Electrical Power and Energy Systems 24 (2002) 387–395

ELECTRICAL POWER & ENERGY SYSTEMS

www.elsevier.com/locate/ijepes

Radial basis function networks for fast contingency ranking

D. Devaraj[a,*], B. Yegnanarayana[b], K. Ramar[a]

[a]Department of Electrical Engineering, Indian Institute of Technology, Madras 600036, India
[b]Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600036, India

Received 1 December 2000; revised 28 February 2001; accepted 2 April 2001

Abstract

This paper presents an artificial neural network-based approach for static-security assessment. The proposed approach uses radial basis function (RBF) networks to predict the system severity level following a given list of contingencies. The RBF networks are trained off-line to capture the nonlinear relationship between the pre-contingency line flows and the post-contingency severity index. A method based on *mutual information* is proposed for selecting the input features of the networks. Mutual information has the advantage of measuring the general relationship between the independent variables and the dependent variable as against the linear relationship measured by the correlation-based methods. The performance of the proposed approach is demonstrated through contingency ranking in IEEE 30-bus test system.

Keywords: Static-security; Radial basis function network; Contingency ranking; Mutual information

1. Introduction

Security assessment is one of the key issues in power system operation and planning. Assessment of static-security of a power system enables us to detect, through simulation, any potential line flow violations or an out-of-limit voltage following a given list of contingencies. For large power systems, due to the time constraint involved in real-time operation, those contingency cases which are potentially harmful to the system must be identified and detailed analysis are carried out for these cases alone. This process of selecting the contingencies according to their severity is referred to as contingency selection.

Over the years a number of algorithmic contingency selection methods have been proposed to speed up the process of contingency analysis. They can be broadly classified into two categories: *sensitivity-based ranking methods* [1,2] and *screening methods* [3,4]. Ranking methods utilize an approximate system-wide scalar performance index (PI) to quantify the severity of each contingency. The PI is used to rank all the contingencies. Screening methods use approximate or partial network solutions. Some of the solutions suggested include DC load flow, one iteration of AC load flow, local solution methods, etc. Sensitivity-based ranking methods are efficient but vulnerable to misrankings, while screening methods are accurate but inefficient.

Recently, artificial neural networks (ANN) have been proposed as tools for contingency screening and ranking [5–9]. Most of the authors have used feedforward neural networks with sigmoidal nonlinearities, for model development. Any continuous function can be approximated to within an arbitrary accuracy by carefully choosing the parameters in the network provided the network structure is sufficiently large. But the shortcoming of this network is that it takes long time for training. Also, feedforward network with sigmoidal activation function in the hidden nodes has no inherent ability to detect the outliers. Even though training is done in off-line, short training time is preferred as one may have to retrain the networks on a regular basis as the topology or the system condition changes. Outliers can occur in practice, because it is hard to produce a complete training set representing all possible operating conditions of a power system.

In this paper, we propose radial basis function (RBF) networks [10] to capture the nonlinear relationship between the pre-contingency system state and the post-contingency severity level following a contingency. RBF networks take less time for training and the distance-based activation function used in the hidden nodes gives the ability to detect the outliers during estimation.

Input feature selection plays an important role in ANN-based approaches. Various statistical methods have been

* Corresponding author. Tel.: +91-44-445-8356.
E-mail address: deva@neuron.iitm.ernet.in (D. Devaraj).

0142-0615/02/$ - see front matter
PII: S0142-0615(01)00041-2

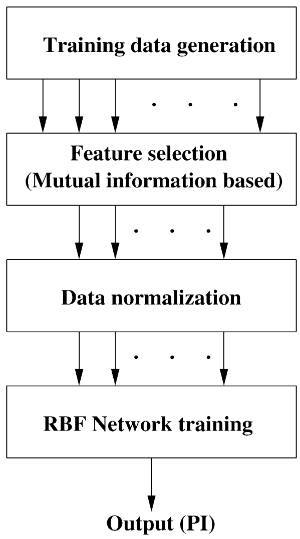

Fig. 1. Schematic representation of the learning stage.

proposed for feature selection in contingency selection models. Pang et al. [11] used class separation capability (secure/insecure) of the variables as the criterion to select the input features of their statistical security classifier. The following index was used to measure the *interclass-distance*:

$$F_i = \left| \frac{m_i^{(S)} - m_i^{(I)}}{\sigma_i^{(S)2} + \sigma_i^{(I)2}} \right| \tag{1}$$

where $m_i^{(S)}$ and $m_i^{(I)}$ are the mean of the variable i for the secure and insecure class and $\sigma_i^{(S)}$ and $\sigma_i^{(I)}$ are their variances. In Refs. [5,9] also the authors have used the same interclass-distance measure for feature selection in their ANN-based contingency selection models. Ghosh and Chowdhury [7] used the *correlation coefficient* between the input variables and the output severity measure to select the input features for their feedforward neural network model.

