J. Indian Inst. Sci., July–Aug. 2006, **86**, 327–342

Power system static security assessment using self-organizing neural network

K. S. SWARUP AND P. BRITTO CORTHIS
Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600 036, India.
email: swarup@ee.iitm.ac.in, shantiswarup@ieee.org; Phone: 91-044-2257-4440; Fax: 91-44-2257-4402.

Received on May 26, 2004; Revised on March 22, 2006.

Abstract

Artificial neural network approach to the problem of static security assessment of power system is presented. This paper utilizes the artificial neural net of Kohonen's self-organizing feature map (SOFM) technique that transforms input patterns into neurons on the two-dimensional grid to classify the secure/insecure status of the power system. SOFM uses the line flows under different component cases as inputs and self-organizes to obtain the cluster of the components based on their loading limits. The output of SOFM provides information about the violation of the constraints from which the operating state of the power system can be identified, which can be classified as secure or insecure. The proposed method of security assessment was initially demonstrated for a model 3 generator 6-bus system and later extended to IEEE-14, -30 and -57 bus systems.

Keywords: Power system static security assessment, artificial neural networks, self-organizing feature map (SOFM), classification and clustering.

1. Introduction

Large interconnected power systems with dispersed and geographically isolated generators and load constitute a majority of present power network. The present-day power systems are dynamic in nature, with network topology changing frequently with load demand. With increase in load demand, the power network is loaded to its limits thus making it susceptible to blackout under minor/major disturbances. In order to operate the power system economically, the state of the system has to be identified as secure/insecure.

Power system security can be defined to remain secure without serious consequences to any one of a preselected list of credible disturbances or contingencies. Security assessment (SA) is the analysis performed to determine whether, and to what extent, a power system is reasonably safe from serious interference to its operation. In other words, it is the process of determining if the power system is in a secure or alert (insecure) state, the secure state implying if the load is satisfied and no limit violation will occur under present operating conditions and in the presence of unforeseen contingencies (i.e. outages of one or several lines, transformers or generators). The alert (or emergency) state implies that some limits

*Author for correspondence.

are violated and/or the load demand cannot be met and corrective action must be taken to bring the power system back to the secure state.

2. Static security assessment

One of the main aspects of power system security is static security. Static security is defined as the ability of the system to reach a state within the specified secure region following a contingency. The standard approach to the security assessment problem is to perform the static security analysis followed by dynamic security analysis. Static security analysis evaluates the post-contingent steady state of the system neglecting the transient behaviour and any other time-dependent variations due to changes in load generation conditions. On the other hand, dynamic security analysis evaluates the time-dependent transition from the pre- to the post-contingent state. Most of the energy management systems perform only static security analysis and hence the focus of this paper is on static security assessment.

3. Conventional techniques

In conventional practice, security assessment is obtained by analytically modeling the network and solving load flow equations repeatedly for all the prescribed outages, one contingency at a time. This normal practice is not entirely satisfactory because the computations are lengthy and are particularly so at load values for which the system is in fact insecure against the occurrences of certain contingencies. To reduce the above computational effort of the security assessment most energy management systems use one or more security assessment predictors such as sensitivity matrix, distribution factors, fast decoupled load flows, or performance indicators to reduce the number of critical contingencies to be calculated.

These analytical techniques are usually time consuming and therefore are not always suitable for real-time applications. Also, these methods suffer from the problem of misclassification or/and false alarm. Misclassification arises when an active contingency is classified as critical. Expert and fuzzy system-based methods are fast; however, they lack versatility as many expert and fuzzy system-based rules are system specific. With recent advancements in information processing and learning techniques, artificial neural network (ANN)-based methods for security assessment are a viable alternative.

4. Application of ANN to static security analysis

Several ANN approaches have been proposed as alternative methods for security assessment in power-system operations. Neural network methodology is applied in areas where conventional techniques have not achieved the desired speed and accuracy. One such area is the classification of system security status. Once the neural network is properly trained, it can interpolate patterns using a limited amount of input data. Since neural networks are quick in response time and can be easily adapted, they become excellent for online application. Considering that the neural networks are good in interpolation but not in extrapolation, training sets have to represent the different states of the power system. This means that they need to comprise the complete pattern space of the secure and insecure power system operation. A large training set is necessary as it is not known how much input data would

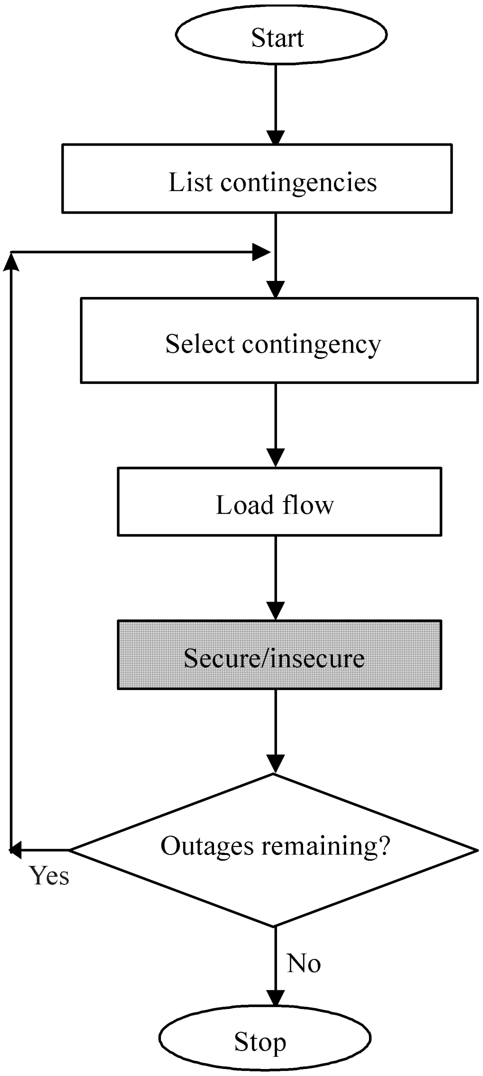


FIG. 1. Conventional approach to security assessment.

