# Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 7 (21): 3208-3216, 2007
ISSN 1812-5654


# Transient Stability Assessment of a Power System Using PNN and LS-SVM Methods

Noor Izzri Abdul Wahab, Azah Mohamed and Aini Hussain
Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan, Malaysia

**Abstract:** This study presents transient stability assessment of electrical power system using two artificial neural network techniques which are Probabilistic Neural Network (PNN) and Least Squares Support Vector Machine (LS-SVM). Transient stability of a power system is first determined based on the generator relative rotor angles obtained from time domain simulation outputs. Simulations were carried out on the IEEE 9-bus test system considering three phase faults on the system. The data collected from the time domain simulations are then used as inputs to the PNN and LS-SVM. Both networks are used as a classifier to determine whether the power system is stable or unstable. To verify the effectiveness of the proposed PNN and LS-SVM methods, they are compared with the Multi Layer Perceptron Neural Network (MLPNN). Results show that the PNN gives faster and more accurate transient stability assessment compared to the LS-SVM network and MLPNN in terms of classification results.

**Key words:** Transient stability assessment, artificial neural network, probabilistic neural network, least squares support vector machines

## INTRODUCTION

Recent blackouts in the USA, some European and Asian countries have illustrated the importance and need of more frequent and thorough power system stability study. Nowadays, power systems have evolved through continuing growth in interconnection, use of new technologies and controls. Due to increased operations which may cause power system to be in highly stressed conditions, the need for dynamic security assessment of power systems is arising. Transient Stability Assessment (TSA) is part of dynamic security assessment of power systems which involves the evaluation of the ability of a power system to remain in equilibrium under severe but credible contingencies. These evaluations aim to assess the dynamic behavior of a power system in a fast and accurate way. Methods normally employed to assess TSA are by using time domain simulation, direct and artificial intelligence methods. Time domain simulation method is implemented by solving the state space differential equations of power networks and then determines transient stability. Direct methods such as the transient energy method determine transient stability without solving differential state space equations of power systems. These two methods are considered most accurate but are time consuming and need heavy computational effort. Presently, the use of Artificial Neural Network (ANN) in TSA has gained a lot of interest among researchers due to its ability to do parallel data processing, high accuracy and fast response.

In transient stability assessment, the Critical Clearing Time (CCT) is a very important parameter in order to maintain the stability of power systems. The CCT is the maximum time duration that a fault may occur in power systems without failure in the system so as to recover to a steady state operation. Earlier ANN works carried out in TSA used the feed forward Multi Layer Perceptron (MLP) with back propagation learning algorithm to determine the CCT of power systems (Pothisarn and Jiriwibhakorn, 2003; Sanyal, 2004). Bettiol *et al.* (2003) proposed the use of radial basis function networks to estimate the CCT. Another method to assess power system transient stability using ANN is by means of classifying the system into either stable or unstable states for several contingencies applied to the system (Krishna and Padiyar, 2000; Sanyal, 2004). ANN method based on fuzzy ARTMAP architecture has also been used for TSA of a power system (Silveira *et al.*, 2003). Boudour and Hellal (2005) proposed the use of combined supervised and unsupervised learning for evaluating dynamic security of a power system based on the concept of stability margin. Sawhney and Jeyasurya (2004) used ANN to map the operating condition of a power system based on a transient stability index which provides a measure of stability in power systems. Support Vector Machine (SVM) is another ANN method used for TSA (Moulin *et al.*, 2004; Wang *et al.*, 2005) in which the method has several advantages such as automatic determination of the number of hidden neurons, fast convergence rate and good generalization capability.

**Corresponding Author:** Noor Izzri Abdul Wahab, Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan, Malaysia

In this study, two new ANN methods are proposed and developed for transient stability assessment of power systems using PNN and Least Squares Support Vector Machine (LS-SVM). Both ANN methods are considered new applications in transient stability assessment of power systems. The procedures of transient stability assessment using PNN and LS-SVM are described and the performance of the PNN and LS-SVM is compared with the MLPNN so as to verify the effectiveness of these two methods. Both the MLP and PNN networks were developed using the MATLAB Neural Network Toolbox, whereas the LS-SVM was developed using the LS-SVM Matlab Toolbox (Suykens *et al.*, 2002).

## MATHEMATICAL MODEL OF MULTIMACHINE POWER SYSTEM

The differential equations to be solved in power system stability analysis using the time domain simulation method are the nonlinear ordinary equations with known initial values. Using the classical model of machines, the dynamic behavior of an *n*-generator power system can be described by the following equations:

$$M_i \frac{d^2\delta_i}{dt^2} = P_{mi} - P_{ei} \tag{1}$$

However,

$$\dot{\delta}_i = \omega_i \tag{2}$$

By substituting (2) in (1), Eq. 1 becomes

$$M_i \dot{\omega}_i = P_{mi} - P_{ei} \tag{3}$$

Where:
$\delta_i$ = Rotor angle of machine i,
$\omega_i$ = Rotor speed of machine i,
$P_{mi}$ = Mechanical power of machine i,
$P_{ei}$ = Electrical power of machine i,
$M_i$ = Moment of inertia of machine i.

Equation 3 is then solved by using a time domain simulation program through step-by-step integration so as to produce time response of all state variables.

## BACKGROUND ANN THEORY

Here the exolanation of two ANN methods, namely the PNN and the LS-SVM are given. Both the ANN methods are used as classifiers to determine the stability of a power system.

