Vulnerability Assessment of a Large Sized Power System Using Neural Network Considering Various Feature Extraction Methods

Ahmed M. A Haidar†, Azah Mohamed*, and Aini Hussian*

Abstract – Vulnerability assessment of power systems is important so as to determine their ability to continue to provide service in case of any unforeseen catastrophic contingency such as power system component failures, communication system failures, human operator error, and natural calamity. An approach towards the development of on-line power system vulnerability assessment is by means of using an artificial neural network (ANN), which is being used successfully in many areas of power systems because of its ability to handle the fusion of multiple sources of data and information. An important consideration when applying ANN in power system vulnerability assessment is the proper selection and dimension reduction of training features. This paper aims to investigate the effect of using various feature extraction methods on the performance of ANN as well as to evaluate and compare the efficiency of the proposed feature extraction method named as neural network weight extraction. For assessing vulnerability of power systems, a vulnerability index based on power system loss is used and considered as the ANN output. To illustrate the effectiveness of ANN considering various feature extraction methods for vulnerability assessment on a large sized power system, it is verified on the IEEE 300-bus test system.

Keywords: Artificial Neural Network, Contingency Analysis, Feature Extraction, Power System Loss, Vulnerability Assessment

1. Introduction

The uncertainties of power system restructuring efforts have led many companies to operate their systems close to the maximum loadability limits, thereby unwittingly pushing their limits. Thus, one of the challenging problems faced by power system operators nowadays is the increase of vulnerability level in the system. This enhances the need to develop a fast and accurate vulnerability assessment technique for real time application. Traditionally, power system security assessment is carried out to estimate the security level of a system and it encompasses the vulnerabilities resulting in voltage insecurity, static insecurity, and dynamic insecurity. Vulnerability assessment takes into account not only the security information but the information of the entire system. Vulnerability assessment of power systems requires analysis of the system behavior under a prescribed set of events known as contingencies such as line outage (LO), generator outage (GO), increase in total load and amount of load disconnected. Conventionally, such analysis is done by simulating all the contingencies, which is very time consuming for a large sized power system. Hence, vulnerability assessment by contingency analysis is not feasible for real-time application.

Artificial neural network (ANN) has been proposed as an alternative method for solving certain difficult power system problems where the conventional techniques have not achieved the desired speed, accuracy, and efficiency. It is considered as an important artificial intelligence technique that has been used successfully in many areas of power systems [1]. Applications of ANN in power systems are such as for predicting security of the power system [2], transient stability assessment [3], topology recognition [4], fault detection, and load modeling [5, 6]. ANN is also applied to power system vulnerability assessment based on vulnerability indices [7]. In this paper, ANN is also applied for vulnerability assessment of the power system but considering various feature extraction methods.

One of the important aspects for achieving good neural network performance is by proper selection of training input features [2]. Power systems encompass significant input information about the system state which includes

† Corresponding Author: Dept. of Electrical, Electronic and Systems Engineering, The National University of Malaysia - UKM, 43600 Bangi, Selangor, Malaysia (ahaidar67@yahoo.com)
* The National University of Malaysia - UKM (azah @eng.ukm.my, aini @ eng.ukm.my)
Received 24 May, 2007; Accepted 12 February, 2008

load flow information such as voltage, real and reactive power flows, voltage angle, generated power, and demand. Other information may include topological data such as transformer setting, switch position, and system topology. For large interconnected power systems, complete state information is too large for effective ANN implementation. Therefore, the data information needs to be reduced to a smaller amount of information.

High dimensional datasets are becoming more and more abundant when solving classification problems. The most common approach to feature management is feature selection and feature extraction. A variety of feature selection methods have been developed to tackle the issue of high dimensionality. Feature selection is the process of identifying those features that contribute most to the discrimination ability of the neural network in which only these features are used to train the neural network and the rest of the features are discarded. Feature extraction is a transformation of the original dataset into a new space of lower dimensions while preserving the needed information. In other words, it is the process of mapping all available features into a composite feature set of lower dimension. It cannot be performed by engineering judgment or physical knowledge only, but it must be implemented according to the statistical property of the various features and the dependency among them [8]. A major challenge in feature extraction is to extract a set of features, as small as possible, that accurately classifies the learning examples. The extracted set of features must represent the entire system, since a loss of information in the reduced set results in loss of performance and accuracy of the ANN. In the literature, there are many different approaches of feature selection and feature extraction methods applied to power systems such as the use of sensitivity analysis [3], K-mean clustering [9], [10], neural network based feature extraction technique [7], feature selection algorithms [11], principle component analysis, and decision tree and genetic algorithm [8].

This paper is organized accordingly; Section 2 describes the ANN theory and the algorithm used in this study, Section 3 provides some background on the generation of training data, and Section 4 outlines and describes the proposed feature extraction technique using Neural Network Weight Extraction (NNWE) based on vulnerability index and several other feature extraction techniques, and Section 5 compares and evaluates the efficiency of the proposed feature extraction technique. Finally, Section 6 summarizes the findings.

2. Artificial Neural Network

Neural networks offer several advantages over traditional techniques including the ability to learn from examples. These can be used for online applications, since most of the computations are done offline and negligible online computation is required. The most popular type of ANN is the multilayer feed-forward neural network with back propagation training algorithm in which one or more layers exist between the input and output layers as shown in Fig. 1. These layers are called hidden layers. Learning is achieved by systematically modifying the weights and biases of the neural network to improve the network's output response to acceptable levels. The back-propagation training algorithm is designed to minimize the mean square error between the actual and the desired outputs. First, a training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until the output layer generates an output pattern. If this pattern is different from the desired output, an error is calculated and then propagates backwards through the network from the output layer back to the input layer. The weights are modified as the error is propagated backward in the second phase [12].