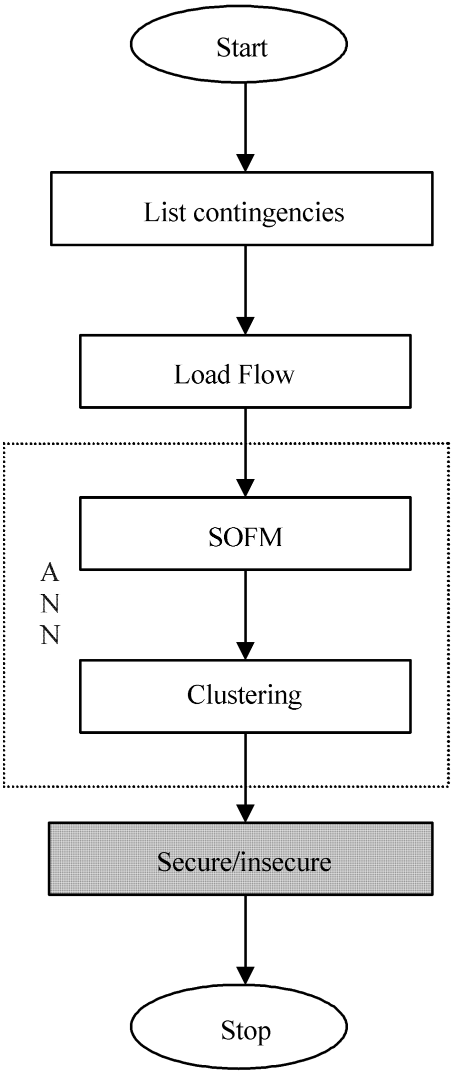


FIG. 2. ANN approach to security assessment.

provide the best results in the output. Most neural network design procedures, whether supervised or unsupervised, are based on a procedure which optimizes the neural network performance and not on those which maximize the accuracy of the security prediction.

Figure 1 shows the conventional approach to security assessment, where the load flow study has to be repeated for selected contingencies. The process is time consuming. The contingencies are ranked and ordered according to their severity.

Figure 2 shows the ANN approach to security assessment. It is shown how the self-organizing map and clustering eliminate the repetitive load flow computation required in the traditional method. This is the important advantage of the ANN approach. Better classification and clustering are attained by the self-organizing feature map (SOFM).

5. Neural network architectures

The most popular choice of neural network is the multilayered perceptron (MLP) which can be trained offline and used online [1]. This is a good method for reducing the problem of selecting the training set and for selecting the inputs. MLP is based on pattern matching technique where a group of neural networks is trained to classify the secure and insecure status of the power system for specific contingencies based on the pre-contingency system variables.

An unsupervised adaptive resonance theory (ART) like neural network is used for the clustering of the input vectors [2]. For this neural network, each cluster has an adaptively determined center, the typical operating state and a radius which has to be determined in advance, usually through experimentation.

Hopfield network is also used for the prediction of the class of violation for post-contingency bus voltages, voltage drops and line flows [3]. The security problem is being reduced to optimization problem when the Hopfield networks are used. The applicability of a multilayered neural network trained with back propagation algorithm to access static security assessment is also in use but it requires huge amount of training data and is extremely time consuming [4]. Owing to the high dimensionality of this function, accuracy is highly dependent on the number of training points. Each contingency has to be viewed as a different function. Hopfield net also faces the same problem as discussed above.

A different quantization approach for identifying the security region is the SOFM [5, 6]. Each neuron has one weight vector which represents the center of a class of operating states. This weight vector is interpreted as a typical operating state which in this application is given by the line powers. This unsupervised training process constructs intermediate classes which do not represent any training vector but may classify unknown system states, thus generalizing information on known states. In addition to the class information, the two-dimensional self-organizing map gives a two-dimensional representation of the m-dimensional operating space. The operating space is presented on the map by secure and insecure regions.

6. SOFM

SOFM ANN is simplified model of the feature to localized region mapping of the brain from which it derives its name. It is a competitive, self-organizing network which learns from the environment without the aid of the teacher. The architecture is quite simple. It consists of the group of geometrically organized neurons in one, two, three or even higher dimensions. The one-dimensional network is a single layer of units that are arranged in a row. In the two-dimensional network cases, the units are arranged as a lattice array and so on for higher dimensions. Figure 3 shows the typical architecture of SOFM. The process of mapping from input (one dimension) to output (two dimensions) surfaces is shown in Fig. 4.

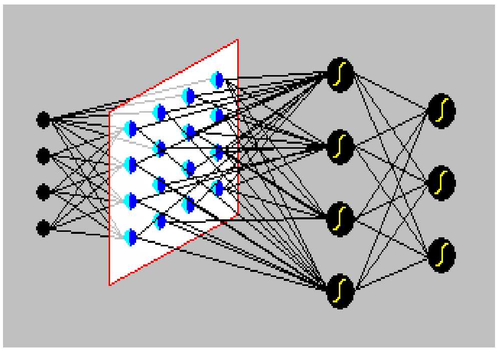


FIG. 3. Typical architecture of SOFM.

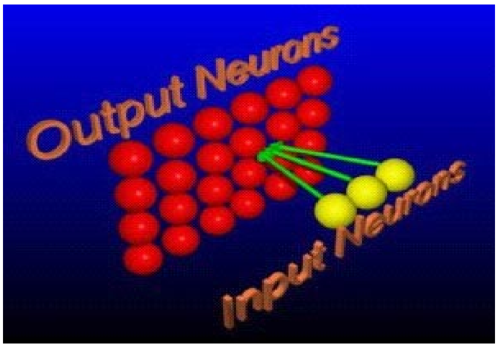


FIG. 4. Mapping of SOFM.

This network learns without supervision; that means the topological order of the input data is not necessarily known a priori. This network architecture was first developed by Kohonen [7]. In his approach, Kohonen proposed a fully laterally connected network with distance related feedback synapses showing inhibition and excitation features, where it was shown that the simplified model, described and used below, exhibits the same self-organization properties with a significantly lower computation efforts. The remarkable property of the Kohenon network is that this neural network learns the topology of the n-dimensional vector space with a given set of training vectors. The set of input vectors, which may amount to several thousands of n-dimensions, will be represented after learning by a smaller number m × n-dimensional weight vectors.