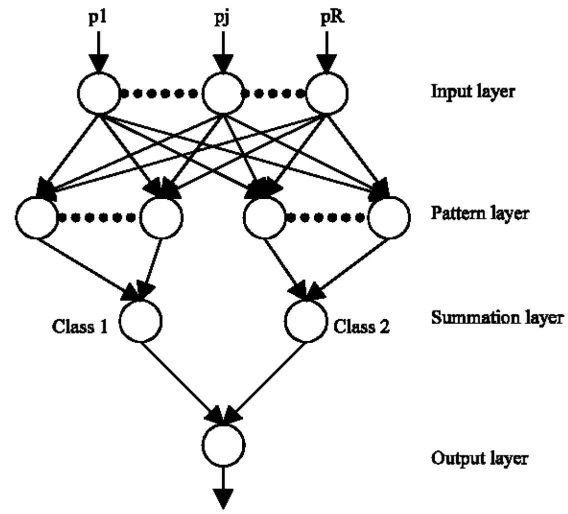


Fig. 1: PNN Architecture

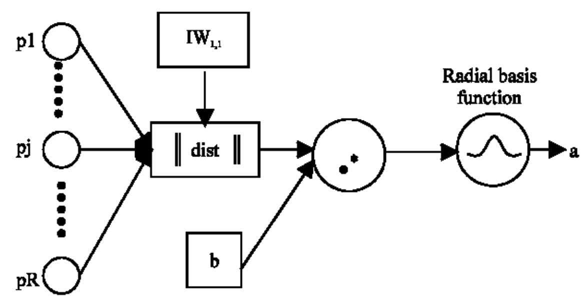


Fig. 2: PNN pattern layer

**Probabilistic Neural Network:** PNN which is a class of Radial Basis Function (RBF) network is useful for automatic pattern recognition, nonlinear mapping and estimation of probabilities of class membership and likelihood ratios (Specht, 1992). It is a direct continuation of the work on Bayes classifiers (Burrascano, 1991) in which it is interpreted as a function that approximates the probability density of the underlying example distribution. The PNN consists of nodes with four layers namely input, pattern, summation and output layers as shown in Fig. 1. The input layer consists of merely distribution units that give similar values to the entire pattern layer.

For this work, RBF is used as the activation function in the pattern layer of the PNN.

The ‖dist‖ box shown in Fig. 2 subtracts the input weights, $IW_{1,1}$, from the input vector, p and sums the squares of the differences to find the Euclidean distance. The differences indicate how close the input is to the vectors of the training set. These elements are multiplied element by element, with the bias, b, using the dot product (.*) function and sent to the radial basis transfer function. The output a is given as:

$$a = \text{radbas}(\ \| IW_{1,1} - p \|\ b) \tag{4}$$

Where, radbas is the radial basis activation function which can be written in general form as:

$$\text{radbas}(n) = e^{n^2} \tag{5}$$

The training algorithm used for training the RBF is the orthogonal least squares method which provides a systematic approach to the selection of RBF centers (Chen *et al*., 1991).

The summation layer shown in Fig. 1 simply sums the inputs from the pattern layer which correspond to the category from which the training patterns are selected as either class 1 or class 2. Finally, the output layer of the PNN is a binary neuron that produces the classification decision. As for this work, the classification is either class 1 for stable cases or class 2 for unstable cases.

**Least Squares Support Vector Machine (LS-SVM):** LS-SVM is a reformulation of the standard SVM (Suykens and Vandewalle, 1999). The reformulation leads to solving a set of linear equations which is easier to solve than SVM quadratic equations. The reformulation does not result in SVM losing any of its advantage. LS-SVM map input vectors to a higher dimensional space where a maximal separating hyperplane is constructed. Its mathematical formulations are described in this section.

Given the training data set, $\{x_k, y_k\}^N_{k=1}$, where, $x_k \in \mathbb{R}^n$ represent k-th input pattern and $y_k \in \mathbb{R}$ is the k-th output pattern, the LS-SVM aims at constructing a classifier of the form,

$$y(x) = \text{sign}\left[\sum_{k=1}^{N} \alpha_k y_k \psi(x, x_k) + b\right] \tag{6}$$

Where, $\alpha_k$ are positive real constant and b is a real constant.

$$\Psi(x,x_k) = \exp\left\{\frac{-\|x - x_k\|^2}{2\sigma^2}\right\}$$

is the RBF kernel which is considered in this study.

The least squares version to the SVM classifier is done by formulating the classification problem as:

$$\min_{w,b,e} J(w,b,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2 \tag{7}$$

subject to equality constraints,

$$y_k \left[ w^T \varphi(x_k) + b \right] = 1 - e_k, k = 1, \ldots, N \tag{8}$$

Where, $\varphi(x_k)$ is a nonlinear function which maps the input space into a higher dimensional space.

By using the Mercer's Theorem, this function is related to $\Psi(x,x_k)$ as follows,

$$\varphi(x)^T \varphi(x_k) = \psi(x, x_k) \tag{9}$$

Equation 7 and 8 lead to Karush-Kuhn-Tucker systems and can be written as the solution to the following set of linear equations,

$$\begin{bmatrix} 0 & -Y^T \\ Y & ZZ^T + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{1} \end{bmatrix} \tag{10}$$

Where:

$Z = [\varphi(x_1)^T y_1; \ldots; \varphi(x_N)^T y_N]$,
$Y = [y_1; \ldots; y_N]$,
$\vec{1} = [1; \ldots; 1]$,
$e = [e_1; \ldots; e_N]$,
$\alpha = [\alpha_1; \ldots; \alpha_N]$.

