Dynamic Classification of Power Systems via Kohonen Neural Network

S.F Mahdavizadeh
Iran University of Science & Technology, Tehran, Iran
M.A Sandidzadeh
Iran University of Science & Technology, Tehran, Iran
M. R. Aghamohammadi
Power & Water University of Technology, Tehran, Iran

Abstract- Dynamic security assessment is one of the most prominent behavioral attributes of power systems. It can be assessed and evaluated by means of various behavioral events. For dynamic analysis of power systems after faults, Critical Clearing Time (CCT) is one of the most important parameters to evaluate. Calculation of CCT requires solving complicated problems which usually involves solving time consuming non-linear equations during fault moments. In this paper we use Kohonen neural network for finding a new pattern. First some sample patterns are created, and this is done in such a way to cover the entire operating space of the system. Then parameters of this pattern are applied to the neural network in order to train it and gain the best input. After these patterns are classified (20 different classes) and based on neural network's outputs, attributes of each class is determined. Finally a program is developed to determine each new behavioral pattern's class and attribute by applying it.

Keywords: Dynamic Security, Kohonen Neural Network, Transient Stability

1. Introduction

Dynamic security assessment is one of the prominent behavioral features of power systems which can be assessed and appraised base on various behavioral phenomenon. Transient stability is one of the dynamic phenomenons which can be used for dynamic security assessment of power systems. Network and generators' stability depends on various parameters such as network structure, generators specifications, load levels and generation pattern of the generators. Generation pattern can be changed during operation of a system and can improve and increase the dynamic security level of that system via a suitable generation pattern [1]. Creating a procedure for dynamic security level assessment of power systems by finding attributes of generation patterns is the purpose of this paper. This can be done by pattern finding, neural network and other methods. Critical Clearing Time (CCT) is the stability and classification index of generation pattern in this paper, and based on that we can classify generation patterns according to their security levels [2].
CCT is the most prominent parameter that can be identified for dynamic analysis of power systems. Calculation of CCT requires complicated computations that include time consuming solving of nonlinear equations when fault occurs. In this paper we want to use Kohonen artificial neural network for pattern extraction. Therefore by using correct inputs and without calculation of CCT, we can assess the dynamic security via neural network [1].

2. Introduction to the Neural Network

2.1. Learning with Supervisor

In learning with supervisor, a set of data pairs called learning data $I(P_i,T_i)$ is assigned to the learning rule in which P_i is the input to the network and T_i is the desired network output for P_i input. After applying P_i input to the network, w_i (weight coefficients) is compared with T_i in network output and then the learning error is calculated and is used to tune network parameters[3]. So that when next time the same P_i input is applied to the network, network output gets closer to T_i, knowing that the teacher is a system with exclusive environmental control (for example it knows that for P_i input the proper output is T_i)

2.2 Unsupervised Learning

In unsupervised or self-organizing learning, neural network parameters are tuned and amended only via system answers. In other words, input vectors are transmitted to the network only via captured data. Compared to learning with supervisor, in this method the desired answer vector is not applied to the network. In other words, no sample of the function the network has to learn is given to it. We will see that in practice learning by supervisor in networks consisting of multiple neural layers is very slow, and in such cases a combination of supervised and unsupervised learning is recommended[3], [4].

2-3 Kohonen Neural Network

In most applications it is better to ask the network to classify the learning data by itself. For this purpose, it is better to have these two main hypotheses in mind: First, membership in a level generally means having shared attributes, and second, the network can discriminate these shared attributes in input data space. Self organizing Kohonen maps are one type of these networks based on the mentioned hypotheses and it uses unsupervised learning for changing inner network state and modeling highlighted attributes of training data. We will investigate this by taking a closer look at Kohonen training algorithm[3],[5].

***2.4* General Operation of Kohonen**

It is postulated that the brain uses local maps for modeling complex structures. Kohonen has used this to his own advantage because this helps Kohonen to compress information via a method called *vector quantization*. This also helps the network to save input data meaningfully while also keeping its topological features.

Compression of information means that data can be kept in a space much smaller than a previous one[6].

The brain cortex is mainly composed of two dimensional surfaces of neurotic connections though it can keep meanings of much higher dimensions inside itself. Application of Kohonen algorithm is also usually two dimensional.

