Towards static-security assessment of a large-scale power system using neural networks

S. Weerasooriya
M.A. El-Sharkawi
M. Damborg
R.J. Marks II

Indexing terms: Power systems and plant, Circuit theory and design, Neural networks

Abstract: A neural-network-aided solution to the problem of static-security assessment of a large scale power system is proposed. It is based on a pattern-recognition technique where a group of neural networks is trained to classify the secure/insecure status of the power system for specific contingencies based on the precontingency system variables. The large dimensionality of the input data is reduced by partitioning the problem into smaller subproblems at different stages. When each trained NN is queried online, it can provide the power-system operator with the security status of the current operating point for a specified contingency. Parallel network architecture and the adaptive capability of the neural networks can be combined to achieve high speeds of execution and good classification accuracy.

1 Introduction

One of the main aspects of power-system security is static security. This is defined as the ability of the system to reach a state within the specified safety and supply quality following a contingency. The time period of consideration is such that the fast-acting automatic-control devices have restored the system load balance, but the slow-acting controls and human decisions have not responded [1–3].

The problem of predicting the static-security status of a large power system is a computationally demanding task [2]. It involves the solution of a nonlinear-programming problem with a large number of variables and limit constraints which define the feasible region of operation [2, 3]. In addition, the amount of memory required to store the steady-state security under different system configurations and contingencies is equally prohibitive. These considerations seriously undermine the application of static-security assessment in real time without the support of large computing capability. In an era when power-system facilities are utilised to their maximum to supply the growing energy demands, a fast on-line security-prediction scheme is imperative in ensuring uninterrupted supply quality to the consumers [4].

The concepts of pattern recognition have long been looked at as a possible means of speeding up these calculations [5]. From a pattern-recognition perspective, the problem of static security assessment (SSA) is considered as a classification problem where the precontingency system attributes are used to predict postcontingency system-security status. Many attempts have been reported where conventional pattern-recognition techniques have been used to solve both static [5] and transient [5, 6] security problems in power systems, but almost all of the techniques suffer from the same problem known as the curse of dimensionality. The computational effort required to formulate the classifier becomes prohibitively large and the mapping to be learned becomes increasingly complicated with the increasing size of the power system [5, 6].

Neural Networks (NNs) have gained renewed popularity as a method of synthetising a mapping between input and output variables by learning a set of arc weights and node thresholds of a connectionist model based on training examples [7, 8]. Certain problems in power systems, with their inherent nonlinear and complex nature, seem amenable to solutions through trained NNs. Several such applications have been documented in the literature: power-factor correction [9], harmonic analysis [10], topological observability [11], identification of static- and dynamic-security regions [12, 13] and post-fault dynamic analysis of interconnected power systems [14].

Layered-perceptron NNs are known to be well suited for pattern classification [7, 8, 12, 13]. In classification of steady-state security, the inputs to the NN are the precontingency-system attributes while the output is the postcontingency-security status. The NN is trained to solve the two-class problem by presenting it with a set of patterns generated by an oracle. An oracle, in this case, is a valid computational model of the power system which can be solved to investigate system performance. A properly trained NN can classify the security of a previously unencountered input pattern with good accuracy. Due to its parallel architecture, the time and computational effort involved in classification are small compared to other conventional schemes [12].

In a large power system, there are many different attributes to choose from, and many contingencies to look at to predict static-security status. Training a single large NN for this task would be almost impossible. Instead, the aim is to split up the analysis into small but well defined tasks in a logical manner and then train a collection of NNs to handle each classification task. The most obvious division is at the contingency evaluation stage where separate NNs can be dedicated to handle specific contingencies. In evaluating the security under a

Paper 8449C (P9, P11), received 30th April 1991
The authors are with the Department of Electrical Engineering, FT-10, University of Washington, Seattle, WA98195, USA

particular contingency, the large-scale power system can be decomposed into a study system and several external systems based on the field of influence of the contingency [2]. Some well-established techniques of feature-selection algorithms [15, 16] can be used to further reduce the dimensionality of each individual NN input space. The collection of NN classifiers can then be integrated to form a composite security-assessment package. The inherent parallel-connectionist architecture of the NNs can be fully exploited this way.