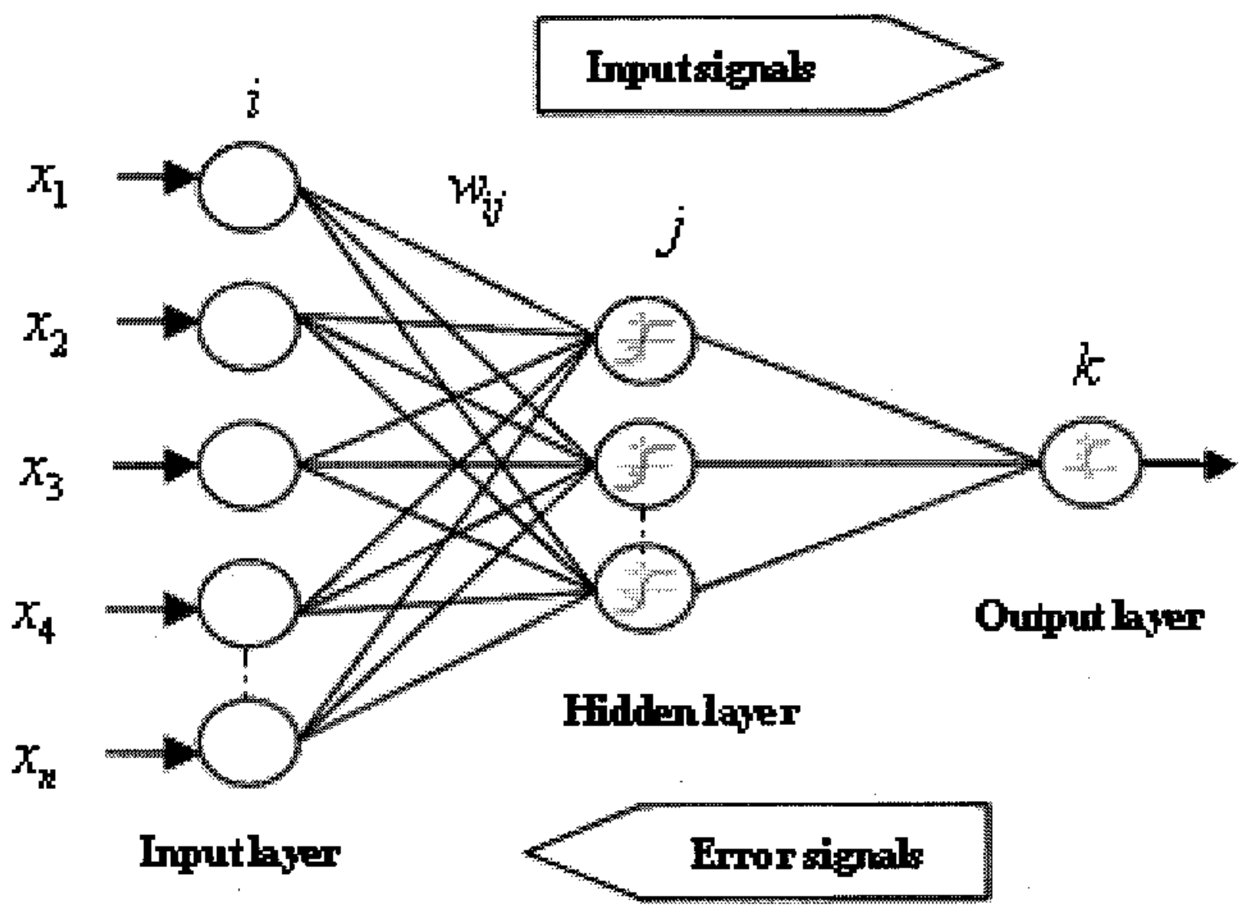


Fig. 1. Multilayer feed-forward ANN with back-propagation algorithm

The back-propagation training algorithm is described as follows:

Step 1: Setting of ANN parameters

Set the values of learning rate parameter η, number of hidden layers, number of hidden neurons, and number of output neurons.

Step 2: Initialization

Set all the weights and threshold levels of the network to random numbers uniformly distributed inside a small range of $(-2.4/F_i, +2.4/F_i)$, where F_i is the total number of input neurons. The weight initialization is done on a neuron-by-neuron basis.

Step 3: Activation

Activate the back-propagation neural network by applying inputs $x_1, x_2, \ldots, x_n$ and desired outputs $y_{d,1}, y_{d,2}, \ldots, y_{d,n}$

i. Calculate the actual values of the neurons in the hidden layer:

$$v_j = sigmoid\left[\sum_{i=1}^{n}[x_i.w_{ij}] - \theta_j\right] \quad (1)$$

Where:

$sigmoid$ - activation function

n - number of inputs to the neurons in the hidden layer

w_{ij} - weight between i -th input and j -th hidden neurons

θ_j - threshold applied to a neuron in the hidden layer

ii. Calculate the actual outputs of the neurons in the output layer:

$$y_k = sigmoid\left[\sum_{j=1}^{m}[v_j w_{jk}] - \theta_k\right] \quad (2)$$

A convenient logistic activation function yields an output which varies continuously from 0-1 as shown in Fig. 2. The output can be represented by the relation:

$$y_k = f(Net_k) = \frac{1}{1 + \exp[(Net_k - \theta_k)/\theta_0]} \quad (3)$$

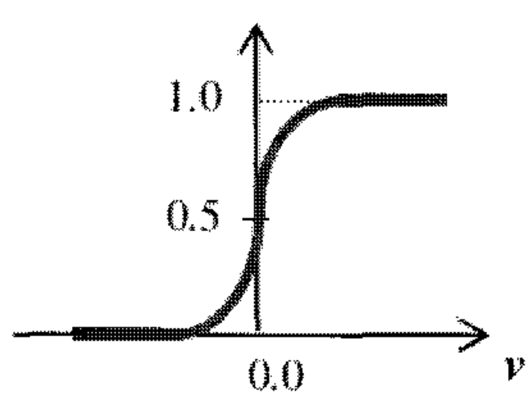
Where:

m - number of neurons in the hidden layer

w_{jk} - weight between j -th hidden neuron and k -th output neuron

θ_k - threshold applied to a neuron in the output layer

θ_0 - to determine the abruptness of the transition



$$Y_{logsig} = \frac{1}{1+\exp(-v)}$$

Fig. 2. Logistic activation function

Step 4: Updating the weights

i. Update the weights in the back-propagation network by propagating backward the errors that are associated with the output neurons.

ii. Calculate the error gradient for the neurons in the output layer:

$$\delta_k = y_k(1 - y_k)e_k \quad (4)$$

where;

$$e_k = y_{d,k} - y_k \quad (5)$$

iii. Calculate the weight correction:

$$\Delta w_{jk} = \eta.v_j\delta_k \quad (6)$$

iv. Update the weights between output and hidden layers

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (7)$$

v. Calculate the error gradient for the hidden neurons:

$$\delta_j = v_j(1 - v_j)\sum_{k=1}^{i}\delta_k w_{jk} \quad (8)$$

vi. Calculate the weight correction:

$$\Delta w_{ij} = \alpha.x_i\delta_j \quad (9)$$

vii. Update the weights between hidden and input layers

$$w_{ij}(new) = w_{ij}(old) + \Delta w_{ij} \quad (10)$$

In Equation (6), η is the leaning rate parameter that can be increased without causing oscillation, by modifying the equation such that a momentum term is added as

$$\Delta w_{jk}(N+1)) = \eta.v_j\delta_k + \alpha.\Delta w_{jk}(N) \quad (11)$$

Where:

N - represents the number of times for which a set of input patterns have been presented to the network and α is the momentum term which determines the effect of past weight changes on the current direction of movement in weights.

Step 5: Iteration

Increase iteration by going back to Step 3 and repeat the process until the selected error criterion is satisfied.