As an example, you can take a look at figure 1. This network is composed of only one two-dimensional surface. The fact to be highlighted here is that neurons, unlike multi-layered perceptrons, are not located on different (input, hidden, output) surfaces; rather they are located on an even level surface. All inputs are connected to output nodes. It is shown that there is no separate output surface. Each node in the network is also an output node.

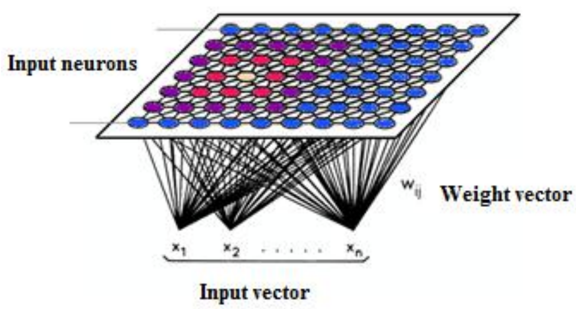


Figure (1): A Kohonen Network

2.5 Kohonen Algorithm

Kohonen training algorithm arranges network nodes as local neighbors, so that it can classify attributes of input information [7]. Network topography map is shaped automatically by periodically comparing inputs of each node with the vector saved by it in its connection lines. No desired answer is defined for training inputs. Whenever inputs of a node match the vector saved in that node, that area of the network is selectively optimized, so that it later represents the average training information of a data surface [8]. If the network is randomly shaped and adjusts itself slowly, it achieves a stable state.

This algorithm is like this:

1- Randomly determine primary weight coefficients and select the primary neighborhood radius (initially set t = 0)

Determine primary weight coefficient values which are the connectors between *n* input nodes to *j* output nodes by random values. $W_{ij}(t)$ presents weight coefficients at time *t* , which is the input, and *j* which is the output node. It should be noticed that at first we assign t = 0

Initially assign a large value to radius of neighborhood of output nodes ($N_j(0)$).

2- Select the input vector randomly from patterns selected for training.

Then apply input vector to the network. $X_i(t)$ is the input value of node *i* at time *t*

3- Determine neuron *c* so that its weight coefficients vector W_C has the shortest distance from the input vector.

Weight coefficient vector is selected as the following:

a) Distance d_j between input and output vector of each node *j* is determined via the following equation :

$$d_j = \sum_{i=0}^{n-1} ((x_i(t) - w_{ij}(t))^2 \qquad (1)$$

b) We Select the shortest distance $min\{d_j\}$
Correct weight coefficient W_C of winner neuron *c* and weight coefficients of its neighbor W_i.
These weight coefficients are corrected via the following equation:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)(x_i(t) - w_{ij}(t)) \qquad (2)$$

With these equations, weight coefficients of all neurons located in neighborhood of $N_C(t)$ are corrected.

Neighborhood function $\alpha(t,i)$ is decreased with time. This reduction rate is logarithmic.
As a result sharing coefficient of weight coefficients is slowed gradually. Consequently, radius of neighborhood is also reduced gradually.

4- Increase the time (t= t+1)
5- Repeat the algorithm by jumping to step 2. [4], [5], [9]
In this paper MATLAB functions are used for neural network simulation.

3. Methods for Creating Operation Patterns

One of the most important parts of dynamic security assessment - when done *offline* and via neural networks- is introducing the system entirely to the neural network[3]. For neural network training we need to define all the work space. Neural network only can interpolate in the work space, and for this reason defining the total performance condition of power system is really necessary. Therefore we tried to define this work apace. In order to produce this work space, minimum and maximum demands in load buses are determined. For example $P_{Lmin}{}^i$ and $P_{Lmax}{}^i$ are minimum demand and maximum demand of *i*th bus. Then this load limitation is divided into ten parts.
Total load of power system can be calculated with relation 3:

$$pload_{Sys}^{i} = \sum_{i=1}^{m} pload(i) \qquad (3)$$

In which $pload_{sys}^{i}$ is load's level of *i*th bus of power system.
After that, with the minimum and maximum power generation of each bus, the generation orders will be extracted.
Finally by taking random patterns into accounts we will have loads of 10 load levels and for each load level 10 random load patterns and 10 random production patterns for our desired network, meaning that 1000 operation patterns are taken into account.
We cannot use all these patterns for network training and the number of patterns used should be as low as possible (so that it won't impact the network training). This is because a high number of patterns will make the training process very time consuming and slow.
It is therefore necessary to select suitable sample patterns with optimal numbers from among all other patterns for this purpose. Our basis for selecting patterns was sample network's load level. So that 202 patterns were selected as primary patterns for network training and these patterns, when compared to the total patterns, have the same load levels. Figures (2) and (3) prove this. By comparing these two graphs it becomes clear that distribution of load levels of the selected patterns and the entire patterns is the same.