In pattern recognition, the compromise for achieving on-line speed is the large amounts of processing required off-line [5–14]. A large number of simulations must be performed off-line to generate a good representative data set for training the NNs. For a selected contingency, each training pattern would require the solution of an AC power flow, to determine the post-contingency security picture. The training-data set should also span the entire demand space brought about by hourly, daily and weekly variations in load. The effects of different contingencies that can occur also have to be taken into account, but once the NNs are successfully trained, on-line SSA can be done with speed and a predicted statistical accuracy.

The successful implementation of this scheme will of course depend heavily upon the availability of neural-network hardware. A software implementation is ruled out owing to the size and the combinatorial complexity of the problem at hand, but there are promising signs of a growing number of parallel architectures and custom devices [17] which can be used for both generic and neural-network implementations. This would pave the way for an actual implementation of the model.

2 Problem formulation

The NN-based pattern-recognition approach for SSA depends on the assumption that there are some characteristics of precontingency system states that give rise to a secure or insecure post-contingency system. The task of the NN is to capture these common underlying characteristics for a set of known operating states and to interpolate this knowledge to classify a previously unencountered state.

The first step in such an application is to obtain a set of training data which represents the different power-system operating modes that are likely to be encountered due to hourly, daily or monthly variations in load demand. This information is derived from a precontingency optimum-power-flow (OPF) study. The load data are obtained either from past operating conditions or from an approximated load model. Once a set of feasible operating points is obtained, a selected contingency is simulated and the post contingency power-flow solution is investigated for line and voltage violations.

As described earlier, the task of SSA of a large-scale power system using pattern recognition is an enormous computational exercise. One way of reducing this complexity is to divide the problem into smaller tasks at different levels and train dedicated NN classifiers to handle each task. Fig. 1 shows a possible break up of the problem into smaller tasks.

2.1 Contingency partition

A power system is vulnerable to different types of contingencies. There are different types of contingency selection, ranking and evaluation algorithms [2, 4] to come up with a list of critical contingencies for a power system. Static security under each specific contingency or a specific class of contingencies should be assessed by a specifically trained NN. It is envisaged that this partition would simplify the task of capturing the diverse effects of individual contingencies on static security, thereby helping towards building more accurate classifiers.

When dealing with a large-scale power systems SSA, the concept of a study system connected to external systems through a set of boundary buses is well known [2]. This is based on the assumption that a contingency within the study system produces the highest repercussions within that system. However, there are always cases where a contingency in one system is strongly felt in another [2]. Two possible ways are proposed to avoid this problem [2]. In the first method, one strives, during offline studies, to obtain study systems that are insensitive to external influences. Failing that, sensitivity-analysis techniques can be employed.

Boundary-bus compensation [2] is a sensitivity-based method used to decouple the external systems from the local system. The concept assumes that the postcontingency boundary-bus injections are based on the first-order sensitivities of the corresponding tieline flows to that particular class of contingencies [2]. Hence, external network effects are strictly represented by the updated boundary-bus injections. If the incremental change in tieline power is significant, the method suggests that the boundaries be pushed a bit deeper into the external network.

2.2 Voltage and line violations

A complete SSA involves checking for both voltage and thermal violations in the postcontingency steady state. It

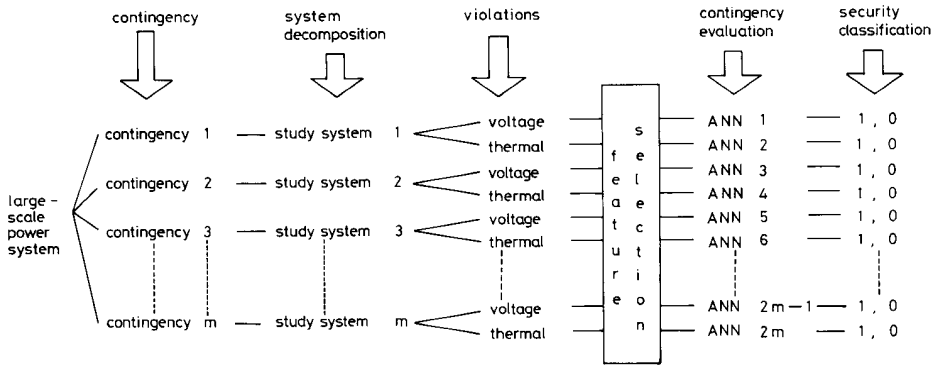

Fig. 1 *Proposed NN approach to SSA*

was observed that efficient prediction of voltage and thermal violations cannot be based on the same set of features. Since the mechanisms leading to thermal and voltage violations are fundamentally different, it was decided that the classification under each class be handled by separate NNs. Partitioning the two tasks helps towards achieving more accurate predictions.

