A Computational Intelligence-based Suite for Vulnerability Assessment of Electrical Power Systems

Ahmed M. A. Haidar[a] Azah Mohamed[b] Federico Milano[c,*]

[a] *University Malaysia Pahang, Pahang, Malaysia*

[b] *University Kebangsaan Malaysia, Selangor, Malaysia*

[c] *University of Castilla-La Mancha, Ciudad Real, Spain*

Abstract

This paper discusses the feasibility of implementing computational intelligence algorithms for power system analysis in an open source environment. The scope is specially oriented to education, training and research. In particular, the paper describes a software package, namely Computational Intelligence Applications to Power System (CIAPS), that implements a variety of heuristic techniques for vulnerability assessment of electrical power systems. CIAPS is based on Matlab and suited for analysis and simulation of small to large size electric power systems. CIAPS is used for solving power flow, optimal power flow, contingency analysis based on artificial neural networks and fuzzy logic techniques. A variety of illustrative examples are given to show the features of the developed software tool.

Key words:
Contingency Analysis, Vulnerability Assessment and Control, Computational Intelligence, Artificial Neural Networks, Fuzzy Logic.

* Corresponding author

Email addresses: ahaidar67@yahoo.com (Ahmed M. A. Haidar), azah@eng.ukm.my (Azah Mohamed), Federico.Milano@uclm.es (Federico Milano).

1 Introduction

1.1 Motivation

Electrical power systems are the biggest and most complex systems ever built by man. In recent years, electrical power systems have faced a deregulation process that has further increased the complexity of common operations such as monitoring, security assessment and emergency control. Due to the huge number of variables and scenarios that have to be taken into account for a proper security assessment of electrical power systems, heuristic techniques based on computational intelligence (e.g., fuzzy logic, artificial neural networks, etc.) are of increasing importance [1]. This paper describes a modular approach for power systems vulnerability assessment based on heuristic techniques. In particular, the paper focuses on a software package, namely Computational Intelligence Applications to Power System (CIAPS). This tool is oriented to education as well as practitioner training and research.

1.2 Literature review

The term *computational intelligence* (CI) generally refers to a group of techniques that attempt to mimic certain aspects of human brain or nature behavior. These include several techniques such as Artificial Neural Network (ANN), Fuzzy Logic (FL), Genetic Algorithms and Particle Swarm Optimization [1]. Reference [2] provides an extensive bibliographic review of recent applications of CI to power system analysis with particular regard to the optimal power flow problem. This paper focuses on ANN and FL, which have proved to be promising methodologies for solving certain complex problems in power systems, where conventional methods have not achieved the desired speed and accuracy.

ANN techniques have been used since late 90s for solving power system security-related problems [3–9]. In recent years, an ANN model known as *radial basis function neural network* (RBFNN) has become increasingly popular due to its structural simplicity and training efficiency. RBFNN has been used for solving power system problems such as for protection of transmission lines [10], locating faults in transmission lines [11] and transient stability assessment of power systems [12].

Fuzzy logic and fuzzy expert systems are also widely used for solving power system security assessment, power system protection and automation systems. Relevant bibliography on this topic is [13–25]. Fuzzy expert system have been also used to assess voltage stability control and the optimum amount of load shedding [26]. Furthermore, fuzzy logic is also used for power system control and stability (e.g. [27]).

Despite the abundance in the literature of proposals of CI-based techniques for power system analysis, there is a lack of a common benchmark software tool that can function as a general-purpose board for CI-based algorithms. This paper addresses this issue through a variety of ANN and FL-based examples.

1.3 Software Development Environment

In the last decade, the Matlab language and scientific environment has become a standard tool for flexible technical computing [28]. Matlab incorporates a large number of domain specific toolboxes such as fuzzy logic toolbox, neural network toolbox, control toolbox real-time workshop, matrix-oriented programming, excellent plotting capabilities, etc. Furthermore Simulink offers a set of tools for modeling, simulating and analyzing dynamic systems [29].

These features make Matlab/Simulink an attractive choice for power systems research and education. Further details on the pros and cons of developing educational power system software applications in Matlab language can be found in [30, 31]. As a matter of fact, a number of Matlab-based proprietary toolboxes, as well as open source research and educational power system tools have been developed. These are Power System Toolbox (PST) [32], MatPower [33], Voltage Stability Toolbox (VST) [34], MatEMTP [35], SimPowerSystems [36], Power Analysis Toolbox [37], and the Educational Simulation Tool [38], and PSAT [39]. Among these, MatPower, VST and PSAT are open source and can be freely downloaded.

None of the Matlab-based tools cited above includes AI techniques.

1.4 Contributions

Due to the lack of publicly-available software packages that provide computational intelligence techniques for power system analysis, this paper proposes and describes the software suite CIAPS. This tool has been developed using Matlab and mainly focuses on vulnerability assessment. The main purpose of CIAPS is education and practitioner training. At this aim, CIAPS comes with a complete graphical user interface that eases assessing power system vulnerability, ANN-based vulnerability relief and FL-based controlled load shedding.

With education in mind, CIAPS is not intended as a closed software package, but rather as a main board for future development and extension of CI algorithms and methods for power system analysis, thus not limited to vulnerability assessment.

The paper also introduces a novel technique for data reduction based on weight extraction using ANN.

1.5 Paper organization

The paper is organized as follows: an overview of the proposed software tool as well as of models and algorithms is given in Section 2. Section 3 illustrates simulations and discusses results of the proposed tools through small and large size power systems. Finally, Section 4 presents relevant conclusions.

2 Outlines of the Proposed Tool and Algorithms

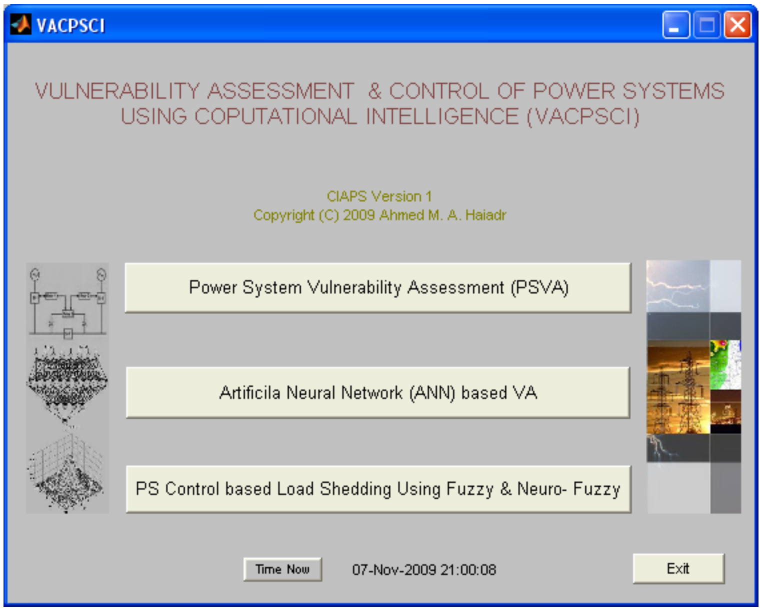
CIAPS is a package of Matlab scripts for solving power flow, optimal power flow, contingency analysis, vulnerability assessment based on ANN and FL simulations. CIAPS is intended as a simulation tool that is easy to use for students and educators. CIAPS was developed based on some of the available Matlab programs for power system analysis [33, 39, 40] and computational intelligence toolboxes [41, 42].