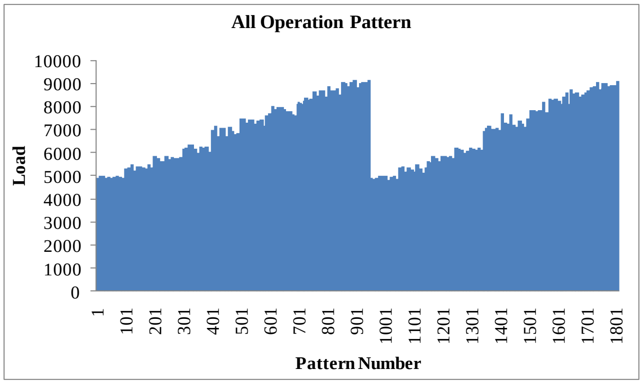


Figure (2) Load levels of all patterns (including 1817 operation patterns)

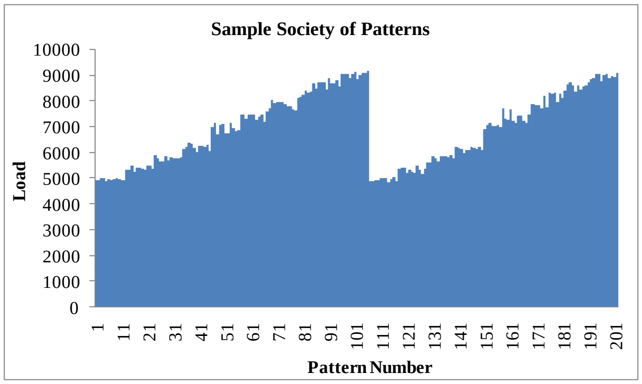


Figure (3) Load level of 202 sampled patterns (patterns with load level distribution similar to that of the entire patterns)

4. Sample Network Introduction

The questioned network in this article is IEEE network with 39 buses. This network comprises 10 buses with generator and 29 load buses. It also has 34 lines.

For simplicity, network lines in this article are displayed like figure 4. It is to be noticed that bus 31 is selected as the reference bus.

