technique [6] etc to predict solar radiation. Here we propose a data model that employs artificial intelligence techniques that mimic the learning process of human brain. This artificial neural network can learn the relationship between input parameters and the output variables by studying previously recorded data. They are fault tolerant in the sense that they are able to handle noisy and incomplete data and once trained, can perform prediction and generalisation at high speed [7]. This makes artificial neural network ideal for modelling non-linear, dynamic and complex systems such as for the prediction of meteorological data such as solar radiation [8] and other Atmospheric Science related fields [9].

In this paper, we use meteorological parameters like Air temperature (°C), Relative humidity (%), Rainfall (mm/h) and solar radiation (W/m<sup>2</sup>) of two consecutive days to predict the solar radiation of the 3<sup>rd</sup> day. The

paper is organized as follows: After brief introduction on the importance of the domain and technique applied, section II describes the methodology of MLR and MLP, section III presents data preparation, section IV gives results and discussions, section V presents Feature Selection technique and shows selected and rejected parameters, section VI and VII gives data preparation and results and discussions after feature selection, and section VIII concludes the study.

## II. METHODOLOGIES

### A. Multiple linear Regression:

Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Regression analysis measures the degree of influence of the independent variables on a dependent variable. In the case of a single independent variable, the dependent variable could be predicted from the independent variable by the simple equation:

$$y = ax + c \quad (1)$$

where  $c$  is a constant

This could be extended for multiple variables as follows:

$$y = a_1x_1 + a_2x_2 + \dots + a_nx_n + c \quad (2)$$

where  $a_1, a_2, \dots, a_n$  are coefficients and are calculated using past data. The value of  $y$  is predicted using the equation 2. To approximate a regression equation for a single variable, the relationship between the variables is to be plotted in the scatter diagram. Next, identify the straight line that represents the trend through the mid-point of the data, a trend which has the 'best fit' is identified. The trend identifies the relationship between the independent and dependent variables. The relationship, thus identified, is used to predict the various values of the dependent variable, given specific values of the independent variable. This predicted relationship is always linear. This concept is utilized to develop an accurate mathematical formulation of the regression analysis. The regression line is defined as a line for which the sum of squares of deviation of the various data points from the line is the least. This is also referred to as the least squares line. In case of a multi-variable problem, the regression equation is arrived at in a sequence of multiple linear regression equations, in a stepwise manner. At each step of the sequence, one variable is added to the regression equation. The variable added is the one that makes the greatest reduction in the error sum of squares of the sample data.

### B. Multilayer Perceptron Model

The architecture of an MLP is a feed forward neural network in which the nonlinear elements are arranged in successive layers. The characteristics of an MLP are as follows:

- i. Has any number of inputs.
- ii. Has any number of hidden layers with any number of nodes.
- iii. Uses linear combination function in the hidden and output layers.
- iv. Uses generally sigmoid activation number in the hidden layer(s).
- v. Has any number of outputs with activation functions.
- vi. There are complete connections between the nodes in successive layers but there is no connection within the layer. A typical MLP is shown in Figure 1

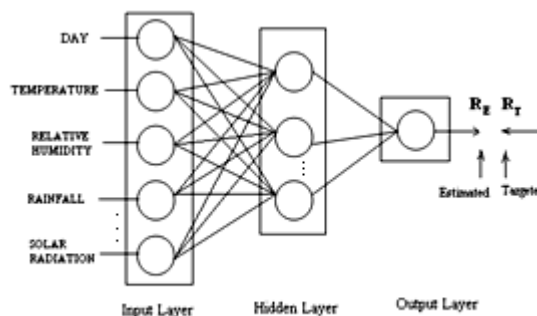


Figure 1 An example of MLP for Solar radiation prediction

The objective of the MLP learning is to set the connection weights such that the error between the network output and the target output is minimized. Back propagation algorithm is an involved mathematical tool which has been widely used as a learning algorithm in feed forward multilayer neural networks. In our research study

we use this algorithm to update the weights of our neural based System. Following is a brief description about the algorithm that is being used in our neural network based model to calculate the output and to update the weights of each layer of the system.

**Algorithm:**

Input and output are described below.

$n_i$  = no. of neurons in input layer

$n_h$  = no. of neurons in the hidden layer

$n_o$  = no. of neurons in the output layer

$h$  = A vector ( $h_1, h_2, \dots, h_{n_h}$ ) of hidden layer

$i$  = A vector ( $i_1, i_2, \dots, i_{n_i}$ ) of input layer

$o$  = A vector ( $o_1, o_2, \dots, o_{n_o}$ ) of output layer

$w^{ih}$  = A weight matrix [ $w_{u,v}^{ih}$ ]  $n_i \times n_h$  for input to hidden layer

$w^{ho}$  = A weight matrix [ $w_{u,v}^{ho}$ ]  $n_h \times n_o$  for hidden to output layer

$F(x) = 1/(1+e^{-x})$ , an activation function

In forward pass, the following steps are executed.