CIAPS is launched at the Matlab prompt using the command `ciaps`. Afterwards, all main structures are initialized and the main window GUI is displayed (see Fig. 1). The main window works as a hub from where the user can solve available AI analyses. In particular CIAPS currently include the following tools:

(1) Power system vulnerability assessment (see Subsection 2.1).
(2) Artificial neural networks-based vulnerability assessment (see Subsection 2.2).
(3) Load shedding control based on fuzzy and neuro-fuzzy logic (see Subsection 2.3).

CIAPS has been designed with the following goals:

(1) To provide a fast and accurate simulation environment using improved computational intelligence techniques.
(2) To allow advanced users building additional models and functions, which is useful for rapid development of prototype models and testing of research ideas.


VACPSCI
Power System Vulnerability Assessment (PSVA)
Artificila Neural Network (ANN) based VA
PS Control based Load Shedding Using Fuzzy & Neuro- Fuzzy
Time Now
Exit

VULNERABILITY ASSESSMENT & CONTROL OF POWER SYSTEMS
USING COPUTATIONAL INTELLIGENCE (VACPSCI)

CIAPS Version 1
Copyright (C) 2009 Ahmed M. A. Haiadr

07-Nov-2009 21:00:08

Fig. 1. Main user interface for the CIAPS software package.

(3) To provide a graphical and easy-to-understand presentation of vulnerability assessment results.

To use CIAPS, the core Matlab and Simulink packages as well as the neural networks and fuzzy toolboxes are needed. The full source code as well as further details on how installing and running CIAPS can be obtained directly contacting the first author of this paper. Furthermore a pre-parsed version of CIAPS is available at [43]. Following subsections describe the algorithms implemented in CIAPS.

2.1 Power System Vulnerability Assessment

The power system vulnerability assessment implemented in CIAPS is based on a simple yet efficient index, namely the power system losses vulnerability index (PSLVI). The main idea underlying PSLVI is that system losses typically increase after the occurrence of a contingency [44]. In particular, the outage of transmission lines, transformers or generators lead to overload remaining lines and cause an increase of power losses. Furthermore, load shedding can, in some idiosyncratic cases, have an effect similar to line outages. In fact, a load outage can cause reactive powers loops among generators and, thus, lead to local loss increase in certain transmission lines [40]. Thus, the more critical is the contingency, the more losses increase. The PSLVI evaluates total power system losses as a measure of power systems vulnerability [45].

The computation of the proposed PSLVI is as follows:

$$\mathrm{PSLVI} = \frac{S_{BCL}}{S_{CCL} + S_{IL} + S_{LD} + \sum\limits_{i=1}^{n} S_{LGO,i} W_{G,i} + \sum\limits_{i=1}^{m} S_{LLO,i} W_{L,i}} \tag{1}$$

where:

S_{BCL}	system power loss in MVA at base case;
S_{CCL}	system power loss in MVA at contingency case;
S_{IL}	increase in total load in MVA;
S_{LD}	amount of load disconnected in MVA;
$S_{LGO,i}$	loss of generated MVA due to generator i outage;
$S_{LLO,i}$	loss of transported MVA due to line i outage;
$W_{G,i}$	weight of generator i power output;
$W_{L,i}$	weight of line i power influence;
n	number of generators;
m	number of lines.

Equation (1) indicates that PSLVI range is $(0, 1)$. In fact, we assume that, for each contingency, total losses are greater than those for the base case. These values can be screened and ranked by a system operator. For example, if the value of PSLVI is close to 1.0, then the system can be considered *invulnerable*, whereas if the PSLVI value is close to 0, then the system can be considered *vulnerable*. The system operator can define a threshold value of PSLVI based on experience and on the system configuration. Observe that, in (1), also the weight of each generator $W_{G,i}$ and line $W_{L,i}$ is chosen based on operator experience.

2.2 Artificial Neural Network-based Vulnerability Assessment

In order to explain the ANN-based vulnerability assessment that has been implemented in CIAPS, this subsection describes two powerful ANN models, namely the generalized regression neural network and the multilayer perceptron neural network.

The performance of each proposed ANN model is measured according to the classification rate [46]. CIAPS evaluates this rate in terms of accuracy, i.e. the error between the actual and the desired test data. Tolerable limits of the absolute error for training and testing data are 7 and 6 % respectively.

Finally this section describes how these models are used by CIAPS for defining the level of vulnerability of a given contingency based on the PSLVI described in the previous section.

2.2.1 Generalized Regression Neural Network

A powerful neural network model is the generalized regression neural network (GRNN) that has a simple architecture of four layers known as *input*, *patterns*, *summation* and *output* layers (see Fig. 2). The number of input units in the first layer is equal to independent factors or variables. The first layer is fully connected to the pattern layer, whose output is a measure of the distance of the input from the stored patterns. Each pattern layer unit is connected to two neurons in the summation layer, known as S summation neuron and D summation neuron. The S summation neuron computes the sum of the weighted outputs of the pattern layer while the D summation neuron calculates the unweighted outputs of the pattern neurons [47]. For D-summation neuron, the connection weight is set to unity. The output layer merely divides the output of each S-summation neuron by that of each D-summation neuron, yielding the predicted value expressed as:

$$\hat{y}_i = \frac{\sum\limits_{i=1}^{n} y_i \exp[-D(x, x_i)]}{\sum\limits_{i=1}^{n} \exp[-D(x, x_i)]} \tag{2}$$

where n is the number of independent input variables, y_i is the target output value corresponding to the i^{th} input pattern and the Gaussian D function is defined as:

$$D(x, x_i) = \sum_{j=1}^{p} \left(\frac{x_j - x_{ij}}{\sigma_j} \right)^2 \tag{3}$$

where p is the total number of training patterns and σ_j is generally referred to as the *smoothing*, *width* or *spread* parameter, whose optimal value is often determined experimentally. A correct tuning of the spread parameter value is important because if the spread is too low, the network requires many neurons and computing time increases, while if the spread if too high, accuracy flaws. At this aim, CIAPS allows adjusting the spread in order to optimize performance and accuracy.

2.2.2 Multilayer Perceptron Neural Network

One of the most popular models of ANN is the multilayer perceptron neural network (MLPNN) with back propagation training algorithm [48]. Typically, one or more layers are included between the input and output layers. These layers are called *hidden layers*. Learning is achieved by systematically modifying the weights and biases of the neural network to improve the networks output response to acceptable levels. The back-propagation training algorithm

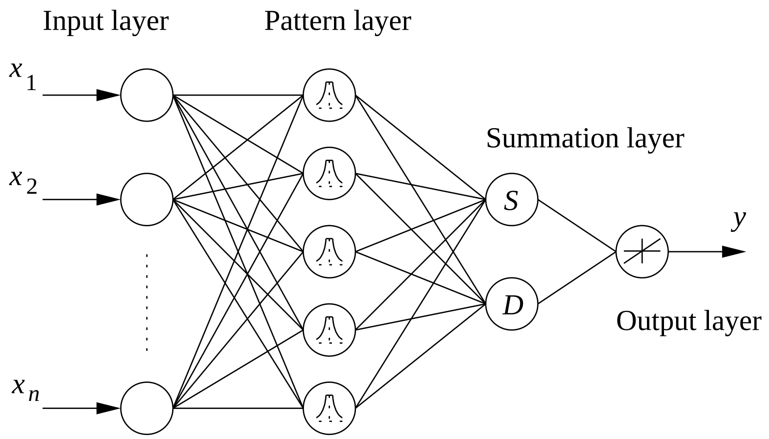


Fig. 2. Four-layer GRNN architecture.

is designed to minimize a sum-of-squares error, as follows:

$$E = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{d} \{y_k(x^n) - x_k^n\}^2 \tag{4}$$

where:

d	number of features;
N	number of training data sets;
x	input vector.

