Journal of Scientific Research and Advances

Modeling of a Thermal Power Plant using Neural Network and Regression Technique

Rajarshi Dixit[1*], Jitendra Kumar[2], Tarun Soota[1]

Cite this article: J. Sci. Res. Adv. Vol. 2, No. 4, 2015, 166-174.

This paper works for modeling of a thermal power plant by using Neural Network and Regression technique. Neural Network can be improved to attain the desired accuracy level by training it on experimental data. Neural Network is an information processing paradigm made up of a set of algebraic equations. In this paper, Feed Forward Back Propagation Neural Network technique is used to train the data. Neural Network modeling described in this study are implemented in Matlab R2011a (Math Works, Inc.) and run under the Microsoft Windows 8 environment. Regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. Regression analysis is widely used for prediction and forecasting. Regression technique described in this paper is implemented in Minitab 14 and run under the Microsoft Windows 8 environment.

ISSN 2395-0226

Introduction

The construction of a new thermal power plant has been reduced relatively due to capital costs, time for installation, and availability of fuel etc. However, thermal power plants account for approximately 65% of the world's power supply. Recently, there have been concerns regarding the efficiency improvement of existing thermal power plants. The efficiency of such type of power plant is very low and great amount of loss in thermal energy may be noticed. In order to generate a required electric energy, the turbine needs an equivalent amount of thermal energy in addition to the loss. Steam is merely a vehicle which transports the energy from the burners to the turbine shaft. The flow of steam is not only the parameter that should be determined, but also the temperature and the pressure must be taken into consideration for a given value of power. The dynamic behavior of thermal plants heavily depends on disturbances and in particular on changes in operating point. This is particularly the case for large coal fired power plants. Such plants represent from the control engineering point of view a time-variant and nonlinear multivariable process with strong interactions. Therefore, they are very difficult to control. Nowadays, modeling of a thermal power plant plays a vital role for optimizing the plant. There are many types of modeling technique (i.e. neural network, fuzzy logic and regression etc.) which are used for optimizing the thermal power plant. An overview of papers which shows various results on different modeling and optimization techniques on thermal power plant. The purpose of this study is to provide background information on the issues to be considered in this paper and give importance to the relevance of the present study. Aurora C. et. Al [1] proposed the paper for verifying the applicability of industrial model predictive control (MPC) to thermal Power Plants for improving the plant efficiency in order to cope up with high levels of competition in market. Cerri G. et. Al [2] proposed the paper which deals with the problem of finding the optimum load allocation on machines and apparatuses in complex Cogeneration Heat and Power (CHP) plants. The methodology of this paper based on Neural Networks (NN) has been developed. Ilamathi P. et. Al [3] proposed the paper for predictive modeling of nitrogen oxides emission from a 210 MW coal fired thermal power plant with combustion parameter optimization. Manke P. and Tembhurne S [4] proposed the paper for the development and testing of a neural network based drum level controller for sub-critical thermal power plant boilers. In this paper, Experimental data obtained from an operational coal fired power plant (500MW Thermal Power Station, Korba, India) is used to train the neural network and The Artificial neural networks (ANN) modeling can significantly reduce the frequency of deviations and the degree of deviation of the water level in the drum. Panda S. et. Al [5] developed an approach to design a controller used for blow down optimization in the losses reduction process of a power plant boiler. In this paper, neural network technology offers a best method for designing a neuron control based on back propagation because the back propagation technique can prove to be a very effective tool for evaluating and maintain boiler efficiency and indirect losses. Zhou H. et. Al [6] develops an efficient NOx emissions model based on support vector regression (SVR), and compares its performance with traditional modeling techniques, i.e., back propagation (BPNN) and generalized regression (GRNN) neural networks. Mandal D. et. Al [7] developed a model and optimize the complex electrical discharge machining (EDM) process using soft computing techniques in the paper. In this paper Artificial neural network (ANN) with back propagation algorithm is used to model the process and non-dominating sorting genetic algorithm-II is used to optimize the process. Hosoz M. et. Al [8] proposed the paper to predict the performance of a cooling tower under a broad range of operating conditions using artificial neural networks (ANN). Melhum I. A. et. Al [9] discussed a comprehensive approach for Short Term Load Forecasting (STLF) using artificial neural network. In this paper they used three layer feed forward back propagation neural network. For this study, four ANN models are implemented and validated with reasonable accuracy on real electric load generation output data.

Methodology

Neural Network

Artificial Neural Networks are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. It is natural proof that some problems that are beyond the scope of current computers are indeed solvable by small energy efficient packages. This brain modeling also promises a less technical way to develop machine solutions. This new approach to computing also provides a more graceful degradation during system overload than its more traditional counterparts. These biologically inspired methods of computing are thought to be the next major advancement in the computing industry. Even simple animal brains are capable of functions that are currently impossible for computers. Computers do rote things well, like keeping ledgers or performing complex math. But computers have trouble recognizing even simple patterns much less generalizing those patterns of the past into actions of the future. Now, advances in biological research promise an initial understanding of the natural thinking mechanism. This research shows that brains store information as patterns. Some of these patterns are very complicated and allow us the ability to recognize individual faces from many different angles. This process of storing information as patterns, utilizing those patterns, and then solving problems encompasses a new field in computing. This field, as mentioned before, does not utilize traditional programming but involves the creation of massively parallel networks and the training of those networks to solve specific problems. This field also utilizes words very different from traditional computing, words like behave, react, self-organize, learn, generalize, and forget.