2.3 Feature selection

It is also well known that some power-system variables are better indicators of static security than others. Established methods of feature selection and extraction algorithms [15, 16] can be used to analyse the internal and boundary variables of the precontingency study system to reduce the large-dimensional pattern space into a smaller but well-representative feature space. NNs can be trained on these selected features. This makes the learning phase fast and less complex.

2.4 Limitations

Each NN is trained by using the appropriate set of features which are selected from a larger set of internal and boundary variables. The feature selection is based on a heuristic statistical measure which is introduced later. During NN training, each contingency will have a separate set of training samples. This is a necessary evil owing to the wide-ranging characteristics associated with different types of contingencies.

The paper restricts itself to predicting SSA for single line outages. Double contingencies are not considered. NNs are trained only to predict voltage violations. The partitioning into subsystems is done with the assumption that there are no mutual intereaction between subsystems, i.e. boundary-bus injections are assumed to remain unchanged in the postcontingency system. However, it should be noted that such interactions can be easily incorporated into the training data.

3 Static-security assessment

Static security of a power system is assessed once the system reaches a steady state after a contingency. Assume the precontingency power-flow solution to be given by

$$f^{(0)}\{X^{(0)}, U^{(0)}, L^{(0)}\} = 0 \tag{1}$$

where X is the state vector (complex bus voltages), U is the control vector (real-power generation and generation voltage), L is the demand vector (real and reactive-power demand), and $\{\cdot\}^{(0)}$ are precontingency variables.

The inputs to the equation are U and L. These are real data from the system, or data generated based on some preconceived model. The control vector $U^{(0)}$ is usually selected to minimise a separate objective function $F\{X^{(0)}, U^{(0)}\}$ which is based on economic considerations. Under most cases, this is the combined cost of generation in the precontingency state space. The resulting Lagrangian function to be minimised takes the form

$$L\{X^{(0)}, U^{(0)}, L^{(0)}, \lambda\} = F\{X^{(0)}, U^{(0)}\} + \lambda^T f^{(0)}\{X^{(0)}, U^{(0)}, L^{(0)}\} \tag{2}$$

where λ is the lagrange multiplier vector. The minimisation process is iterative with respect to $X^{(0)}$, U, and λ. A gradient-based search technique is used for the process. The control vector U which could be both the real power output and the voltage of generator buses is bounded by the constraint

$$U_{min} \leqslant U^{(0)} \leqslant U_{max} \tag{3}$$

based on generator ratings and system considerations. A solution to the constrained-optimisation problem should satisfy the Kuhn-Tucker corner conditions. This procedure is commonly known as an optimal power flow (OPF) [1–3].

Security of the postcontingency power system under the kth contingency is determined by solving for $X^{(k)}$ in the load flow equations

$$f^{(k)}\{X^{(k)}, U^{(k)}, L^{(k)}\} = 0 \tag{4}$$

$L^{(k)}$ is assumed to remain at its precontingency value $L^{(0)}$. The postcontingency control vector $U^{(k)}$ is calculated based on the type of fault. For a sizeable disruption of real power, such as the loss of a tieline or a generator, the outputs of the remaining generator are adjusted on the basis of their individual speed-drop characteristics [2]. Otherwise, only the swing bus absorbs the slack generation. The specifics used in this paper are explained in Section 6.

The line flows and bus voltages are then checked against their safe operating limits specified by

$$G\{X^{(k)}, U^{(k)}\} \leqslant 0 \tag{5}$$

Depending on whether the postcontingency operating condition satisfies or violates any one of the operating limits, the corresponding precontingency power system is appropriately labelled secure or insecure.