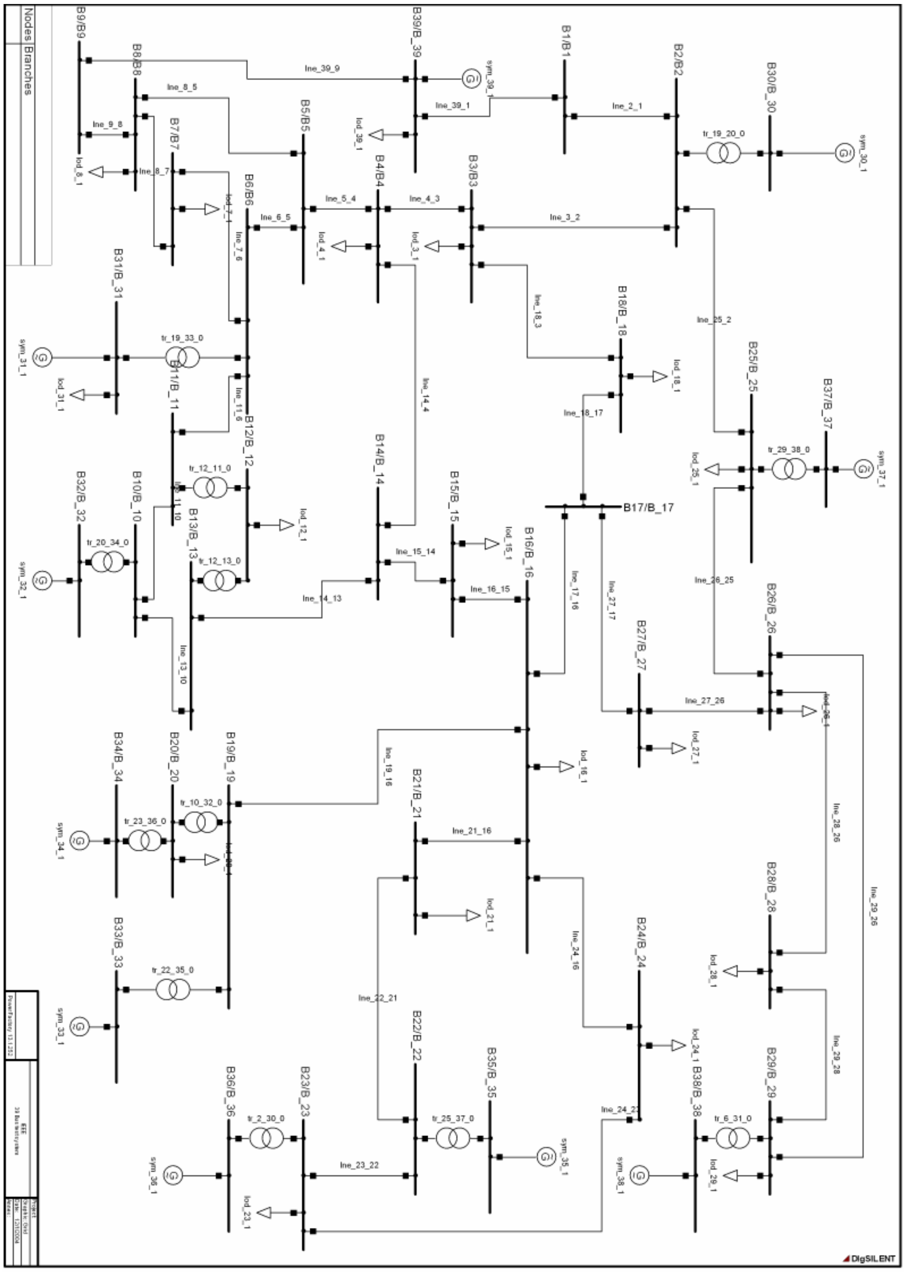


Figure (4): Sample Network with 39 buses

5. Combination of utilized parameters

Combination of inputs can be a better way for describing primary state of the system. So we decided to use various combinations of parameters. Since some parameters yield to analogous behaviors in the power system, we have not used combination of all of them. Here we have used 12 different combinations that include:

1) Lines active flow power (P_{Line})
2) Lines active flow power and each generator's active generation power (P_{Line} & P_G)
3) Lines active flow power, each generator's active generation power and voltage of buses (P_{Line} & P_G & V)
4) Lines active flow power, each generator's active generation power and inertia constant (P_{Line} & P_G &H)
5) Lines active flow power, each generator's active generation power and lines reactive flow power (P_{Line} & P_G &Q_{Line})

6) Lines active flow power, each generator's active generation power, lines reactive flow power and voltage amount (P_{Line} & P_G & Q_{Line} & V)
7) Lines active flow power and voltages of buses (P_{Line} & V)
8) Lines active flow power, voltages of buses and inertia constant (P_{Line} & V & H)
9) Lines active flow power, lines reactive flow power and voltages of buses (P_{Line} & Q_{Line} & V)
10) Lines active flow power, each generator's active generation power, lines reactive flow power, load of buses, voltage amount and inertia constant (H & P_{Line} & P_G & Q_{Line} & Q_G & P_{Load} & V)
11) Load active power and generators' generation power (*PLoad & PG*)
12) Active and reactive load power (*PLoad & QLoad*)

To indentify the best input combination we should first define a principle. Our principle here is classification based on behavioral patterns (dynamical or CCT). It means that we take one input load of the neural network as CCT of lines and after training we define it in form of classes. After this we classify the results gained from training of operation patterns. Notice that the number of classes in both classification methods (behavioral and operational) should be the same. Finally we compare equivalent classes and measure coordination percent of each class. By calculating average coordination percent of each class we achieve total coordination percent. Finally we take total coordination percent as the comparison principle for various inputs.
Coordination percent was achieved between each one of the mentioned combinations and CCT. The gained results are demonstrated in table (1). As it is shown in this table, the highest percents were achieved for (*PLine & QLine & V*) and (*PLine & V*) combinations. In table (1) total coordination percent is achieved by surveying each individual pattern's coordination. It means coordinated patterns in patterns classifications is determined based on operation neural network and is divided by the number of total patterns. Also coordination percent of each class is calculated separately and classes with the highest and lowest coordination are indicated in this table. Finally by calculating the average weight of coordination percent of the entire classes, the average weight coordination of classes is calculated.
We have used equation (4) for calculating the average coordination percent of classes:

$$CP_{av} = \frac{\sum_{k=1}^{k=7} CP_k \times N_k}{\sum_{k=1}^{k=7} N_k} \qquad (4)$$