Mercer's Theorem can be applied again to the matrix $\Omega_{kl} = ZZ^T$ where,

$$\begin{aligned} \Omega_{kl} &= y_k y_l \varphi(x_k)^T \varphi(x_l) \\ &= y_k y_l \psi(x_k, x_l) \end{aligned} \tag{11}$$

Hence, the solution to the classifier as given in Eq. 6 can be found by solving the linear set of Eq. 10 and 11 instead of using quadratic programming for solving the equation as is the case with SVM. The LS-SVM network developed in this work uses the LS-SVM Matlab Toolbox (Suykens *et al*., 2002) in which the training of LS-SVM is based on the iterative solver conjugate gradient algorithm.

**Performance evaluation of PNN and LS-SVM networks:** Performance of the developed PNN and LS-SVM networks can be gauged by calculating the error of the actual and desired test data. Firstly, error is defined as:

$$\text{Error, } E_n = |\text{Desired output, } DO_n - \text{Actual output, } AO_n| \tag{12}$$

Where:

n = The test data number.

The desired output is the known output data or target data used for comparing with the neural network output. Meanwhile, the actual output (AO) is the output obtained from the trained neural network.

From Eq. 12, the mean error can be calculated using,

$$\text{Percentage Mean Error(ME)} = \sum_{n=1}^{N} \frac{E_n}{N} \times 100 \qquad (13)$$

Where:
N = The total number of test data.

The percentage classification error is given by,

$$\text{Percentage classification error} = \frac{\text{No. of misclassification ot test data}}{N} \times 100 \qquad (14)$$

## MATERIALS AND METHODS

In the PNN and LS-SVN methods used for transient stability assessment, the IEEE 9-bus test system is used for verification of the methods. Before the PNN and LS-SVM implementation, time domain simulations considering several contingencies were carried out for the purpose of gathering the training data sets. Simulations were done by using the MATLAB-based PSAT software (Milano, 2005). Time domain simulation method is chosen to assess the transient stability of a power system because it is the most accurate method compared to the direct method. In PSAT, power flow is used to initialize the states variable before commencing time domain simulation. The differential equations to be solved in transient stability analysis are nonlinear ordinary equations with known initial values. To solve these equations, the techniques available in PSAT are the Euler and trapezoidal rule techniques. In this work, the trapezoidal technique is used considering the fact that it is widely used for solving electro-mechanical differential algebraic equations (Milano, 2007).

The type of contingency considered is the three-phase balanced faults created at various locations in the system at any one time. When a three-phase fault occur at any line in the system, a breaker will operate and the respective line will be disconnected at the Fault Clearing Time (FCT) which is set by a user. The FCT is set randomly by considering whether the system is stable or unstable after a fault is cleared. According to (Anderson and Fouad 2003), if the relative rotor angles with respect to the slack generator remain stable after a fault is cleared, it implies that FCT < CCT and the power system is said to be stable but if the relative angles go out of step after a fault is cleared, it means that FCT > CCT and the system is unstable.

**Transient stability simulation on the test system:** Figure 3 shows the IEEE 9-bus system in which the data used for this work is obtained from Anderson and Fouad

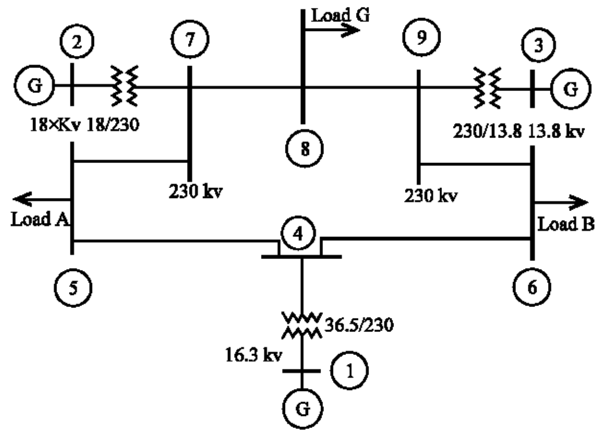


Fig. 3: IEEE 9 bus System

(2003). The system consists of three Type-2 synchronous generators with AVR Type-1, six transmission lines, three transformers and three loads.

Figure 4 shows examples of the time domain simulation results illustrating stable and unstable cases. A three phase fault is said to occur at time t = 1 second at bus 7. In Fig. 4a the FCT is set at 1.08 sec while in Fig. 4b the FCT is set at 1.25 sec. The relative rotor angles of the generators oscillate and the system is said to be stable (Fig. 4) whereas the relative rotor angles of the generators go out of step after a fault is cleared and the system becomes unstable (Fig. 4b). It can be deduced from Fig. 4 that the FCT setting is an important factor to determine the stability of power systems. If FCT is set at a shorter time than the CCT of the line, the system is stable; otherwise the system will be unstable.

**Data preprocessing:** The simulation on the system for a fault at each line runs for 5 sec at a time step $\Delta t$, set at 0.001 sec. The fault is set to occur at one second from the beginning of the simulation. Data for each contingency is recorded in which one steady state data is taken before a fault occurs and 20 sampled data are taken for one second duration after a fault occurs. There are 25 contingencies simulated on the system and this gives a size of 25×21 or 525 data collected.