The interclass-distance measure assumes Gaussianity in the input domain. If a variable x has normal or Gaussian distribution, its distribution is completely characterized by its mean and variance. If this assumption is not true, serious errors may occur in feature selection. Also, methods based on linear dependence (like correlation) cannot measure arbitrary relations between the independent and dependent variables. In this paper, we propose *mutual information* [12] between the dependent and independent variables as the criterion to select the input features of the RBF networks.

2. Proposed methodology for contingency selection

The proposed method for contingency selection is based on RBF neural networks. The objective is to estimate the severity levels for each contingency. The study presented in this paper focuses on single line outages. The severity of a contingency to line overload is expressed by the following scalar PI:

$$\mathrm{PI}_{\mathrm{MVA}} = \sum_{l=1}^{N_l} \left[\frac{S_l^{\mathrm{post}}}{S_l^{\mathrm{max}}} \right]^{2m} \tag{2}$$

where S_l^{post} is the post-contingency MVA flow in line l, S_l^{max} the MVA rating of line l, N_l the no of lines in the system and m is the integer exponent.

The contingencies can then be ranked according to the order of their severity. Following a contingency, any line which is overloaded will make a contribution of greater than unity to the PI, whereas a line whose flow is below its rating will make a contribution of less than unity. A small value of m in Eq. (2) would result in 'masking effect' and a very high value of m may cause the final ordering to become worse due to increased nonlinearity [7]. For the IEEE 30-bus system considered in this paper, we have fixed the value of m as 5. These PI values will be used as target values for the ANN during training.

It is possible to train a single network to estimate the PI values of all the contingencies by taking the system state variables as the input and the system severity levels as the output of the network. But, as the dimension of the input vector increases, the number of basis functions (hidden layer nodes) required to approximate the given function rises exponentially [10]. Such large networks are inefficient and sensitive to over fitting and exhibit poor performances. So, the problem of estimating the post-contingency severity level using ANN is decomposed into several networks, with each one dealing with one contingency. While training a network dedicated to a contingency, the localized nature of contingencies could be exploited in the form of dimension reduction through proper feature selection.

The schematic representation of the learning stage of the model for PI estimation is shown in Fig. 1. For model development, a large number of training data is generated through off-line power system simulation. Pre-contingency state power flows are the input to the models and the PI value following a contingency is the output of the model. A mutual information-based feature selection technique is applied to identify the relevant attributes from the set of system state variables for each contingency model. By selecting only the relevant variables as input features and excluding irrelevant ones, higher performance is expected with smaller computational effort. The selected input features and the output are normalized between 0 and 1 and presented to the RBF networks for training. Once the networks are trained, they are ready for contingency ranking at various load conditions. The details of mutual information-based feature selection and the architecture and training of RBF network are presented in the following sections.

3. Feature selection

One of the important issues in ANN-based approach is the proper selection of input features to the model. The problem of feature selection is stated as follows: given an initial set of n features, find the subset with $k < n$ features that is 'maximally informative' about the output.

As most of the contingencies are localized in nature, all the variables in the input vector may not exert equal influence on the post-contingency PI values. Irrelevant and redundant attributes in the input not only complicate the network structure, but also degrade the performance of the networks. By selecting only the relevant variables as input features and excluding irrelevant ones, higher performance is expected with smaller computational effort. This section presents the details of input feature selection based on mutual information.

