Novelty Journals

ISSN 2394-9678

International Journal of Novel Research in Electrical and Mechanical Engineering
Vol. 2, Issue 3, pp: (46-56), Month: September-December 2015, Available at: www.noveltyjournals.com

On-Line Voltage Security Analysis Using Enhanced Radial Basis Function Neural Network

[1]Poornima Vinodiya, [2]Dr. L.S. Titare

JEC Jabalpur, India

Abstract: **Power system security is one of the major concerns in recent years due to the deregulation of power systems which are forced to operate under stressed operating conditions. This paper presents an enhanced radial basis function neural network (ERBFNN) to examine whether the power system is secure under steady-state operating conditions. Hidden layer units have been selected with the proposed algorithm, which has the advantage of being able to automatically choose optimal unit centers and distances. The proposed approach to contingency analysis was found to be suitable for fast voltage and line-flow contingency screening. The generalization capability of the proposed method was able to identify unknown contingencies with large range of operating conditions and changes in network topology. The advantages of this method are simplicity of algorithm and high accuracy in classification. A case study with IEEE 14-bus power system is used to illustrate the good performance of the proposed method.**

Keywords: **Power system security; Static evaluation; Correlation coefficient; voltage stability; Radial basis function.**

1. INTRODUCTION

Power system security assessment can be classified into as static security and transient security [1]. Static security evaluation of a power system copes with analyzing the system steady state operation after disturbances and detects any potential overload of a system branch or an out of limit voltage following a given list of contingencies [3]. Transient security analysis entails evaluation of power system's ability to withstand a set of severe but credible contingencies and to survive transition to an acceptable steady state condition [2]. The severity of contingencies is assessed on the basis of a scalar performance index (PI) in the static security evaluation. Several PI based methods have been reported in literatures [4–6]. A straight approach to this problem would involve performing full AC load flow for each contingency event followed by operating limit violations has been reported in [7–9]. These conventional approaches are hard to put through online due to high computational requirements. With the development of artificial intelligence based techniques such as expert system techniques [10], and fuzzy logic approaches [11] in recent years, there is growing trend in applying these methods for the operation and control of power system. Pattern Recognition (PR) techniques have shown great important as a means of estimating the severity of contingencies [12]. Artificial neural network systems obtained popularity in respect of the conventional methods as they are efficient in discovering similarities among large bodies of data and synthesize complex mappings accurately and rapidly. Research work related to static security analysis [13, 14] and dynamic security analysis [15–17] have been reported in literature. Most of the published work in this area utilizes multilayer perceptron (MLP) model based on back propagation (BP) algorithm, which usually encounters to local minima and over fitting problems. The contingency analysis process by conventional load flow solution using FDLF can give the solution to the contingency selected only for one loading and generating condition at one go. But practically a power

International Journal of Novel Research in Electrical and Mechanical Engineering
Vol. 2, Issue 3, pp: (46-56), Month: September-December 2015, Available at: www.noveltyjournals.com

system can have a varying level of operating conditions and to predict the solution of contingency analysis for all the possible and future operating points is an impossible task. In such cases, the use of Artificial Neural Networks (ANN) can be useful. Since ANN has the ability to predict the output for unseen input set once it gets trained with sufficient number of training patterns. It has been used for dynamic nature problems in power system like contingency analysis, load forecasting, component fault detection etc. The ability of Neural Networks to predict the outcome for a new pattern in a fast and accurate manner makes them suitable for online analysis also. This paper discusses the brief review of ANN, its application to power system and its ability to solve the contingency analysis problem. The one case study is used to investigate the solution of contingency selection problem using Radial Basis Function Neural network and the results are discussed thoroughly.

2. REVIEW OF ARTIFICIAL NEURAL NETWORKS

The aim of neural network is to mimic the ability of biological neuron or to perform the same kind of processing ability as that of a human brain. It is known that human brain learns by example to which it has experienced in the past and it has the ability to apply this experience for predicting the future situations. The capability of an artificial neural network (ANN) is same as that of a digital computer and it enables parallel processing. The major advantage of using ANN is that once trained the network maps well to any kind of input data and provides an accurate fault tolerant system.

2.1 Choice of Neural Network for Contingency Analysis:

Out of several neural network topologies available, it is essential to choose an appropriate neural network, which perfectly solves the contingency analysis problem. The use of multi layer perceptron network trained by using error back propagation algorithm is a popular choice for analysis of a complex mapping problem but this suffers from slow rate of convergence and local minima problem. However, the combined use of supervised learning and unsupervised learning can alleviate the problem of local minima. Unsupervised networks can be viewed as a data pre-processing step which reduces the data before learning the data characteristics with supervised learning. Among the networks like self organizing, progressive learning, counter propagation the best non linear mapping capability is provided by the Radial Basis Function Neural Network [15]. It has excellent convergence characteristics on a huge dimensionality and its capability to augment new data without any retraining makes it a robust tool. Further it has advantages like its structural simplicity, training efficiency and no local minima problem.