The collected data are further analyzed and trimmed down to 468 due to repetitions of data. The one steady state data taken before all faults occur are reduced to one since the values will be the same for all faults. Next, the repetitions are due to faults that occur on the same line. The FCT of the same line are set at four different times, two for stable cases and two for unstable cases. At the start of a fault, same values of data are recorded for all the four faults. A few milliseconds after a fault, the recorded data differ from each other due to different FCT settings. Due to repetitions of data recorded, one data out of the

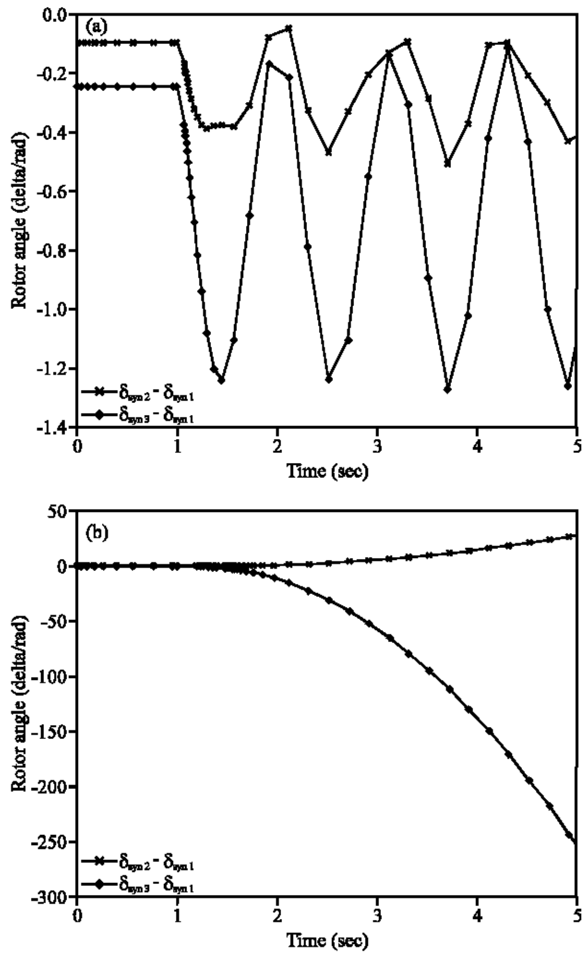


Fig. 4: Relative rotor angle curves of generators for (a) stable and (b) unstable cases

Table 1: Input features selected

| Name of input features | No. of features |
|---|---|
| Relative rotor angles ($\delta_{i-1}$) | 2 |
| Generator speed ($\omega_i$) | 3 |
| $P_{gen}$ and $Q_{gen}$ | 6 |
| $P_{line}$ and $Q_{line}$ | 12 |
| $P_{trans}$ and $Q_{trans}$ | 6 |
| Total No. of features | 29 |

four different FCT settings are kept. These data are denoted as data for stable cases. The data collected are normalized so that they have zero mean and unity variance.

There are 468 sets of data collected from simulations in which a quarter of the data which is 117 are randomly selected for testing and the remaining 351 data are selected for training the neural network.

**Input features selection:** The selection of input features is an important factor to be considered in the ANN implementation. The input features selected for this study are relative rotor angles ($\delta_{i-1}$), motor speed ($\omega_i$), generated real and reactive powers ($P_{gen}$, $Q_{gen}$), real and reactive power flows on transmission line ($P_{line}$, $Q_{line}$) and the transformer powers ($P_{trans}$, $Q_{trans}$). Overall there are 29 input features to the ANN (Table 1). The breakdown of the input features selected for the neural network.

## TEST RESULTS

Here, the results obtained from the PNN and LS-SVM for transient stability assessment are presented. Initially, the PNN results using 29 input features are given and discussed. Then, results obtained from LS-SVM using the same input features as PNN are presented and discussed. For the purpose of evaluating the effectiveness of the PNN and LS-SVM, the results of the multi layer perceptron neural network (MLPNN) are also presented. Finally, comparisons are made between the PNN, LS-SVM and MLPNN results for transient stability assessment.

**PNN results for transient stability assessment:** The PNN developed in this study is used for classifying power system transient stability states in which the PNN classifies 1 for stable cases and 2 for unstable cases. The architecture of the PNN is such that it has 29 input neurons, the hidden layer neurons equal the number of training data which is 351 and with a single output neuron. The PNN testing results using the 29 input features. The shaded cells in the table denote the misclassification of test data (Table 2). From the Table 2 it can be deduced that the false alarm rate is 0.86% and the false dismissal rate is 0.86%. False dismissal rate is the rate of unstable cases assigned to the stable cases and the false alarm rate is the rate of stable cases assigned to the unstable cases. Thus, the total error of misclassification (false alarm rate + false dismissal rate) and the mean error are both 1.71%.

**LS-SVM results for transient stability assessment:** The developed LS-SVM is also used for classifying power system transient stability states in which it classifies 1 for stable cases and 2 for unstable cases similar to that of PNN. The architecture of LS-SVM is such that it has 29 input neurons, 351 hidden neurons which is the same as the number of training data and a single output neuron. The trained 351 hidden neurons are used to classify the 117 test data. Table 3 shows the LS-SVM testing results in which the shaded cells in the table denote the misclassification of test data. From the table, it can be deduced that the false alarm rate is 1.71% and the false dismissal rate is 1.71. Thus, the percentage error of misclassification and the mean error are both 3.42%.