3. Data Generation

Before applying ANN for power system vulnerability assessment, the first step is to create an appropriate training dataset for the ANN. The approach used is by carrying out simulations on a power system that is subjected to various disturbances and gathering a set of system features along with the corresponding system vulnerability index. These input features are processed by using various feature extraction methods before presenting to the ANN. The next step is to normalize the inputs and targets by transforming each feature in the dataset. It can be transformed either to the range [0, 1] or [-1, 1], or it can normalize the input and target to obtain a zero mean and unity variance [3, 8]. However, the standardized z-score is computed by the deviation feature vector from its mean normalized by its standard deviation. The z-score, called z_j, is computed for every feature vector x_j, including p pattern, and it is given by

$$z_j = \frac{x_j - \bar{x}_j}{\sigma_x} \tag{12}$$

With the standard deviation of feature vector x_j given as

$$\sigma = \sqrt{\frac{1}{p-1}\sum_{i=1}^{p}(x_i - \bar{x})^2} \tag{13}$$

and the mean of feature vector x_j given as

$$\bar{x} = \frac{1}{p}\sum_{i=1}^{p} x_i \tag{14}$$

To implement the ANN method for vulnerability assessment in power systems, the IEEE 300-bus system is considered as the test system in which it is divided into three areas. The test system consists of 69 generators, 116 transformers, 295 lines, and 198 loads. Fig. 3 presents a single line diagram of the first area of the 300-bus system. The first area is selected to represent the system because it has the largest number of generators in which it contains 26 generators and it is linked to the second and the third areas of the system.

A training database was collected by first analyzing the system behavior at the base case condition. The next step is to analyze the system behavior when subjected to credible system contingencies such as line outage (LO), generator outage (GO), load increase, and disconnection of loads. For each set of training data, the vulnerability index based on power system loss (PSL) is then calculated. PSL considers total system loss, generation loss due to generation outage, power line loss due to line outage, increase in total load, and amount of load disconnected. The rationale for considering PSL is due to the fact that losses in a power transmission system are a function of not only the system load but also of the generation [13]. PSL is used as the ANN output and it is given as,

$$PSL = \frac{S_{BCL}}{S_{CCL} + S_{IL} + S_{LD} + \sum_{i=1}^{n} S_{LGO} W_{G,i} + \sum_{i=1}^{m} S_{LLO} W_{L,i}} \tag{15}$$

Where:

S_{BCL} - apparent power loss at base case

S_{CCL} - apparent power loss at contingency case

S_{IL} - increase in total load

S_{LD} - amount of load disconnected

$S_{LGO,i}$ - loss of generated MVA due to generator outage

$S_{LLO,i}$ - loss of transported MVA due to line outage

$W_{G,i}$ - weight of individual generator power output

$W_{L,i}$ - weight of individual line power influence

n - number of generators

m - number of lines

From Equation (15), it can be seen that the proposed vulnerability index, PSL, will have values in the range of 1 – 0 in which these values can be categorized based on vulnerability boundaries characterized by the values of PSL. If the PSL value is high, for example, in the range (0.6 – 1), it indicates that the system is invulnerable, whereas if the PSL value is low, that is (0 – 0.59), it indicates that the system is vulnerable. These index values can be readjusted by a control operator based on any new system configuration considering the weights of power system components.

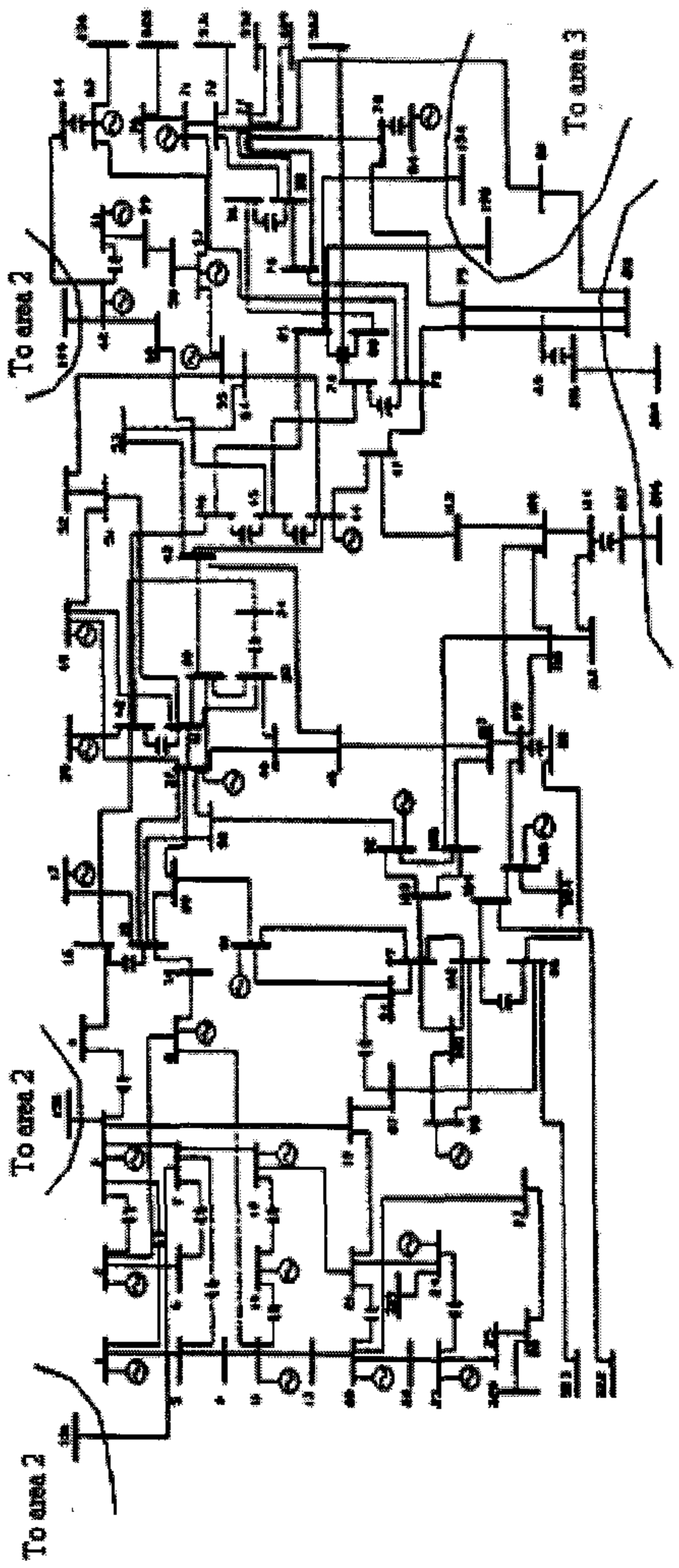


Fig. 3. First area of the IEEE 300-bus test system

4. Feature Extraction Methods

Feature extraction is the process of mapping all available features into a composite feature set of lower dimension. Here, the dimensionality of a feature set is reduced by combining features while retaining characteristics that allow for accurate classification. The first step before applying feature extraction is to collect as many data as possible from a power system, which is assumed to be of physical interest for vulnerability assessment based on engineering judgment. Such data should be both measurable in a real power system and available from power utilities. In this section, four different feature extraction methods used in this work are discussed.