In CIAPS, the MLPNN training is achieved by providing the input pattern to the network input layer. The neural network automatically 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, the 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 [49]. CIAPS determines the optimum number of hidden neurons and the activation function using a pruning strategy [50].

2.2.3 Feature Extraction

An important aspect to be considered for achieving an adequate neural network performance is a proper selection and extraction of training data. For large-scale interconnected power systems, the complete state information is too large to efficiently train the ANN and therefore, the training data size must be reduced while preserving relevant information [20]. In general, the reduced set of features must represent the entire system, since a loss of information in the reduced set results in a loss of performance and accuracy of the ANN. The dimensionality reduction cannot be achieved only based on experience, but it should be implemented according to statistical information and correlations among features [51].

There exist several feature extraction methods. In this paper we use the neural network based feature extraction technique since it can be used effectively for non-linear dimensionality reduction. However, a disadvantage of this technique is that the training time for extracting the features is relatively high, especially for large-scale power systems. To overcome this issue, a novel feature extraction technique that we call *neural network weight extraction* is proposed in this section.

Feature extraction is the process of mapping all available features into a composite feature set of lower dimension [45]. The strategy consists in combining features while retaining the characteristics that allow for an accurate classification. The procedure is as follows.

(1) The first step is to define a set of physical quantities (features) that, based on experience, are related to power system vulnerability. These features have to be both measurable in a real power system and available from power utilities.
(2) The next step is to carry out simulations on a power system subjected to a several disturbances and gathering the set of defined features and computing the corresponding system vulnerability index.
(3) Then, inputs and output are normalized in order to fall in the range $[0, 1]$ or $[-1, 1]$, or in order to obtain a zero mean and unity variance. This step is required in order to reduce the possibility of saturation within the ANN structure.
(4) The normalized input and output features are then processed by using the proposed NNWE extraction method before presenting to the ANN for vulnerability assessment. By using the NNWE method, the dimension of the input features can be reduced at a high reduction rate and minimal loss of information.
(5) Finally, the reduced input features are used as input features of the ANN developed for vulnerability assessment of power system. In CIAPS, the features concern the vulnerability assessment of the network based on the PSLVI.

The proposed procedure of the neural network weight extraction (NNWE) method is as follows:

(1) The starting point is to determine the original (i.e., full size) training data sets $\boldsymbol{p}$ ($\boldsymbol{p} \in \mathbb{R}^n$) of the ANN-2 for vulnerability assessment.
(2) From the original training data sets, sub-data sets $\boldsymbol{x}$ ($\boldsymbol{x} \in \mathbb{R}^n$) are selected based on the PSLVI vulnerability measure. These sub-data sets are used as inputs for training the ANN-1 that is used for the following feature

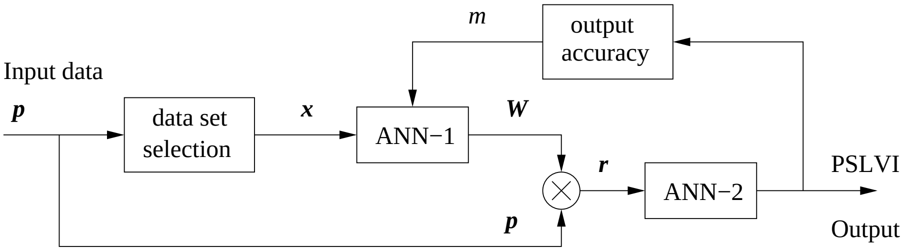


Fig. 3. Proposed NNWE scheme.

extraction.

(3) The ANN-1 is trained by considering different numbers m of hidden neurons at a fixed accuracy.

(4) The weight matrix $\boldsymbol{W}$ ($\boldsymbol{W} \in \mathbb{R}^m \times \mathbb{R}^n$) is computed using sub-data sets $\boldsymbol{x}$. The hidden neurons $\boldsymbol{v}$ ($\boldsymbol{v} \in \mathbb{R}^m$) of the ANN-1 are a linear combinations of the input variable vector $\boldsymbol{x}$ and the weight matrix $\boldsymbol{W}$, as follows:

$$\boldsymbol{v} = \boldsymbol{W}\boldsymbol{x} \tag{5}$$

(5) Using the weights matrix $\boldsymbol{W}$ with the original sets $\boldsymbol{p}$ of input variables, one can determine the values of the reduced feature sets $\boldsymbol{R}$, as follows:

$$\boldsymbol{R} = \boldsymbol{W}\boldsymbol{P} \tag{6}$$

where $\boldsymbol{P}$ ($\boldsymbol{P} \in \mathbb{R}^m \times \mathbb{R}^p$) is a matrix whose columns are the vectors of full size training data $\boldsymbol{p}$, and $\boldsymbol{R}$ ($\boldsymbol{R} \in \mathbb{R}^n \times \mathbb{R}^p$) is a matrix whose columns are the resulting vectors $\boldsymbol{r}$ of reduced training data.

(6) The ANN-2 is trained by means of sets $\boldsymbol{r}$. The most accurate ANN-2 output determines the optimum number m of reduced input feature sets.

The structure of the proposed feature reduction method is illustrated in the synoptic scheme of Fig. 3. Observe that the number of reduced features depends on the number m of hidden neurons in the ANN-1. Clearly, m has to be lower than the number n of the input variables to the ANN-1 in order to reduce computing time when solving the ANN-2 for vulnerability assessment. Once the dimension of the original input features is reduced, the computing time needed for training and getting the ANN-2 convergence is also generally considerably reduced. Observe that this method is a novel contribution of the paper.

2.3 Fuzzy Logic-based Load Shedding Control

The objective of the load shedding problem is to minimize the difference between the generated power at the base case and the generated power considering an $N-1$ contingency criterion. This difference can be expressed as:

$$\Delta S = S_{CC} - S_{BC} \tag{7}$$

where:

S_{CC} generated power in MVA for a given contingency;
S_{BC} generated power in MVA at the base case.

The difference ΔS between the generated power for a given contingency and the generated power at the base case gives the amount of load to be shed so that the power system can remain in a secure state. Thus ΔS is the amount of load to be shed for maintaining the desired security level.

In order to determine the optimal amount of load shedding, CIAPS implements a fuzzy and a neuro-fuzzy logic controllers. The most important issue in order to define such controllers are the input parameters given to these controllers. At this aim, we used the index PSLVI and bus voltage magnitudes computed for each considered contingency. The following subsections briefly described the fuzzy and the neuro-fuzzy logic controllers implemented in CIAPS.