2.2 Classical Radial Basis Function Neural Network:

RBFNN have been used in the solution of problems such as pattern classification and nonlinear functional approximation. The construction of RBFNN, in its most basic form, involves three layers with entirely different roles. The output layer gives the network output vector which is a linear combination of the basis function outputs. The form of the radial basis function, $O_j(x)$ is strictly positive and symmetric with a unique maximum at the center. The most commonly used form is the Gaussian basis function and given as:

$$O_j(x) = exp^{-\frac{(x-t_j)^2}{2\sigma_j^2}} \qquad (1)$$

where $t_j = [t\_j1, t\_j2, \dots, t\_jm]^T$ is the field centers matrix and σ_j is the radius in the function. ***j*** **-** takes the value from ***1 to m***, m **-** signifies the total number of input pattern, $\boldsymbol{t_j}$ **-** is bias centre vector associated with $\boldsymbol{j^{th}}$ hidden unit, ***j*** **-** takes the value from ***1 to m***, ***m*** **-** signifies the total number of bias centers, **σ** - is the spread width whose value is being kept greater than 0. The symbol $\|.\|$ represents the Euclidean 2- norm used in the Gaussian function. Compared to the well-known multi-layer perceptron (MLP) neural networks trained by the back-propagation method (BP), the RBFNN has the advantages of having a substantially faster training procedure, and it does not encounter the problem of local-minima. This is due to its typical two-stage training scheme. In the first stage the basic functions are determined by using the input vectors. The weights connected to the output layer can then be derived by applying both input and output data samples in the second stage. If the number of neurons in hidden layer is too small the generalized output vectors may be in low accuracy. Conversely too large a number may cause over-fitting of the input data, and hence also upsetting the global generalization performance.

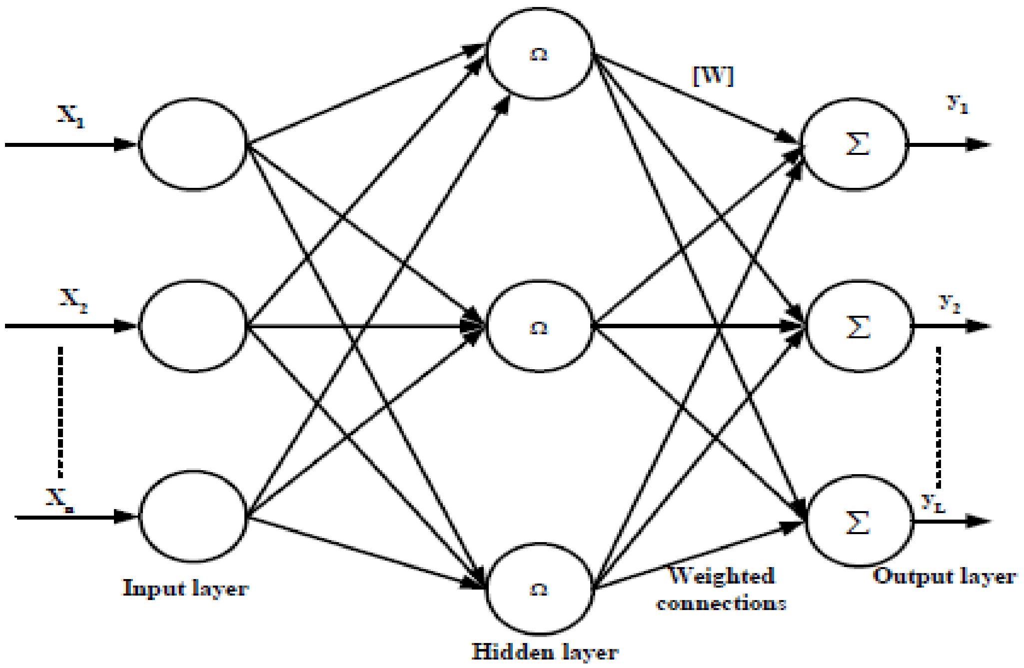

Fig. 1 General structure of Radial Basis Function Neural Network

It is to be noted that the input is passed from the input layer to the hidden layer without any transformation. After computing the signal in accordance to the Gaussian function, the activated output from the hidden neuron is passed on to the linear output neurons through the weighted connected nodes. The size of the weight matrix is **L x J** and the output from the activity vector **y** is given by eqn. 2

$$y_l = \sum_{j=1}^{J} W_{lj} O_j \tag{2}$$

Where,

W_{lj} is the synaptic weights connecting hidden neuron *j* to output neuron *l* and j is the number of neuron in the hidden layer. There is no formal method for specifying the required number of hidden units in an RBF network. It is to be noted that the weight of the RBF network is determined by a combination of supervised and unsupervised learning, the learning which takes place between the input and hidden layer is unsupervised one and the learning that takes between the hidden and output layer is supervised learning. As the network is required to fit the training data with the desired output $\boldsymbol{d_j}$, where $\boldsymbol{j}$ takes the value from ***1 to n***. It is required to fit the training data as per the eqn. 3.

$$y_l = d_j \tag{3}$$

Putting the value of eqn. 3 in eqn. 2 we obtain the eqn. 4.

$$d_j = \sum_{j=1}^{J} W_{lj} O_j \tag{4}$$

Writing eqn. 4 in the matrix form we obtain eqn. 5 given by

$$H.W = d \tag{5}$$

The required weight between the hidden and output unit can be found directly from the above equation. The **H** matrix is not a square matrix, therefore no unique inverse exists for the **H** matrix. Therefore to calculate the weight **W**, the minimum norm solution as explained below is used;

$$W = H^{+} d$$

$$= (H^T H)^{-1} H^T d \tag{6}$$

International Journal of Novel Research in Electrical and Mechanical Engineering
Vol. 2, Issue 3, pp: (46-56), Month: September-December 2015, Available at: www.noveltyjournals.com

Here $\mathbf{H}^{+}$ is called the pseudo inverse of matrix **H**

2.3 Proposed Algorithm Based On A Growing And Pruning Training Method:

In the traditional RBFNN, if the number of neurons in hidden layer is too small the generalized output vectors may be in low accuracy and Conversely, too large a number may cause over fitting of the input data, and hence also upsetting the global generalization performance. A significant problem in RBFNN design, however, is selecting the appropriate number and positions of the radial basis functions in the hidden layer space. If the number of RBF neurons is not chosen properly, the network may present poor global generalization capability, slow training speed, and large memory space request. Another problem, from a classification point of view, is the boundary patterns where clusters of each class contain data from other classes. In the proposed algorithm, the boundary region separates different classes and the patterns of each class lie in their corresponding clusters.