**STEP 1:**

Compute the hidden layer neuron activation as follows:

$$h = F(iw^{ih}) \quad \text{where } h_k = F(a_k) = 1/(1+e^{-a_k}) \quad \text{and } a_k = \sum_{l=1}^{n_i} w_{lk}^{ih} i_l \quad \text{for } k = 1, 2, \dots, n_h$$

**STEP 2:**

Compute the output layer neuron activation as follows:

$$o = F(hw^{ho}) \quad \text{where } o_k = F(b_k) = 1/(1+e^{-b_k}) \quad \text{and } b_k = \sum_{l=1}^{n_h} w_{lk}^{ho} h_l \quad \text{for } k = 1, 2, \dots, n_o$$

In backward pass, the following steps are executed:

**STEP 3:**

Compute the output layer error as follows:

$$d_k = o_k (1 - o_k) (t_k - o_k) \quad \text{for } k = 1, 2, \dots, n_o$$

**STEP 4:**

Compute the hidden layer error as follows:

$$e_k = h_k (1 - h_k) \sum_{i=1}^{n_o} w_{ik}^{ho} d_i \quad \text{for } k = 1, 2, \dots, n_h, \quad \text{where } e_k = \text{error in hidden layer neuron } k. \quad e = (e_1, e_2, \dots, e_{n_h})$$

**STEP5:**

Adjust the weights for  $w^{ho}$  of synapses:

$w^{ho}(t+1) = w^{ho}(t) + \Delta w^{ho}(t)$ , Where  $\Delta w^{ho}(t)$  is a matrix representing the change in  $\Delta w^{ho}$ .

$\Delta w^{ho}(t) = \alpha h d + \beta \Delta w^{ho}(t-1)$ , where  $\alpha$  = the learning rate, and,  $\beta$  = momentum factor for weight changes.

**STEP6:**

Adjust the weights for  $w^{ih}$  of synapses:

$w^{ih}(t+1) = w^{ih}(t) + \Delta w^{ih}(t)$ , Where  $\Delta w^{ih}(t)$  is a matrix representing the change in  $\Delta w^{ih}(t)$

$\Delta w^{ih}(t) = \alpha i e + \beta \Delta w^{ih}(t-1)$ , where  $\alpha$  = the learning rate,  $\beta$  = momentum factor for weight changes.

**STEP7:**

Repeat STEP1 to STEP6 for all input output pairs until error  $d$  is within limit.

**STEP8:**

Stop.

### III. DATA PREPARATION

In this paper, the meteorological parameters that have been used as input parameters are Air temperature ( $^{\circ}\text{C}$ ), Relative humidity (%), Rainfall (mm/h) and solar radiation ( $\text{W}/\text{m}^2$ ). We used these parameters of two

consecutive days to predict the solar radiation of the 3<sup>rd</sup> day. For a particular day t, we have observations of seven variables namely:

1. Day of the year (D (t))
2. Max temperature (M<sub>1</sub>T (t))
3. Min temperature (M<sub>2</sub>T (t))
4. Max Relative humidity (R<sub>1</sub>H (t))
5. Min Relative humidity (R<sub>2</sub>H (t))
6. Rainfall (RF (t))
7. Solar Radiation (SR (t))

The solar radiation for a t<sup>th</sup> day, SR (t) is determined as follows:

$$SR(t) = f(D(t-2), M_1T(t-2), M_2T(t-2), R_1H(t-2), R_2H(t-2), RF(t-2), SR(t-2), D(t-1), M_1T(t-1), M_2T(t-1), R_1H(t-1), R_2H(t-1), RF(t-1), SR(t-1))$$

Initially all the aforesaid parameters have been taken into consideration. The data that are used in the training and testing are acquired from Dumdum Meteorological Centre; Kolkata.

An important observation is found from the input data set, which shows that the solar radiation pattern over a year is repetitive and sinusoidal in nature. This pattern is depicted in Figure 2

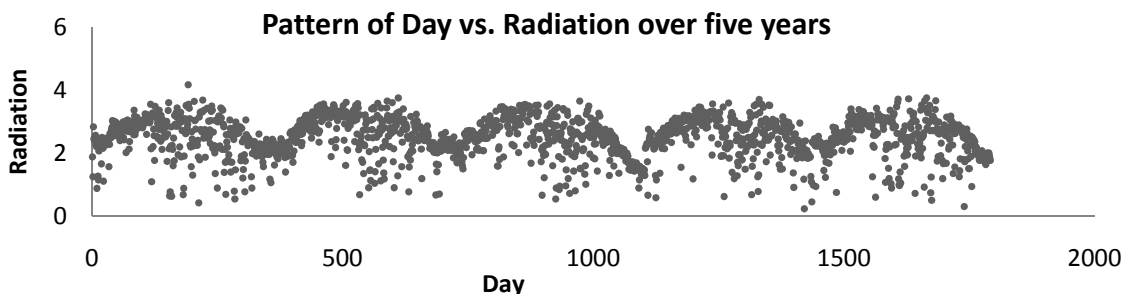


Figure 2 Graph showing the day vs. radiation pattern

The input file with all the initial parameters is shown in the following **Table 1**.