Table 2: PNN testing results

| Test data | Desired output | PNN output | Test data | Desired output | PNN output | Test data | Desired output | PNN output |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 40 | 1 | 1 | 79 | 1 | 1 |
| 2 | 1 | 1 | 41 | 1 | 1 | 80 | 2 | 1 |
| 3 | 1 | 1 | 42 | 2 | 2 | 81 | 2 | 2 |
| 4 | 1 | 1 | 43 | 2 | 2 | 82 | 2 | 2 |
| 5 | 1 | 1 | 44 | 2 | 2 | 83 | 2 | 2 |
| 6 | 2 | 2 | 45 | 2 | 2 | 84 | 1 | 1 |
| 7 | 2 | 2 | 46 | 1 | 1 | 85 | 1 | 1 |
| 8 | 2 | 2 | 47 | 1 | 1 | 86 | 1 | 1 |
| 9 | 2 | 2 | 48 | 1 | 1 | 87 | 1 | 1 |
| 10 | 1 | 1 | 49 | 1 | 1 | 88 | 1 | 1 |
| 11 | 1 | 1 | 50 | 1 | 1 | 89 | 2 | 2 |
| 12 | 1 | 1 | 51 | 2 | 2 | 90 | 2 | 2 |
| 13 | 1 | 1 | 52 | 2 | 2 | 91 | 2 | 2 |
| 14 | 1 | 1 | 53 | 2 | 2 | 92 | 2 | 2 |
| 15 | 2 | 2 | 54 | 2 | 2 | 93 | 2 | 2 |
| 16 | 2 | 2 | 55 | 1 | 1 | 94 | 1 | 1 |
| 17 | 2 | 2 | 56 | 1 | 1 | 95 | 1 | 1 |
| 18 | 2 | 2 | 57 | 1 | 1 | 96 | 1 | 1 |
| 19 | 1 | 1 | 58 | 1 | 1 | 97 | 1 | 1 |
| 20 | 1 | 1 | 59 | 1 | 1 | 98 | 1 | 1 |
| 21 | 1 | 1 | 60 | 2 | 2 | 99 | 1 | 1 |
| 22 | 1 | 1 | 61 | 2 | 2 | 100 | 1 | 1 |
| 23 | 1 | 1 | 62 | 2 | 2 | 101 | 1 | 1 |
| 24 | 2 | 2 | 63 | 2 | 2 | 102 | 1 | 1 |
| 25 | 2 | 2 | 64 | 2 | 2 | 103 | 1 | 1 |
| 26 | 2 | 2 | 65 | 1 | 1 | 104 | 2 | 2 |
| 27 | 2 | 2 | 66 | 1 | 1 | 105 | 2 | 2 |
| 28 | 1 | 2 | 67 | 1 | 1 | 106 | 2 | 2 |
| 29 | 1 | 1 | 68 | 1 | 1 | 107 | 2 | 2 |
| 30 | 1 | 1 | 69 | 1 | 1 | 108 | 1 | 1 |
| 31 | 1 | 1 | 70 | 2 | 2 | 109 | 1 | 1 |
| 32 | 1 | 1 | 71 | 2 | 2 | 110 | 1 | 1 |
| 33 | 2 | 2 | 72 | 2 | 2 | 111 | 1 | 1 |
| 34 | 2 | 2 | 73 | 2 | 2 | 112 | 1 | 1 |
| 35 | 2 | 2 | 74 | 2 | 2 | 113 | 2 | 2 |
| 36 | 2 | 2 | 75 | 1 | 1 | 114 | 2 | 2 |
| 37 | 1 | 1 | 76 | 1 | 1 | 115 | 2 | 2 |
| 38 | 1 | 1 | 77 | 1 | 1 | 116 | 2 | 2 |
| 39 | 1 | 1 | 78 | 1 | 1 | 117 | 2 | 2 |

**MLPNN results for transient stability assessment:** The architecture of the MLPNN is such that it has 29 input neurons representing the 29 input features, one hidden layer with 13 neurons using the hyperbolic tangent transfer function and a single output neuron. The mean squared error is used as a goal for training the neural network which is set at 0.03. The training algorithm used for this network is the resilient back propagation algorithm (Riedmiller and Braun, 1993). The performance goal was met at 41,050 epochs with a training time of 25 min 32 sec.

From the Table 4 the calculated mean error is 6%. As shown in Table 4, some of the MLPNN outputs are not crisp 0 or 1 but in the range 0 to 1, where 0 indicates the system is stable and 1 when the system is stable. So for classification purpose, a decision rule is used such that if the MLPNN output is in the range of 0.9 to 1.1 (±10%), it will indicate that the system is stable (class 1) whereas if the MLPNN output is in the range of -0.1 to 0.1 (±10%), it means that the system is unstable (class 2). For MLPNN output outside this range of values, it is considered as misclassified. The column indicated by C in the table shows the classification of the converted MLPNN outputs so that they can be easily compared with the desired outputs to determine the accuracy of the MLPNN. Classes 1 and 2 are used in column C instead of 1 and 0 for stable and unstable classification so that the results conformed to the results obtained from PNN and LS-SVM. By using this decision rule the number of misclassified data is 13 out of 117 test data, which is 11.1%. The shaded cells in the table are the respective misclassified data which are denoted as x in the column C.

**Comparison of neural network results in transient stability assessment:** It can be concluded that the performance of PNN is better compared with LS-SVM and MLPNN in transient stability assessment of the 9 bus power system (Table 5).