4.1 Principle Component Analysis

Obtaining the eigenvalues and eigenvectors of the covariance matrices for normal distributions is known as the PCA or the Karhunen–Loeve transforms. PCA is also known as an optimal linear dimensionality reduction and it is a classical method often used for pattern recognition and data analysis. The goal is to map vectors a_i $(i = 1,2,...., m)$ in an n -dimensional space onto another set of vectors in a k -dimensional subspace where $k < n$. But this usually causes some loss of the information which discriminates the different classes. However, the projection points on the new subspace still keep the characteristics of the original information. After the projection, similar points will be closer together and dissimilar points will be farther away from one another [8], [14]. The covariance matrix Φ of the feature vectors belonging to a word-class is defined as

$$\Phi = \sum_{1}^{m}(a_i - a_{ave})(a_i - a_{ave})^T \tag{16}$$

Where:

T - denotes the transpose

a_{ave} -average vector of all feature vectors in the word-class

$$a_{ave} = (a_1 + ... + a_m)/m \tag{17}$$

PCA is applied to the input variables by transforming the sample points of original variables into a transformed space such that variances of the variables are maximum along the coordinates of the transformed space. Projections of the variables on axes, on which variances are high, contain much information regarding variation of the data.

4.2 K-mean Clustering

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because different locations cause different results to occur. So, the better choice is to place them as far away as possible from each other. The next step is to take each point belonging to a given dataset and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group is formed. At this point, it is needed to re-calculate k new centroids as bar centers of the clusters resulting from the previous step. After getting these k new centroids, a new binding has to be done between the same dataset points and the nearest new centroid. A loop has been generated. As a result of this loop it can be noticed that the k centroids change their location step by step until no more changes are made. In other words, centroids no longer move. This algorithm aims at minimizing an objective function, in this case a squared error function. The objective function is given as

$$J=\sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)}-c_j\right\|^2 \tag{18}$$

Where:

$\left\|x_i^{(j)}-c_j\right\|^2$ - is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j. It is an indicator of the distance of the n data points from their respective cluster centers.

Clustering of the input data space is achieved by using the k-mean method, and then takes only a discriminant sample from the resulting clustering scheme to perform the learning process. Unfortunately, there is no general theoretical solution to find the optimal number of clusters for any given dataset. A simple approach is to compare the results of multiple runs with different k classes and choose the best one according to a given criterion. By definition, by increasing k, smaller error function values result, but there is also the increased risk of over-fitting [8-10].

4.2 Neural Network Feature Extraction (NNFE)

NNFE can be used effectively for non-liner dimensionality reduction, thereby overcoming some of the limitation of linear PCA. Fig. 4 shows a schematic NN structure used to achieve dimensionality reduction. This NN has d input neurons, d output neurons, and m hidden neurons, with $m<d$. The training dataset is represented by d dimensional vector x, and is presented sequentially as the input and output of the NN which is designed and trained to reproduce its own input vector [7]. This is done by means of mapping each input vector onto itself. The optimization goal is to minimize the sum of square error of the form

$$E=\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{d}\{y_k(x^n)-x_k^n\}^2 \tag{19}$$

Where:

d - is the number of features

N - is the number of training datasets

After the neural network is trained, the vector V of the hidden variables represents the extracted feature of the system. This work can be understood as a two stage successive mapping operation F_1 and F_2. F_1 is a projection from the large d - dimensional input onto a small m - dimensional subspace S. This is done from the input to the hidden layer of the NN. F_2 is a mapping from that subspace back to the full d - dimensional space. This process is represented by the hidden to output layer of the network.

Fig. 5 reveals a simple geometric interpretation for the case $d=3$ (three dimensional spaces) and $m=2$ (two dimensional space). The mapping F_1 defines a projection of points from the original three dimensional space onto the two dimensional subspace S. Points in $S(F2)$ are then mapped through F_2 back to the three dimensional space of the original data. Because learning in this network is not linear, there is the risk of finding a suboptimal local minimum for the error function. Also, one must take care to choose an appropriate number of m - neurons. Assuming that two-stage mapping has been successively achieved, the middle layer of this network