Fig. 1 Creating new data

Fig. 2 Feed input data

Networks Training Process

There are two types of training which can be done by using MATLAB R2011a.

1. By Neural Network toolbox
2. By Command

Training by Neural Network Toolbox

There are generally four steps in the feed forward neural network as shown in figures 1-5:

1. Assemble The Training Data.
2. Create The Network.
3. Train The Network.
4. Simulate The Network Response To New Inputs.

Command for feed forward neural network is: nn tool

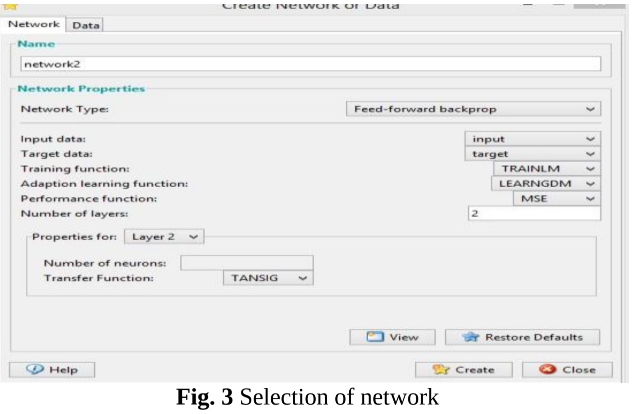

Fig. 3 Selection of network

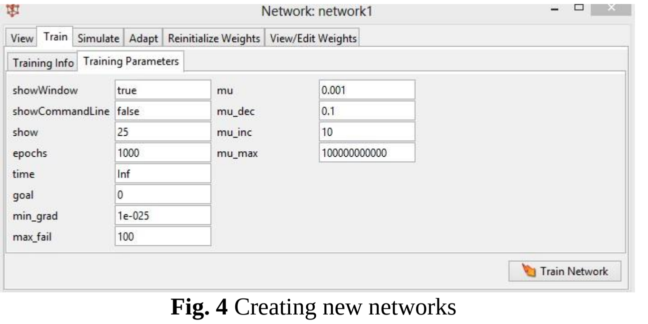

Fig. 4 Creating new networks

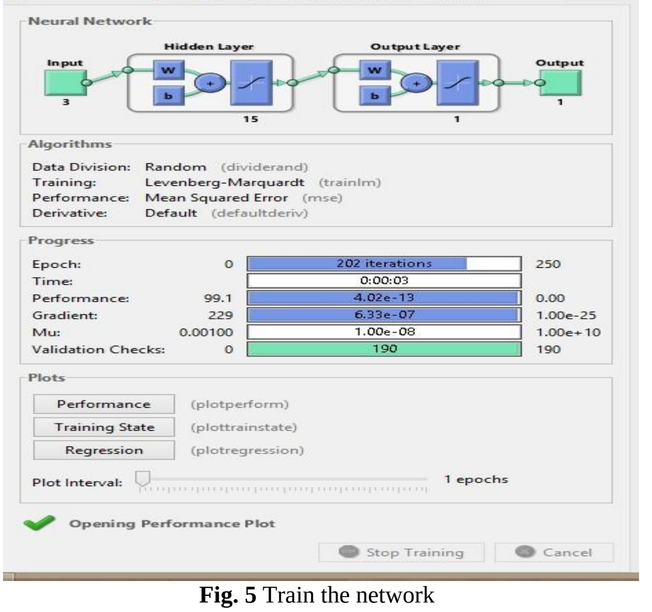

Fig. 5 Train the network

Training by Command

It is a feed forward network which has two layers and the number of neurons in hidden layer is 10.the default number of layers is 2, and the default training function is **trainlm**. The default transfer function for hidden layers is **tansig** and the default for the output layer is **purelin**.

```
load plant_data set
net = feedforwardnet;
net = configure(net,plantInputs,plantTargets);
net = init(net);
[net,tr] = train(net,plantInputs,plantTargets);
Y = sim(net,plantInput)
```

The next task is optional. On some occasions you may wish to alter the training parameters before training. The following line of code displays the default Levenberg-Marquardt training parameters (defined when you set net.trainFcn totrainlm).

```
net.trainParam
```

Improving Results

If the network is not sufficiently accurate, you can try initializing the network and the training again. Each time your initialize a feed forward network, the network parameters are different and might produce different solutions.

```
net = init(net);
net = train(net,plantInputs,plantTargets);
```

As a second approach, you can increase the number of hidden neurons above 20. Larger numbers of neurons in the hidden layer give the network more flexibility because the network has more parameters it can optimize. (Increase the layer size gradually. If you make the hidden layer too large, you might cause the problem to be under-characterized and the network must optimize more parameters than there are data vectors to constrain these parameters.)

Regression Method

Regression analysis models the relationship between one or more response variables (also called dependent variables, explained variables, predicted variables, or regressands) (usually named Y), and the predictors (also called independent variables, explanatory variables, control variables, or regressors) usually named (X1,...,Xp). Multivariate regression describes models that have more than one response variable.

Basically, there are two type of regression model-

Linear Regression

Multiple Regression

For,

1. Linear Regression: $Y = a+bX$
2. Multiple Regressions: $Y = a + b_1 X_1 + b_2 X_2 + \ldots\ldots\ldots\ldots + b_t X_t$

Where:

Y= the variable that we are trying to predict

X= the variable that we are using to predict Y

a= the intercept

b= the slope.