Table 1: Training input file

| D (t-2)  | M <sub>1</sub> T (t-2) | M <sub>2</sub> T (t-2) | R <sub>1</sub> H (t-2) | R <sub>2</sub> H (t-2) | RF (t-2) | SR (t-2) | D (t-1)  | M <sub>1</sub> T (t-1) | M <sub>2</sub> T (t-1) | R <sub>1</sub> H (t-1) | R <sub>2</sub> H (t-1) | RF (t-1) | SR (t-1) | SR (t) |
|----------|------------------------|------------------------|------------------------|------------------------|----------|----------|----------|------------------------|------------------------|------------------------|------------------------|----------|----------|--------|
| 01011991 | 27                     | 13                     | 96                     | 86                     | 0        | 1.36     | 02011991 | 27                     | 18                     | 94                     | 74                     | 16.3     | 1.05     | 1.35   |
| 02011991 | 27                     | 18                     | 94                     | 74                     | 16.3     | 1.05     | 03011991 | 23.0                   | 15.6                   | 96                     | 83                     | 14.4     | 1.35     | 1.05   |
| .        | .                      | .                      | .                      | .                      | .        | .        | .        | .                      | .                      | .                      | .                      | .        | .        | .      |
| 30121995 | 27.2                   | 24                     | 86                     | 78                     | 11.3     | 1.09     | 31121995 | 25.4                   | 20.7                   | 92                     | 79                     | 12.4     | 1.21     | 1.11   |

#### IV. RESULT AND DISCUSSION

In this study, at first the conventional approach is used to estimate SR(t), then soft computing approach is used to estimate SR(t).

##### A. Conventional approach for estimation of solar radiation using Multiple Linear Regression (MLR)

MLR is a statistical method that can be used to estimate a particular variable with inputs as a set of variables. The result of MLR is computed using eq. (2). Initially 80% data is used to estimate the regression coefficients and the rest 20% data is used for the estimation of SR(t). The obtained result is shown in following **Table 2**. And the graph corresponding to target radiation and computed radiation is shown in Figure 3.

Table 2: Error Table

| Target Radiation | Computed Radiation | Error    |
|------------------|--------------------|----------|
| 0.547847         | 0.481988           | 0.065859 |
| 0.471292         | 0.487124           | -0.01583 |
| 0.54067          | 0.480798           | 0.059872 |
| 0.583732         | 0.485456           | 0.098276 |
| 0.576555         | 0.483382           | 0.093173 |
| 0.519139         | 0.47751            | 0.041629 |
| 0.535885         | 0.476541           | 0.059344 |
| 0.4689           | 0.484329           | -0.01543 |
| 0.507177         | 0.475716           | 0.031461 |
| 0.471292         | 0.484769           | -0.01348 |
| 0.279904         | 0.487221           | -0.20732 |
| 0.483254         | 0.495085           | -0.01183 |
| 0.770335         | 0.55356            | 0.216775 |
| 0.696172         | 0.556039           | 0.140133 |
| 0.80622          | 0.538469           | 0.267751 |
| 0.782297         | 0.554853           | 0.227444 |
| 0.787081         | 0.558205           | 0.228876 |
| 0.748804         | 0.543057           | 0.205747 |
| 0.452153         | 0.567378           | -0.11522 |
| 0.566986         | 0.569764           | -0.00278 |
| 0.803828         | 0.577737           | 0.22609  |
| 0.80622          | 0.541066           | 0.265154 |
| 0.782297         | 0.533863           | 0.248434 |
| 0.389952         | 0.528421           | -0.13847 |
| 0.54067          | 0.548841           | -0.00817 |
| 0.356459         | 0.498106           | -0.14165 |
| 0.645933         | 0.499235           | 0.146698 |
| 0.645933         | 0.516399           | 0.129534 |
| 0.444976         | 0.512887           | -0.06791 |
| 0.358852         | 0.509421           | -0.15057 |

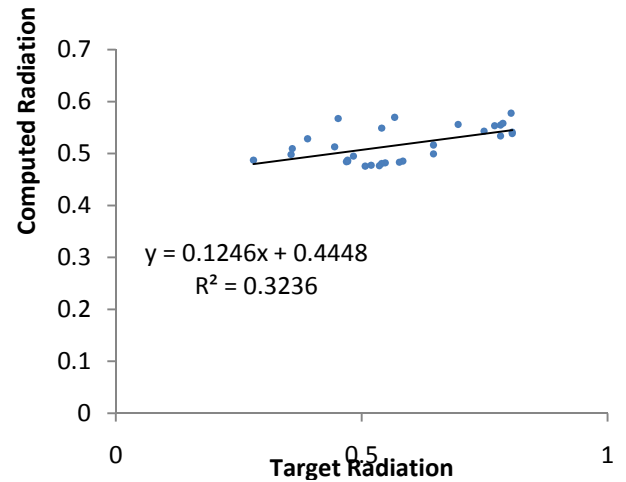


Figure 3 Graphical representation of Target vs. Computed Radiation

Above **Table 2** indicates that the average error is 21%. The accuracy of the system is 79% (approx.), which is low.