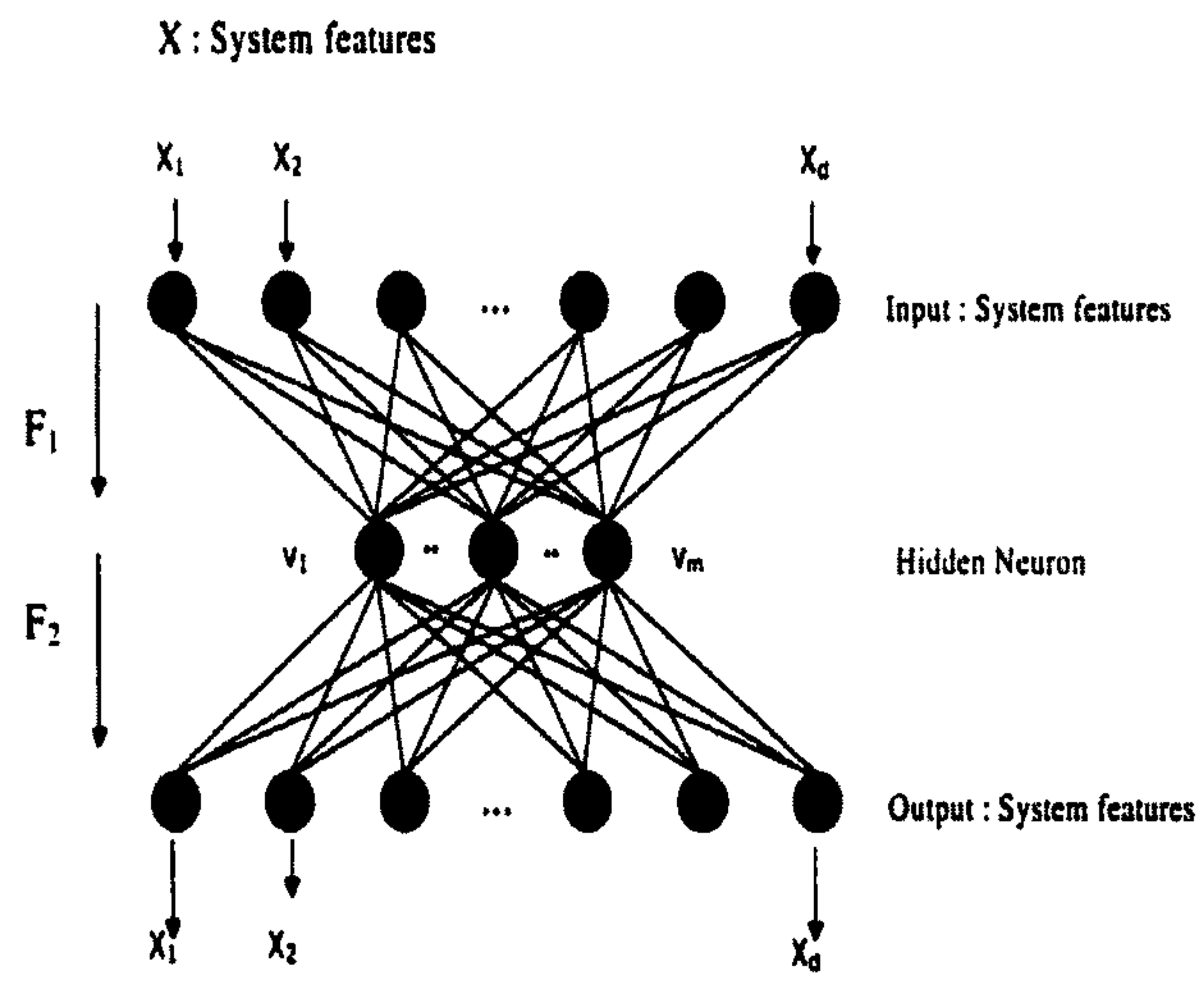


Fig. 4. Neural network feature extraction

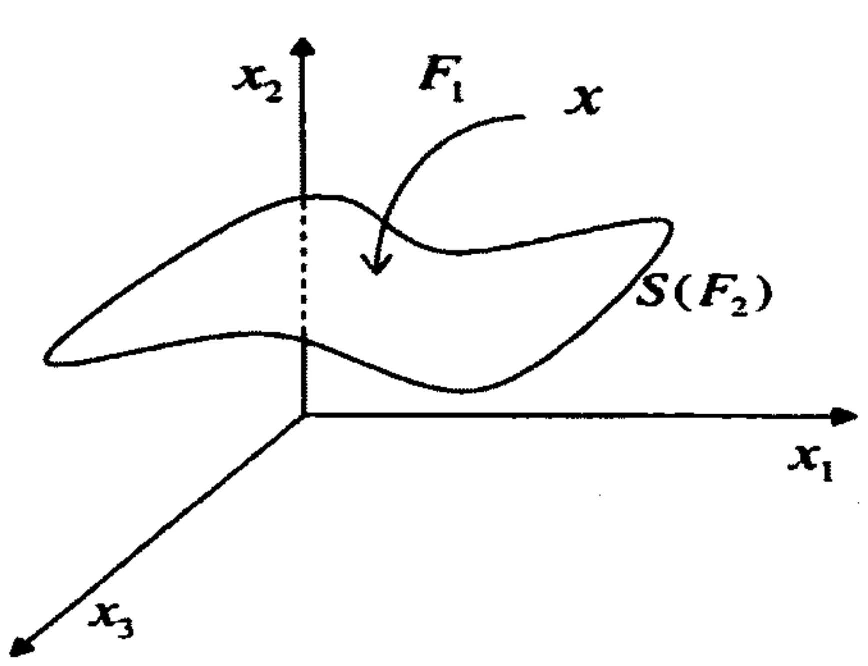


Fig. 5. Geometric interpretation of the mapping

represents the extracted features which are the availed representation of the original system. Such a network is said to be an auto-encoder. This feature extraction method works with any activation function [7].

4.2 Proposed Neural Network Weight Extraction (NNFE) Method

This method is similar in some ways to the NNFE which is the feature extraction based ANN method. Here, the ANN is trained by minimizing the mean secure error according to the back-propagation training algorithm. The procedure of the proposed NNWE is described as follows:

i. Determine the original training datasets of the neural network for vulnerability assessment.
ii. Use a vulnerability index based on PSL to select the critical contingence. From the original training datasets, select sub-datasets whose PSLs show that the system is vulnerable. These sub-datasets are applied as training sets to train the first ANN.
iii. Obtain the weights matrix which are represented by the following equation:

$$\begin{bmatrix} v_1 = x_1w_{11} + x_2w_{21} + x_3w_{31} + x_4w_{41} + \ldots + x_nw_{n1} \\ v_2 = x_1w_{12} + x_2w_{22} + x_3w_{32} + x_4w_{42} + \ldots + x_nw_{n2} \\ \vdots \\ v_m = x_1w_{1m} + x_2w_{2m} + x_3w_{3m} + x_4w_{4m} + \ldots + x_nw_{nm} \end{bmatrix} \quad (20)$$

Where, v_m is the value of the hidden neuron and x_n is the input variable.

The weights matrix w extracted from (20) is given as,

$$w = \begin{bmatrix} w_{11} & w_{21} & w_{31} & w_{41}\cdots & w_{n1} \\ w_{12} & w_{22} & w_{32} & w_{42}\cdots & w_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{1m} & w_{2m} & w_{3m} & w_{4m}\cdots & w_{nm} \end{bmatrix} \quad (21)$$

iv. Using the weights matrix given by Equation (21), determine the values of the reduced feature sets by using the following equation:

$$R = w\,p = \begin{bmatrix} w_{11} & w_{21} & w_{31} & w_{41}\cdots & w_{n1} \\ w_{12} & w_{22} & w_{32} & w_{42}\cdots & w_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{1m} & w_{2m} & w_{3m} & w_{4m}\cdots & w_{nm} \end{bmatrix} \times \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \\ \vdots \\ x_{n1} \end{bmatrix} \quad (22)$$

R is a set of data that has the number of reduced features which are similar to the number of hidden neurons of the first ANN. In other words, the number of reduced features in this case depends on the number of hidden neurons selected in the first ANN. After determining these reduced features, they are then used as the input features to the second ANN for assessing vulnerability of the power systems.

The NNWE method uses a simple approach to find the optimal number of m - hidden neurons so as to reduce the n - dimension of the original training datasets by comparing the ANN results of multiple runs with different number of hidden neurons and choose the best one according to the performance of the second ANN. An advantage of using NNWE compared to NNFE is that training time is reduced in which the method does not require training of ANN using the original datasets to extract the reduced feature dimension. The NNFE method, however, needs to train the ANN using all the original datasets to determine the extracted features and this increases the ANN training time.

5. Results and Discussion

The multilayer feed-forward NN with back propagation training algorithm is trained using the dataset presented by each of four feature extraction methods previously described in Section 4. In this study, the total data generated from simulations is comprised of 413 features, which are real and reactive power flows and total generated real and reactive powers. After feature extraction, the features are reduced to 40 to 90 percent of the original features. In the selection of number of hidden neurons, there is no fixed rule to determine the number of neurons. The neurons are increased gradually until a satisfactory NN performance is obtained. It is found that the most accurate NN result can be achieved by using 50 hidden neurons. As for the output neuron, only 1 neuron is considered to represent the vulnerability index, PSL. The NN training was implemented using the Matlab version 7 on a Pentium4 1.50 GHz, with 384 Mb of RM memory. Fig. 6 shows the training error during the NN learning process.