2.3.1 Fuzzy Logic Controller

In fuzzy logic, a typical fuzzy inference system (FIS) is developed to map a given input to an output using fuzzy rules and membership functions which are often chosen arbitrarily. Every FIS consists of three main components, namely, *fuzzification*, *fuzzy inference mechanism* and *defuzzification* [41]. The implemented fuzzy logic controller (FLC) for load shedding has two inputs and one output. The PSLVI and the lowest voltage magnitude (V) in a power system are considered as the inputs and the amount of load shed (LS) is considered as the output of the FLC. The inputs of FLC are fuzzified according to its membership functions. For each input variable, five linguistic variables are defined as *more vulnerable* (MV), *vulnerable* (V), *more alert* (MA), *alert* (A) and *invulnerable* (I). Furthermore, for the output variable, LS, five linguistic variables are defined as *high* (H), *medium* (M), *low* (L), *very low* (VL) and *very very low* (VVL). For example, a rule relating two inputs and one output defined as a logical statement, as follows:

```
if (PSL is MV) and (V is MV) then (LS is H)
```

The inference mechanism is used to compute the FLC output membership levels. Finally, the defuzzification methods is the centroid technique that finds the point where a vertical line would slice the aggregate set into two equal masses or, using a physical analogy, in the centre of gravity. The final output of FLC is the estimated amount of load shed which is given by [41]:

$$S_{FLC} = \frac{\sum\limits_{i=1}^{n} ls_i \mu(ls_i)}{\sum\limits_{i=1}^{n} \mu(ls_i)} \tag{8}$$

where $\mu(ls_i)$ is the output membership level for the i^{th} rule.

2.3.2 Neuro-Fuzzy Logic Controller

The rules governing a fuzzy logic system should be defined first, but one may have no knowledge about the behavior of a power system for defining these rules. Therefore, automatically tuning parameters through a neural network embedded within a fuzzy system could avoid the need for a previous knowledge about the power system response. For load shedding estimation using neuro-fuzzy logic controller (NFLC), the output of an adaptive neural network based fuzzy inference system (ANFIS) [41] is associated with the amount of load shed. Similar to the FLC, the inputs to the NFLC are the PSLVI and the lowest bus voltage magnitude. A multilayer feed forward neural network trained by using the back propagation algorithm is used to adjust the membership function parameters according to the input-output characteristic of the training patterns. The overall output of the ANFIS is the estimated amount of load shed by NFLC that calculates the sum of outputs of all defuzzification neurons and is given by [41]:

$$S_{NFLC} = \sum_{i=1}^{n} \bar{\mu}_i (k_{i0} + k_{i1} x_1 + k_{i2} x_2) \tag{9}$$

where k_{i0}, k_{i1} and k_{i2} are the sets of parameters following the i^{th} rule, and μ_i is the normalized firing strength.

CIAPS implements a typical ANFIS structure with five layers and two inputs, each of which has two membership functions. The five layers of the ANFIS are connected by adequate weights. The first layer is the input layer which receives input data that are mapped into membership functions so as to determine the membership of a given input. The second layer of neurons represents association between input and output, by means of fuzzy rules. In the third layer, the output are normalized and then passed to the fourth layer. The output data are mapped in the fourth layer to give output membership function based on the pre-determined fuzzy rules. The outputs are summed in the fifth layer to give a single-value output.

3 Case Study

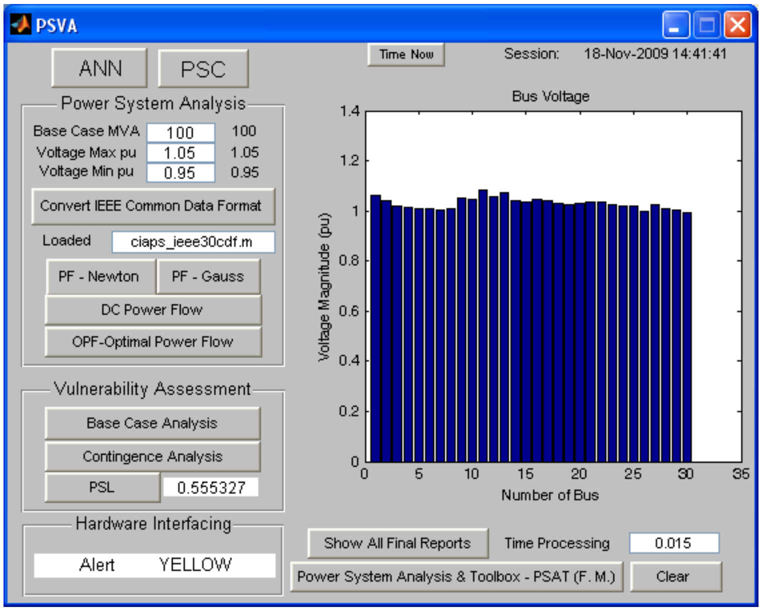
In this section, we validate the accuracy and the performance of the vulnerability assessment techniques described in the previous section based on a variety of tests systems. These are the IEEE 30 bus system, the IEEE 300 bus system, and a 87-bus model of the Malaysian high voltage transmission system. The 30-bus test system consists of 6 generators, 34 lines, 7 transformers and 21 loads with voltages at 11 kV, 33 and 132 kV levels. This system consists of 69 generators, 116 transformers, 295 lines and 198 loads and voltage levels varying from 0.6 to 345 kV. Finally, the 87-bus system consists of 22 generators, 177 lines and 56 loads with a voltage rate of 275 kV.

Before applying the proposed ANN technique for power system vulnerability assessment, one has to create an appropriate training data set for the ANN. At this aim, the system behavior is analyzed when subjected to a set of credible contingencies. Power flow simulation results for each contingency are used to check voltage limits and transmission line thermal limits. Then the vulnerability index based on PSLVI is calculated for each contingency. Finally, to implement the load shedding scheme for vulnerability control of power systems, the outputs of the ANN (i.e., the desired PSLVI values and the bus voltage magnitudes obtained from the $N-1$ contingency analysis) are used as input variables for the fuzzy-based load shedding controllers.

3.1 Power System Vulnerability Assessment

Figure 4 depicts the main graphical user interface (GUI) for power flow, contingency and vulnerability analysis. The behavior of CIAPS can be customized through this GUI. For example, one can select different methods for the power flow method and optimal power flow. As for contingency analysis, the vulnerability index based on PSL is used to analyze the system behavior at the base case and for the $N-1$ contingency analysis. The kind of the contingency (line outage, generator outage, etc.) as well as the weights can be chosen by the user.

Figure 4 also depicts the contingency analysis results for the IEEE 30-bus test system. Each contingency is represented by a bar in the plot. However, results can also be printed as a plain text report file. Some examples of such reports is given in following subsections.



Session: 18-Nov-2009 14:41:41

Base Case MVA 100 100
Voltage Max pu 1.05 1.05
Voltage Min pu 0.95 0.95

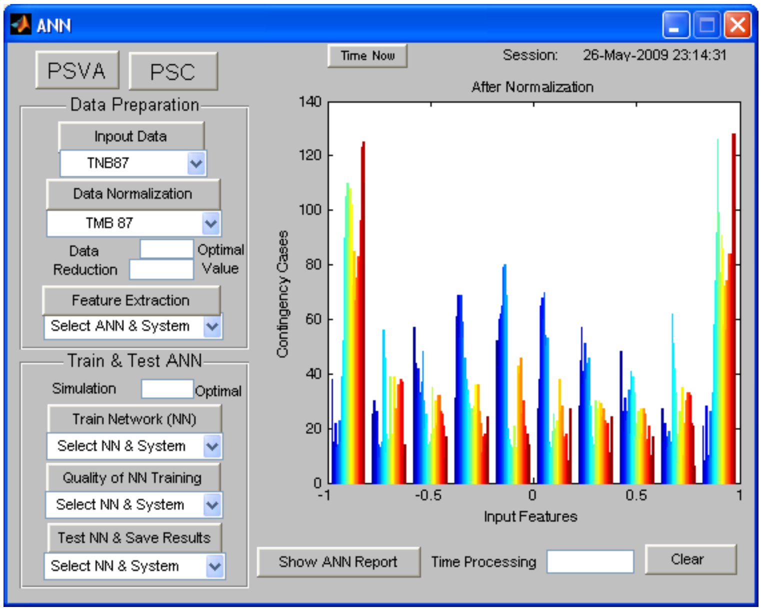
Loaded ciaps_ieee30cdf.m

PSL 0.555327

Alert YELLOW

Time Processing 0.015

Fig. 4. Main GUI for power flow, contingency and vulnerability analysis.