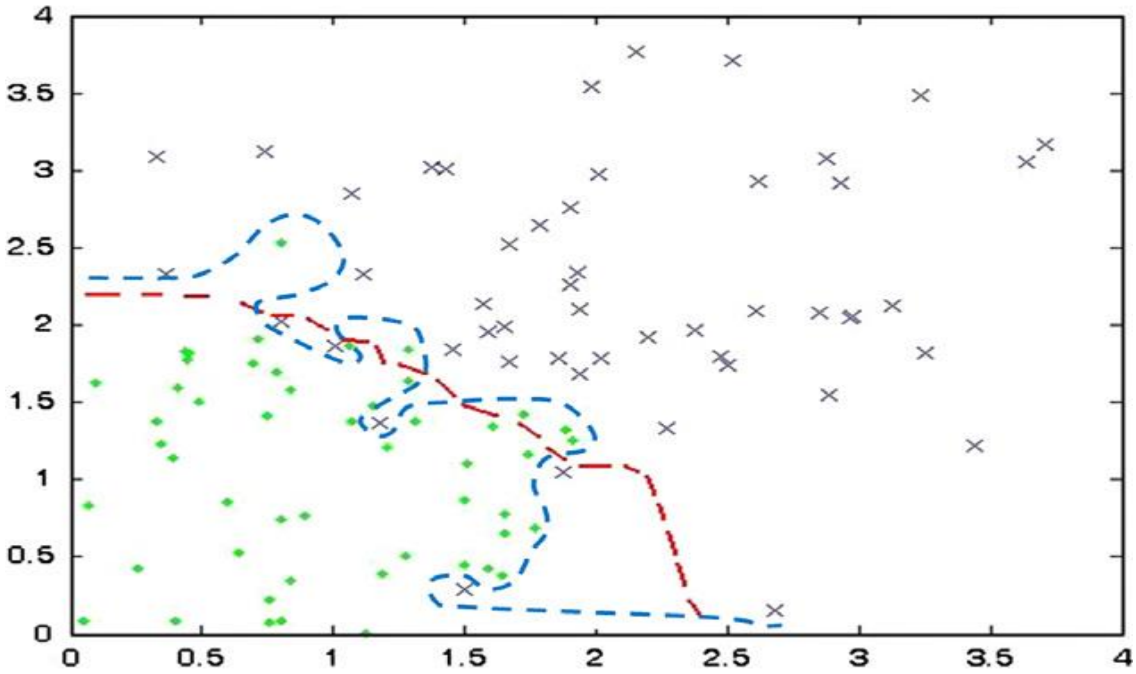

Fig. 2. Boundary of two classes

The red dash line is an improper classification and the blue is an exact classification. Fig.2 illustrates two boundary regions that if the red dash line place into boundary, the different data of classes are classified inaccuracy and for splitting the two classes by distance of proposed algorithm for clusters partitioning accurately similar to the blue dash line. In the growing and pruning training algorithm, these imperfections have been eliminated. The incremental training algorithm is outlined as follows:

(1) Suppose the classes A and B have m and n patterns respectively.

(2) calculate the distance between m patterns in class A and n patterns in class B:

$$P_i = \left\{ \begin{array}{c|c} \|a_1 - b_1\| & \|a_1 - b_n\| \\ . & . \\ . & . \\ . & . \\ \|a_m - b_1\| & \|a_m - b_n\| \end{array} \right\} \qquad (7)$$

(3) Find the minimum distance from pattern ai to patterns of class B in every row of matrix Pi:

$$Dis(a_1;\ B) = \min(\ \|a_1 - b_1\|, \ldots \|a_1 - b_n\|)$$

$$\ldots$$

$$.. \qquad (8)$$

$$Dis(a_m;\ B) = \min(\ \|a_m - b_1\|, \dots \|a_m - b_n\|)$$

(4) Determine the maximum distance from (8), therefore, consider R_{aj} and a_j as the radius and center for first neuron and cluster of class A:

$$R_{aj} = \max(Dis(a_1, B), \dots, (Dis(a_m, B)$$

(5) Calculate the distance between patterns of class A to each other:

$$Q_i = \left\{ \begin{array}{c|c} \|a_1 - b_1\| & \|a_1 - b_n\| \\ . & . \\ . & . \\ . & . \\ \|a_m - b_1\| & \|a_m - b_n\| \end{array} \right\} \tag{9}$$

(6) Sort the distance of aj from other patterns in class A:

$$\|a_j - a_1\|, \|a_j - a_2\|, \dots \|a_j - a_m\| \tag{10}$$

(7) Remove the patterns of class A for smaller distances than R_{aj} (Pruning) and label as the first cluster, for remainder of the patterns of class A which have larger distances than R_{aj} , continue the process till all patterns in class A have been eliminated.