Procedure for Finding a Regression Equation in Minitab

The basic procedure is

1. Collect the data and fit into the Minitab.

2. Select **Stat> Regression > Regression (**feed the variables in response and predictor).

And regression equation is:

$Y = a + b_1 X_1 + b_2 X_2$

Results and Discussion

Thermal power plants have some inputs and outputs. The main input variables of a thermal power plant are fuel flow, feed water flow, load etc. In this paper output of the system is superheater temperature. These input and output parameters are adequate for the thermal power plant.

Implementation of methodology

Company profile

The data being used in this paper work is being collected from Parichha Thermal Power Station Jhansi of 210MW Plant Capacity.

Table 1 shows the actual data of the 210MW plant.

Table 1 210 MW Plant data

Sample No.	Input			Actual Output
	FUEL FEED	FEED WATER FLOW	LOAD	SUPERHEATER TEMPERATURE
1	149.06	591.99	195.81	520.11
2	149.21	615.96	199.17	517.15
3	150.41	595.69	197.29	524.08
4	150.33	602.66	195.81	523
5	152.6	612.96	198.92	514.89
6	152.44	614.62	201.3	515.47
7	152.98	608.83	200.42	522.9
8	154.11	600.67	198.3	524
9	154.06	606.2	198.18	520.37
10	154.17	614.52	197.29	519.36
11	155.39	608.8	197.41	518.29
12	159.44	612.59	198.54	519.09
13	156.2	607.78	197.42	519.26
14	156.29	606.32	197.04	520.39
15	156.34	590.32	195.92	523.74
16	156.63	590.87	194.43	522.8
17	156.74	603.69	194.55	520.17
18	156.79	600.2	193.93	517.2
19	156.56	592.5	193.3	518.03
20	156.78	591.53	194.55	520.67
21	156.78	586.2	192.06	523.61
22	156.92	587.85	189.45	519.28
23	156.71	577.39	190.32	521.92
24	156.95	575.2	190.56	526.53

25	156.9	570.31	195.44	527.83
26	156.5	549.49	185.33	526.2
27	156.02	576.99	187.45	524.28
28	155.08	595.06	188.7	519.14
29	155.08	543.08	181.21	517.91
30	155.17	541.58	182.46	521.68
31	155.15	561.43	184.2	521.17
32	154.83	569.64	184.7	520.72
33	151.31	604.18	185.21	503.27
34	153.17	568.05	185.46	518.25
35	154.82	518.38	171.62	522.9
36	154.68	569.82	183.22	519.49
37	155.04	557.18	182.96	523.91
38	154.89	549.76	182.05	525.73
39	154.55	556.31	183.58	525.21
40	154.77	563.4	184.59	527.24
41	154.48	550.05	183.72	531.26
42	154.54	576.41	189.2	517.68
43	153.59	567.62	183.71	515.97
44	154.98	577.25	188.32	519.02
45	155.08	566.48	186.08	525.01
46	154.77	564.15	184.7	526.57
47	156.6	532.67	175.1	522.07
48	156.84	596.95	187.31	513.59
49	156.97	587.16	189.07	519.22
50	155.22	574.37	193.93	524.33

Where,

Fuel Feed is in tonne/15min.
Feed Water Flow is in m^3/15min.
Load is in MW
Superheater Temperature is in ^{0}C.

Neural Network

Take the 20 values of the table 1 to train the network

Feed the Plant Data

Table 2 shows the plant data which is used for training the network. These data values are feed into the MATLAB. After feeding the data in MATLAB, the input values are feed into input folder and output value feed into the target folder in neural network toolbox.

In neural toolbox, data is randomly selected for training, testing and validation. The data division is set to default in the toolbox [Mark Hudson Beale et. Al, Neural Network Toolbox User's Guide] which are –

60 % data is used for **Training**
20 % data is used for **Testing**
20 % data is used for **Validation**

Table 2 Feed the plant data

Sample No.	Input			Actual Output
	FUEL FEED	FEED WATER FLOW	LOAD	SUPERHEATER TEMPERATURE
1	151.31	604.18	185.21	503.27
2	156.84	596.95	187.31	513.59
3	152.6	612.96	198.92	514.89
4	152.44	614.62	201.3	515.47
5	153.59	567.62	183.71	515.97
6	149.21	615.96	199.17	517.15
7	156.79	600.2	193.93	517.2
8	154.54	576.41	189.2	517.68
9	155.08	543.08	181.21	517.91
10	156.56	592.5	193.3	518.03
11	153.17	568.05	185.46	518.25
12	155.39	608.8	197.41	518.29
13	154.98	577.25	188.32	519.02
14	159.44	612.59	198.54	519.09
15	155.08	595.06	188.7	519.14
16	156.97	587.16	189.07	519.22
17	156.2	607.78	197.42	519.26
18	156.92	587.85	189.45	519.28
19	154.17	614.52	197.29	519.36
20	154.68	569.82	183.22	519.49

Train the Network

After feeding the data in neural toolbox,
Select the feedback propagation network is network type.
Training Function is trainlm.
Number of layers is 2.
First Layer has the 15 Neurons and Transfer Function is transig and transfer function of the second layer is also transig and then, creates the network.
After creating the network, change the training parameter are shown in figures 6 and 7.