CP_{av}: Coordination Percent Average
N_k: Number of Patterns in Kth class
CP_k: Coordination Percent in Kth class

Table (1): Coordination percent between classification by selected combinations and CCT

Parameter	P_{Line}	V	Q_{Line}	Q_G	P_G
Coordination percent to CCT of par ameters	*43.4*	*15.7*	*25.2*	*7.21*	*6.18*

6. Selecting Optimal Combination by Correlation Coefficient Method

As previously mentioned, for optimal training of neural networks their inputs should be selected from those operation quantities that have the highest impact on changes of CCT values of lines. Correlation Coefficient method for determining these quantities is described in the following part.
In this method, correlation between one line's CCT values and each operation quantity of system is calculated via equation

(5):

$$r = \frac{S_{XY}}{S_Y S_X} = \frac{\sum XY - (1/n)\sum X \sum Y}{\sqrt{[\sum X^2 - (1/n)(\sum X)^2][\sum Y^2 - (1/n)(\sum Y)^2]}} \qquad (5)$$

In equation (5) we have:

X & Y: are quantities we want to find their correlation, which can represent CCT and the desired operation variable respectively.
n: Number of samples
Neural Network input is selected from those operation quantities that have the highest correlation with all lines CCT.
In table (2) total correlation percent of lines CCT to different parameters is presented.
It can be seen in this table that P_{Line}, V and Q_{Line} have a better condition and lines CCT have the highest correlations with these variables. Now by using the results from the two mentioned methods and by taking the experimented coordination percent into account, (P_{Line} & Q_{Line} & V) is selected as the optimized combination.

Table (2) Percent of Lines CCT correlation to different parameters

	Input Parameter	Total coord. Percent	Max. Coord. Classes	Min. Coord. Classes	verage coord. i Classes
1	$_{ine}$	76.8	95	35	75
2	$_{ine}$ & P_G	74.8	100	36	76
3	$_{ine}$ & P_G & H	75.3	100	35	73
4	$_{ine}$ & P_G & V	72.8	100	35	74
5	$_{ine}$ & P_G & Q_{Line}	74	100	37	76
6	$_{ine}$ & P_G & Q_{Line} &	77.3	100	37	78
7	$_{ine}$ & V	82	100	55	83
8	$_{ine}$ & V & H	71.4	94	37	67
9	$_{ine}$ & Q_{Line}_V	79.8	100	50	79.5
10	$_{ine}$ & P_G & Q_{Line} & & P_Lo & V	75.8	100	42	78
11	$_{oad}$ & P_G	61	95	32	61.5
12	$_{oad}$ & Q_Load	47.5	94	28	49.5

7. Choosing the Best Number of Neurons

As it was previously mentioned, the exact number of security classes of sample network is not clear to us and therefore we can't estimate the exact number of neurons. Consequently we have to train the network for various numbers of neurons and find the results. We will compare each gained state of our selected operation patterns with their analogous dynamic parameters and select the optimal number of neuron from a state that has the highest coordination rate.
In table (3) we have the coordination results for states with 16, 25, 100, 196, 225, 256 and 400 neurons. It is seen that an increase in the number of neurons results in an increase in coordination percent. Since large a number of neurons extends the training time and also we observe no significant boost in coordination rate when this number exceeds 196, we have decided to choose 196 as the number of neurons. We also tested 100 neuron sets, though since there was a low coordination in neighborhood neurons, we decided not to use this configuration of neurons.