(8) Repeat the algorithm to the n patterns of the class B. Steps (2)–(4) compare the patterns of A with B and search for the optimum center and radius:

$$\forall a_i \& b_j \epsilon\ B = \ R_{ai} = Max_i Min_i\ norm_2(a_i, b_{j=1,2,\dots n}) \tag{11}$$

where, R_{ai} and a_i are the selected radii and centers set for class A. Steps (5)–(7) compare the pattern selected as the center with other patterns of class A and if distances are smaller than the selected radius, put them in one cluster and iterate the algorithm for the remainder of the patterns of class A which are outside the cluster. So, the proposed algorithm is repeated until all patterns of class A sit in their optimum clusters. Fig. 3 graphically illustrates the procedure of clustering stages for an artificial data set for classes A and B. The presented algorithm also can be utilized for several classes and it has not limitation of the number of the classes.

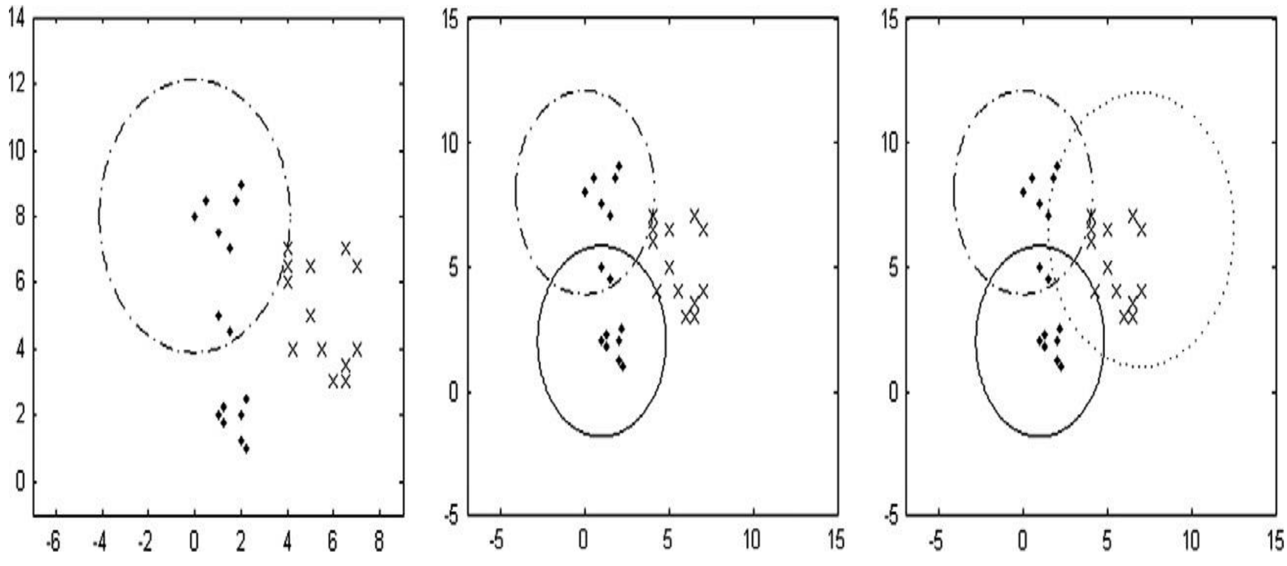

Fig.3. Classification of an artificial dataset

Fig. 4 shows the 6 classes classification. The construction of improved RBFNN, in its most basic form, involves three layers with entirely different roles as shown in Fig. 5(A-B). The first layer is an input layer of which each node corresponds to an attribute of an input pattern. The second layer is a hidden layer and the transformation from the input layer to the hidden layer is nonlinear.

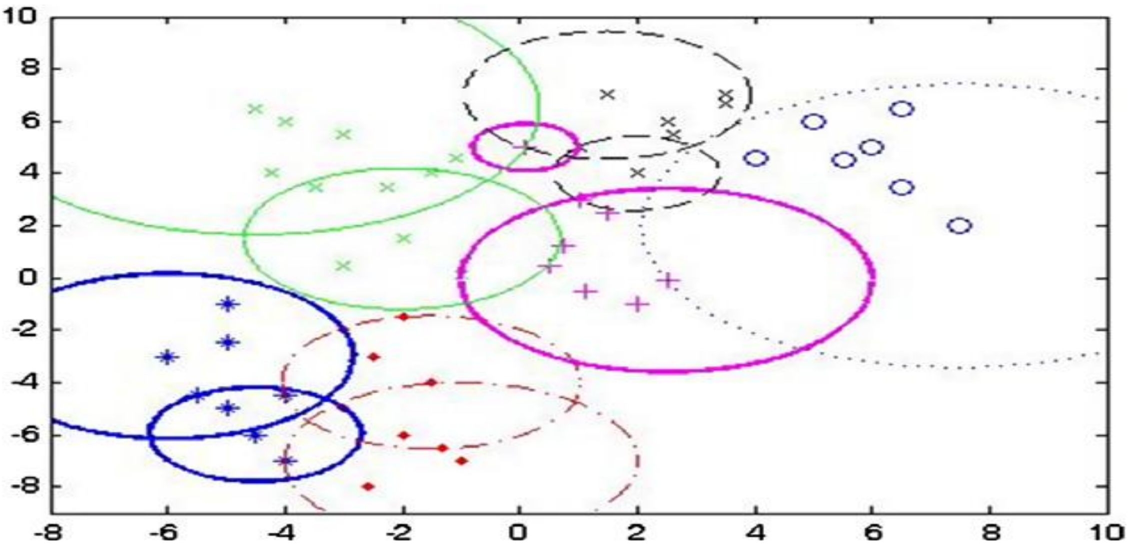

Fig. 4. Classification of an artificial dataset for 6 classes

Whereas, it is constructed using the proposed algorithm. The third layer gives the network output vector which is a linear combination of the basis function outputs. Thus, for the improved RBFNN having l-neurons, the relationship between a n-dimensional output vector $Y = [y_1, y_2, \dots, y_n]^T$ and a m-dimensional input vector