7. Algorithm for SOFM

The following procedure implements the method adopted for unsupervised self-organization.

1. Initialize the weights from m inputs to n output units to small random values. Initialize the size of the neighborhood region $R(0)$.
2. Present a new input vector a.
3. Compute the distance d_i between the input and the weight on each output unit i:

$$d_i = \sum_{j=1}^{M} [a_j(t) - w_{ij}(t)]^2, \text{ for } i = 1, 2, \ldots, n, \tag{1}$$

   where $a_j(t)$ is the input to the jth input unit at time t and $a_{ij}(t)$ is the weight from the jth input unit to the ith output unit.

4. Select the output unit k with minimum distance defined by $k = \min_i(d_i)$.
5. Update weight to node k and its neighbors

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\{a_j(t) - w_{ij}(t)\} \text{ for } i \; R_k(t) \text{ and } j = 1, 2, \ldots, m, \tag{2}$$

where $\eta(t)$ is the learning rate parameter ($0 < \eta(t) < 1$) that decreases with time. $R_k(t)$ gives the neighborhood region around the node k at time t.

6. Repeat steps 2 through 5 for all input several times.

8. Algorithm for classification

In the classification phase, the network maps a vector with unknown features to the cluster where its closet neighbors have been mapped to.

The algorithm consists of the following:

1. Present the input vector x.
2. Select the neuron c with the weight vector closest to the input vector.

$$\| w_c(t_{\max}) - a \| = \min \| w_{ij}(t_{\max}) - a \| \tag{3}$$

9. Implementation aspects

9.1. *Simulation data*

Figure 5 illustrates the structure of the six-bus eleven line system. For 11 lines, the system state is defined by 11 complex quantities (real and reactive power) totaling 22 components corresponding to 11 active and 11 reactive line power flows. These input vectors were obtained by offline load flow simulations.

9.2. *Selection of number of neurons*

The number of neurons depends on the number of contingencies taken. Here 14 number of single lines are taken into consideration.

They are,

Number of generator outages	:	3
Number of line outages	:	11
One base case	:	1
Total number of neurons	:	15

To represent in a square matrix, 4 × 4 matrix is assumed and therefore 16 neurons are taken to represent the given power network.

10. Simulation data

A simulator for the Kohonen learning algorithm was developed in MATLAB. All parameters can be defined and changed interactively; the size of the input vector and of the Kohonen network, the neighborhood function and their decrease with the learning steps. Vectors are drawn randomly from the set of input training vectors and are presented several times as inputs to a two-dimensional Kohonen network of size of 4 × 4. Neighborhood and learning rate decrease exponentially with the number of training steps (Fig. 6). The adaptation of the weight vectors with respect to the learning step is also illustrated in Fig. 6. After nearly 20,000 steps of unsupervised learning, the network is already organized. i.e. the weight vec-

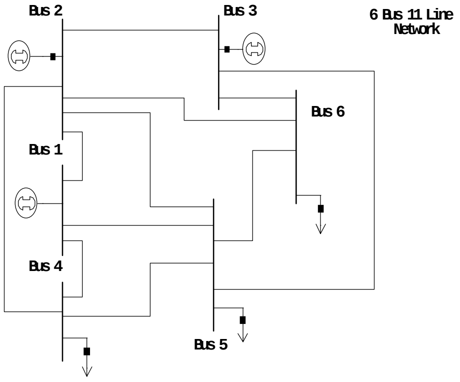


FIG. 5. Model six-bus–eleven line system considered.

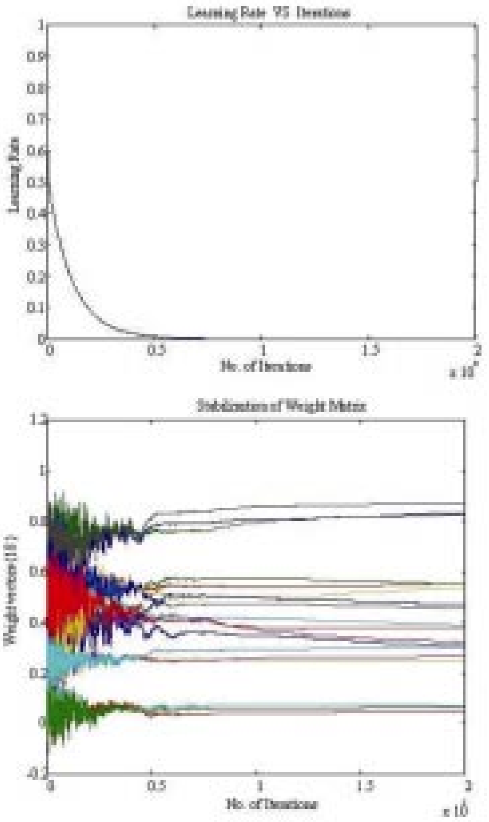


FIG. 6. Stabilization of weight vectors as the learning rate decreases.

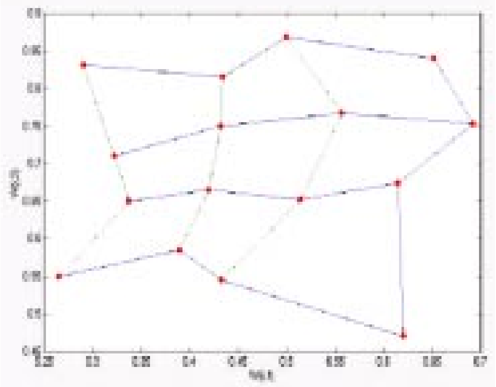


FIG. 7. Self-organization of neurons.

tors do not change significantly. Weight vectors whose norm is represented by a straight line correspond to those neurons which are never chosen as the nearest weight vector for any training vector and which therefore do not change any more once the neighborhood order function is smaller than 1. Figure 7 provides the plot of the self-organization of the neurons, where the neurons are located at the x and y components of its weight vector. The structure of the grid represents the uniform distribution of the input vectors. It is evident that the classification is not perfect in this case and a more regular pattern can be obtained for larger training sets and more slowly decreasing neighborhood function.