Table (3): Assessing the best configuration of neurons by comparing behavioral and working neural network classification

Number of Neurons	*inimum neur coordination*	*aximum neur coordination*	*Average neuron coordination in the first state*	*Average neuron coordination in the second state*	*inal average coordination*
16	*37*	*100*	*66*	*67*	*66..5*
25	*40*	*100*	*66*	*67.5*	*66.75*
100	*37*	*100 (34)*	*82*	*87*	*84.5*
196	*41*	*100 (34)*	*84*	*88*	*86.5*
225	*41*	*100 (61)*	*84*	*90*	*87.5*
256	*41*	*100 (61)*	*86*	*91*	*89*
400	*41*	*100*	*88*	*92*	*90*

8. Final Classification

By finding the appropriate combination and optimal number of neurons for neural network training we should now train the specified patterns.
We start network training by applying the resultant inputs and the optimal number of neurons. We have assumed 5000 iterations. It was consequently observed that 68 neurons won in the training process and a number of patterns were dedicated to them. Each one of these neurons can represent one class. The remaining (128) neurons were not excited enough in this process and they are considered dead neurons.

Since winning neurons are numerous and neighbor neurons might have similar behaviors, it might be possible to put some of them in the same classes. Therefore it is necessary to first study patterns located in each neuron and then define each neuron's characteristics. After defining characteristics of all neurons we study these characteristics' shared points in neighboring neurons, and if they have the same characteristics we put them in same classes.
The criterions for classification of patterns included in neurons are:

1- The amount of similarity of neurons' characteristics
2- Order of neighborhood of neurons

We classified the gained results with the aim of defined principles. Finally we have achieved 20 security classes for operation patterns.

9. Feature Extraction of Classes

Classification via neural network is not based on a specific physical context. In other words, Kohonen neural network classifies patterns only based on mathematical principles and in fact it is us that have to look for specific physical characteristics in classes created by neural network. Finding the mentioned characteristic(s) depends on our purpose of classifying the desired inputs. Since assessing dynamic security is our goal in this paper, then the basis for patterns classification is dynamic security. It is clear that CCT of lines is a clear indication of network dynamic security.
For investigating condition of patterns CCT we indentify lines with the lowest CCT in each pattern and record CCT value of these lines. Now we investigate CCT condition of patterns located inside each neuron.
It can be observed that patterns located inside neurons with the worst lines CCT are identical, and the amount of CCT in patterns of each neuron is in an identical range. It should be noticed that order of lines criticality is considered in all of them.
Hence after classification (or putting neighbor neurons with the same behavior in the same class), orders of the worst lines was also considered for patterns of each class; meaning that the patterns located in a class have the same worst lines and their CCT value is in the same range, though the amount of coordination in this case was reduced when compared to the previous case.
Another characteristic of the resulting classes is that all patterns located in a class have almost identical load levels.
In figure (5) conformity percent of each class is presented. Also number of patterns of each class is indicated as redundancy of each class:

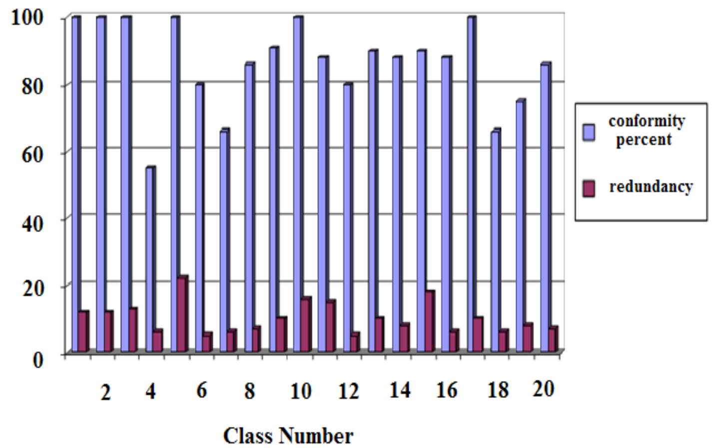
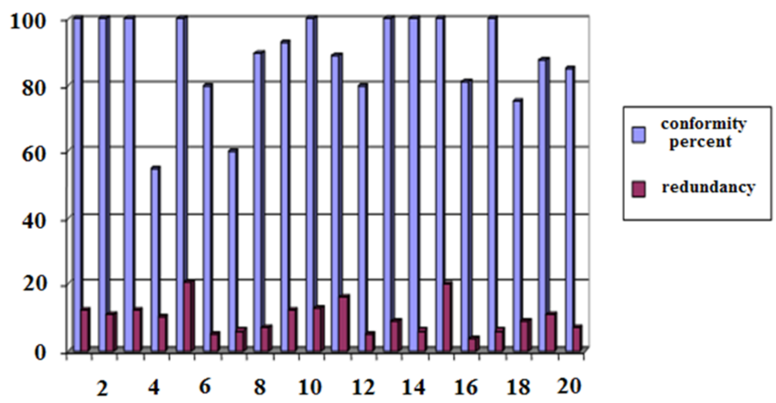