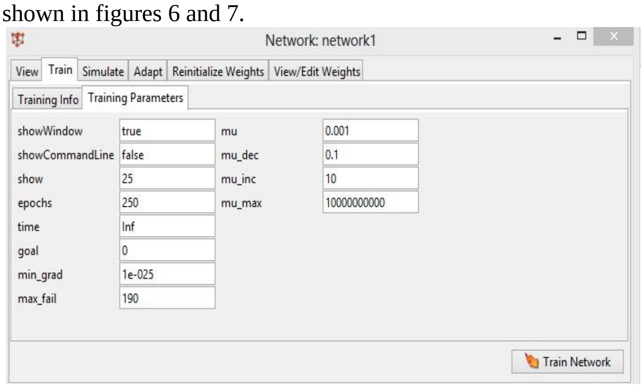

Fig. 6 Training Parameter After changing the parameter train the network

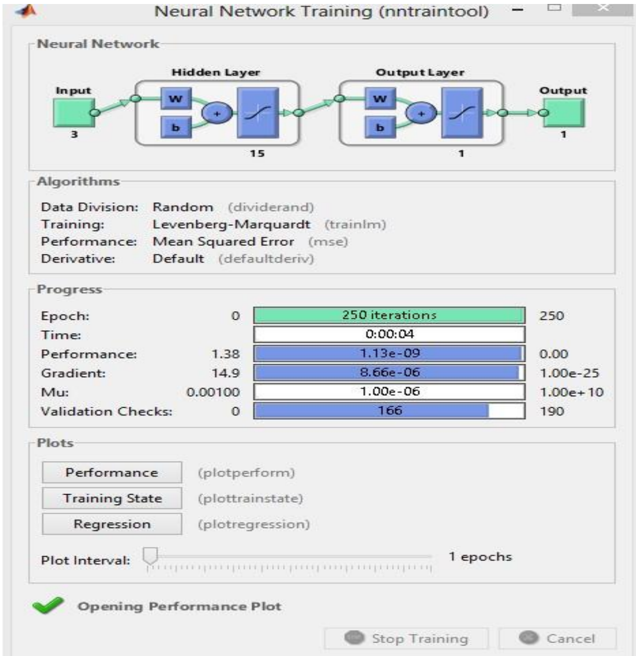

Fig. 7 Training of the network

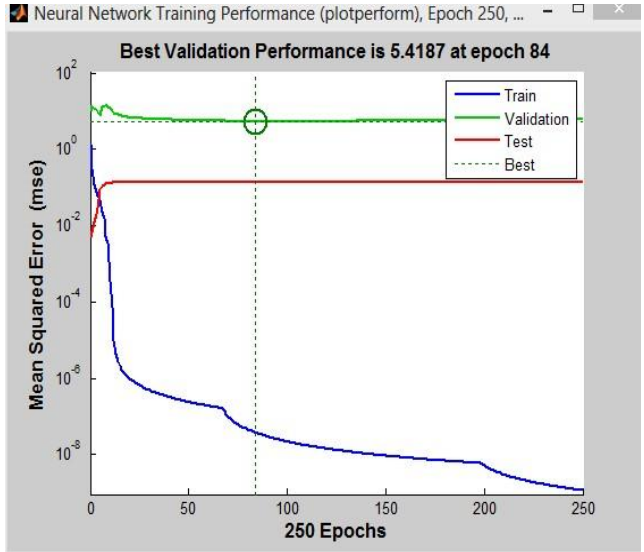

Fig. 8 Performance Curve for 3-15-1 NN

Table 3 Predicted output and error

Sample No.	Error	Predicted Output
1	-0.006868842	503.2768688
2	0.008108742	513.5818913
3	-0.02097725	514.9109773
4	-0.008332819	515.4783328
5	0.01817882	515.9518212
6	0.000925378	517.1490746
7	-1.5184	518.7184
8	0.004959487	517.6750405
9	2.779	515.131
10	-0.001662385	518.0316624
11	0.004989976	518.24501
12	-0.006917454	518.2969175
13	-0.46697	519.48697
14	-0.39507	519.48507
15	2.4955	516.6445
16	-0.18739	519.40739
17	-0.0011404	519.2611404
18	0.006485487	519.2735145
19	0.003250666	519.3567493
20	0.00071903	519.489281

Max. Positive Error = 2.779, Max. Negative Error = 1.5184
Min. Positive Error = 0.00071903, Min. Negative Error = 0.0011404

Performance curve shows the variation between the training line, validation line and test line to the best line as shown in figure 8.

Predicted Output and Error

After training, predicted output and error is collect from the toolbox and shown in Table 3.