Session: 26-May-2009 23:14:31

Inpout Data: TNB87

Data Normalization: TMB 87

Data Reduction Optimal Value

Feature Extraction: Select ANN & System

Simulation Optimal

Train Network (NN): Select NN & System

Quality of NN Training: Select NN & System

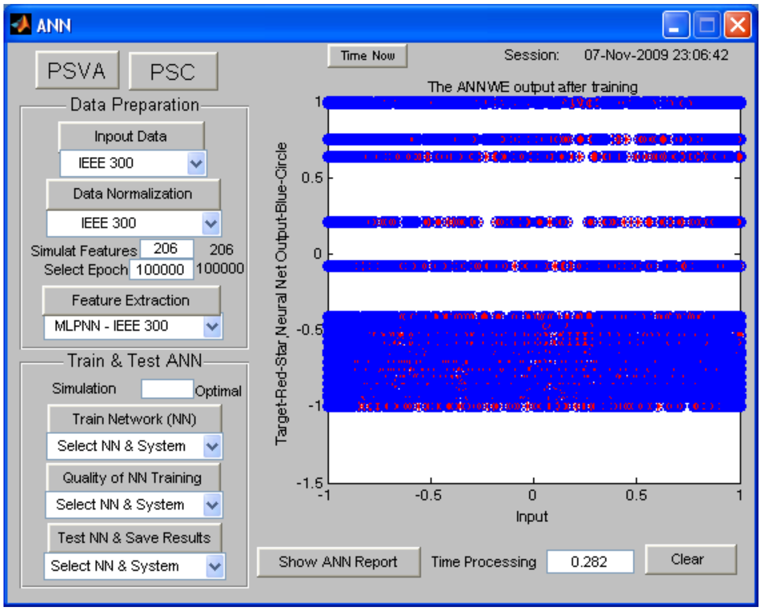
Test NN & Save Results: Select NN & System

Time Processing

Fig. 5. Data preparation of ANN.

3.2 Artificial Neural Network-based Vulnerability Assessment

The GUI for feature extraction and ANN-based vulnerability assessment is shown in Fig. 5. The input data of the selected system and normalized data are plotted to provide to the user an idea about the input data before training the ANN. The plot in Fig.5 means that the inputs fall in the interval $[-1, 1]$. This step will reduce the possibility of saturation within the ANN structure.



Session: 07-Nov-2009 23:06:42

Inpout Data: IEEE 300

Data Normalization: IEEE 300

Simulat Features 206 206

Select Epoch 100000 100000

Feature Extraction: MLPNN - IEEE 300

Simulation Optimal

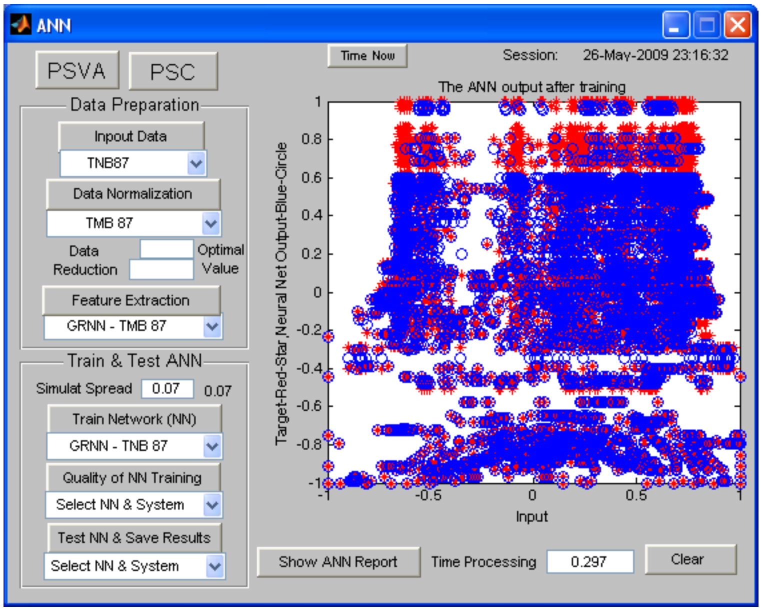
Time Processing 0.282

Fig. 6. Classification rate of the reduced features.

According to Fig. 6, the original total number of features is 413, considering real and reactive power flows and total generated real and reactive powers as features. After applying the proposed feature extraction, the features are reduced at least by a 40%. This reduction allows speeding up the ANN training process. Figure 6 shows that the 206 reduced features provide the best classification rate. Thus, the reduced features can represent the entire system, with a minimum loss of information.

Once the dimensionality reduction by means of feature extraction is completed as illustrated in Fig. 7, the ANNs are simulated in order to improve the accuracy of the ANN outputs.Here, the plot evaluates the reduced data in terms of accuracy after training. Several options are available to allow adjusting the performance of the ANNs by using the typical values of MLPNN and GRNN which are usually recommended for large size power systems. Fig.8 illustrates these options. 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 performance is obtained. The activation functions for hidden and output neurons are linear or sigmoid-nonlinear either of logistic or hyperbolic type, depending on what combination achieves the most accurate result for a given input and output.

Usually, the selection of input features is based on experience. In CIAPS, given that the output of the ANN is the proposed vulnerability index PSLVI, the input features selected are based on variables that influence the output such as the load flow information. Such information includes voltage, transmission real and reactive power flows, voltage angle, generated powers and demands.

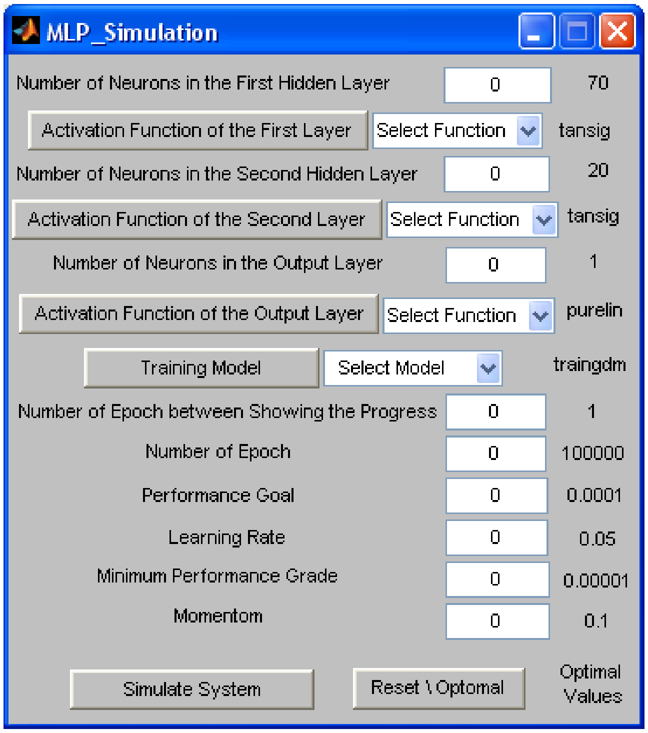


Session: 26-May-2009 23:16:32

Simulat Spread 0.07 0.07

Time Processing 0.297

Fig. 7. ANN simulation results.