3.1. Definition of mutual information

Consider a stochastic system with input X and output Y. Let the discrete variable X has N_x possible values and Y has N_y possible values. Now the initial uncertainty about Y is given by the entropy $H(Y)$ which is defined as [13]

$$H(Y) = -\sum_{j=1}^{N_y} p_j \log p_j \tag{3}$$

where $p_j = P(Y = y_j)$ is the probability of occurrence of the event $Y = y_j$. The amount of uncertainty remaining about the system output Y after knowing the input X is given by the conditional entropy $H(Y|X)$ which is defined as

$$H(Y|X) = -\sum_{i=1}^{N_x} p_i \left(\sum_{j=1}^{N_y} p_{ji} \log p_{ji} \right) \tag{4}$$

where $p_i = P(X = x_i)$ is the probability of occurrence of the event $X = x_i$ and p_{ji} is the conditional probability for output y_j given the input x_i. Now the difference $H(Y) - H(Y|X)$ represents the uncertainty about the system output that is resolved by knowing the input. This quantity is called the mutual information between the random variables X and Y. Denoting it by $I(Y;X)$, we may thus write

$$I(Y;X) = H(Y) - H(Y|X) \tag{5}$$

The mutual information is therefore the amount by which the knowledge provided by the feature vector decreases the uncertainty about the output.

3.2. Selecting the features with the mutual information

Mutual information between two random variables measures the amount of common information contained in these variables. The problem of selecting input features which contain much of the information of output can be solved by computing the mutual information between each variable and output and selecting those variables having higher mutual information values.

To compute mutual information the probability distribution function of variables is needed which in practice is not known and the best we can do is to use the histogram of the data. The steps involved in calculating the mutual information from the histogram of the data are given below:

1. Arrange all the PI values in the descending order and divide them into N_y classes equally.
2. Calculate the initial entropy using Eq. (3).
3. Sort the data points in the first input variable in the descending order. Divide the sorted patterns into N_x groups equally.
4. Compute the conditional entropy, given the input vector 1 using Eq. (4) and calculate the mutual information using Eq. (5).
5. Repeat steps 3 and 4 for the remaining variables also.

4. RBF networks

RBF network is a class of single hidden layer feed forward neural network [10,14]. The input nodes pass the input to the hidden nodes directly and the first layer connections are not weighted. The transfer functions in the hidden nodes are similar to the multivariate Gaussian density function

$$\phi_j(x) = \exp\left(-\frac{\|x - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2} \right) \tag{6}$$

where $\boldsymbol{\mu}_j$ is the vector determining the center of basis function ϕ_j and σ_j are their widths. Each RBF unit has a significant activation over a specific region determined by $\boldsymbol{\mu}_j$ and σ_j; thus each RBF represents a unique local neighborhood in the input space.

The connections in the second layer are weighted and the output nodes are linear summation units. The value of the kth output node y_k is given by

$$y_k(x) = \sum_{j=1}^{h} w_{kj}\phi_j(x) + w_{k0} \tag{7}$$

where w_{kj} is the connection weight between the output and jth hidden node and w_{k0} is the bias term.

The training in RBF networks can be decomposed quite naturally and the learning is done in three sequential stages as against the single optimization procedure followed in backpropagation network training. The first stage of the learning consists of determining the unit centers $\boldsymbol{\mu}_j$ by the K-means clustering algorithm, (see Appendix A). Next, we determine the unit widths σ_j using a heuristic approach that ensures the smoothness and continuity of the fitted function. The width of any hidden node is taken as the maximum Euclidean distance between the identified centers. Finally, the weights of the second layer connections are determined by linear regression using a least-squares objective function.

RBF networks can be viewed as an alternative tool for learning in neural networks. While RBF networks exhibit