X = $[X = [x_1, x_2, \dots, x_m]^T$, can be expressed as:

$$y_l(x) = \sum_{j=1}^{J} W_{lj} O_j(x) \quad (12)$$

With minimizing the Sum of Square Error (SSE), weight matrix (W) is defined as:

$$SSE = \sum_{p=1}^{P} \left\| d_p - W x O_p \right\|^2 \quad (13)$$

where d_p is a desirable output and P is number of patterns. In the matrix form:

$$\text{SSE}=(\underline{D} - Wx\underline{O})^T \text{x} (\underline{D} - Wx\underline{O}) \quad (14)$$

where $\underline{D} = [\underline{d}_1, \underline{d}_2, \dots, \underline{d}_P]_{nxP}$ and $\underline{O} = [\underline{o}_1, \underline{o}_2, \dots, \underline{o}_P]_{lxP}$ are the desirable matrix and output matrix of hidden layer respectively. Finally, objective function is minimizing the SSE and can be expressed as:

$$\frac{\partial SSE}{\partial W}=0=\text{W}=\underline{D}X\underline{O}^T\text{X}((\underline{O}X\underline{O}^T)^{-1} \quad (15)$$

3. ENHANCED RBF APPLIED TO CONTINGENCY ANALYSIS

The two performance indices parameters PI_P and PI_V indicating a measure about the severity of the line contingency as discussed in section 2.6, have been evaluated by the use of RBF-ANN. It is clear that when any neural network is applied for a practical problem two modes of operation namely training and testing are performed. In training, sets of training data are used to adjust the weights of the network. Whereas, in testing the network response is being seen for the new data which had not been used in training. For the contingency analysis problem the training data has been obtained by using conventional load flow solution for different loading levels and generation scenarios. The training phase has been carried out till the error between the desired and the actual output is small. Whereas, testing is done using the data of loading levels that had not been used in the training phase. The RBF-ANN consists of three layers namely input, hidden and output layer, it's the hidden layer which computes the output by an exponential measure between the input data and a sample data. A generalized adopted model for the input, hidden and output layer which has been used for Contingency analysis is shown in Fig. 2, the salient features are:

The Input layer should include as many neurons as required for the desired input information. Generally the power injections for generator and load bus are chosen as the raw inputs to the ANN. It is advantageous to choose the power

International Journal of Novel Research in Electrical and Mechanical Engineering
Vol. 2, Issue 3, pp: (46-56), Month: September-December 2015, Available at: www.noveltyjournals.com

injections as the input data since they are readily available where as the parameters like bus voltages and phase angles cannot be obtained directly. The input layer [x] consists of power injections, P and Q at the generator and load buses, and the value Ki which represents the outage of line *i*, a sample of input pattern has been shown:

$$[x] = [P_{G1}, Q_{G1},\ldots\ldots\ldots P_{Gg}, Q_{Gg}, P_{L1}, Q_{L1},\ldots\ldots\ldots P_{Ln}, Q_{Ln}, K_i] \quad (16)$$

where,

G is the generator bus, g is the number of generators in the power system, L is a load bus, n is the number of load buses, K_i is the number denoting the outaged line.

The Middle layer does not have any criteria for selecting the number of neurons and the choice of the number of neurons in this layer is based on experimentation and simulation, in general the more the number of neurons in the middle layer the better the network can fit the targets while too many neurons in the middle layer can result in over fitting hence the process of experimentation is purely followed for selecting the number of neurons.

The Output layer consists of an output vector [O] with two elements which are the active and reactive power performance indices PI_P and PI_V respectively, i.e.,