The results of NN performance using the four extraction methods, namely, PCA, K-means clustering (KMC), NNFE, and NNWE are evaluated in terms of absolute errors as shown in Table 1 and Fig. 7. Table I indicates the comparison of NN results corresponding to different numbers of data features obtained from using the four different feature extraction methods. Results shown in Table I indicate that the absolute errors calculated by using NNFE and NNWE are smaller than the PCA and KMC methods. This proves that the PCA and KMC feature

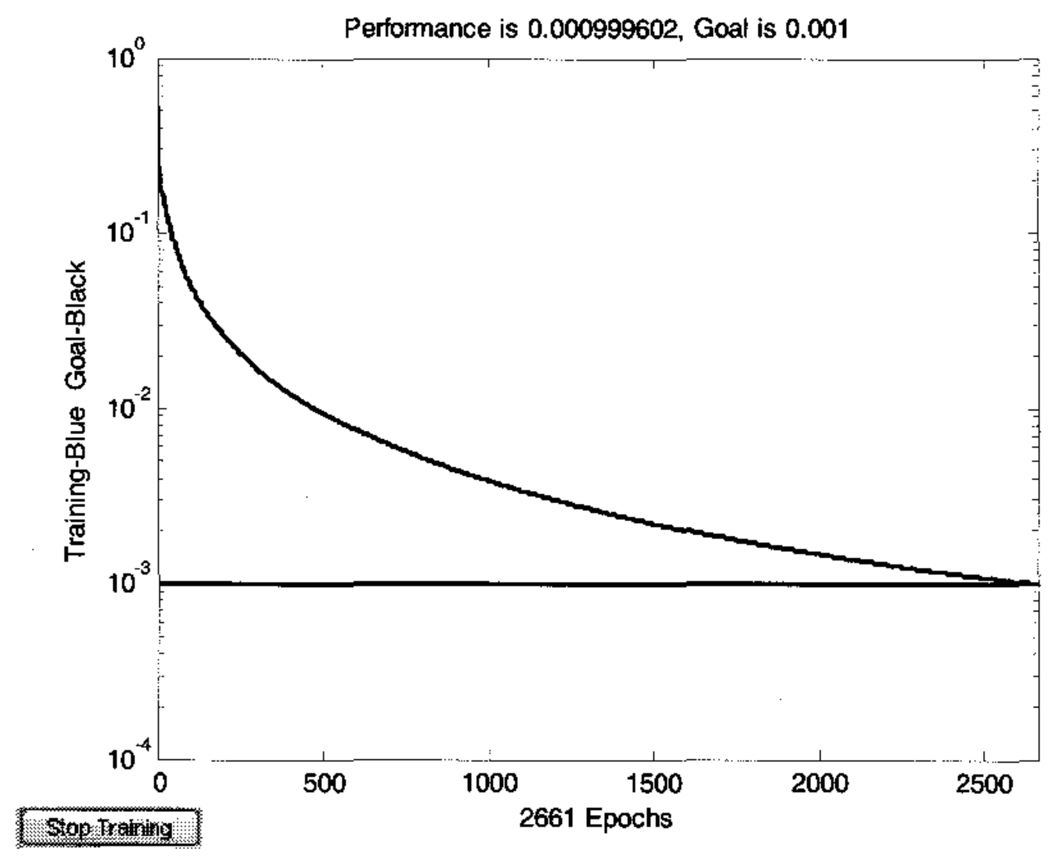


Fig. 6. Training errors during the learning process

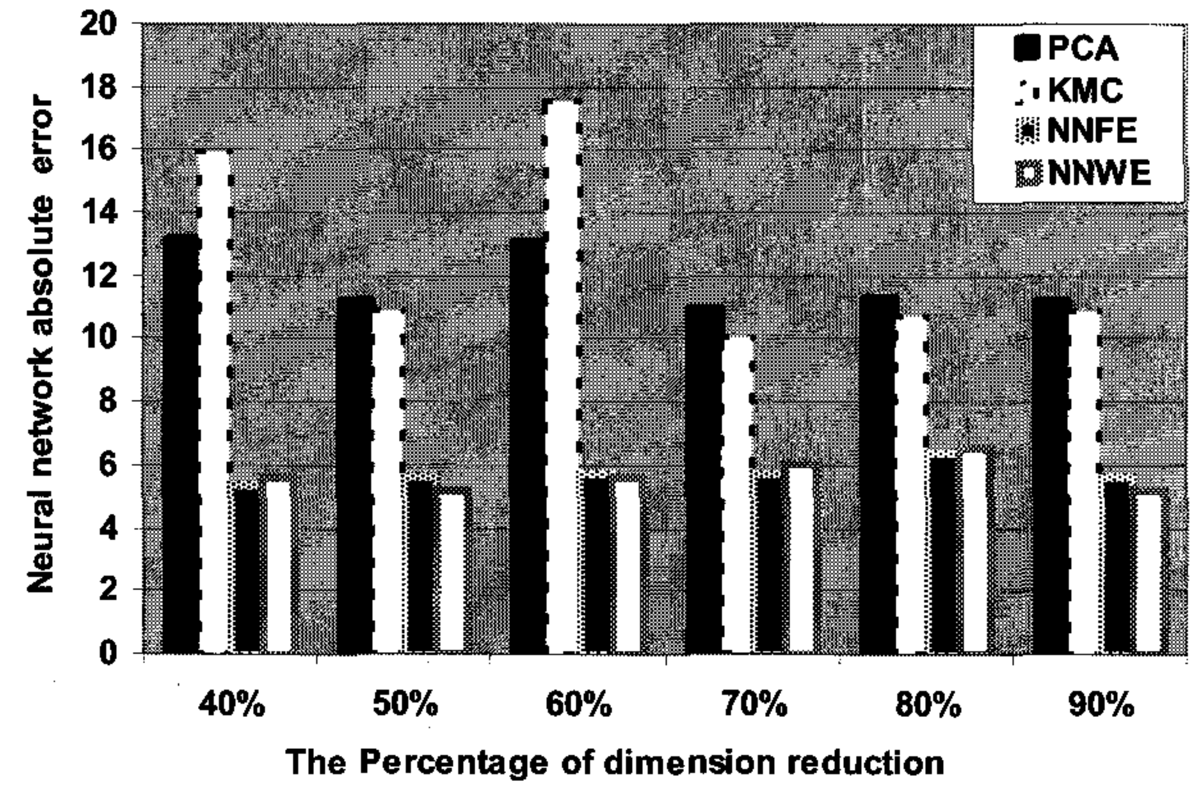


Fig. 7. Comparison of various feature extraction methods

Table 1. Comparison of various feature extraction methods

The Percentage of	PCA	KMC	NNFE	NNWE
Dimension Reduction	Neural Network Absolute Error %			
40% (247 Features)	13.16	15.93	5.36	5.63
50% (206 Features)	11.22	10.90	5.63	5.22
60% (165 Features)	13.06	17.56	5.70	5.64
70% (123 Features)	10.97	10.04	5.70	6.06
80% (82 Features)	11.31	10.74	6.40	6.59
90% (41 Features)	11.22	10.90	5.63	5.22

extraction methods are not suitable for use in assessing the vulnerability of power systems. The NN performance results are considered acceptable because in practice, it is not always an appropriate strategy to try to find a neural network with a 100 classification rate because a high classification rate for the training patterns sometimes leads to poor performance of NN testing. This observation is known as over-fitting of NN during training.