Number of Neurons in the First Hidden Layer 0 70

Activation Function of the First Layer Select Function tansig

Number of Neurons in the Second Hidden Layer 0 20

Activation Function of the Second Layer Select Function tansig

Number of Neurons in the Output Layer 0 1

Activation Function of the Output Layer Select Function purelin

Training Model Select Model traingdm

Number of Epoch between Showing the Progress 0 1

Number of Epoch 0 100000

Performance Goal 0 0.0001

Learning Rate 0 0.05

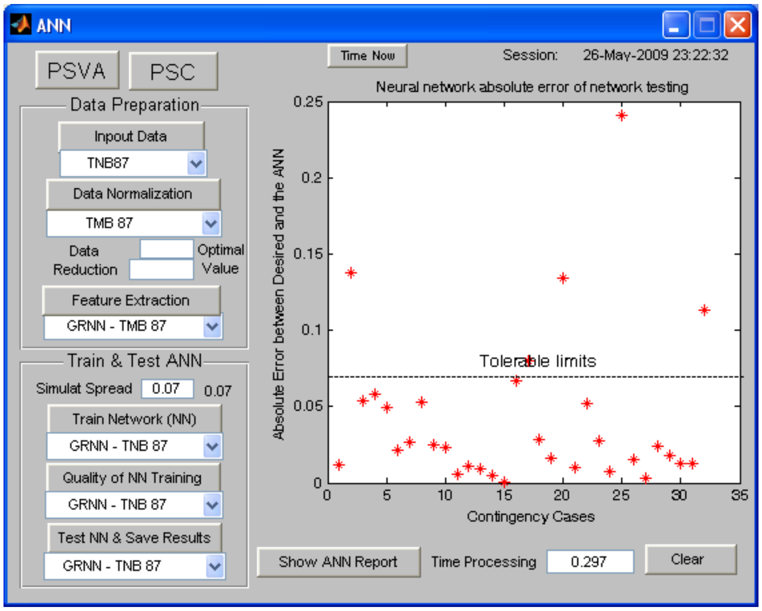
Minimum Performance Grade 0 0.00001

Momentom 0 0.1

Optimal Values

Fig. 8. Typical settings of multilayer perceptron neural network.

The number of input variables and training data sets depend on the size of a power system. In CIAPS, the training data sets are obtained by simulating the system in response to various disturbances. A set of system features along with the corresponding system vulnerability index are considered as the ANN input and output variables, respectively. The input variables are real and reactive power flows and total power generation while the output variable is the



Session: 26-May-2009 23:22:32

Simulat Spread 0.07 0.07

Time Processing 0.297

Fig. 9. GRNN testing results.

Table 1
ANN vulnerability assessment results.

ANN VI	Desired VI	Absolute Error	Vulnerability
0.5385	0.5265	0.0120	Alert
0.5383	0.4006	0.1380	Alert
0.4087	0.3801	0.0286	Vulnerable
0.1328	0.1556	0.0228	Vulnerable
0.2144	0.1932	0.0212	Vulnerable
0.2835	0.2868	0.0030	Vulnerable
0.8052	0.8722	0.0670	Invulnerable
0.8029	0.8187	0.0158	Invulnerable

proposed vulnerability index, PSLVI. The results of vulnerability assessment using GRNN for the 87 test system (TNB) are shown in Fig. 9 and Table 1. Figure 9 illustrates the performance of the ANN in terms of accuracy. The tolerable limits of the absolute error for training and testing data are 7 and 6 %, respectively.

3.3 Fuzzy Logic-based Load Shedding

The GUI that handles fuzzy and neuro-fuzzy controllers for load shedding is illustrated in Fig. 10. Several options allow adjusting the performance of

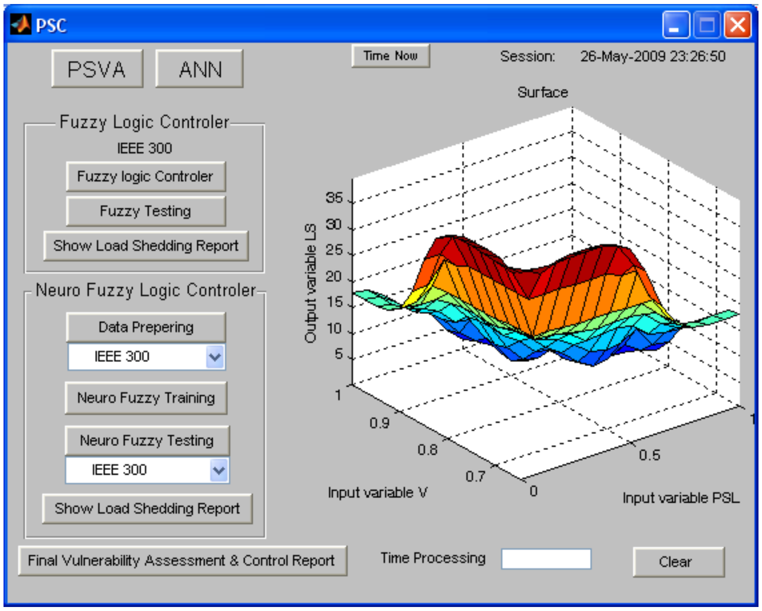


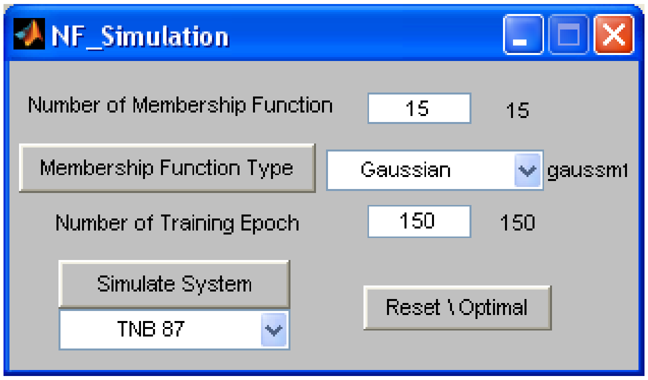
Fig. 10. Fuzzy controller GUI.

the controllers. A standard fuzzy inference mechanism is used for the fuzzy logic controller (FLC) if the output does not depend directly on the input. Otherwise, a smoother membership function (e.g., the Gaussian function) is used. Afterwards, the center-of-gravity defuzzification method is applied to convert the fuzzy output into a crisp value in which the output of FLC is the estimated amount of load shed. As usual, results can be displayed as both plots and report files.

Figure 10 also shows the results of load shedding for the IEEE 300 bus system using the proposed FLC. This plot shows the control surface for the FLC. The vertical axis is the estimated load shedding while horizontal axes are PSLVI and bus voltage magnitudes. The neuro-fuzzy controller can be simulated by adjusting the number of membership function and training epoch. [1]

Figure 11 shows the recommended typical setting used for large scale power systems while Fig. 12 illustrates optimal setting of the membership function and the epoch. Each Adaptive Neuro Fuzzy Inference System (ANFIS) is trained by means of 150 iterations. The ANFIS is generated using 15 Gaussian membership functions for each input. The number of inputs and the number of membership functions determine the number of fuzzy rules and therefore the training time. Table 2 shows the final vulnerability assessment and control results of the simulated contingency cases for the IEEE 300 bus system.