*B. Soft Computing approach for estimation of solar radiation using Artificial Neural Network (MLP)*

The initial training is done with all the parameters of the input file. We have made several runs of the MLP neural network with different hidden nodes ( $n_h$ ). The following table (**Table 3**) reports the average performance, maximum & standard deviation on the test data for  $n_h = 2, 4, 6, 8, 10$ . Table 2 shows the cumulative percentage of prediction within different ranges. For example, the column with  $n_h = 10$  shows that on the test data the network could make prediction with  $\leq \pm 0.02$  error in only 6.67% cases. It is interesting to note that the networks with  $n_h = 8, 10$  perform reasonably well but the performance degrades with increase in the number of hidden nodes beyond 20.

Table 3: Cumulative frequency of error for different no. of nodes in hidden layer

| Error Range (W/m <sup>2</sup> )   | % frequency of test data |                       |                       |                       |                        |
|-----------------------------------|--------------------------|-----------------------|-----------------------|-----------------------|------------------------|
|                                   | For n <sub>h</sub> =2    | For n <sub>h</sub> =4 | For n <sub>h</sub> =6 | For n <sub>h</sub> =8 | For n <sub>h</sub> =10 |
| ± 0.02                            | 15.4                     | 13.4                  | 6.7                   | 6.6                   | 6.67                   |
| ± 0.04                            | 26.7                     | 20                    | 16.7                  | 20.7                  | 10                     |
| ± 0.06                            | 43.4                     | 36.6                  | 23.4                  | 40.4                  | 26.8                   |
| ± 0.08                            | 60                       | 43.4                  | 26.7                  | 56.7                  | 40                     |
| ± 0.1                             | 73.3                     | 50                    | 36.6                  | 66.7                  | 46.8                   |
| ± 0.3                             | 90.6                     | 86.6                  | 83.3                  | 89.6                  | 87                     |
| <b>Average Standard Deviation</b> | 0.024                    | 0.027                 | 0.022                 | 0.029                 | 0.027                  |
| <b>Maximum Standard Deviation</b> | 0.073                    | 0.086                 | 0.066                 | 0.0843                | 0.075                  |

From the above mentioned table it is visible that for n<sub>h</sub> =2 the MLP system gives better result in comparison with other MLP having no. of neuron in hidden layer as 4, 6, 8, 10. Having Average Standard Deviation as 0.024 and Maximum Deviation as 0.073 we take n<sub>h</sub> =2 as our final no. of neuron in hidden layer. After the training, testing is done with rest 20% of data. The table containing the error is shown in table 3. And the graph regarding the solar radiation (target and computed) is shown in the following Figure 4.

Table 4: Error table

| Target Radiation | Computed Radiation | Error    |
|------------------|--------------------|----------|
| 0.54784689       | 0.541465365        | 0.006382 |
| 0.471291866      | 0.554963328        | 0.083671 |
| 0.540669856      | 0.507954112        | 0.032716 |
| 0.583732057      | 0.508582589        | 0.075149 |
| 0.576555024      | 0.531572338        | 0.044983 |
| 0.519138756      | 0.532098237        | 0.012959 |
| 0.535885167      | 0.493352609        | 0.042533 |
| 0.468899522      | 0.495989482        | 0.02709  |
| 0.507177033      | 0.497001888        | 0.010175 |
| 0.471291866      | 0.483440123        | 0.012148 |
| 0.279904306      | 0.48892539         | 0.209021 |
| 0.483253589      | 0.439352663        | 0.043901 |
| 0.770334928      | 0.573970156        | 0.196365 |
| 0.696172249      | 0.662097191        | 0.034075 |
| 0.806220096      | 0.667082979        | 0.139137 |
| 0.782296651      | 0.675898979        | 0.106398 |
| 0.78708134       | 0.674537014        | 0.112544 |
| 0.748803828      | 0.672964087        | 0.07584  |
| 0.45215311       | 0.772762913        | 0.32061  |
| 0.566985646      | 0.609339864        | 0.042354 |
| 0.803827751      | 0.619238094        | 0.18459  |
| 0.806220096      | 0.674136118        | 0.132084 |
| 0.782296651      | 0.67693669         | 0.10536  |
| 0.389952153      | 0.547043823        | 0.157092 |
| 0.540669856      | 0.510705158        | 0.029965 |
| 0.35645933       | 0.523883779        | 0.167424 |
| 0.645933014      | 0.355534206        | 0.290399 |
| 0.645933014      | 0.602755818        | 0.043177 |
| 0.444976077      | 0.597668843        | 0.152693 |
| 0.358851675      | 0.531274874        | 0.172423 |
| 0.54784689       | 0.541465365        | 0.006382 |