11. Classification of static security states

In the present work, the power system static security index proposed classifies the power system states based on the condition, secure or insecure, and in turn on the operating state, i.e. whether normal, alert, emergency or in network splitting. This is calculated by calling an output neuron where the estimated index is assigned. After an output neuron on the grid responds to an input pattern, the output calls the estimated index or performance index [14] which is calculated as follows.

11.1. *Estimated index*

$$EI_{MW} = \sum_{i=1}^{N_L} \left(\frac{W_{li}}{2n} \right) \left(\frac{P_L}{P_L^{\lim}} \right)^{2n} \tag{4}$$

where,

P_L	=	MW flow of line l
$P_L^{\lim}$	=	MW capacity of line l
N_L	=	Number of lines of the system
W_{li}	=	Real non-negative weighting factor (= 1)
n	=	Exponent of penalty function (= 2 preferred)

11.2. *Security index*

The proposed method is based on the fact that the weight vector or reference vector represents the inherent input pattern corresponding to the output neuron. In other words, the features of the output neuron are expressed in the weights between the output neuron and the input units. The security index is obtained using the weight vectors after finishing the learning process to calculate the index μ [15].

The algorithm to calculate the security index is as follows:

Step 1: Prepare learning patterns for SOFM and the estimated index for each learning pattern.

Step 2: Carry out SOFM.

Step 3: Calculate μ^i with the weight vector at each output neuron i.

Step4: Give an unknown input pattern to the constructed network. Then, the evaluated output neuron i* that is closest to the input pattern.

Step 5: Call μ_e^i* at output neuron i*.

$$\mu_e^{i^*} = \frac{1}{N_i} \sum \mu_k^i, \tag{5}$$

where, $\mu_e^{i^*}$ is the security index at output neuron, i; N_i, the number of input patterns classified into output neuron i and μ_k^i, the index of input pattern k classified into output neuron i.

The calculated security index has a small value, when all line flows are within their limits, and a high value when there are line overloads. Thus, it provides a measure of the severity of line overloads for a given state of the power system.

11.3. *Power system static security levels and their security index*

To quantify the concept of secure and insecure operating states, four security levels have been determined—normal, alert, emergency 1, and emergency 2 [16]. For these operating states, the values of the security index are also given (Table I).

12. Simulation results

Simulation results are obtained by the proposed scheme for different power system networks to assess the security level of the network. The proposed method is trained and tested on 6-bus system.

Table I
Severity level and security index

Severity level	Line flow (%)	Security index
Normal (N)	< 100	0.25
Alert (A)	100–150	0.25–1.27
Emergency 1 (E1) (correctable)	150–200	1.27–4.00
Emergency 2 (E2) (non-correctable)	> 200	4.00

Table II
Neural network parameters

Input dimensions	22
Kohonen neurons	16
Initial learning rate	1
Training patterns	330

12.1. *Classification of known training vectors*

The neural network is trained using the database obtained from 100% loading and the same is tested for classification of the loading patterns. Table II shows the neural network test results, Table III, the classification of training patterns, and Table IV, the classification results for 15 training vectors. For Table IV, the following notations have been used: All generators are named with letter G followed by the bus number to which the generator is connected. All lines with letter L followed by the pair of bus numbers, to which they are connected. Therefore, G1 denotes generator outage on Bus 1 and L12, the line outage between Buses 1 and 2. The ‘Base Case’ (BC) denotes the case without any line or generator outages. ANN is trained for this base case and tested for 100% loading and various other loadings above and below 100%.

Table IV shows the classification of 15 cases by 16 neurons. Initially, before training each neuron represents one contingency at a time. For example, Neuron 1 stands for BC loading and Neuron 2 for G1 outage, etc. After training, the contingencies having similar line loading patterns are grouped together to each of the neurons. For example, Neuron 6 has four contingencies grouped together such as BC, L23, L45 and L56, and Neuron 16 two contingencies such as G3 and L36, etc. Figure 8 shows the cluster map for the classification of 15 trained vectors. Four different clusters can be distinguished. Neurons containing empty cases do not classify any of the training vectors. That means their weight vector has never been chosen as closest vector to any of the training vectors. Figure 9 shows the leg-

Table III
Classification of training patterns

Neuron number	No. of patterns responded	Neuron number	No. of patterns responded
1	2	6	4
2	1	12	1
4	1	13	3
5	1	16	2

Table IV
Classification of 15 cases by 16 neurons

1 G1 L35	2 L25	3	4 L14
5 L26	6 BC L23 **L45, L56**	7	8
9	10	11	12 L15
13 G2 L12 L24	14	15	16 G3 L36

▒ #1	▨ #2	#3	▦ #4
▨ #5	░ #6	#7	#8
#9	#10	#11	▦ #12
▒ #13	#14	#15	▦ #16

FIG. 8. Kohonen's feature map.

░	Normal (N)
▨	Alert (A)
▒	Emergency 1 (E1) [Correctable]
▦	Emergency 2 (E2) [Non-correctable]

FIG. 9. Legend showing the states of the neurons.

ends of the neuron of the Kohonen map. Figure 10 shows the neuron number and its corresponding security index.

From the cluster map, the states of the power system are very clear. Normal state is represented by Neuron 6 and in this state the contingencies that come are BC, L23, L45, and L56. Alert state is represented by Neurons 2 and 5 and the contingencies that come are L25 and L26. Emergency 1 state is represented by Neurons 1 and 13 and the contingencies that come are G1, G2, L12, L24 and L35. Emergency 2 state is represented by Neurons 4, 12 and 16 and the contingencies that come under this are G3, L14, L15 and L36. The system is said to be in secure state for the contingencies of BC, L23, L45, L56 depicted by Neuron 6 and for the remaining contingencies the system is in insecure state. Table V shows the activated neurons for the testing data, and Table VI, the contingency list for each operating state.

12.2. *Classification of unknown vectors*

To generate vectors which have not been trained, the load of the base case was uniformly changed by 10% for lower loads and 25% increase for higher loads. Thus, the proposed system is tested successfully for 70, 80, 90, 150 and 175% loading. The following results show the cluster map for the classification of vectors which have not been trained.