[1] An *epoch* is a set of training data with a fixed accuracy.

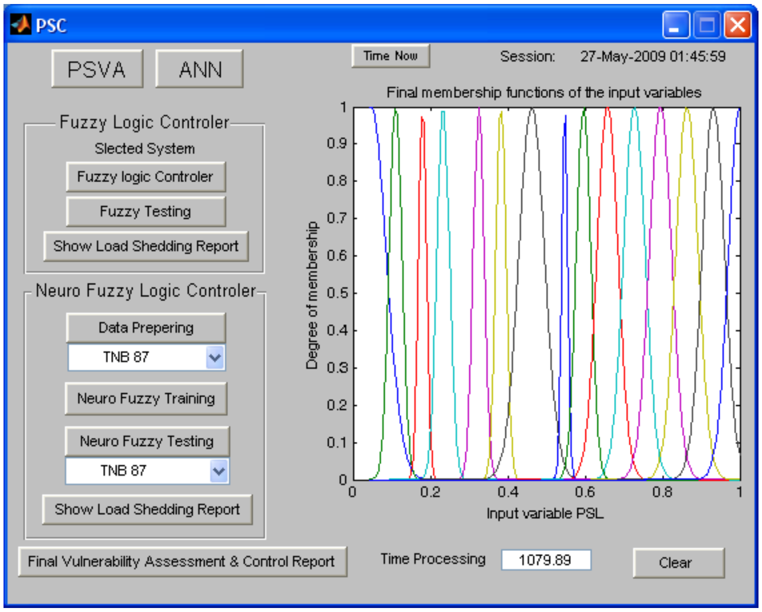

NF_Simulation

Number of Membership Function 15 15

Membership Function Type: Gaussian gaussmf

Number of Training Epoch 150 150

Fig. 11. Typical settings for Fuzzy controller.


PSC

Session: 27-May-2009 01:45:59

Time Processing 1079.89

Fig. 12. Neuro-Fuzzy controller GUI.

4 Conclusions

The paper describes a computational intelligence application for power system (CIAPS) simulation tool. The proposed tool is the outcome of the research conducted on vulnerability assessment and control of large scale interconnected power system and its implementation using computational intelligence. CIAPS integrates numeric and symbolic computations with user-friendly graphical interfaces. Results show that the proposed tool is promising for contingency analysis and computational intelligence studies, and very helpful to understand vulnerability assessment phenomena. CIAPS can be used for illustration purposes during the lectures and by students when preparing personal assignments.

CIAPS is an open project and any contribution in term of computational

Table 2
Neuro-Fuzzy vulnerability assessment results.

Contingency	Load Shed	Weak Buses
Cases	[p.u.]	#
LO-93	7.65	47, 43, 44, 113
LO-177	14.14	159, 157, 122, 121, 120, 118, 117, 115
LO-182	9.40	159, 157, 122, 121, 118, 117, 115
LO-242	1.79	9038, 9033, 9032, 9031
LO-305	1.63	225, 224, 223, 192
GO-84	2.49	9042, 9038, 9035, 9033, 9032, 9031
GO-213	1.33	9038, 9033, 9031
GO-7061	4.56	9038, 9033, 9031, 7061, 61, 59, 58
GO-7166	5.57	9035, 9033, 9032, 9031
GO-7166	5.57	9035, 9033, 9032, 9031
LI-2.6 Percentage	8.99	9071, 9052, 9043, 9042, 9041, 9038, 9037, 9036, 903, 9033, 9032, 9031, 9004
LI-3.1 Percentage	15.04	9072, 9071, 9052, 9044, 9043, 9042, 9041, 9038, 903, 9035, 9034, 9033, 9032, 9031, 9007, 9004, 900, 52

intelligence-based modules for power system analysis is very welcome.

References

[1] K. Y. Lee, A. El-Sharkawi, Modern Heuristic Optimization Techniques, John Wiley & Sons, Hoboken, New Jersey, 2008, IEEE Press Series on Power Engineering.

[2] M. R. AlRashidi, M. E. El-Hawary, Applications of Computational Intelligence Techniques for Solving the Revived Optimal Power Flow Problem, Electric Power System Research 79 (4) (2009) 694–702.

[3] M. Aggoune, M. A. El-Sharkawi, D. C. Park, M. J. Dambourg, R. J. Marks, Preliminary Results on Using Artificial Neural Networks for Security Assessment [of Power Systems], in: Power Industry Computer Application Conference, Seattle, Washington, 1989, pp. 252 – 258.

[4] M. Aggoune, L. E. Atlas, D. A. Cohn, M. J. Damborg, M. A. El-Sharkawi, R. J. Marks, Artificial Neural Networks for Power System Static Security Assessment,

in: IEEE International Symposium on Circuits and Systems, Vol. 1, Portland, Oregon, 1989, pp. 490 – 494.

[5] A. A. F. Q. Zhou, J. Davidson, Application of Artificial Neural Networks in Power System Security and Vulnerability Assessment, IEEE Transactions on Power Systems 9 (1) (1994) 525–532.

[6] R. Aresi, B. Delfino, G. B. Denegri, S. Massucco, A. Morini, A Combined ANN/Simulation Tool for Electric Power System Dynamic Security Assessment, in: IEEE Power Engineering Society Summer Meeting, Vol. 2, Edmonton, Alberta, 1999, pp. 1303 – 1309.

[7] L. Srivastava, S. N. Singh, J. Sharma, Knowledge-based Neural Network for Voltage Contingency Selection and Ranking, IEE Proceedings on Generation, Transmission & Distribution 146 (6) (1999) 649–656.

[8] F. Ying, T. S. Chung, An Innovative Fast Voltage Security Assessment based on a Hybrid ANN External Equivalent Approach, in: IEEE Power Engineering Society Winter Meeting, Vol. 2, Singapore, 2000, pp. 987 – 992.

[9] P. B. C. K. S. Swarup, ANN Approach Assesses System Security, IEEE Computer Applications in Power 15 (2) (2002) 32–38.

[10] P. K. Dash, A. Pradhan, G. Panda, Application of Minimal Radial Basis Function Neural Network to Distance Protection, IEEE Transactions on Power Delivery 16 (1) (2001) 68–74.

[11] M. Joorabian, A. Taleghani, R. Aggarwal, Accurate Fault Locator for EHV Transmission Lines Based on Radial Basis Function Neural Networks, Electric Power Systems Research 71 (2004) 195–202.

[12] P. K. Dash, S. Mishra, G. Panda, A Radial Basis Function Neural Network Controller for UPFC, IEEE Transactions on Power Systems 15 (4) (2000) 1293–1299.

[13] Y. Y. Hsu, H. C. Kuo, Fuzzy-Set Based Contingency Ranking [Power System Security], IEEE Transactions on Power Systems 7 (3) (1992) 1189–1196.

[14] K. H. Abdul-Rahman, S. M. Shahidehpour, Application of Fuzzy Sets to Optimal Reactive Power Planning with Security Constraints, IEEE Transactions on Power Systems 9 (3) (1994) 589–597.

[15] K. H. Abdul-Rahman, S. M. Shahidehpour, Static Security in Power System Operation with Fuzzy Real Load Conditions, IEEE Transactions on Power Systems 10 (1) (1995) 77–87.