4 Feature selection

One of the classical problems in pattern recognition is to reduce the dimensionality of the measurement vector [15]. One advantage of this concept of dimensionality reduction is that classification in the lower-dimensional space is faster and less complex [15, 16]. A simple example of this concept of dimensionality reduction is when d minimally correlated elements of each n-dimensional measurement vector (normalised between 0 and 1)

$$Y_j = [y_1, y_2, \ldots, y_{nj}]^T \quad j = 1, 2, \ldots, N \tag{6}$$

are selected where $d \ll n$ and the classification is based on these (d-dimensional) patterns.

An acceptable simple criterion for selecting a variable as a feature is that it should provide more information for classification than those not selected [5, 6]. The heuristic notion of interclass distance is used to accomplish this task. Given a set of patterns with dimension n, it is reasonable to assume that the pattern vectors for each of the two classes occupy a distinct region in the observation space [5, 6, 16]. The average pairwise distance between the patterns is a measure of class separability in the region with respect to the particular variable. The index F_i provides a measure of this class separation with respect to the ith variable.

$$F_i = \left| \frac{m_i^{(S)} - m_i^{(I)}}{\sigma_i^{(S)2} + \sigma_i^{(I)2}} \right| \quad 1 \leqslant i \leqslant n \tag{7}$$

$$m_i^{(S)} = \frac{1}{N^{(S)}} \sum_{j=1}^{N^{(S)}} y_{ij}^{(S)} \quad m_i^{(I)} = \frac{1}{N^{(I)}} \sum_{j=1}^{N^{(I)}} y_{ij}^{(I)}$$

$$\sigma_i^{(S)2} = \frac{1}{N^{(S)}} \sum_{j=1}^{N^{(S)}} \{y_{ij}^{(S)} - m_i^{(S)}\}^2$$

$$\sigma_i^{(I)2} = \frac{1}{N^{(I)}} \sum_{j=1}^{N^{(I)}} \{y_{ij}^{(I)} - m_i^{(I)}\}^2$$

$m_i^{(\cdot)}$ and $\sigma_i^{(\cdot)2}$ are the mean and variance of the ith variable corresponding to class $(\cdot)$. The superscript (S) stands for 'secure' while (I) stands for 'insecure'. $N^{(S)}$ and $N^{(I)}$ indicate the number of secure and insecure patterns that form the training set $\{N = N^{(S)} + N^{(I)}\}$. Variables with higher values of F carry more information about class separability than others. Therefore classification can be based on d ($\ll n$) selected variables which will be referred to as features. These features are selected as follows:

(*a*) Calculate F_i for all i such that $0 \leqslant i \leqslant n$;

(*b*) Rank them according to the descending order of F_i;

(*c*) Go to the first ranked variable;

(*d*) Calculate the correlation coefficients (C_C) of all lower-ranked variables with respect to this variable;

(*e*) Eliminate all lower-ranked variables which have $|_CC| > 0.9$; and

(*f*) Go to the next-highest-ranked variable and go to step (*d*).

The correlation coefficient between the ith and the jth variable is defined as

$$C_{C_{ij}} = \frac{E\{y_i y_j\} - E\{y_i\}E\{y_j\}}{\sigma_i \sigma_j} \quad i, j = 1, 2, \ldots, n$$

$$E\{y_i y_j\} = \frac{1}{N}\sum_{k=1}^{N} y_{ik} y_{jk} \quad E\{y_i\} = \frac{1}{N}\sum_{k=1}^{N} y_{ik}$$

$$\sigma_i^2 = \frac{1}{N}\sum_{k=1}^{N} (y_{ik} - E\{y_i\})^2$$

The value of 0.9 in step (*e*) is selected arbitrarily. The process is repeated until all (n) variables either ranked or discarded. Subsequently a set of d variables from the top of the ranked list is selected as the key features for training the NN classifier. The value d is the minimum number of features required to obtain the specified classification accuracy. Hence each original pattern Y_j given by eqn. 6 will now be represented by a reduced d ($\ll n$) dimensional pattern

$$\tilde{}Y_j = [y_{k_1 j}, y_{k_2 j}, \ldots, y_{k_d j}]^T \tag{8}$$

The values $k_1, k_2, \ldots, k_d$ are common for all patterns. Selecting a suitable value for d is a tradeoff between classification accuracy and classifier design. This is discussed further in Sections 5 and 6.