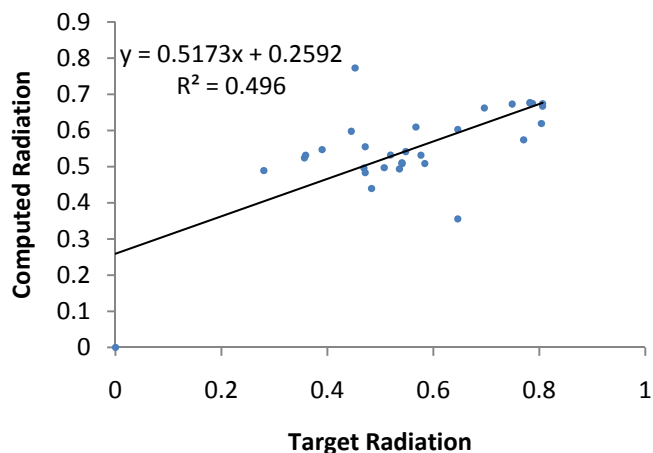


Figure 4 Graphical representation of target & computed radiation

The result indicates that the average error (Table 4) is 10.11%. The accuracy of the system is 89.9% (approx.) which is moderate.

So, **Table 2** and **Table 4** signifies that soft computing approach is advantageous over statistical method as the accuracy of the system has increased from 79% that is achieved in MLR (**Table 2**) to 89.9% that has been achieved using MLP (**Table 4**). The RMSE value in case of MLP is 0.12 whereas the RMSE for MLR is 0.14.

V. ONLINE FEATURE SELECTION TECHNIQUE

To increase the accuracy of the system we need to eliminate some of the input parameters that do not contribute much for the estimation. An online feature selection network identifies a subset of relevant, non-redundant parameters by estimating the relative contribution of the features to the output neurons [14]. Feature selection can be implemented by associating an adaptive gate to each input node. The gate should be modeled in such a manner that a good feature can pass completely but a bad feature cannot. A simple way of identifying useful gate functions is to use s-type (or sigmoid) functions with a tunable parameter which can be learnt using training data. Let  $F: R \rightarrow (0,1)$  be an attenuation function associated with each of the  $p$  input nodes. If  $x$  is the node input then  $x F(\gamma)$  is the node output. Thus,  $x F(\gamma_i)$  can be viewed as the activation function of the  $i$ -th node in input layer, where  $\gamma_i$  is a parameter (not a connection weight) of the activation function. Thus, the input layer nodes act as “neurons” (i.e., have internal calculations). Notice that  $F(\gamma_i)$  acts as a fixed multiplier for all input values of the  $i$ -th feature once  $\gamma_i$  is known. The function  $F$  can have various forms. In the experiments described below we use the attenuation function

$$F(\gamma) = \frac{1}{1 + e^{-\gamma}} \tag{3}$$

Thus, the  $i$ -th input node attenuates  $x_i$  by an amount  $F(\gamma_i) \in (0,1)$ , where the input “weight”  $\gamma_i$  is a parameter to be learnt during training. If  $F(\gamma_i)$  is close to 0, we may choose to eliminate input feature  $x_i$ : this is how feature selection technique works. Let

- Q = number of nodes in the first hidden (not input) layer;
- $\mu$  = learning rate for the parameters of the attenuator membership functions;
- $\eta$  = learning rate for the connection weights;
- $w_{ij}^{ih}(t)$  = weight connecting  $i$ -th node of the input layer to the  $j$ -th node of the first hidden layer for the  $t$ -th iteration; and
- $\delta_j^1$  = error term for the  $j$ -th node of the first hidden layer.
- $F'(\gamma_i)$  = derivative of  $F$  at  $\gamma_i$ ;

It can be easily shown that the learning rule for connection weights remains the same for all layers except for  $w_{ij}^{ih}(t)$ . The update rule for  $w_{ij}^{ih}(t)$  and  $\gamma_i$  are:

$$w_{ij}^{ih}(t) = w_{ij}^{ih}(t - 1) - \eta x_i \delta_j^1 F(\gamma_i(t - 1)) \tag{4}$$

$$\gamma_i(t) = \gamma_i(t - 1) + \mu x_i \left( \sum_{j=1}^q w_{ij}^{ho} \delta_j^1 F'(\gamma_i(t - 1)) \right) \tag{5}$$

We have taken all the input parameters as the input set for feature selection. Result of Feature selection Technique is shown in **Table 5**.