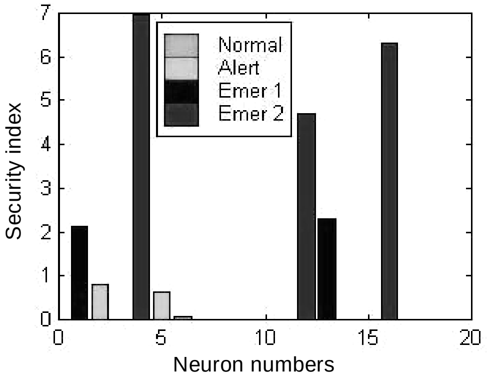


FIG. 10. Neuron number and its security index.

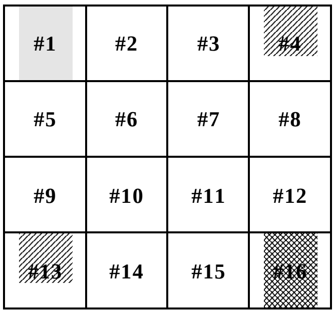
Table V
Activated neurons for the testing data

Activated neurons (secure)	Activated neurons (insecure)
# 6	#1, #2, #4, #5, #12, #13, #16

Table VI
Security levels and their contingencies

Security level	Contingency list
Normal	BC, L23, L45, L56
Alert	L25, L26
Emergency 1	G1, G2, L12, L24, L35
Emergency 2	G3, L14, L15, L36

12.2.1. *Test case 1–70% loading*



#1	#2	#3	#4
#5	#6	#7	#8
#9	#10	#11	#12
#13	#14	#15	#16

FIG. 11. Kohonen's feature map for 70% loading.

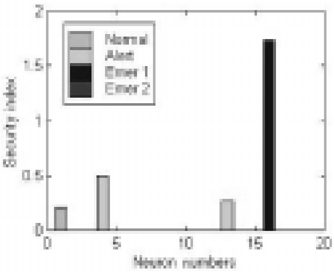
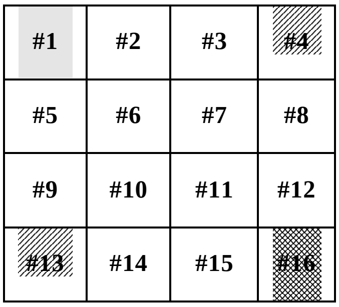


FIG. 12. Security index of neurons for 70% loading.

Table VII
Security levels and their contingencies for 70% loading

Security level	Contingency list
Normal	G1, L12, L15, L23, L25, L26, L36, L45, L56
Alert	G2, G3, L14, L24
Emergency 1	G3, L36
Emergency 2	Nil

12.2.2. *Test case 2–80% loading*



#1	#2	#3	#4
#5	#6	#7	#8
#9	#10	#11	#12
#13	#14	#15	#16

FIG. 13. Kohonen's feature map for 80% loading.

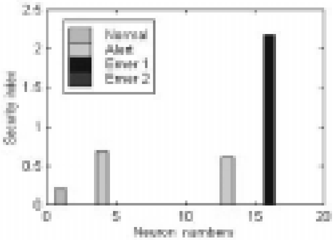
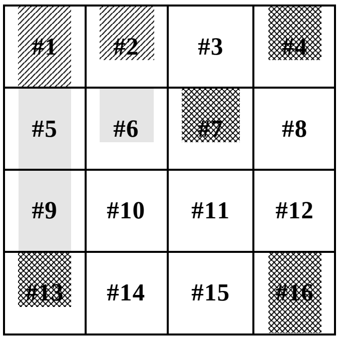


FIG. 14. Security index of neurons for 80% loading.

Table VIII
Security levels and their contingencies for 80% loading

Security level	Contingency list
Normal	G1, L12, L15, L23, L25, L26, L36, L45, L56
Alert	G2, G3, L14, L24
Emergency 1	G3, L36
Emergency 2	Nil

12.2.3. *Test case 3–90% loading*



#1	#2	#3	#4
#5	#6	#7	#8
#9	#10	#11	#12
#13	#14	#15	#16

FIG. 15. Kohonen's feature map for 90% loading.

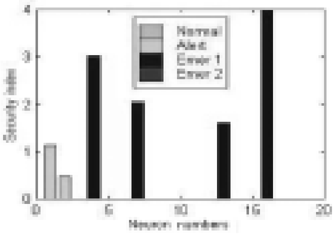


FIG. 16. Security index of neurons for 90% loading.

Table IX
Security levels and their contingencies for 90% loading

Security level	Contingency list
Normal	BC, L12, L23, L26, L45, L56
Alert	G1, L25, L35
Emergency 1	G2, G3, L14, L15, L24, L36
Emergency 2	Nil

12.2.4. *Test case 5–150% loading*

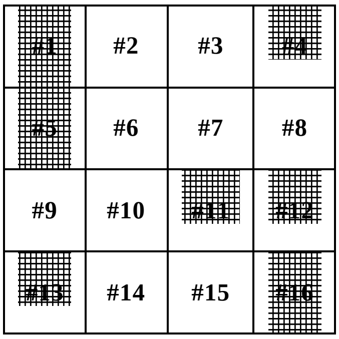


FIG. 17. Kohonen's feature map for 150% loading.

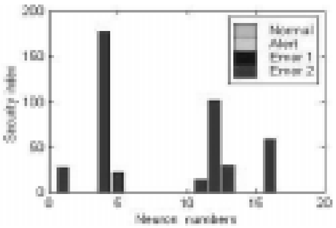


FIG. 18. Security index of neurons for 150% loading.