Interclass-distance measures are the only family of feature-selection-criterion functions that do not depend on the estimation of probability-density functions. These heuristic measures are therefore attractive mainly for computational reasons [15, 16]. However, in general whether or not a feature can be selected on the basis of its individual effectiveness is problem dependent.

5 Neural networks

NNs have been found to be effective systems for learning discriminants for patterns from a body of examples [7, 8]. Once a set of training patterns is generated and an optimal set of features is selected, a NN classifier can be made to learn the mapping associated with them. The feedforward NN architecture used in the paper is commonly known as the multilayer perceptron NN and is given in Fig. 2. It consists of sets of nodes arranged in layers. Activation signals of nodes in one layer are transmitted to the next layer through links which either attenuate or amplify the signal [8].

Representation of an L-layer NN can be described by the two equations

$$u_i(l+1) = \sum_{j=1}^{N_l} w_{ij}(l+1) y_j(l) + \theta_i(l+1)$$
$$(j = 1, 2, \ldots, N_l)$$
$$y_i(l+1) = \Phi\{u_i(l+1)\} \tag{9}$$

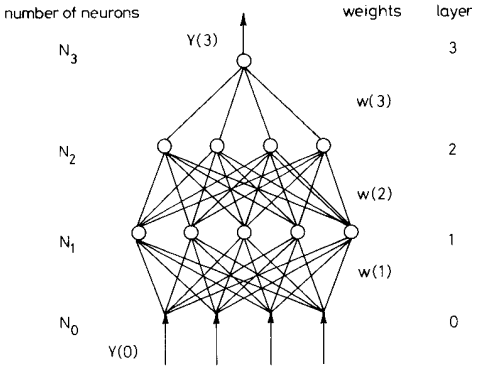

Fig. 2 *Topology of a 3-layer NN*

where $y_i(l+1)$ is the activation value of the jth neuron of the $(l+1)$th layer; $u_i(l+1)$ is the net input to the jth neuron in the $(l+1)$th layer; $w_{ij}(l+1)$ is the weight between the jth neuron of the lth layer and the ith neuron of the $(l+1)$th layer; Φ is the sigmoid activation function $[1/(1+e^{-x})]$; $\theta_i(l+1)$ is the external input to the jth neuron in the $(l+1)$th layer; and the indices i and l are such that $1 \leqslant i \leqslant N_{l+1}$ and $0 \leqslant l \leqslant L-1$. It is important to note that the $y_i(0)$ denote the inputs and the $y_i(L)$ the outputs of the NN.

5.1 Back-propagation learning

The back-propagation training technique adjusts the weights in all connecting links and thresholds in the nodes of the NN so that the difference between the actual output and the target output are minimised for all patterns. This is done by minimising the energy function E given by,

$$E = \frac{1}{2P}\sum_{i=1}^{P}\sum_{j=1}^{N_L}\{t_{ij} - y_{ij}(L)\}^2 \tag{10}$$

with respect to all the weights and thresholds. P is the number of training patterns while N_L is the number of output neurons. y_{ij} and t_{ij} denote the jth output and the corresponding target for the ith training pattern, respectively. The update for the weights are calculated using the iterative-gradient-descent technique where

$$w_{ij}(l) = w_{ij}(l) + \eta\frac{\partial E}{\partial w_{ij}(l)} + \nu\Delta w_{ij}(l) \tag{11}$$

Constant η is the iteration step while constant ν is the momentum factor. $\Delta w_{ij}(l)$ indicates the weight change in the previous iteration. The choice of η and ν is critical for satisfactory learning and they are usually selected based on experience.

For SSA, the inputs to the NN consist of the features selected as described in Section 4 and are denoted by $Y(0) = \tilde{}Y_j$ as given in eqn. 8. The target for the output $Y(L)$ is a 1 or a 0 depending on whether the pattern is secure or insecure. The value d in eqn. 8 is the minimum

number of input features required to achieve the specified classification accuracy. Selecting the minimum possible input dimension gives a faster learning rate for the corresponding NN classifier.