Mean Squared Error and Relative Error

Table 4 shows the squared error and relative error

Relative Error = |Error| / Known Value (Actual Output)

Table 4 Squared errors and relative errors

Sample No.	Squared Error	Relative Error
1	29.52085756	0.010796014
2	2.215245257	$2.897973091 \times 10^{-3}$
3	6.504744194	$4.953368681 \times 10^{-3}$
4	10.42967025	$6.265156071 \times 10^{-3}$
5	$9.529859641 \times 10^{-3}$	$1.891989844 \times 10^{-3}$
6	4.826677181	$4.248225853 \times 10^{-3}$
7	3.660295976	$3.69912993 \times 10^{-3}$
8	3.506555856	$3.617253902 \times 10^{-3}$
9	12.03042288	$6.697090228 \times 10^{-3}$
10	4.712980484	$4.190761153 \times 10^{-3}$
11	0.148425267	$7.433863965 \times 10^{-4}$
12	0.391713256	$1.207567192 \times 10^{-3}$
13	5.597956×10^{-4}	$4.558591191 \times 10^{-5}$
14	4.928755206	$4.276869136 \times 10^{-3}$
15	13.63721498	$7.113418346 \times 10^{-3}$
16	0.437225112	$1.273506413 \times 10^{-3}$
17	0.134380896	$7.059661826 \times 10^{-4}$
18	0.393593116	$1.208153597 \times 10^{-3}$
19	6.261455244	$4.818026032 \times 10^{-3}$
20	7.448641808	$5.253652621 \times 10^{-3}$
Sum	**111.1989439**	

Mean Squared Error (MSE) = Sum of squared error/Total no. of sample

MSE = 16.66626933/20= 0.833313466

These values are of 3-15-1 NN configuration which is the best configuration as compare to 3-10-1 NN configuration.

Regression Method

Regression analysis is doing by the MINITAB 14. The table 1 data is feed into the MINITAB.

Finding a Regression Equation

Superheater Temperature is taking as Response Variable. And Fuel Feed, Feed Water Flow and Load are taking as Predictor Variable.

The regression equation is,

SUPERHEATER TEMPERATURE = 417 + 0.594 FUEL– 0.218 FEED WATER FLOW + 0.721 LOAD........................ (1)

Table 5 shows the regression analysis in MINITAB.

Table 5 Regression analysis in MINITAB

Predictor	Coefficient	Se Coefficient	T	P
CONSTANT	416.58	44.27	9.41	0.000
FUEL	0.5938	0.2526	2.35	0.032
FEED WATER FLOW	- 0.21833	0.05801	- 3.76	0.002
LOAD	0.7209	0.1880	3.83	0.001

Table 6 Predicted outputs and error

Sample No.	Predicted Output	Error
1	508.70331	-5.43331
2	515.07837	-1.48837
3	517.44044	-2.55044
4	518.6995	-3.2295
5	516.94621	-0.97621
6	514.95303	2.19697
7	519.11319	-1.91319
8	519.55258	-1.87258
9	521.37849	-3.46849
10	520.20094	-2.17094
11	517.86474	0.38526
12	518.91587	-0.62587
13	518.99634	0.02366
14	521.31008	-2.22008
15	515.44714	3.69286
16	518.55877	0.66123
17	519.62658	-0.36658
18	518.65263	0.62737
19	516.85771	2.50229
20	516.76078	2.72922

Predicted Output and Error

The new output values find out by the use of regression equation (1), are shown in Table 6.

Where

Max. Positive Error = 3.69286,

Max. Negative Error = 5.43331

Min. Positive Error = 0.02366,

Min. Negative Error = 0.36658

Mean Squared Error and Relative Error

Table 7 shows the squared error and relative error.

Table 7 Squared errors and relative error.

Sample No.	Squared Error	Relative Error
1	29.52085756	0.010796014
2	2.215245257	$2.897973091*10^{-3}$
3	6.504744194	$4.953368681*10^{-3}$
4	10.42967025	$6.265156071*10^{-3}$
5	$9.529859641*10^{-3}$	$1.891989844*10^{-3}$
6	4.826677181	$4.248225853*10^{-3}$
7	3.660295976	$3.69912993*10^{-3}$
8	3.506555856	$3.617253902*10^{-3}$
9	12.03042288	$6.697090228*10^{-3}$
10	4.712980484	$4.190761153*10^{-3}$
11	0.148425267	$7.433863965*10^{-4}$
12	0.391713256	$1.207567192*10^{-3}$
13	$5.597956*10^{-4}$	$4.558591191*10^{-5}$
14	4.928755206	$4.276869136*10^{-3}$
15	13.63721498	$7.113418346*10^{-3}$
16	0.437225112	$1.273506413*10^{-3}$
17	0.134380896	$7.059661826*10^{-4}$
18	0.393593116	$1.208153597*10^{-3}$
19	6.261455244	$4.818026032*10^{-3}$
20	7.448641808	$5.253652621*10^{-3}$
Sum	**111.1989439**	

$$\text{Mean Squared Error (MSE)} = \frac{\text{Sum of squared error}}{\text{Total no. of sample}}$$

MSE = 111.1989439/20 = 5.559947195

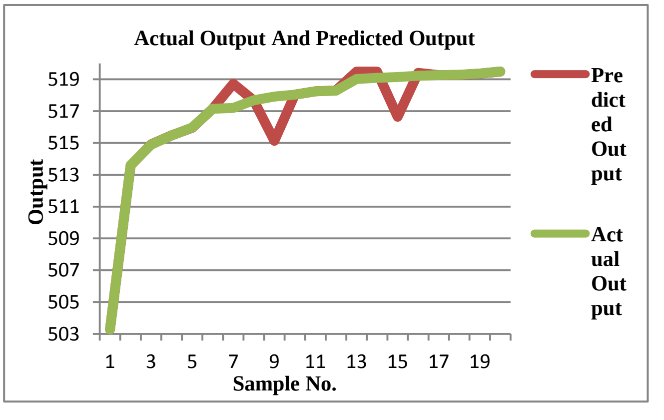

Fig. 9 Output versus sample number

Final Results of Neural Network

Table 8 shows neural network results (predicted output, error, and relative error).