Table X
Security levels and their contingencies for 150% loading

Security level	Contingency list
Normal	Nil
Alert	Nil
Emergency 1	Nil
Emergency 2	G1, G2, G3, L12, L14, L15, L23, L24, L25, L26, L35, L35, L45, L56

12.2.5. *Test Case 6–175% loading*

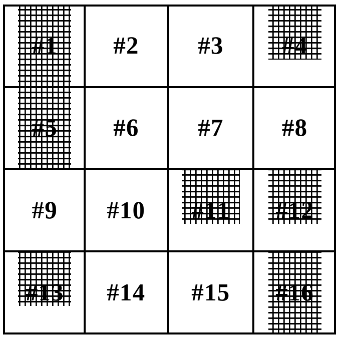


FIG. 19. Kohonen's feature map for 175% loading.

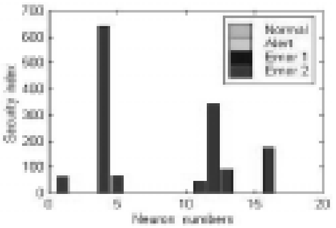


FIG. 20. Security index of neurons for 175% loading.

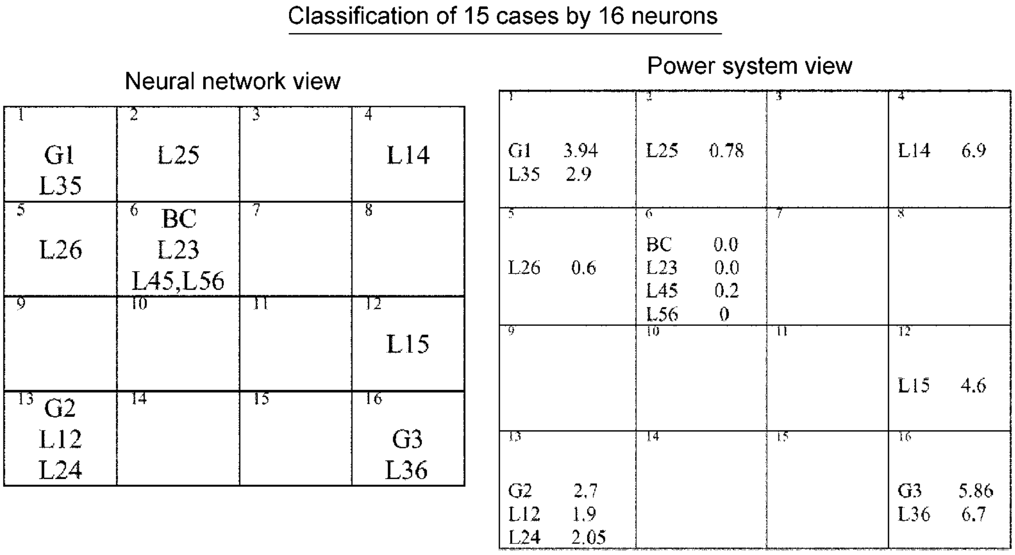
Table XI
Security levels and their contingencies for 175% loading

Security level	Contingency list
Normal	Nil
Alert	Nil
Emergency 1	Nil
Emergency 2	G1, G2, G3, L12, L14, L15, L23, L24, L25, L26, L35, L35, L45, L56

Figure 21 shows the important inference obtained from the self-organizing neural network. Classification results from both the neural network and power system point of view are presented. It is important to observe how the neural network has self-organized the neuron locations to as to match the security index of the outages. For instance, the contingencies G1 (3.94) and L35(2.9) having similar indices are grouped together and are placed in neuron cell number 1, while G3(5.86) and L36(6.7) are grouped together in neuron cell number 16. Similar clustering can be observed for neuron cell numbers 2, 4, 5, 6, 12, 13. Another important observation is that all severe contingencies are placed along the corners (1, 4, 12 and 16) of the Kohonen feature map.

12.3. *Extension to large-scale systems*

The working of the self-organizing method for clustering and classification has been demonstrated for a model six-bus system in Section 9. The neural network point perspective

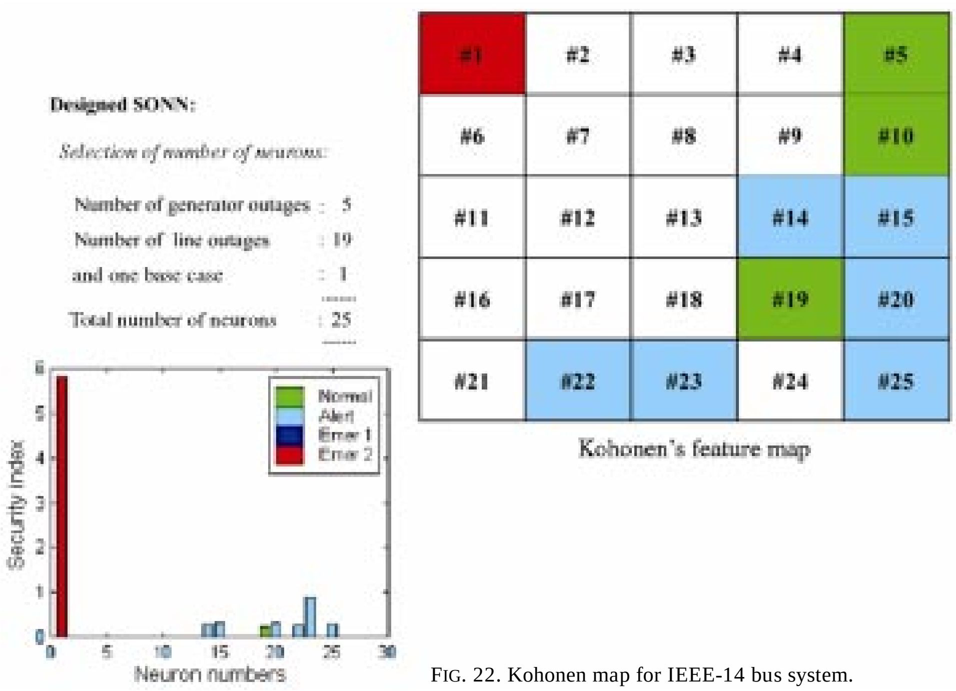


1 G1 L35	2 L25	3	4 L14
5 L26	6 BC L23 L45,L56	7	8
9	10	11	12 L15
13 G2 L12 L24	14	15	16 G3 L36

1 G1 3.94 L35 2.9	2 L25 0.78	3	4 L14 6.9
5 L26 0.6	6 BC 0.0 L23 0.0 L45 0.2 L56 0	7	8
9	10	11	12 L15 4.6
13 G2 2.7 L12 1.9 L24 2.05	14	15	16 G3 5.86 L36 6.7