Deciding a proper NN architecture for a classification task is still an open question. Although there have been numerous efforts to clarify this issue, no generally adopted treatment can yet provide clear answers. It is currently accepted that a single-layer feedforward network can form an arbitrary classification boundary. The number of hidden neurons is the minimum required to ensure the desired convergence criteria. The particular network structure is usually chosen based on the experience gained during previous trials.

6 Simulation results

The test system consists of the AEP 8-bus study system [12] connected to two external systems through two tielines as shown in Fig. 3. The study system includes N_g

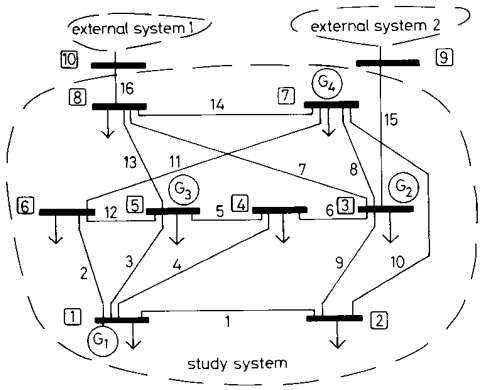

Fig. 3 *Test power system*

(=4) generators, N_b (=8) loads and N_t (=16) transmission lines. The influence of the external networks is modelled by the bi-directional power flow at the boundary buses 9 and 10, respectively.

Table 1 shows the permitted perturbation in the real and reactive loads at each bus on a 100 MVA base. The

Table 1: Range of load parameters

Bus	Bus type	Range of real-load variations (p.u.MW)	Range of reactive-load variations (p.u.MVAr)
1	Slack	9.0–11.0	0.0–1.0
2	Load	11.2–16.8	0.0–1.0
3	Generation	13.5–16.5	0.0–1.0
4	Load	14.0–26.0	0.0–1.0
5	Generation	13.5–16.5	0.0–1.0
6	Load	15.4–28.6	9.1–16.9
7	Generation	9.0–11.0	0.0–1.0
8	Load	0.0–2.0	5.0–15.0
9	Boundary	−7.5–7.5	−7.5–7.5
10	Boundary	−7.5–7.5	−7.5–7.5

tieline flows are considered either positive or negative depending on the demand schedule. Different power-system-loading conditions within the specified range are generated by randomly perturbing each of the real and reactive loads with a uniformly distributed random variable. The perturbations are uncorrelated. This is done in the absence of real operational data. If such data are available, they could be used in training and testing the NNs.

The precontingency optimal-dispatch strategy is to minimise the cost of generation given by

$$F\{X^{(0)}, U^{(0)}\} = \sum_{i=1}^{N_g} (C_{2i} P_{gi}^2 + C_{1i} P_{gi} + C_{0i}) \tag{12}$$

where C_{2i}, C_{1i} and C_{0i} are the constant coefficients of the quadratic cost function of the ith generator. The control vector U is given by

$$\begin{aligned} U^{(0)} &= [P_g] \\ [P_g] &= [P_{g1}, P_{g2}, \ldots, P_{gNg}]^T \end{aligned} \tag{13}$$

The generator bus voltages are not considered as control variables. Also, for tripping of a tie line, the generation $[P_g]$ is updated based on the droop characteristics of the generators. The droop at each individual generator is assumed to be proportional to its maximum ratings. Therefore, if tripping of a tieline causes a deficit of real power Δp, the individual generator power settings are adjusted as

$$U^{(k)} = U^{(0)} + \Delta U \tag{14}$$

where

$$\Delta U = \left[\frac{\Delta p}{\sum_{i=1}^{N_g} P_{g(max)i}} \right] [P_g] \tag{15}$$

$P_{g(max)i}$ is the maximum allowable real generation of the ith generator.