[16] J. Heydeman, R. Reijntjes, R. Babuska, U. Kaymak, H. R. van Nauta Lemke, Fuzzy Logic Based Security Assessment of Power Networks, in: ISAP '96 - International Conference on Intelligent Systems Applications to Power Systems, Vol. 4, Orlando, Florida, 1996, pp. 405–409.

[17] M. A. Matos, N. D. Hatziargriou, J. A. Pecas Lopes, Multicontingency Steady State Security Evaluation using Fuzzy Clustering Techniques, IEEE Transactions on Power Systems 15 (1) (2000) 177–183.

[18] H. Mori, E. Ando, Two-Staged Simplified Fuzzy Inference for Dynamic Contingency Screening in Power Systems, in: IEEE Power Engineering Society Summer Meeting, Vol. 4, Seattle, Washington, 2002, pp. 1996–2001.

[19] H. Sun, D. C. Yu, Y. Xie, Flexible Steady-State Security Region of Power System with Uncertain Load Demand and Soft Security Limits, in: IEEE Power Engineering Society Summer Meeting, Vol. 4, Seattle, Washington, 2000, pp. 2008–2013.

[20] C. A. Jensen, M. A. El-Sharkawi, R. J. Marks, Power System Security Assessment Using Neural Networks: Feature Selection using Fisher Discrimination, IEEE Transactions on Power Systems 16 (1) (2001) 757–763.

[21] J. Z. Zhu, Optimal Power System Steady-State Security Regions with Fuzzy Constraints, in: IEEE Power Engineering Society Winter Meeting, Vol. 2, New York, New York, 2002, pp. 1095–1099.

[22] M. Dumitrescu, T. Munteanu, A. P. Ulmeanu, Fuzzy Logic in Power System Performability, in: Proceeding of the Second IEEE International Conference on Intelligent System, Vol. 1, 2004, pp. 326–330.

[23] J. M. G. Alvarez, P. E. Mercado, Online Inference of the Dynamic Security Level of Power Systems Using Fuzzy Techniques, IEEE Transactions on Power Systems 22 (2) (2007) 717–726.

[24] T. M. L. Assis, D. M. Falcao, G. N. Taranto, Dynamic Transmission Capability Calculation Using Integrated Analysis Tools and Intelligent Systems, IEEE Transactions on Power Systems 22 (4) (2007) 1760–1770.

[25] T. M. L. Assis, G. N. Taranto, D. M. Falcao, Secure Economic Dispatch Determination through Integrated Tools and Fuzzy Inference Systems, in: iREP Symposium: Bulk Power system Dynamics and Control - VII, Charleston, South Carolina, 2007.

[26] A. Sallam, A. Khafaga, Fuzzy Expert System Using Load Shedding for Voltage Instability Control, in: Proceedings of the IEEE Large Engineering Systems Conference on Power Engineering, Halifax, NS, 2002, pp. 125–132.

[27] S. Al-Osaimi, A. Abdennour, A. Abdullaziz, Hardware Implementation of A Fuzzy Logic Stabilizer on A Laboratory Scale Power System, Electric Power Systems Research 74 (2005) 9–15.

[28] The MathWorks, Inc., Matlab Programming, available at http://www.mathworks.com (2005).

[29] The MathWorks, Inc., Simulink: Dynamic System Simulation Software, available at http://www.mathworks.com (2001).

[30] F. Milano, L. Vanfretti, J. C. Morataya, An Open Source Power System Virtual Laboratory: The PSAT Case and Experience, IEEE Transactions on Education 51 (1) (2008) 17–23.

[31] L. Vanfretti, F. Milano, The Experience of PSAT as a Free and Open Source Software for Power System Education and Research, accepted for publication on the International Journal of Electrical Engineering Education (May 2008).

[32] J. H. Chow, K. W. Cheung, A Toolbox for Power System Dynamics and Control Engineering Education and Research, IEEE Transactions on Power Systems 7 (4) (1992) 1559–1564.

[33] R. D. Zimmerman, C. E. Murrillo-Sánchez, D. Gan, Matpower, Version 3.0.0, User's Manual, Power System Engineering Research Center, Cornell University, available at http://www.pserc.cornell.edu/matpower/matpower.html (2005).

[34] S. Ayasun, C. O. Nwankpa, H. G. Kwatny, Voltage Stability Toolbox for Power System Education and Research, IEEE Transactions On Education 49 (4) (2006) 432–442.

[35] J. Mahseredjian, F. Alvarado, Creating an Electromagnetic Transient Program in Matlab: MatEMTP, IEEE Transactions on Power Delivery 12 (1) (1997) 380–388.

[36] G. Sybille, SimPowerSystems User's Guide, Version 4, published under sublicense from Hydro-Québec, and The MathWorks, Inc., available at http://www.mathworks.com (Oct. 2004).

[37] K. Schoder, A. Hasanović, A. Feliachi, A. Hasanović, PAT: A Power Analysis Toolbox for Matlab/Simulink, IEEE Transactions on Power Systems 18 (1) (2003) 42–47.

[38] C. D. Vournas, E. G. Potamianakis, C. Moors, T. Van Cutsem, An Educational Simulation Tool for Power System Control and Stability, IEEE Transactions on Power Systems 19 (1) (2004) 48–55.

[39] F. Milano, An Open Source Power System Analysis Toolbox, IEEE Transactions on Power Systems 20 (3) (2005) 1199–1206.

[40] H. Saadat, Power System Analysis, MaGraw-Hill, New York, 1999.

[41] M. Negnevitsky, Artificial Intelligence - A Guide to Intelligent System, Addison Wesley, Boston, 2002.

[42] H. Demuth and M. Beale and M. Hagan, Neural Network Toolbox v. 5 for use with Matlab, available at http://www.mathworks.com (2006).

[43] Computational Intelligence Applications to Power System (CIAPS), available at http://notes.ump.edu.my/fkee/Dr. Ahmed M. A. Haidar.

[44] P. M. Anderson, A. A. Fouad, Power System Control and Stability, The Iowa State University Press, Ames, Iowa, 1977.

[45] A. M. A. Haidar, M. Azah, H. Aini, Vulnerability Assessment of a Large Sized Power System using Neural Network Considering Various Feature Extraction Methods, Journal of Electrical Engineering and Technology 3 (2) (2008) 167–176.

[46] K. M. Faraoun, A. Boukelif, Neural Networks Learning Improvement Using the K-means Clustering Algorithm to Detect Network Intrusions, International Journal of Computational Intelligence 3 (2006) 161–168.

[47] B. Kim, B. Junki, L. Byung, Modeling of Silicon Oxynitride Etch Microtrenching using Genetic Algorithm and Neural Network, Journal of Microelectronic Engineering 83 (2006) 513–519.

[48] A. Jain, S. Tripathy, R. Balasubramanian, Y. Kawazoe, Stochastic Load Flow Analysis Using Artificial Neural Networks, in: IEEE Power Engineering Society General Meeting, Monteral, QC, 2006, pp. 1–6.

[49] S. Haykin, Neural Network: A Comprehensive Foundation, Prentice Hall, Upper Saddle River, New Jersey, 1999.

[50] R. Duda, P. Hart, D. Stork, Pattern Classification, John Wiley & Sons, Singapore, 2000, Wiley-Interscience Publication.

[51] P. Simon, Oscillatory Stability Assessment of Power System using Computational Intelligence, Ph.D. thesis, Universität Duishdurg-Essen, Germany (2005).