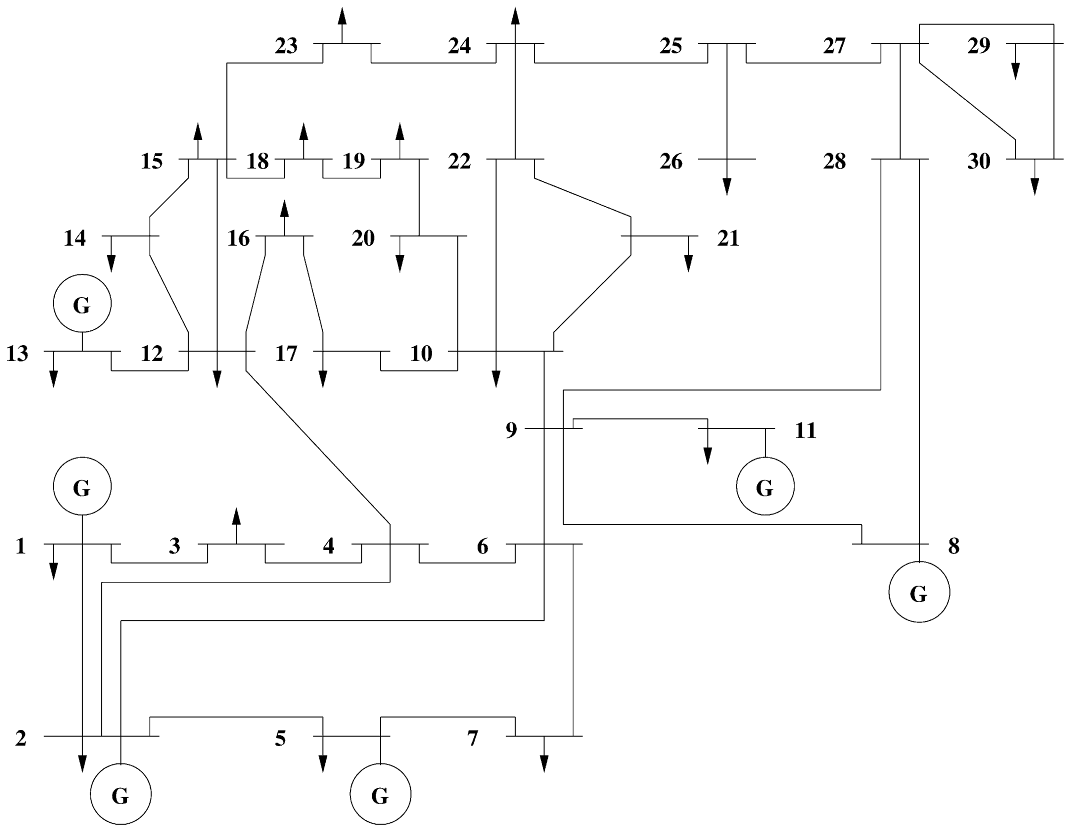

Fig. 2. IEEE 30-bus test system.

the same properties as backpropagation networks such as generalization ability and robustness, they also have the additional advantage of fast learning and ability to detect outliers during estimation.

5. Simulation results

To demonstrate the applicability of the proposed RBF network-based approach for contingency ranking, IEEE 30-bus system shown in Fig. 2 is selected as the test system. The transmission line parameters, generator ratings and base load condition are given in Ref. [15]. The system has six generators and 41 transmission lines. Forty single line outages except lines 25–26 are chosen for contingency analysis. The various steps involved in developing and evaluating the system for contingency ranking and selection are presented below.

5.1. Training data generation

In machine learning approaches training data is the only available information to build the model, and so they should represent the complete operating conditions of the system. For contingency analysis model development, input–output patterns are generated as per the following procedure:

(a) First, a range of situations is generated by randomly perturbing the load at all buses between 70 and 140% of their base case value and by adjusting the generator output in proportion to the output in the base case condition.
(b) For each load-generation pattern pre-contingency line flows are obtained by solving the load flow equations using Newton–Raphson algorithm.
(c) Also, for each load-generation pattern, the single line-outages specified in the contingency list are simulated sequentially and their PI values evaluated by conducting AC load flow.

Based on the above simulation procedure, a training set consisting of 750 input–output pairs was created. Additionally, a test set of 250 data pairs was generated in order to evaluate the learning and generalization abilities of the networks. In all the 1000 patterns it was noticed that the PI_{MVA} values are very low for 31 contingencies. These contingencies are not considered for model development and the models are developed for the remaining nine cases alone.

5.2. Feature selection

Pre-contingency line flows in all the lines are chosen as the input to the networks and they are 41 in number. The PI corresponding to a contingency is the output of the network.

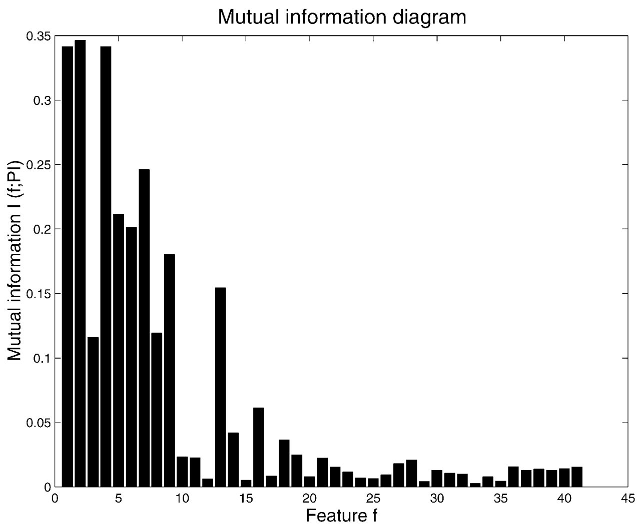

Fig. 3. Mutual information for variables in model 1–3.