FIG. 21. Inference of security assessment from neural network and power system view point.

and the power system point perspective of clustering or grouping of the similar contingencies are shown in Fig. 21. It can be observed from the figure that the SOFM performs clus-


Designed SONN:

Selection of number of neurons:

Number of generator outages	:	5
Number of line outages	:	19
and one base case	:	1
Total number of neurons	:	25

#1	#2	#3	#4	#5
#6	#7	#8	#9	#10
#11	#12	#13	#14	#15
#16	#17	#18	#19	#20
#21	#22	#23	#24	#25

Kohonen's feature map

FIG. 22. Kohonen map for IEEE-14 bus system.

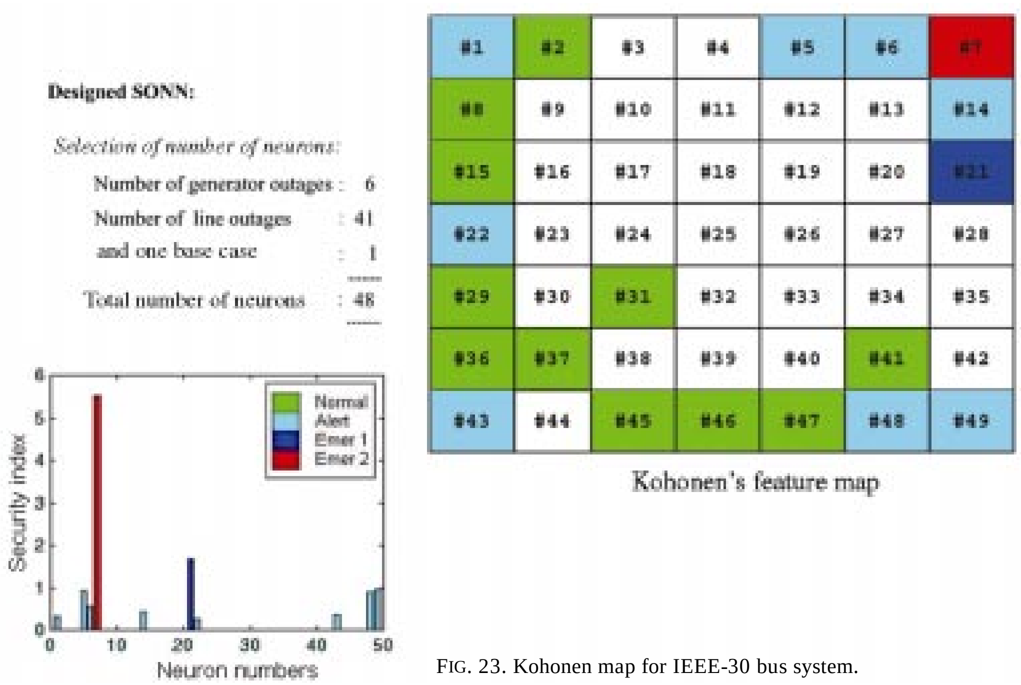


FIG. 23. Kohonen map for IEEE-30 bus system.

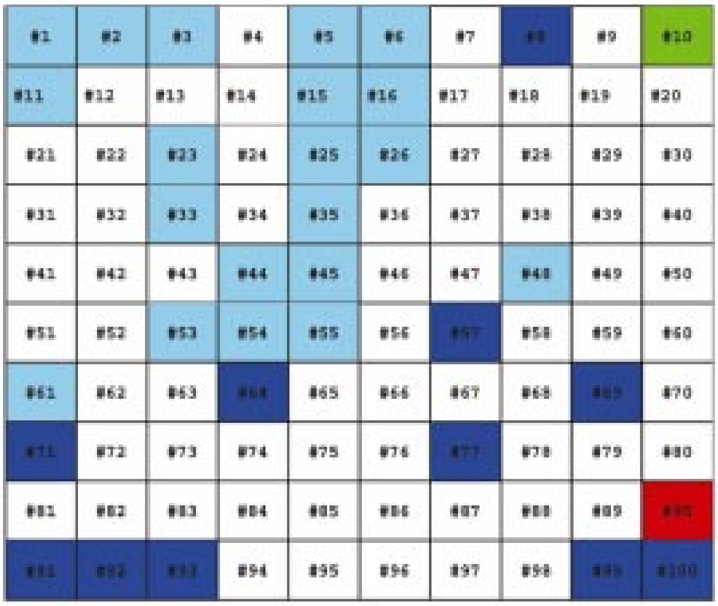
tering/grouping of identical contingencies in a way similar to the conventional method employed by the operator in a control centre. In other words, SOFM–ANN provides the same level of inference as that of the operator’s heuristics.

The generalization of the technique and its suitability for large-scale systems has been reported by considering three standard benchmark systems, namely, the IEEE-14, -30 and the –57 bus systems. Only the final Kohonen map result of these systems has been provided to illustrate the applicability of the method. Other related results corresponding to different loading conditions are not provided due to space limitations. The scalability of the proposed method for large-scale systems could be observed.

The application of SOFM for power system security assessment has been successfully extended to large systems like the IEEE-14, -30 and -57 bus systems, respectively. Figures 22–24 show the Kohonen maps for the IEEE-14, -30 and -57 bus systems. SOFM was found to identify the severe contingencies and determine the operating state of the power system.

12.4. *Important observations*

The neural net is trained offline with base cases and several contingencies. SONN classifies any unknown operating state correctly base cases. The Kohonen classifier serves as a monitoring tool. Training process takes a long time, approximately 5–10 min for smaller system and for large systems 15–20 min on a 700 MHz processor. Testing process requires only a few seconds for any system.