Table 5: Showing the selected and rejected feature list

| Input Parameters            | Exp1        | Exp2        | Exp3        | Exp4        | Frequency | Decision        |
|-----------------------------|-------------|-------------|-------------|-------------|-----------|-----------------|
| D(t- 2)                     | 0.27        | 0.22        | 0.32        | 0.31        | 0         | Rejected        |
| M <sub>2</sub> T (t-2)      | 0.35        | 0.35        | 0.37        | 0.36        | 0         | Rejected        |
| <b>M<sub>1</sub>T(t-2)</b>  | <b>0.60</b> | <b>0.71</b> | <b>0.74</b> | <b>0.7</b>  | <b>4</b>  | <b>Selected</b> |
| <b>R<sub>1</sub> H(t-2)</b> | <b>0.76</b> | <b>0.68</b> | <b>0.60</b> | <b>0.47</b> | <b>3</b>  | <b>Selected</b> |
| R <sub>2</sub> H (t-2)      | 0.49        | 0.48        | 0.40        | 0.47        | 0         | Rejected        |
| RF(t-2)                     | 0.46        | 0.48        | 0.30        | 0.4         | 0         | Rejected        |
| <b>SR(t-2)</b>              | <b>0.66</b> | <b>0.79</b> | <b>0.70</b> | <b>0.47</b> | <b>3</b>  | <b>Selected</b> |
| D(t- 1)                     | 0.37        | 0.48        | 0.51        | 0.40        | 1         | Rejected        |
| M <sub>2</sub> T(t-1)       | 0.4         | 0.43        | 0.32        | 0.36        | 0         | Rejected        |
| <b>M<sub>1</sub>T(t-1)</b>  | <b>0.77</b> | <b>0.88</b> | <b>0.57</b> | <b>0.86</b> | <b>4</b>  | <b>Selected</b> |
| <b>R<sub>1</sub> H(t-2)</b> | <b>0.76</b> | <b>0.68</b> | <b>0.60</b> | <b>0.47</b> | <b>3</b>  | <b>Selected</b> |
| <b>R<sub>2</sub> H(t-2)</b> | <b>0.69</b> | <b>0.58</b> | <b>0.50</b> | <b>0.77</b> | <b>4</b>  | <b>Selected</b> |
| <b>RF(t-1)</b>              | <b>0.76</b> | <b>0.60</b> | <b>0.70</b> | <b>0.50</b> | <b>4</b>  | <b>Selected</b> |
| <b>SR(t-1)</b>              | <b>0.76</b> | <b>0.66</b> | <b>0.80</b> | <b>0.47</b> | <b>3</b>  | <b>Selected</b> |

By Feature Selection technique, six Input Parameters (**D(t-2)**, **M<sub>2</sub>T(t-2)**, **Min R<sub>2</sub>H(t- 2)**, **RF(t-2)**, **D(t-1)**, **M<sub>2</sub>T(t-1)**) are found to be irrelevant on the basis of voting scheme and therefore they are rejected in the training process.

After Feature Selection method, the radiation of t<sup>th</sup> day is given as:

$$SR(t) = f(M_1T(t-2), R_1H(t-2), SR(t-2), M_1T(t-1), R_1H(t-1), R_2H(t-1), RF(t-1), SR(t-1))$$

## VI. DATA PREPARATION AFTER FEATURE SELECTION

After eliminating the six irrelevant features the input file is also modified. Modified input file is shown in the following **Table 6**:

Table 6: Showing the modified input file

| M <sub>1</sub> T<br>(t-2) | R <sub>1</sub> H<br>(t-2) | SR<br>(t-2) | M <sub>1</sub> T<br>(t-1) | R <sub>1</sub> H<br>(t-1) | R <sub>2</sub> H<br>(t-1) | RF<br>(t-1) | SR<br>(t-1) | SR<br>(t) |
|---------------------------|---------------------------|-------------|---------------------------|---------------------------|---------------------------|-------------|-------------|-----------|
| 21                        | 96                        | 1.36        | 21.4                      | 82                        | 21.7                      | 16.3        | 1.50        | 1.35      |
| 17.5                      | 82                        | 1.50        | 28.6                      | 90                        | 21.6                      | 14.4        | 1.35        | 1.05      |
| .                         | .                         | .           | .                         | .                         | .                         | .           | .           | .         |
| .                         | .                         | .           | .                         | .                         | .                         | .           | .           | .         |
| 18.5                      | 86                        | 1.30        | 23.5                      | 81                        | 24.5                      | 11.4        | 1.34        | 1.11      |

## VII. RESULTS AND DISCUSSION AFTER FEATURE SELECTION

After rejecting the irrelevant parameters and altering the Input file, the final training is done. Here also we have made several runs of the MLP net with different hidden nodes ( $n_h$ ). The following table (**Table 7**) reports the average performance, maximum & standard deviation on the test data for  $n_h=2, 4, 6, 8, 10$ . **Table 7** shows the cumulative percentage of prediction within different ranges. For example, the column with  $n_h=10$  shows that on the test data the network could make prediction with  $\leq \pm 0.02$  error in only 16.67% cases. It is interesting to note that the performance of the networks with  $n_h=8, 10$  is also good.