To select the input features of each model, input feature space is partitioned into five and the output is divided into three groups. Mutual information of each variable with respect to the output is evaluated following the steps given in Section 3.2. For illustration, the mutual information between the input variables and the output for contingency model 1–3 is shown through a bar graph in Fig. 3. From this figure it is evident that only a few variables are having significant information to estimate the PI, and the remaining variables have very less amount of information. The first few variables that have high mutual information value are selected as features to train the networks. The features selected for satisfactorily training the networks are given in Table 1.

5.3. Data normalization

The first stage of RBF network learning is the identification of the cluster centers through *K*-means clustering algorithm, which uses Euclidean distance as a measure of dissimilarity. Distance norms are sensitive to variations in the numerical ranges of the different features. For example, the Euclidean distance assigns more weighting to features with wide ranges than to those with narrow ranges. To overcome this problem, input data is normalized before presenting it to the clustering algorithm. The input data is normalized between 0 and 1 using

$$x_n = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} \tag{8}$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum of the variable x. Similarly, output data is also normalized between 0 and 1.

5.4. Network training and evaluation

For the prediction of PI for each line outage, separate RBF network is trained. The selected variables after normalization are presented to the network. Twenty iterations of the clustering algorithm followed by linear regression are performed to estimate the parameters of the network. As the value of h is not known in advance, a trial-and-error procedure is followed to select the optimum number of basis functions.

After training, the generalization performance of the networks are evaluated with the 250 test data. The results of training and testing phase for all the nine models are presented in Table 1. The results clearly show that the training of the RBF networks has been successful and the correct estimation of PI has been achieved by the RBF network even for previously unseen data.

Table 2 presents the PI values estimated for one particular load condition with the ranking of the contingencies given in the parenthesis. For comparison, the actual values of PI calculated from AC load flow study are also presented. The result shows the agreement between the actual ranking and the ranking based on the output of the RBF networks.

5.5. Comparison with multilayer perceptron network

To compare the performance of the proposed RBF network-based approach with the commonly used neural network architecture, multilayer perceptron (MLP) networks are developed to estimate the PI values. The networks are trained with the conjugate gradient algorithm [10] to reach the same error level achieved by the RBF

Table 1
Results of RBF networks

S. No.	Line outage	Selected features ($S_l; l = 1, 2, \ldots, 41$)	No. of basis functions	Training time (s)	Testing error (mse)
1	1-2	1,2,4,5	15	0.50	4.4×10^{-5}
2	1-3	1,2,4,7	15	0.50	9.7×10^{-6}
3	3-4	1,2,4,7	15	0.50	1.6×10^{-5}
4	2-5	1,5,8,9	20	0.70	5.2×10^{-5}
5	4-6	1,2,4,6,7,13	20	0.75	2.6×10^{-4}
6	6-7	1,5,8,9	20	0.65	1.5×10^{-4}
7	9-10	12,14,18,21,27,28	25	1.00	3.5×10^{-4}
8	19-20	2,14,18,22,23,24,25	35	1.40	3.9×10^{-4}
9	28-27	31,36,37,38,39,41	25	0.95	5.8×10^{-4}

Table 2
PI estimation

Line outage	PI value	
	RBFN output	Load flow result
1-2	16.00 (3)	15.92 (3)
1-3	6.40 (4)	6.41 (4)
3-4	5.63 (5)	5.76 (5)
2-5	40.27 (2)	40.73 (2)
4-6	1.56 (8)	1.58 (8)
6-7	1.72 (7)	1.75 (7)
9-10	2.26 (6)	2.17 (6)
19-20	0.47 (9)	0.53 (9)
28-27	44.40 (1)	48.54 (1)

Table 3
Results of MLP networks

Line outage	Hidden nodes	Training time (s)	Testing error (mse)
1-2	8	9.1	9.6×10^{-5}
1-3	6	5.3	9.6×10^{-5}
3-4	6	5.4	9.6×10^{-5}
2-5	8	17.7	9.3×10^{-5}
4-6	8	18.3	3.0×10^{-4}
6-7	6	15.5	1.5×10^{-4}
9-10	6	16.0	4.4×10^{-4}
19-20	6	16.2	2.0×10^{-4}
28-27	6	16.1	6.2×10^{-4}

networks during training. After training, the networks are tested with the test data. The results of training and testing for the MLP networks are presented in Table 3. Based on the information presented in Tables 1 and 3, it is observed that RBF networks take less time for training, but they require more number of hidden nodes as compared to MLP networks. Apart from that RBF network exhibits better generalization performance than the MLP network in most of the cases.