#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
#21	#22	#23	#24	#25	#26	#27	#28	#29	#30
#31	#32	#33	#34	#35	#36	#37	#38	#39	#40
#41	#42	#43	#44	#45	#46	#47	#48	#49	#50
#51	#52	#53	#54	#55	#56	#57	#58	#59	#60
#61	#62	#63	#64	#65	#66	#67	#68	#69	#70
#71	#72	#73	#74	#75	#76	#77	#78	#79	#80
#81	#82	#83	#84	#85	#86	#87	#88	#89	#90
#91	#92	#93	#94	#95	#96	#97	#98	#99	#100

FIG. 24. Kohonen map for IEEE-57 bus system.

13. Conclusion

An NN-based static security assessment technique for a model power system is proposed. The proposed work demonstrates the feasibility of classification of load patterns for power system static security assessment using Kohonen and SOFM. The most important aspect of this network is its generalization property. Using 15 different line loading patterns for training, the network successfully classifies the unknown loading patterns. This powerful and versatile feature is especially useful for power system operation.

References

1. S. Weerasooriya, M. W. El-Sharkawi, M. Damborg, and R. J. Marks II, Towards static security assessment of a large-scale power system using neural networks, *IEE Proc. C*, **139**, 64–70 (1992).
2. D. J. Sobaiic, Y. H. Pao, Wanto Njo, and J. C. Dolce, Real-time security monitoring of electric power systems using parallel associative memories, *IEEE Int. Symp. on Circuits and Systems, ISCAS–1990*, pp. 2929–2932 (1990).
3. H. H. Yan, J. C. Chow, M. Kam, C. R. Sepich, and R. Fischl, Design of a binary neural network for security classification in power system operation, *IEEE Int. Symp. on Circuits and Systems, ISCAS–1991*, Vol. 2, pp. 1121–1124 (1991).
4. R. Fischl, Application of NN to power system security. Technology and trends, *Proc. ICNN'94, Int. Conf. on Neural Networks*, Piscataway, NJ, 1994, pp. 3719–3723 (1994).
5. Dagmar Niebur, and A. J. Germond, Power system static security assessment using the Kohonen network classifier, *IEEE Trans. Power Systems*, **7**, 865–872 (1992).
6. Dagmar Niebur, and A. J. Germond, Unsupervised neural net classification of power system static security states, *Int. J. Electl Power Energy Systems*, **14**, 233–242 (1992).

7. Teuvo Kohonen, The self-organizing map, *Proc. IEEE*, **78**, 1464–1476 (1990).

8. B. Scott, Security analysis and optimization, *Proc. IEEE*, **75**, 1623–1644 (1987).

9. Yoh-Han Pao, Pattern recognition and machine intelligence techniques for electric power system security assessment, CWRU Technical Report (1988).

10. G. C. Ejebe, and B. F. Wollenberg, Automatic contingency selection, *IEEE Trans. Power Apparatus Systems*, **98**, 92–104 (1979).

11. Hiroyuki Mori, Yoshihito Tamaru, and Senji Tsuzuki, An artificial neural-net-based technique for power system dynamic stability with the Kohonen model, *IEEE Trans. Power Systems*, **7**, 856–864 (1992).

12. R. Fischl, D. Niebur, and M. A. El-Sharkawi, Security assessment and enhancement, *Proc. IEEE Conf. Artificial Neural Networks with Application to Power Systems* (M. A. El-Sharkawi, and D. Niebur, eds), 1996, IEEE Catalog no. 96-TP112-0, Ch. 9, pp. 104–127 (1996).

13. M. B. Zyan, M. A. El-Sharkawi, and N. R. Prasad, Comparative study of feature extraction techniques for neural network classifier (power system simulation), *Proc. Int. Conf. on Intelligent Systems Applications to Power Systems*, 1996, *ISAP '96*, Jan. 28–Feb. 02, 1996, Orlando, FL, USA, pp. 400–404.

14. Atteri Kuppurajulu, and P. Ossowski, An integrated real-time closed loop controller for normal and emergency operation of power systems, *IEEE Trans. Power Systems*, **1**, 242–249 (1986).

15. Damminda Alahakoon, Saman K. Halgamuge, and Bala Srinivasan, Dynamic self-organizing maps with controlled growth for knowledge discovery, *IEEE Trans. Neural Networks*, **11**, 601–614 (2000).

16. Neal Balu, Timothy Bertram, Anjan Bose, Vladimir Brandwajn, Gerry Cauley, David Curtice, Aziz Fouad, Lester Fink, Mark G. Laubybruce, F. Wollenberg, and Joseph N. Wrubel, On-line power system security analysis, *Proc. IEEE*, **80**, 262–280 (1992).

17. Richard P. Lippmann, An introduction to computing with neural nets, *ACM SIGARCH Computer Architecture News*, **16**, 7–25 (1988); *IEEE ASSP Mag.*, **4**, 4–22 (1987).

18. M. A. El-Sharkawi, and S. J. Huang, Development of genetic algorithm embedded Kohonen neural network for dynamic security assessment, *Int. Conf. on Intelligent System Application to Power Systems*, Orlando, Florida, Jan. 28–Feb. 2, 1996.

19. M. P. de Arizon, and J. R. Marti, Real-time power system security classifier, *Proc.* 1998 *Second IEEE Int. Caracas Conf. on Devices, Circuits and Systems,* March 2–4, 1998, pp. 320–326.

20. T. E. Dy Liacco, System security: the computer's role, *IEEE Spectrum*, **15**, 43–50 (1978).

21. T. S. Chung, and Fu Ying, An ANN-based network equivalent approach for power system on-line voltage security assessment, *Proc. POWERCON'98, Int. Conf. on Power System Technology, IEEE*, Aug. 18–21, 1998, Vol. 2, pp. 1504–1507 (1998).

22. R. N. Dhar, *Computer aided power system operation and analysis*, Tata McGraw Hill (1983).

23. A. J. Wood, and B. F. Wollenberg, *Power generation, operation and control*, Wiley (1984).

24. J. M. Zurada, *Introduction to artificial neural systems*, PWS Publishing (1992).

25. B. Yegnanarayana, *Artificial neural networks*, Prentice-Hall India (1999).

26. D. Hanselman, and B. Littlefield, *Mastering Matlab 5*, Prentice-Hall Int. (1998).