Table 3: LS-SVM testing results

| Test data | Desired output | LS-SVM output | Test data | Desired output | LS-SVM output | Test data | Desired output | LS-SVM output |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 40 | 1 | 1 | 79 | 1 | 1 |
| 2 | 1 | 1 | 41 | 1 | 1 | 80 | 2 | 2 |
| 3 | 1 | 1 | 42 | 2 | 2 | 81 | 2 | 2 |
| 4 | 1 | 1 | 43 | 2 | 2 | 82 | 2 | 2 |
| 5 | 1 | 1 | 44 | 2 | 2 | 83 | 2 | 2 |
| 6 | 2 | 2 | 45 | 2 | 2 | 84 | 1 | 1 |
| 7 | 2 | 2 | 46 | 1 | 1 | 85 | 1 | 1 |
| 8 | 2 | 2 | 47 | 1 | 1 | 86 | 1 | 1 |
| 9 | 2 | 2 | 48 | 1 | 1 | 87 | 1 | 1 |
| 10 | 1 | 1 | 49 | 1 | 1 | 88 | 1 | 1 |
| 11 | 1 | 1 | 50 | 1 | 1 | 89 | 2 | 2 |
| 12 | 1 | 1 | 51 | 2 | 2 | 90 | 2 | 2 |
| 13 | 1 | 1 | 52 | 2 | 2 | 91 | 2 | 2 |
| 14 | 1 | 1 | 53 | 2 | 2 | 92 | 2 | 2 |
| 15 | 2 | 2 | 54 | 2 | 2 | 93 | 2 | 2 |
| 16 | 2 | 2 | 55 | 1 | 1 | 94 | 1 | 1 |
| 17 | 2 | 2 | 56 | 1 | 1 | 95 | 1 | 2 |
| 18 | 2 | 2 | 57 | 1 | 1 | 96 | 1 | 1 |
| 19 | 1 | 1 | 58 | 1 | 1 | 97 | 1 | 1 |
| 20 | 1 | 1 | 59 | 1 | 1 | 98 | 1 | 1 |
| 21 | 1 | 1 | 60 | 2 | 1 | 99 | 1 | 1 |
| 22 | 1 | 1 | 61 | 2 | 2 | 100 | 1 | 1 |
| 23 | 1 | 1 | 62 | 2 | 1 | 101 | 1 | 1 |
| 24 | 2 | 2 | 63 | 2 | 2 | 102 | 1 | 1 |
| 25 | 2 | 2 | 64 | 2 | 2 | 103 | 1 | 1 |
| 26 | 2 | 2 | 65 | 1 | 1 | 104 | 2 | 2 |
| 27 | 2 | 2 | 66 | 1 | 1 | 105 | 2 | 2 |
| 28 | 1 | 2 | 67 | 1 | 1 | 106 | 2 | 2 |
| 29 | 1 | 1 | 68 | 1 | 1 | 107 | 2 | 2 |
| 30 | 1 | 1 | 69 | 1 | 1 | 108 | 1 | 1 |
| 31 | 1 | 1 | 70 | 2 | 2 | 109 | 1 | 1 |
| 32 | 1 | 1 | 71 | 2 | 2 | 110 | 1 | 1 |
| 33 | 2 | 2 | 72 | 2 | 2 | 111 | 1 | 1 |
| 34 | 2 | 2 | 73 | 2 | 2 | 112 | 1 | 1 |
| 35 | 2 | 2 | 74 | 2 | 2 | 113 | 2 | 2 |
| 36 | 2 | 2 | 75 | 1 | 1 | 114 | 2 | 2 |
| 37 | 1 | 1 | 76 | 1 | 1 | 115 | 2 | 2 |
| 38 | 1 | 1 | 77 | 1 | 1 | 116 | 2 | 2 |
| 39 | 1 | 1 | 78 | 1 | 1 | 117 | 2 | 2 |