Table 8 Neural network result

Sample No.	Predicted Output	Actual Output	Error	Relative Error
1	503.2768688	503.27	-0.006868842	$1.364842331*10^{-5}$
2	513.5818913	513.59	0.008108742	$1.578835647*10^{-5}$
3	514.9109773	514.89	-0.02097725	$4.074122628*10^{-5}$
4	515.4783328	515.47	-0.008332819	$1.616547811*10^{-5}$
5	515.9518212	515.97	0.01817882	$3.523231971*10^{-5}$
6	517.1490746	517.15	0.000925378	$1.789380257*10^{-6}$
7	518.7184	517.2	-1.5184	$2.935808198*10^{-3}$
8	517.6750405	517.68	0.004959487	$9.580217509*10^{-6}$
9	515.131	517.91	2.779	$5.365797146*10^{-3}$
10	518.0316624	518.03	-0.001662385	$3.209051599*10^{-6}$
11	518.24501	518.25	0.004989976	$9.628511336*10^{-6}$
12	518.2969175	518.29	-0.006917454	$1.334668622*10^{-5}$
13	519.48697	519.02	-0.46697	$8.997148472*10^{-4}$
14	519.48507	519.09	-0.39507	$7.610818933*10^{-4}$
15	516.6445	519.14	2.4955	$4.806988481*10^{-3}$
16	519.40739	519.22	-0.18739	$3.609067447*10^{-4}$
17	519.2611404	519.26	-0.0011404	$2.196202288*10^{-6}$
18	519.2735145	519.28	0.006485487	$1.248938338*10^{-5}$
19	519.3567493	519.36	0.003250666	$6.258984134*10^{-6}$
20	519.489281	519.49	0.00071903	$1.38410749*10^{-6}$

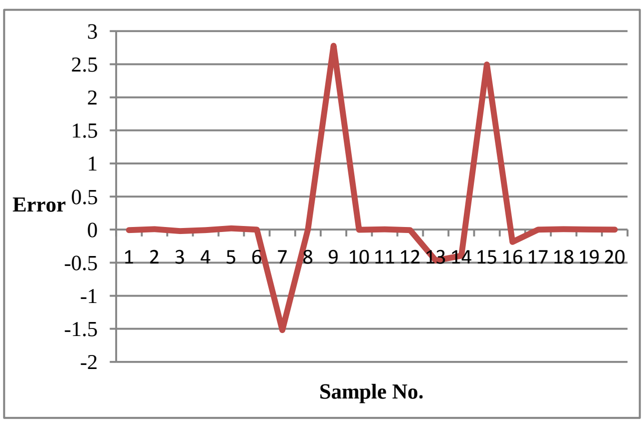

Fig. 10 Error versus sample number

Output, Error and Relative Error versus Sample No. Curve

Figures 9-11 show the output, error and relative error versus sample no. curves in which a graph is being plotted to show the comparison between actual output and predicted output.

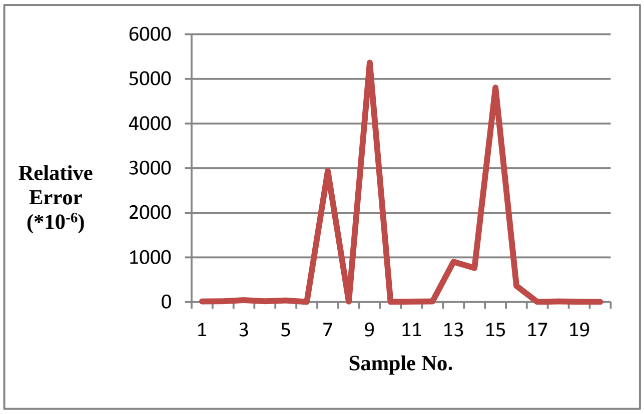

Fig. 11 Relative error versus sample number

Table 9 Regression result

Sample No.	Predicted Output	Actual Output	Error	Relative Error
1	508.70331	503.27	-5.43331	0.010796014
2	515.07837	513.59	-1.48837	$2.897973091*10^{-3}$
3	517.44044	514.89	-2.55044	$4.953368681*10^{-3}$
4	518.6995	515.47	-3.2295	$6.265156071*10^{-3}$
5	516.94621	515.97	-0.97621	$1.891989844*10^{-3}$
6	514.95303	517.15	2.19697	$4.248225853*10^{-3}$
7	519.11319	517.2	-1.91319	$3.69912993*10^{-3}$
8	519.55258	517.68	-1.87258	$3.617253902*10^{-3}$
9	521.37849	517.91	-3.46849	$6.697090228*10^{-3}$
10	520.20094	518.03	-2.17094	$4.190761153*10^{-3}$
11	517.86474	518.25	0.38526	$7.433863965*10^{-4}$
12	518.91587	518.29	-0.62587	$1.207567192*10^{-3}$
13	518.99634	519.02	0.02366	$4.558591191*10^{-5}$
14	521.31008	519.09	-2.22008	$4.276869136*10^{-3}$
15	515.44714	519.14	3.69286	$7.113418346*10^{-3}$
16	518.55877	519.22	0.66123	$1.273506413*10^{-3}$
17	519.62658	519.26	-0.36658	$7.059661826*10^{-4}$
18	518.65263	519.28	0.62737	$1.208153597*10^{-3}$
19	516.85771	519.36	2.50229	$4.818026032*10^{-3}$
20	516.76078	519.49	2.72922	$5.253652621*10^{-3}$

Regression Method

Table 9 shows the regression result (predicted output, error, and relative error).