Fig. 8 indicates the performance of the NN using the proposed NNWE feature extraction method. The performance is measured according to the classification rate (CR) which is given by the following expression:

$$CR = \left(\frac{C}{U}\right)100 \tag{23}$$

Where:

C - number of testing sets classified correctly

U - total number of testing datasets

Referring to Fig. 8, the CR for all reduced datasets in the range of (80-92) % gives an absolute error of less than 0.06. However, reduced dataset 206 provides the best classification rate (92%). It can be concluded that the NNWE method gives NN performance with reasonable accuracy by using less number of features. Here the reduced sets of features represent the entire system, with a minimum loss of information. This is an advantage of the proposed feature extraction method which can be used for solving NN based vulnerability assessment of large size power systems.

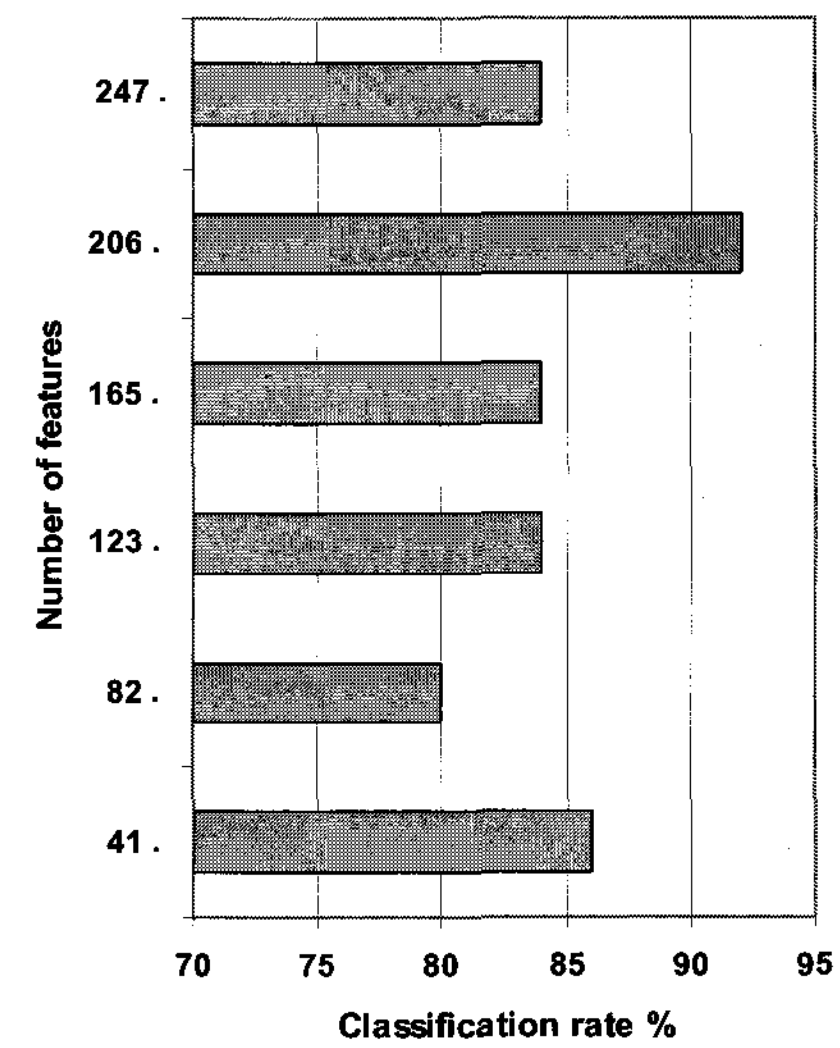


Fig. 8. NN Percentage of classification using NNWE

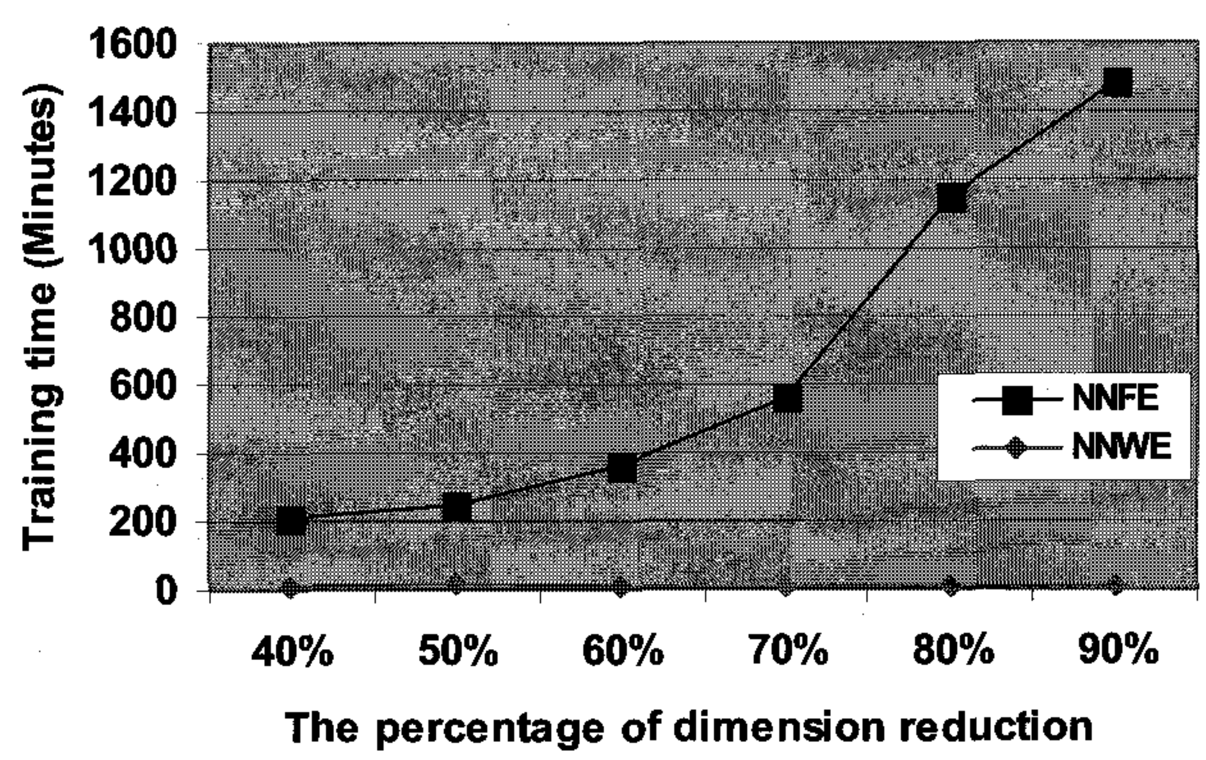


Fig. 9. NN training times considering NNFE and NNWE feature extraction methods

Table 2. Training time for feature extraction

Dimension Reduction	Training Time (minute)	
The original vector has 413 features	NNFE	NNWE
40% (247 - Features)	206.13	12.86
50% (206 - Features)	250.44	16.11
60% (165 - Features)	359.03	11.52
70% (123 - Features)	560.07	11.56
80% (82 - Features)	1148.49	8.8
90% (41 - Features)	1484.7	10.84

Fig. 9 and Table 2 reveal the training times taken in training the NN using NNFE and NNWE methods. Comparing the training times, it can be seen that training times for the NN using NNWE are much smaller than the NN using NNFE.

6. Conclusion

This paper has presented the application of ANN for vulnerability assessment on a large sized power system for the purpose of developing a fast vulnerability assessment tool that can be used for real-time applications. Various feature extraction methods have been considered so as to evaluate the effectiveness of the proposed NNWE method. These feature extraction methods can reduce the number of input features to the ANN thus speeding up the ANN training process. Test results prove that the ANN gives best performance in terms of classification accuracy when implemented with the proposed NNWE method. Thus, the proposed NNWE method is suitable as a feature extraction method for use with ANN for assessing the vulnerability of power systems. The advantages of the NNWE method are not only in terms of dimensionality reduction and faster ANN training but also better ANN performance.

References

[1] V. L. Paccar and M. J. Rider, "Artificial neural networks for solving the power flow problem in electric power systems", *Electric Power Systems Research Elsevier,* Vol. 62, pp. 139–144, June 2002.

[2] C. A. Jensen, Mohamed A. El-Sharkawi and R. J. Marks, "Power system security assessment using neural networks: feature selection using Fisher discrimination", *IEEE Transaction on Power System,* Vol. 16, no 1, pp. 757–763, November 2001.

[3] H. Sawhney and B. Jeyasurya, "On-line transient stability assessment using artificial neural network", *Proceedings of the IEEE on Large Engineering Systems Conference,* Vol. 2, pp. 76–80, July 2004.

[4] M. Delimar, I. Pavii and Z. Hebel, "Artificial neural networks in power system topology recognition", *Proceedings of the IEEE EUROCON Computer as a Tool,* Ljubljana, Slovenia. pp. 287–291, September 2003.