$$[O] = [PI_P, PI_V] \quad (17)$$

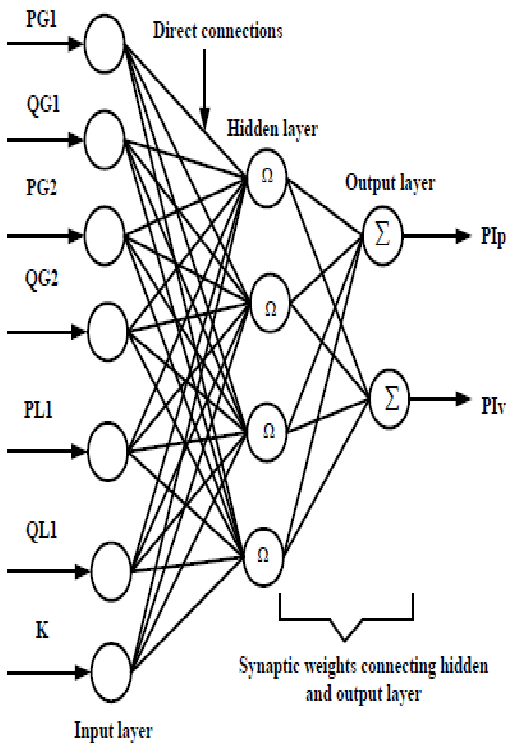

Fig. 5-A RBF network used for Contingency Analysis

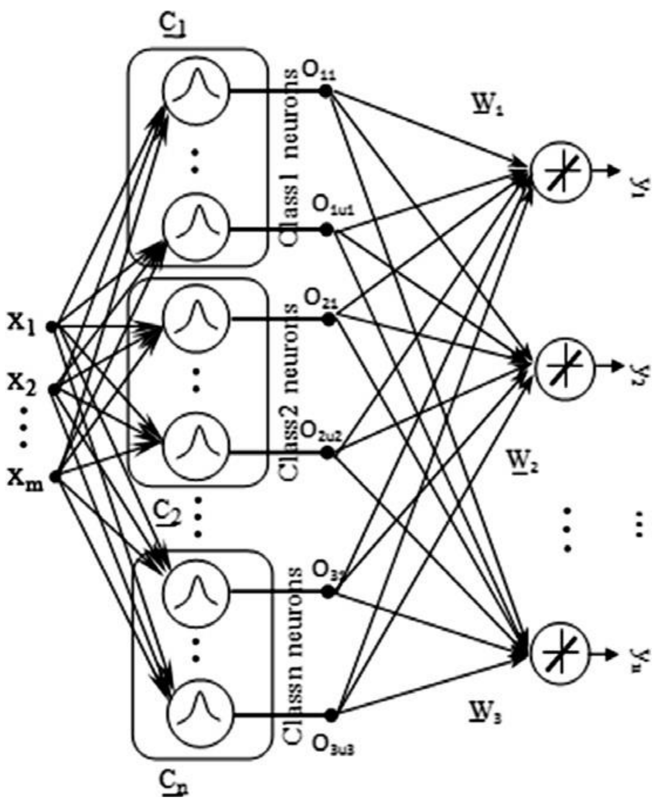

Fig. 5-B. Structure of ERBF neural network

Algorithm for PI_P and PI_V Indices prediction using RBF-ANN

The solution algorithm for the prediction of the active and reactive power performance indices (PI_P & PI_V) is given below:

1. Carry out the fast decouple load flow analysis for all the single line outage cases and calculate the PI_P and PI_V. The indices obtained from this stage are considered as the desired output.

2. Select inputs for the RBFN $[x] = [P_{G1},Q_{G1},\ldots\ldots,P_{Gg},Q_{Gg}, P_{L1},Q_{L1},\ldots\ldots,P_{Ln},Q_{Ln},K_i]$.

International Journal of Novel Research in Electrical and Mechanical Engineering
Vol. 2, Issue 3, pp: (46-56), Month: September-December 2015, Available at: www.noveltyjournals.com

3. Select the number the number of neurons in the middle layer and determine the bias centres **tj** from the input training vector.

4. Select the appropriate value of the spread **σ**.

5. Calculate the output of the j^{th} hidden unit using the eq. 1.

6. Calculate the output of the l^{th} unit of the output neuron yl using eq. 2.

7. Calculate the weight matrix that solves the network specification using eq. 6.

8. Select the test data for the network, these data are chosen different from the data that had been used for the training purpose.

9. Use the weight matrix obtained in step (7) to compute the output from the output layer.

10. Check if the output obtained in step (9) is close to the desired output, if yes then stop, else change the value of number of bias centres and spread and repeat steps (5-9).

4. RESULTS AND DISCUSSION

In this section, the results of contingency analysis problem using enhanced RBF neural network have been presented. The algorithms are implemented in MATLAB for the above. The main objective is to determine the active and reactive power performance indices which form an important part of contingency analysis for different bus systems. The algorithms have been evaluated on two set of bus systems which are being referred as

14- Bus Test System The results of active power performance index PI_P and reactive power performance indices PI_V obtained by using the RBF neural network for a loading level of 10% above the base case i.e. at 2849 MW are given in Table 1.

Table 1 Performance Indices & Contingency Ranking using RBF-ANN for 14-Bus System

Contingency number	PI_P	PI_V	Ranking
1	1.2028	10.2083	7
2	0.9598	8.6043	9
3	1.1377	9.3682	8
4	1.1606	6.5453	12
5	1.1470	8.2748	10
6	1.1533	11.1543	4
7	1.0093	0.4456	20
8	1.0716	1.0975	19
9	1.0348	12.3597	2
10	1.2497	5.1175	13
11	0.8854	10.8292	5
12	0.9590	2.5526	15
13	1.0512	1.9572	18
14	1.1175	8.1495	11
15	1.0464	3.2304	14
16	**1.0425**	**12.8313**	**1**
17	1.0432	2.4524	16
18	1.1353	10.5126	6
19	0.9874	12.0493	3
20	1.0500	2.2493	17

Table 2 shows the comparative results of the active power performance index PI_P obtained using fast decoupled load flow solution and using RBF neural network. Fig 6 shows the graphical representation of the closeness of the results obtained

Novelty Journals

ISSN 2394-9678

International Journal of Novel Research in Electrical and Mechanical Engineering
Vol. 2, Issue 3, pp: (46-56), Month: September-December 2015, Available at: www.noveltyjournals.com

using FLDF and RBF-ANN. It is found that the network fits the desired data well for fifteen hidden neuron in the hidden layer and for a spread value of **σ** = 10.