Figure (5): Conformity percent and redundancy of patterns of each class under training

To obtain the accuracy levels of each class we have calculated conformity levels of classes for experimental patterns presented in figure (6).
Classes displayed here with low accuracy have small number of patterns. It can be seen that we have attained the promised accuracy here.



Class Number
Figure (6): Conformity percent and number of patterns for each class for experimental patterns

Finally with the aim of equation (6), the total conformity degree of neural network turnover in assessment phase is achieved.

$$cd_{Test} = \frac{\sum_{k=1}^{k=20} cd_k \times N_k}{\sum_{k=1}^{k=20} N_k} = 0.91 \qquad (6)$$

In which:
CP_{Test} : Neural Network weigh coordination percent

N_k : Number of Patterns in kth class

CP_k : Weigh coordination percent in kth class

10. Neural Network Assessment

In order to make sure that the operation of neural network is valid we should investigate some examples here. We did select an overall of 200 samples from operation patterns and investigated the results after applying them to the network. It was observed that the neural network had correctly responded with the accuracy we expected from it.

11. Conclusion

With rapid expansion of the power network in Iran and its operational complexity, it has become necessary to identify this system's operation security limits. Time is of an ultimate significance when identifying these security limits. Having live data of network security status will help us to find better ways for operating the network. In this paper we did pursue these goals and a method for extracting these security borders and power system security classification was presented.

The proposed method helps to investigate limits of generators' generation order configurations in various operation patterns of power systems when compared to transient stability phenomenon. It should be noted that the aim of this method is not to estimate or calculate CCT; rather it helps to present an overall picture of system security status and also early detection of critical lines.

To reach the mentioned goals we used Kohonen neural network in this paper and it was shown that Kohonen neural network provides a solution for our mentioned goals. In this method by classifying patterns into various security groups we presented network security status by 20 different security classes.

Attributes of each class was identified as its characteristic. It was observed that patterns inside each class have identical worst lines and follow identical critical patterns. Also we found that patterns in each class have the similar load levels.

References

[1]. P. Kundur , "Power System Stability and Control" International Edition ed. California, McGraw Hill Inc., 1994.

[2]. Hiroshi Takada Yoshinori Kato Shinichi Iwamoto, "Transient Stability Preventive Control Using CCT and Generation Margin", Power Engineering Society Summer Meeting, IEEE 2001, Vol. 2, p. 881-886

[3]. Mahmoud B. Zayan, Mohamed A. El-Sharkawi and Nadipuram R. Prasad, "Comparative Study of Feature Extraction Techniques for Neural Network Classifier", IEEE Proc. of ISAP'96, Orlando, Florida, January-February 1996, p. 400-404

[4]. Alain J. Germond, "Power System Static Security Assesment Using The Kohonen Neural Network Classifier", IEEE Transactions on Power Systems, Vol. 7, No. 2, May 1992, p. 865-872

[5]. R.Beale, T.Jackson, "Neural Computing: An Introduction", Institute of Physics Publishing

[6]. M. Boudour, A. Hellal, "Large Scale Power System Dynamic Security Assessment using the Growing Hierarchical Self-organking Feature Maps", IEEE International Conference on Industrial Technology (ICIT), 2004.

[7]. E.S. Karapidakis and N.D. Hatziargyriou, "Application of Artificial Neural Networks for Security Assessment of Medium Size Power System", 10th Meditenmean Electrotechnical Conference, MEleCon 2000, Vol. 3.

[8]. Dejan J. Sobajic, Yoh-Han Pao, Artificial Neural-Net Based Dynamic Security Assessment for Electric Power Systems, IEEE Transactions on Power Systems, February 1989, Vol. 4, No. 1, p.220-228

[9]. K.W.Chan, A.R.Edwards, R.W.Dunn and ARDaniels, "On-line dynamic security contingency screening using artificial neural networks", IEE proc-Gener Transm Distrib, November 2000, Vol 147, No 6, p. 367-372.