**Table 7:** Showing the cumulative frequency of error after feature selection

| Error Range (W/m <sup>2</sup> )   | % frequency of test data |                       |                       |                       |                        |
|-----------------------------------|--------------------------|-----------------------|-----------------------|-----------------------|------------------------|
|                                   | For n <sub>h</sub> =2    | For n <sub>h</sub> =4 | For n <sub>h</sub> =6 | For n <sub>h</sub> =8 | For n <sub>h</sub> =10 |
| ± 0.02                            | 36.7                     | 23.4                  | 20                    | 20                    | 16.67                  |
| ± 0.04                            | 63.4                     | 53.4                  | 40                    | 36.6                  | 33.33                  |
| ± 0.06                            | 68.7                     | 70                    | 56.7                  | 43.3                  | 60                     |
| ± 0.08                            | 76.4                     | 76.7                  | 66.7                  | 56.6                  | 66.67                  |
| ± 0.1                             | 83.4                     | 80                    | 73.4                  | 70.3                  | 73.33                  |
| ± 0.3                             | 100                      | 96.7                  | 93.4                  | 91.7                  | 93.67                  |
| <b>Average Standard Deviation</b> | 0.022                    | 0.023                 | 0.026                 | 0.025                 | 0.023                  |
| <b>Maximum Standard Deviation</b> | 0.059                    | 0.0644                | 0.068                 | 0.073                 | 0.069                  |

From the above **Table 7** it can be stated that for n<sub>h</sub>=2 the MLP system gives better result in comparison with other MLP having no. of neuron in hidden layer as 4, 6, 8, 10. Having Average Standard Deviation as 0.022 and Maximum Deviation as 0.059 we take n<sub>h</sub>=2 as our final no. of neuron in hidden layer for the neural network system.

After the training, testing is done with rest 20% of data. The table containing the error is shown in **Table 8** along with the graphical representation of target vs. computed radiation in Figure 5.

Table 8: Error table after Feature Selection

| Target Radiation | Computed Radiation | Error    |
|------------------|--------------------|----------|
| 0.54784689       | 0.542448056        | 0.005399 |
| 0.471291866      | 0.545658487        | 0.074367 |
| 0.540669856      | 0.518382181        | 0.022288 |
| 0.583732057      | 0.530856029        | 0.052876 |
| 0.576555024      | 0.546234126        | 0.030321 |
| 0.519138756      | 0.544649164        | 0.02551  |
| 0.535885167      | 0.532804026        | 0.003081 |
| 0.468899522      | 0.531456072        | 0.062557 |
| 0.507177033      | 0.516026108        | 0.008849 |
| 0.471291866      | 0.526991179        | 0.055699 |
| 0.279904306      | 0.318569278        | 0.038665 |
| 0.483253589      | 0.482178183        | 0.001075 |
| 0.770334928      | 0.789266358        | 0.018931 |
| 0.696172249      | 0.737886764        | 0.041715 |
| 0.806220096      | 0.794813923        | 0.011406 |
| 0.782296651      | 0.776143562        | 0.006153 |
| 0.78708134       | 0.761808776        | 0.025273 |
| 0.748803828      | 0.754552604        | 0.005749 |
| 0.45215311       | 0.526092109        | 0.073939 |
| 0.566985646      | 0.58996692         | 0.022981 |
| 0.806220096      | 0.782493418        | 0.023727 |
| 0.782296651      | 0.786106558        | 0.00381  |
| 0.540669856      | 0.533607769        | 0.007062 |
| 0.645933014      | 0.634751902        | 0.011181 |
| 0.358851675      | 0.354190443        | 0.004661 |
| 0.54784689       | 0.542448056        | 0.005399 |
| 0.471291866      | 0.545658487        | 0.074367 |
| 0.540669856      | 0.518382181        | 0.022288 |
| 0.583732057      | 0.530856029        | 0.052876 |
| 0.576555024      | 0.546234126        | 0.030321 |
| 0.519138756      | 0.544649164        | 0.02551  |

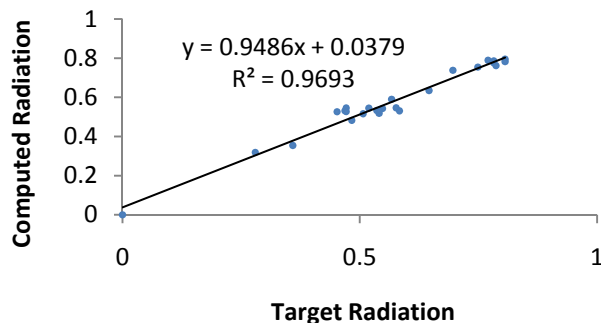


Figure 5 Graphical representation of target & computed radiation after feature selection

The result indicates that the performance (average error 7.14%) is very good. The above results (**Table 8**) indicate that the percentages of error are moderate. The accuracy (approximately 93%) of estimation of solar radiation is quite good. This proposed neural network system will function efficiently to predict solar radiation at a particular place.