Table 2 Active Power Performance Index using FDLF& RBF ANN for 14-BUS SYSTEM

Contingency Number (Line Outage No.)	PIP by FDLF	PIP by RBF ANN	Error
1	1.1693	1.2028	-0.0335
2	0.9807	0.9598	0.0209
3	1.1654	1.1377	0.0277
4	0.9999	1.1606	-0.1607
5	0.9820	1.1470	-0.1650
6	0.9640	1.1533	-0.1893
7	0.9915	1.0093	-0.0178
8	1.0747	1.0716	0.0031
9	0.9807	1.0348	-0.0541
10	1.2396	1.2497	-0.0101
11	1.0142	0.8854	0.1288
12	1.0127	0.9590	0.0537
13	1.0569	1.0512	0.0057
14	1.0072	1.1175	-0.1103
15	1.0759	1.0464	0.0295
16	1.0114	1.0425	-0.0311
17	1.0164	1.0432	-0.0268
18	1.0030	1.1353	-0.1323
19	1.0008	0.9874	0.0134
20	1.0076	1.0500	-0.0424

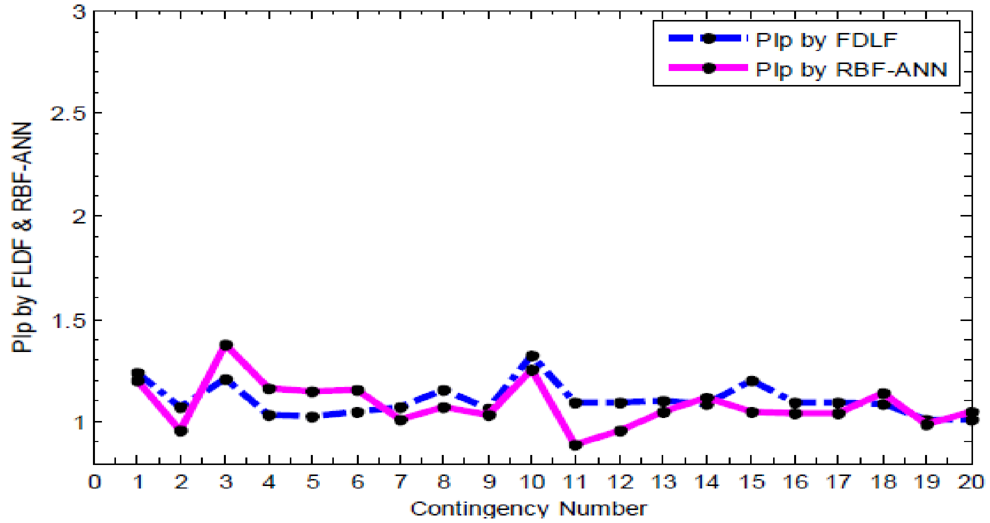

Fig. 6 Curves representing PIP obtained by FDLF & RBFN-ANN

Table 3 shows the comparative results of the reactive power performance index PI_V obtained using fast decoupled load flow solution and using RBF neural network. Fig 7 shows the graphical representation of the closeness of the results obtained using FLDF and RBF-ANN.

International Journal of Novel Research in Electrical and Mechanical Engineering
Vol. 2, Issue 3, pp: (46-56), Month: September-December 2015, Available at: www.noveltyjournals.com

Table 3 Reactive Power Performance Index using FDLF& RBF ANN for 14-BUS SYSTEM

Contingency Number (Line Outage No.)	PIV by FDLF	PIV by RBF ANN	Error
1	7.2973	10.2083	-2.9110
2	7.6650	8.6043	-0.9393
3	10.0117	9.3682	0.6435
4	7.3190	6.5453	0.7737
5	8.8736	8.2748	0.5988
6	13.2526	11.1543	2.0983
7	0.3520	0.4456	-0.0936
8	1.1707	1.0975	0.0732
9	10.5730	12.3597	-1.7867
10	1.6001	5.1175	-3.5174
11	9.5884	10.8292	-1.2408
12	1.8043	2.5526	-0.7483
13	1.3646	1.9572	-0.5926
14	10.4472	8.1495	2.2977
15	0.0798	3.2304	-3.1506
16	13.3418	12.8313	0.5105
17	2.3436	2.4524	-0.1088
18	10.5171	10.5126	0.0045
19	12.5492	12.0493	0.4999
20	2.2845	2.2493	0.0352