Table 4: MLPNN results using 29 input features

| Test data | Desired output | MLPNN output | Error | C | Test data | Desired output | MLPNN output | Error | C | Test data | Desired output | MLPNN output | Error | C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1.000 | 0.000 | 1 | 40 | 1 | 1.000 | 0.000 | 1 | 79 | 1 | 0.989 | 0.011 | 1 |
| 2 | 1 | 1.000 | 0.000 | 1 | 41 | 1 | 0.999 | 0.001 | 1 | 80 | 0 | 0.030 | 0.030 | 2 |
| 3 | 1 | 1.000 | 0.000 | 1 | 42 | 0 | 0.006 | 0.006 | 2 | 81 | 0 | 0.001 | 0.001 | 2 |
| 4 | 1 | 1.000 | 0.000 | 1 | 43 | 0 | 0.000 | 0.000 | 2 | 82 | 0 | 0.000 | 0.000 | 2 |
| 5 | 1 | 1.000 | 0.000 | 1 | 44 | 0 | 0.000 | 0.000 | 2 | 83 | 0 | -0.002 | 0.002 | 2 |
| 6 | 0 | 0.000 | 0.000 | 2 | 45 | 0 | 0.000 | 0.000 | 2 | 84 | 1 | 0.946 | 0.054 | 1 |
| 7 | 0 | 0.000 | 0.000 | 2 | 46 | 1 | 0.996 | 0.004 | 1 | 85 | 1 | 1.000 | 0.000 | 1 |
| 8 | 0 | 0.000 | 0.000 | 2 | 47 | 1 | 1.061 | 0.061 | 1 | 86 | 1 | 0.999 | 0.001 | 1 |
| 9 | 0 | 0.000 | 0.000 | 2 | 48 | 1 | 0.144 | 0.856 | X | 87 | 1 | 1.002 | 0.002 | 1 |
| 10 | 1 | 0.996 | 0.004 | 1 | 49 | 1 | 0.960 | 0.040 | 1 | 88 | 1 | 1.003 | 0.003 | 1 |
| 11 | 1 | 0.997 | 0.003 | 1 | 50 | 1 | 0.663 | 0.337 | X | 89 | 0 | 0.004 | 0.004 | 2 |
| 12 | 1 | 1.002 | 0.002 | 1 | 51 | 0 | 0.006 | 0.006 | 2 | 90 | 0 | -0.012 | 0.012 | 2 |
| 13 | 1 | 1.002 | 0.002 | 1 | 52 | 0 | 0.010 | 0.010 | 2 | 91 | 0 | 0.005 | 0.005 | 2 |
| 14 | 1 | 1.095 | 0.095 | 1 | 53 | 0 | -0.002 | 0.002 | 2 | 92 | 0 | -0.020 | 0.020 | 2 |
| 15 | 0 | 0.004 | 0.004 | 2 | 54 | 0 | -0.003 | 0.003 | 2 | 93 | 0 | -0.077 | 0.077 | 2 |
| 16 | 0 | 0.002 | 0.002 | 2 | 55 | 1 | 0.999 | 0.001 | 1 | 94 | 1 | 0.666 | 0.334 | X |
| 17 | 0 | 0.001 | 0.001 | 2 | 56 | 1 | 0.997 | 0.003 | 1 | 95 | 1 | 0.330 | 0.670 | X |
| 18 | 0 | -0.001 | 0.001 | 2 | 57 | 1 | 1.002 | 0.002 | 1 | 96 | 1 | 1.873 | 0.873 | X |
| 19 | 1 | 1.006 | 0.006 | 1 | 58 | 1 | 1.000 | 0.000 | 1 | 97 | 1 | 0.618 | 0.382 | X |
| 20 | 1 | 1.000 | 0.000 | 1 | 59 | 1 | 1.007 | 0.007 | 1 | 98 | 1 | 1.032 | 0.032 | 1 |
| 21 | 1 | 1.000 | 0.000 | 1 | 60 | 0 | 0.221 | 0.221 | X | 99 | 1 | 1.000 | 0.000 | 1 |
| 22 | 1 | 1.000 | 0.000 | 1 | 61 | 0 | 0.194 | 0.194 | X | 100 | 1 | 0.998 | 0.002 | 1 |
| 23 | 1 | 0.999 | 0.001 | 1 | 62 | 0 | 0.250 | 0.250 | X | 101 | 1 | 1.000 | 0.000 | 1 |
| 24 | 0 | -0.033 | 0.033 | 2 | 63 | 0 | -0.010 | 0.010 | 2 | 102 | 1 | 1.000 | 0.000 | 1 |

Table 4: Continued

| Test data | Desired output | MLPNN output | Error | C | Test data | Desired output | MLPNN output | Error | C | Test data | Desired output | MLPNN output | Error | C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 25 | 0 | -0.005 | 0.005 | 2 | 64 | 0 | 0.005 | 0.005 | 2 | 103 | 1 | 1.000 | 0.000 | 1 |
| 26 | 0 | 0.004 | 0.004 | 2 | 65 | 1 | 0.999 | 0.001 | 1 | 104 | 0 | 0.022 | 0.022 | 2 |
| 27 | 0 | -0.001 | 0.001 | 2 | 66 | 1 | 0.997 | 0.003 | 1 | 105 | 0 | 0.004 | 0.004 | 2 |
| 28 | 1 | 0.257 | 0.743 | X | 67 | 1 | 1.002 | 0.002 | 1 | 106 | 0 | -0.004 | 0.004 | 2 |
| 29 | 1 | 1.046 | 0.046 | 1 | 68 | 1 | 1.003 | 0.003 | 1 | 107 | 0 | -0.004 | 0.004 | 2 |
| 30 | 1 | 0.975 | 0.025 | 1 | 69 | 1 | 1.000 | 0.000 | 1 | 108 | 1 | 0.151 | 0.849 | X |
| 31 | 1 | 1.125 | 0.125 | X | 70 | 0 | 0.272 | 0.272 | X | 109 | 1 | 1.000 | 0.000 | 1 |
| 32 | 1 | 1.000 | 0.000 | 1 | 71 | 0 | -0.009 | 0.009 | 2 | 110 | 1 | 1.000 | 0.000 | 1 |
| 33 | 0 | -0.033 | 0.033 | 2 | 72 | 0 | -0.001 | 0.001 | 2 | 111 | 1 | 1.002 | 0.002 | 1 |
| 34 | 0 | 0.018 | 0.018 | 2 | 73 | 0 | -0.001 | 0.001 | 2 | 112 | 1 | 1.000 | 0.000 | 1 |
| 35 | 0 | -0.006 | 0.006 | 2 | 74 | 0 | -0.001 | 0.001 | 2 | 113 | 0 | 0.004 | 0.004 | 2 |
| 36 | 0 | 0.002 | 0.002 | 2 | 75 | 1 | 1.000 | 0.000 | 1 | 114 | 0 | 0.000 | 0.000 | 2 |
| 37 | 1 | 0.999 | 0.001 | 1 | 76 | 1 | 1.001 | 0.001 | 1 | 115 | 0 | 0.000 | 0.000 | 2 |
| 38 | 1 | 1.000 | 0.000 | 1 | 77 | 1 | 1.001 | 0.001 | 1 | 116 | 0 | 0.000 | 0.000 | 2 |
| 39 | 1 | 1.000 | 0.000 | 1 | 78 | 1 | 1.001 | 0.001 | 1 | 117 | 0 | 0.000 | 0.000 | 2 |