Output, Error and Relative Error versus Sample No. Curve

Figures 12-14 show the output versus sample no. curve in which a graph is being plotted to show the comparison between actual output and predicted output.

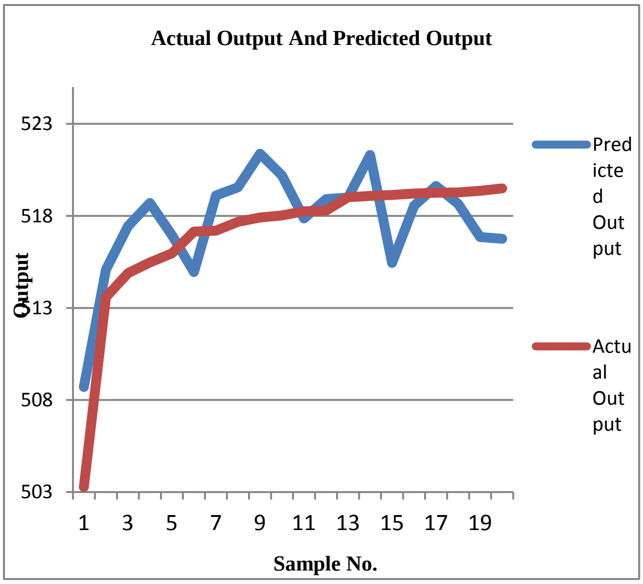

Fig. 12 Output versus sample number

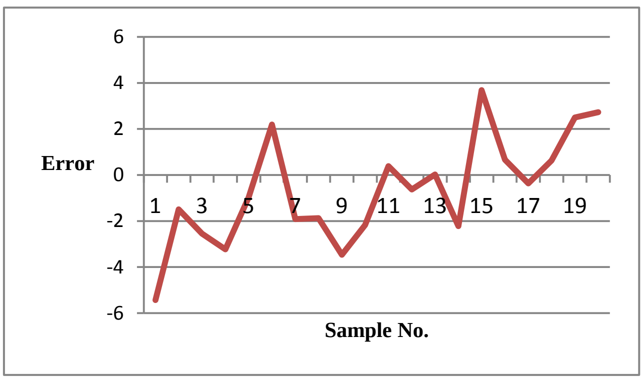

Fig. 13 Error versus sample number

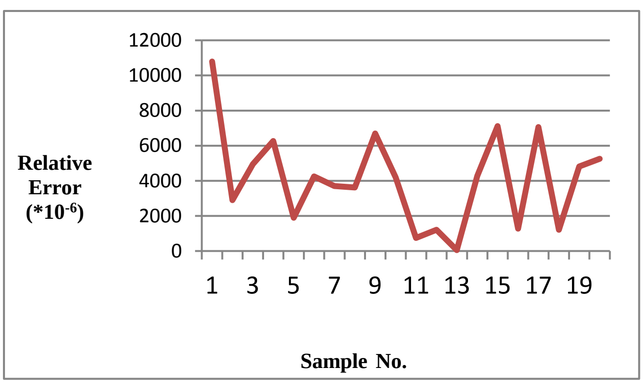

Fig. 14 Relative error versus sample number

Comparison between Neural Network and Regression Method

Figure 15 shows the curve between the outputs which is being obtained by neural network and by regression. Here we compare these outputs with the actual output by the plant. After comparison, we found that the output curve obtained by neural network is much closer to actual output of plant in comparison with the output curve obtained by regression method which implies that the output obtained by neural network best suits to the actual output.

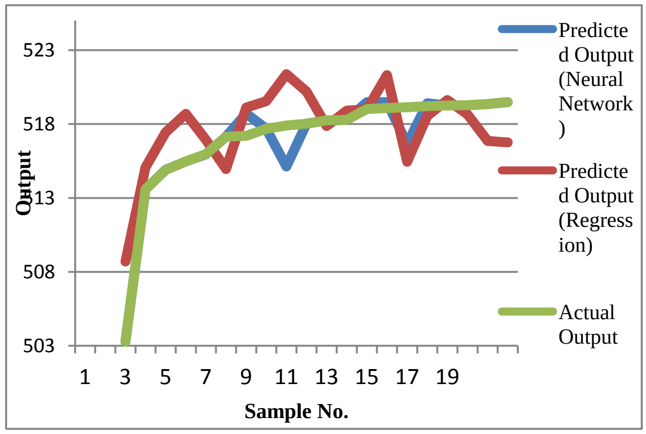

Fig. 15 Comparison Curve between Neural Network and Regression Method Output

Figure 16 shows the curve between the error which is being obtained by neural network and by regression method. Here, we can see that the curve obtained by neural network is much closer to the base line in comparison to the error line obtained by regression method. As from the curve we notice the variations from the base line is less in the error line of neural network which depicts that neural network technique is best suited for this work in comparison with regression technique.