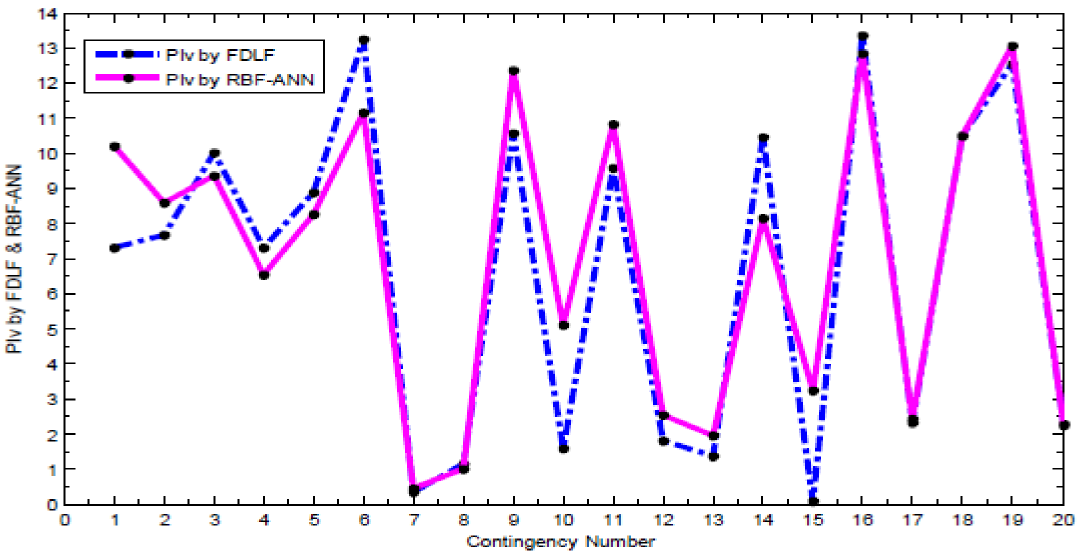

Fig. 7 Curves representing PIV obtained by FDLF & RBFN-ANN

Comparison of computation time: The computation time taken for contingency selection by FDLF and RBF-ANN method for the two bus systems have been detailed in Table 4.

Table 4 Computation time by FDLF& RBF- ANN

Bus System	By FDLF	By RBF
14-Bus	17.4 Sec	0.04 Sec

From the above table it can easily inferred that the contingency selection process by the use of RBF-ANN is much faster as compared to the FDLF method.

5. CONCLUSION

Radial basis function neural network based on Growing and Pruning algorithm has been proposed to evaluate static security in IEEE 14-bus, power systems. In the presented algorithm, optimum centers and radii of neurons in each class is obtained. Another advantage of this, the training of RBFN based on Growing and Pruning algorithm requires less computation time as compared to the Multi-Layer Perceptron model and the testing accuracy of RBFN has been made higher by applying Growing and Pruning algorithm. Test results of the sample system reveal that the proposed network involves less time and is suitable for online static assessment under uncertain loading conditions and is expected to perform similarly on even larger systems and handle even greater number of contingencies than reported here. The main focus has been to perform the fast contingency selection for the possible line contingencies by calculating the two types of performance indices namely PI_P and PI_V. It has been observed that when correct number of hidden neurons and bias centres are chosen for the network the results obtained using RBF-ANN are very much close to that which has been obtained using FDLF. But the neural network has the ability to perform the contingency selection for any loading and generating conditions once it gets correctly trained.

REFERENCES

[1] Shahidepour S M. Communication and control in electric systems. Wiley Inter science, John Wiley and Sons; 2003.

[2] Mansour Yakout, Vaahedi Ebrahim, El-Sharkawi Mohammed A. Neural network based pattern recognition for power system security assessment. IEEE Trans. Neural Network 1997; 8(4):942–50.

[3] Mahadev Pramod M, Christie Richard D. Envisioning power system data: vulnerability and severity representations for static security assessment. IEEE Trans Power System 1994; 9(4):1915–20.

[4] Singh S N, Srivastava S C. Improved contingency selection algorithm for voltage security analysis. Electr. Mach. Power System 1998; 26(8):855–71.

[5] Wollenberg BF. Automatic contingency selection. IEEE Trans Power Apparatus System 1979;PAS-98.

[6] Verma Kusum, Niazi K R. Supervised learning approach to online contingency screening and ranking in power systems. Int J Electr Power Energy System 2012;38(1):97–104.

[7] Scott B, Alsac O. Fast decoupled load flow. IEEE Trans. PAS 1974;PAS-93:859–69.

[8] Mori H, Tanaka H, Kanno J. A preconditioned fast decoupled power flow method for contingency screening. IEEE Trans Power Syst 1996;11:357–62.

[9] Brandwajn V, Lauby M L. Complete bounding method for AC contingency screening. IEEE Trans. Power System 1989; 4:724–9.

[10] James J W, Cote W, Chen-Ching liu C C. Voltage security assessment using generalized operational planning knowledge. IEEE Trans. Power System 1993; 8:28–37.

[11] Matos M A, Hatziargyriou N D, Pecaslopes J A. Multi-contingency steady state security evaluation using fuzzy clustering techniques. IEEE Trans. Power System 2000;15:177–83.

[12] Lo KL, Peng L J, Macqueen J F, Ekwue A O, Cheng D T Y. Fast real power contingency ranking using a counter – propagation network. IEEE Trans. Power System 1998;13(4):1259–64.

[13] P Niebur Dagmar, Germond Alain J. Power system security assessment using the Kohonen neural networks classifier. IEEE Trans. Power System 1992; 7(2):865–72.

[14] El-Keib A A, Ma X. Application of artificial neural networks in voltage stability assessment. IEEE Trans. Power System 1995;10(4):1890–6.

[15] Sobajic D I, Pao Y H. Artificial neural-net based dynamic security assessment for electric power system. IEEE Trans Power Syst 1989;4(1):220–8.

[16] Karami A. Power system transient stability margin estimation using neural networks. Int J Electr Power Energy Syst 2011;33(4):983–91.

[17] Lin Yu-Jen. Prevention of transient instability employing rules based on back propagation based ANN for series compensation. Int J Electr Power Energy Syst 2011;33(10):1776–83. ower Syst 1998;13(4):1259–64.