6. Conclusions

This paper has presented a neural network-based fast contingency selection method for power system static-security assessment. A set of RBF networks has been trained to map the nonlinear relationship between the pre-contingency operating state and the post-contingency security indices. An effective feature selection method has been proposed to reduce the dimension of the input patterns. Simulation results on the IEEE 30-bus test system shows the proposed RBF network-based approach that provides an accurate estimation of post-contingency PI values for various operating conditions. When compared with MLP networks trained with backpropagation algorithm, the proposed approach significantly reduces the development time with improved estimation accuracy.

Appendix A

*A.1. **K**-means clustering algorithm*

The algorithm partition the data points x^n, $n = 1, 2, \ldots, N$, into K disjoint clusters C_j containing N_j data points, in such a way as to minimize the sum-of-squares clustering function given by

$$J = \sum_{j=1}^{K} \sum_{n \in C_j} \|x^n - \boldsymbol{\mu}_j\|^2 \tag{A1}$$

where $\boldsymbol{\mu}_j$ is the mean of the data points in cluster C_j.

The algorithm iteratively determines the cluster centers $\boldsymbol{\mu}_j$ as follows:

1. Initialize the cluster centers $\boldsymbol{\mu}_j$, $j = 1, 2, \cdots, K$ by randomly selecting K data points from among all of the data points.
2. Generate a new partition by assigning each pattern to its closest cluster center.
3. Compute new cluster centers as the centroids of the clusters.
4. Compute the cost function according to Eq. (A1). Stop if either it is below a certain tolerance value or its improvement over previous iteration is below a certain threshold, otherwise go to step 2.

References

[1] Ejebe GC, Wollenberg BF. Automatic contingency selection. IEEE Trans Power Apparatus Syst 1979;PAS-98(1):97–109.
[2] Irisarri G, Sasson AM, Levner D. Automatic contingency selection for on-line security analysis-real-time tests. IEEE Trans Power Apparatus Syst 1979;PAS-98(5):1552–9.
[3] Brandwajn V. Efficient bounding method for linear contingency analysis. IEEE Trans Power Syst 1988;3(1):38–43.
[4] Brandwajn V, Lauby MG. Complete bounding method for A.C. contingency screening. IEEE Trans Power Syst 1989;4(2):724–9.
[5] Weerasooriya S, El-Sharkawi MA, Damborg M, Marks RJ. Towards static-security assessment of a large-scale power system using neural networks. IEE Proc-C 1992;139(1):64–70.
[6] Ray D, Chakravorti S, Mukherjee PK. An application of artificial neural network for power system security evaluation. Inst Engr (India) 1994;75:131–5.
[7] Ghosh S, Chowdhury BH. Design of artificial neural network for fast line flow contingency ranking. Electrical Power Energy 1996;18(5):271–7.
[8] Testa A, Menniti D, Picardi C, Sorrent N. Steady state security prediction in presence of load uncertainty. Eur Trans Electric Power 1998;8(2):97–104.
[9] Lo KL, Peng LJ, Macqueen JF, Ekwue AO, Cheng DTY. Fast real power contingency ranking using a counterpropagation network. IEEE Trans Power Syst 1998;13:1259–64.
[10] Bishop CM. Neural networks for pattern recognition. Oxford: Oxford University Press, 1995.
[11] Pang CK, Prabhakara FS, El-Abiad AH. Security evaluation in power systems using pattern recognition. IEEE Trans Power Apparatus Syst 1974;PAS-93(2):969–76.

[12] Battiti R. Using mutual information for selecting features in supervised neural net learning. IEEE Trans Neural Networks 1994;5(4):537–50.

[13] Haykin S. Neural networks — a comprehensive foundation. New Jersey: Prentice Hall, 1999.

[14] Yegnanarayana B. Artificial neural networks. India: Prentice Hall, 1999.

[15] Alsac O, Scott B. Optimal load flow with steady state security. IEEE Trans Power Syst 1974;PAS-93:745–51.