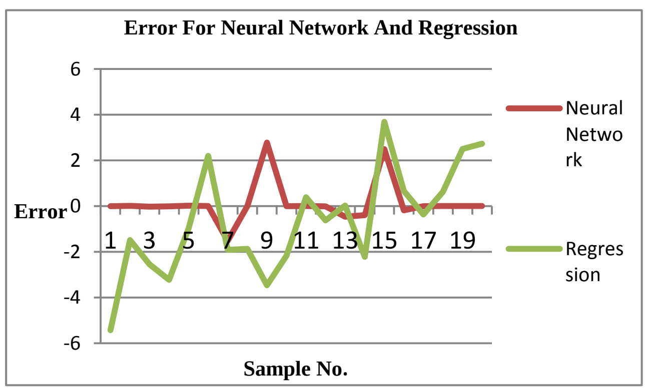

Fig. 16 Comparison Curve between Neural Network and Regression Method Errors

Figure 17 shows the curve between the relative error and sample no. which is being obtained by neural network and by regression method. As from the curve we notice the variations from the base line is less in relative error line obtained by neural network in comparison with regression method which depicts that neural network technique is best suited for this work in comparison with regression technique.

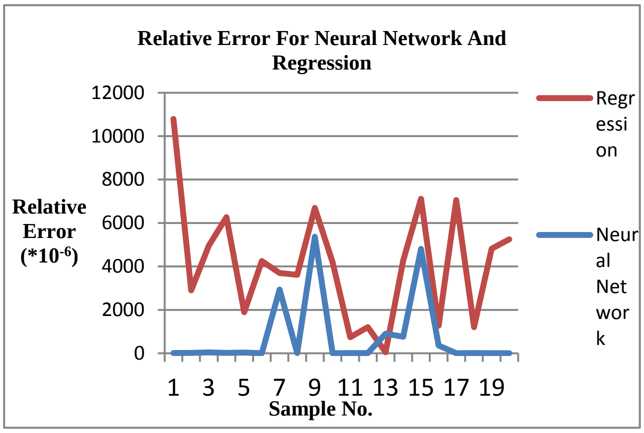

Fig. 17 Comparison Curve Between Neural Network and Regression Method Relative Errors

Mean Squared Error (MSE)

Finally after having all errors we calculated mean squared error whose results are shown in table 10. The table shows a vast difference between the error obtained by neural network and regression technique. Since we know that error should always be less which shows that neural network technique gives a small amount of MSE which clearly shows neural network technique will be the best opted technique for this research work.

Table 10 Mean squared error

Method	Neural Network	Regression
MSE	0.833313466	5.559947195

Conclusion

This paper represents the modeling and optimization of thermal power plant using the techniques such as neural network and regression and comprises the input parameters are fuel feed, feed water flow and load and output parameter is super heater temperature. The predicted outputs from the model, error, relative error and mean squared error are being calculated using neural network and regression techniques. The performance of the thermal power plant is predicted by using these techniques.

The output curve obtained by neural network is much closer to actual output of plant in comparison with the output curve obtained by regression method. The error curve shows that the variations from the base line is less in the error line of neural network which depicts that neural network technique is best suited for this work in comparison with regression technique. Neural Network technique gives a small amount of mean squared error (MSE) and the relative error as compare to regression technique.

These results shows neural network technique will be the best opted technique for this research work.

Notes and References

[1] *Department of Mechanical Engineering, Bundelkhand Institute of Engineering & Technology, Jhansi*

[2] *Department of Mechanical Engineering, Government Engineering College, Banda*

Email: *jitendrarajput1988@gmail.com*

1. Aurora C., Magni L., Scattolini R., Colombo P., Pretolani F. and Villa G. (2004) "Predictive control of thermal Power Plants". International Journal of Robust and Nonlinear Control Int. J. Robust Nonlinear Control 2004; 14:415 – 433.
2. Cerri G., Borghetti S. and Salvini C. (2006) "Neural management for heat and power cogeneration plants". Engineering Applications of Artificial Intelligence 19 (2007) 721–730.
3. Ilamathi P., Selladurai V. and Balamurugan K. (2012) "Predictive Modeling and Optimization of Power Plant Nitrogen Oxides Emission". IAES International Journal of Artificial Intelligence Vol. 1, No. 1, March 2012, pp. 11-18.
4. Manke P. and Tembhurne S. (2012) "Application of Back Propagation Neural Network to Drum Level Control in Thermal Power Plants". IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 1, March 2012.
5. Panda S., Swain B. and Mishra S. (2013) "Blow down Losses Control in Thermal Power Plants Using Neural Network". International Journal of Advancements in Research & Technology, Volume 2, Issue5, May-2013.
6. Zhao H., Zhao J. P. , Zheng L. G., Wang C. L. and Cen K. F. (2011) "Modeling NOx emissions from coal-fired utility boilers using support vector regression with ant colony optimization". Engineering Applications of Artificial Intelligence 25 (2012) 147–158.
7. Mandal D., Pal S. K. and Saha P. (2006)"Modeling of electrical discharge machining process using back propagation neural network and multi-objective optimization using non-dominating sorting genetic algorithm-II". Journal of Materials Processing Technology 186 (2007) 154–162.
8. Hosoz M., Ertunc H. M. and Bulgurcu H. (2006) "Performance prediction of a cooling tower using artificial neural network". Energy Conversion and Management 48 (2007) 1349–1359.
9. Melhum I. A., Omar L. A. B. and Mahmood S. A. (2013) "Short Term Load Forecasting using Artificial Neural Network". International Journal of Soft Computing and Engineering Volume-3, Issue-1, March 2013.