Table 5: Summary of PNN, LS-SVM and MLPNN results

| Network | PNN | LS-SVM | MLPNN |
|---|---|---|---|
| Input features | 29 | 29 | 29 |
| False alarms (%) | 1 (0.86%) | 2 (1.71%) | - |
| False dismissals (%) | 1 (0.86%) | 2 (1.71%) | - |
| Mean error | 0.0171 | 0.0342 | 0.0600 |
| Misclassification | 2 (1.71%) | 4 (3.42%) | 13(11.1%) |
| Training time | 1.32 sec | 1.7 sec | 25 min 32 sec |

The mean error for PNN is 0.017 compared to 0.0342 for LS-SVM network and the percentage classification errors are also less for PNN (1.71%) compared to 3.42% for LS-SVM, respectively. For MLPNN, there are no false alarms and false dismissals but the mean error and misclassification percentage are higher than both PNN and LS-SVM which are 0.06 and 11.1% respectively. In terms of training time, the PNN has the shortest training time (1.32 sec) compared to the time taken to train the LS-SVM (1.7 sec) and MLPNN (25 min 32 sec). The difference in training time for PNN and LS-SVM is insignificant compared to the time taken to train the MLPNN. In general, the performance of PNN and LS-SVM are better compared to MLPNN and that PNN gives the best performance among the three methods.

## CONCLUSION

The use of PNN and LS-SVM has been proposed for transient stability assessment of the 9-bus power system by means of classifying the system into either stable or unstable states for several three phase faults applied to the system. Time domain simulations were first carried out to generate training data for both neural networks and to determine transient stability state of a power system by visualizing the generator relative rotor angles. The PNN and LS-SVM networks are then compared with the MLPNN so as to evaluate its effectiveness in transient stability assessment. The performances of PNN and LS-SVM compared to the MLPNN are better in terms of mean and misclassification errors and training time. Results also show that among the three methods used in this work, the PNN gives the best performance in terms of accuracy in classifying the transient stability states. Thus, the PNN and LS-SVM networks are promising methods for transient stability assessment of power systems.

## REFERENCES

Anderson, P.M. and A.A. Fouad, 2003. Power System Control and Stability. 2nd Edn. IEEE Press, USA.

Bettiol, A.L., A. Souza, J.L. Todesco and J.R. Tesch Jr., 2003. Estimation of critical clearing times using neural networks. Proc. IEEE Bologna Power Technol. Conf., 3: 6.

Boudour, M. and A. Hellal, 2005. Combined use of supervised and unsupervised learning for power system dynamic security mapping. Eng. Applic. Artificial Intell., 18: 673-683.

Burrascano, P., 1991. Learning vector quantization for the probabilistic neural network. IEEE Trans. Neural Networks, 2: 458-461.

Chen, S., C.F.N. Cowan and P.M. Grant, 1991. Orthogonal least squares learning algorithm for radial basis function networks. IEEE Trans. Neural Networks, 2: 302-309.

Krishna, S. and K.R. Padiyar, 2000. Transient stability assessment using artificial neural networks. Proc. IEEE Int. Conf. Ind. Technol., 2000: 627-632.

Milano, F., 2005. An open source power system analysis toolbox. IEEE Trans. Power Syst., 20: 1199-1206.

Milano, F., 2007. Documentation for Power System Analysis Toolbox (PSAT). http://www.power.uwaterloo.ca/~fmilano/downloads.htm.

Moulin, L.S., A.P.A. da Silva, M.A. El-Sharkawi and R.J. Marks, 2004. Support vector machines for transient stability analysis of large-scale power systems. IEEE Trans. Power Syst., 19: 818-825.

Pothisarn, C. and S. Jiriwibhakorn, 2003. Critical clearing time determination EGAT system using artificial neural networks. Proc. IEEE Power Eng. Soc. Gen. Meeting, 2: 731-736.

Riedmiller, M. and H. Braun, 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. IEEE Int. Conf. Neural Networks, 1: 586-591.

Sanyal, K.K., 2004. Transient stability assessment using neural network. IEEE Int. Conf. Electric Utility Deregulation, Restructuring and Power Technologies, Hong Kong, pp: 633-637.

Sawhney, H. and B. Jeyasurya, 2004. On-Line transient stability assessment using artificial neural network. Large Engineering Systems Conference on Power Engineering, pp: 76-80.

Silveira, M.C.G., A.D.P. Lotufo and C.R. Minussi, 2003. Transient stability analysis of electrical power systems using a neural network based on fuzzy ARTMAP. Proc. IEEE Bologna PowerTech Conf., 3: 7.

Specht, D.F., 1992. Enhancements to probabilistic neural networks. Int. Joint Conf. Neural Networks, 1: 525-532.

Suykens, J.A.K. and J. Vandewalle, 1999. Least squares support vector machines classifier. Neural Processing Lett., 9: 293-300.

Suykens, J.A.K., T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle, 2002. Least Squares Support Vector Machines. World Scientific. Singapore.

Wang, X., S. Wu, Q. Li and X. Wang, 2005. v-SVM for Transient stability assessment in power systems. Proc. Autonomous Decentralized Syst., pp: 356-363.