[5] S. C. Tan and C. P. Lim, "Application of an adaptive neural network with symbolic rule extraction to fault detection and diagnosis in a power generation plant", *IEEE Transaction on Energy Conversion,* Vol. 19, no 2, pp. 369–377, June 2004.

[6] G. R. Yousefi. "An application of artificial neural networks in power load modeling", *Proceedings of the IEEE Power Engineering Society General Meeting.* Iran Vol. 1, pp. 308–313, June 2005.

[7] M. Kim "Application of computational intelligence to power system vulnerability assessment and adaptive protection using high speed communication", PhD Thesis, University of Washington, USA, 2002.

[8] Simon P. Teeuwsen, "Oscillatory stability assessment of power system using computational intelligence" PhD Thesis. University Duishdurg-Essen, Germany 2005.

[9] K. Wagsta, C. Cardie, S. Rogers and S. Schroed. "Constrained K-means Clustering with Background Knowledge", *Proceedings of the Eighteenth International Conference on Machine Learning,* pp. 577-584, 2001.

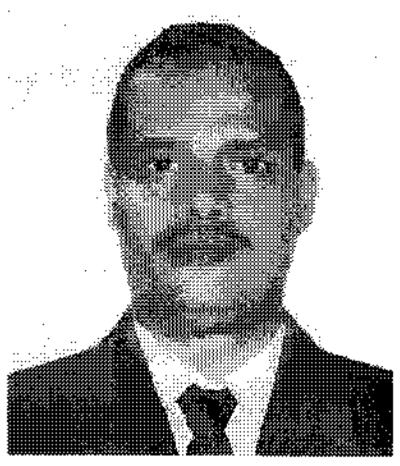
[10] K. M. Faraoun and A. Boukelif "Neural networks learning improvement using the K-means clustering algorithm to detect network intrusions", *International Journal of Computational Intelligence,* Vol. 3, pp. 161–168, June 2006.

[11] A. Kalousis, J. Prados and M. Hilario, "Stability of feature selection algorithms", *Proceedings of the Fifth IEEE International Conference on Data Mining.* pp. 8, November 2005.

[12] Simon Haykin, "Neural Network: A comprehensive foundation," Prentice-Hall, 1999, pp. 156–175.

[13] A. M. A. Haidar, A. Mohamed, A. Hussain. "Vulnerability assessment of a large sized power system using a new index based on power system loss," *European Journal of Scientific Research.* Vol. 17, no. 1, pp. 61–72, 2007.

[14] M. B. Gülmezo˘glu, V. Dzhafarov, and A. Barkana, "The common vector approach and its relation to principal component analysis", *IEEE Transaction on Speech and Audio Processing.* Vol. 9, no. 6, pp. 655–662, September 2001.



Ahmed M. A Haidar

He received his B.S and M.S degrees in Electrical Engineering from Donetsk Polytechnic Institute and Donetsk State University of Technology, Ukraine, Russia, in 1993 and 1994, respectively. He is a Ph.D. candidate in Electrical Engineering at the National University of Malaysia (UKM). His employment experience includes machines, transformers, and power system operation and control. His areas of interest are power system vulnerability and computational intelligence. He is a member of IEEE.

Azah Mohamed

She received her B.Sc from the University of London in 1978 and her M.Sc and Ph.D. degrees from University Malaya in 1988 and 1995, respectively. She is currently a Professor at the Department of Electrical, Electronic and Systems Engineering, National University of Malaysia (UKM). Her main research interests are in power system security, power quality, and artificial intelligence. She is a Senior Member of IEEE.

Aini Hussain

She received her BSEE, M.Sc., and Ph.D. degrees in Electrical Engineering from Louisiana State University, USA; University of Manchester Institute of Science and Technology (UMIST), England; and National University of Malaysia (UKM), respectively. She is currently a Professor in the Department of Electrical, Electronic and Systems Engineering, National University of Malaysia (UKM). Her research interests include signal processing, pattern recognition, and neural networks.