## VIII. CONCLUSION

This paper depicts a neat data model for estimating solar radiation at a particular place. Here a comparative study between statistical approach and soft computing has been shown. Between Multiple Linear Regression (MLR) which has been chosen as the statistical method and Multi Layer Perceptron Neural Network (MLP) which is a soft computing technique, MLP based system proves to be advantageous and gives more accurate results than the MLR as, with MLR the RMSE value is 0.14 (with accuracy 79%) and with MLP the RMSE value decreases to 0.12 (with accuracy 89.9% approx (without feature selection)).

Also a comparative study is done between MLP without feature selection and MLP with feature selection. Comparing **Table 4** and **Table 8**, then it's clearly visible that in **Table 4** the efficiency level of the neural network system was low. The average error in the **Table 4** is 10.11%. So using feature selection some irrelevant features has been eliminated. By using significant features, the predicted result becomes quite good as shown in **Table 8**. The average error in **Table 8** is 7.14%. The RMSE value before feature selection is **0.12**, where as the RMSE value after feature selection is **0.033**.

So feature selection technique (FSMLP) can be used to increase the accuracy of the estimated result of neural network based prediction system.

## REFERENCES:

- [1] Abed El-Aziz A.M., S.G.A. Mouhamed and S.A.H. Ouda, 2006, 'Predicting the interaction between temperature, solar radiation and nitrogen fertilizer on maize yield in South Delta', *Journal of Applied Sciences Research*, Vol.2(11) pp. 926-930.
- [2] Wu Ji; Chan, C.; Loh, J.; Choo, F.; Chen, L.H., 'Solar radiation prediction using statistical approaches', *Information, Communications and Signal Processing*, 2009. Pp. 1 – 5
- [3] M.S. Okundamiya, A.N. Nzeako, 'Empirical Model for Estimating Global Solar Radiation on Horizontal Surfaces for Selected Cities in the Six Geopolitical Zones in Nigeria', *Journal of Applied Sciences, Engineering and Technology* 2(8): 805-812, 2010
- [4] Nalin K. Gautam, N. D. Kaushika, 'A Model for the Estimation of Global Solar Radiation Using Fuzzy Random Variables', *Journal of Applied Meteorology* Vol. 41 pp.1267-1276
- [5] R. Iqdour, A. Zeroual, 'A rule based fuzzy model for the prediction of solar radiation', *Revue des Energies Renouvelables*, Vol. 9 N°2 (2006) 113 – 120
- [6] Shafiqur Rehman, Saleem G. Ghori, 'Spatial estimation of global solar radiation using geostatistics', *Renewable Energy* 21 (2000) 583-605
- [7] Kalogirou, S.A., 'Artificial neural networks in renewable energy systems applications: a review', *Renewable and Sustainable Energy Reviews* Volume 5, Issue 4, December 2001, pp. 373-401
- [8] Mubiru, J., 'Predicting total solar irradiation values using artificial neural networks', *Renewable Energy*, Volume 33, Issue 10, pp. 2329-2332 (2008)
- [9] M.W Gardner, S.R Dorling, 'Artificial neural networks (the multilayer perceptron)—a review of applications in the
- [10] atmospheric sciences', *Atmospheric Environment*, Volume 32, Issues 14-15, 1 August 1998, pp. 2627-2636
- [11] Dennis W. Ruck, Steven K. Rogers, Matthew Kabrisky, 'Feature Selection Using a Multilayer Perceptron', *Journal of Neural Network Computing*, Volume 2, Number 2, 1990, pp. 40-48.
- [12] Verikas, M. Bacauskiene, 'Feature selection with neural networks', *Pattern Recognition Letters*, Vol 23, Issue 11, September
- [13] 2002, pp. 1323-1335
- [14] E. Gasca, J.S. Sánchez, R. Alonso, 'Eliminating redundancy and irrelevance using a new MLP-based feature selection method', *Pattern Recognition*, 39 (2006) 313 – 315
- [15] Rudy Setiono; Huan Li, 'Neural-network feature selector', *Neural Networks* Vol: 8 Issue:3 pp. 654 - 662
- [16] Ledesma, S., Cerda, G., Aviña, G., Hernández, D., and Torres, M., 'Feature Selection Using Artificial Neural Networks', *Advances in Artificial Intelligence*, 2008, Volume 5317, pp. 351-359