Next, the postcontingency power-system states are obtained by solving a regular power flow by taking into account the changes in the system topology and the control-variable settings if any. The ensuring bus voltages are tested for violations by checking against their respective limits, i.e.

$$V_{j(min)} \leqslant V_j \leqslant V_{j(max)} \quad \forall\, j = 1, \ldots, N_b \tag{16}$$

When any one of the above constraints is violated, that particular operating point is labelled insecure. Around 3000 random patterns are generated under each contingency. Once the key features are selected, some of the data are used for training the NN while some are used for testing the trained NN. To avoid memorising, training is stopped when the classification error on the test set becomes a minimum. In some cases, training is stopped when the desired classification accuracy is reached. To evaluate the performance of the trained NN classifier, the following definitions are introduced:

False alarm: When a true secure operating point, as described by the oracle, is classified as insecure by the NN.

False dismissal: When a true insecure operating point as described by the oracle, is classified as secure by the NN.

The following percentages are also introduced to obtain a quantitative measure of the classification performance. The percentage false alarms, false dismissals and false classifications are calculated using the definitions:

$$\text{false alarms (\%)} = \frac{\text{number of false alarms}}{\text{total true secure states}} \times 100$$

$$\text{false dismissals (\%)} = \frac{\text{number of false dismissals}}{\text{total true insecure states}} \times 100$$

false classifications (%)

$$= \frac{\text{false alarms} + \text{false dismissals}}{\text{true secure} + \text{true insecure states}} \times 100$$

6.1 Case 1: outage of a tie line

In this case, the contingency is the tripping of tieline 16, between the boundary bus 10 and the local bus 8. The precontingency operating states are defined by the real and reactive loads at all local buses and the direction and magnitude of complex-power flow in the two tielines 15 and 16. The generation is set based on an economic-dispatch strategy by solving an OPF to minimise the energy costs.

Before the postcontingency power flow is run, the real power generation is adjusted as described above. Then the power flow is reconfigured according to the current generation settings. The bus voltages are now checked for security. For different randomly perturbed loads, approximately 3000 patterns are generated. A single pattern contains 30 attributes which include the real and reactive injections (P_i, Q_i) at all buses and the voltage magnitudes (V_i) at all buses. All boundary bus injections are considered as loads in solving the power flow.

Next, the key features for training the NN are selected as described in Section 4. Table 2 shows the top nine features from the list of ranked variables, their class statistics, and the corresponding F values as derived from eqn. 7. It can be seen from Table 2 that there is a significant change in the value of F between the first and second ranked and seventh and eighth ranked variables. Therefore, as a first attempt, the first seven variables, namely Q_8, V_8, Q_5, Q_7, V_3, Q_{10} and Q_3, are selected as the features for training the NN. Similar discontinuity in the F values may not always be visible in the list of ranked variables. In such an event the optimum number of features can be selected by consequent training of the NNs using a recursively increasing number of features until the minimum required accuracy is obtained.

Table 2: Class statistics of the key variables

Variable	$m^{(S)}$	$m^{(I)}$	$\sigma^{(S)2}$	$\sigma^{(I)2}$	F index
Q_8	0.5848	0.1319	0.0591	0.0275	5.2258
V_8	0.5535	0.3276	0.0439	0.0326	2.9555
Q_5	0.4452	0.5697	0.0254	0.0216	2.6519
Q_7	0.4619	0.5705	0.0261	0.0254	2.1085
V_3	0.9928	0.9318	0.0026	0.0281	1.9889
Q_{10}	0.5533	0.7168	0.0546	0.0305	1.9194
Q_3	0.5614	0.7338	0.0518	0.0489	1.7102
P_5	0.5041	0.5182	0.0293	0.0323	0.2287
V_2	0.5351	0.5151	0.0466	0.0528	0.2011

The training and testing statistics for the NN which predicts voltage violations for this contingency are given in Table 3. It is seen that, starting from 30 dimensional patterns, classification is done using only seven directly measurable features. It is also seen that reasonable classification accuracy is obtained even with a widely varying operating point. Note that identical proportions of secure and insecure data are used in both the training and testing sets. This is to minimise any bias towards a particular class during NN training. It is also seen that the selected features are in the vicinity of buses 8 and 10 which makes intuitive sense. Voltage insecurities are caused by violations at buses 2, 4, 8 and 9.

Table 3: Training and testing statistics for the NN in case 1

NN architecture and training information		Testing statistics	
Inputs	7	Testing data	500
Outputs	1	True secure patterns	241
Hidden layers	1	True insecure patterns	248
Hidden neurons	9	False alarms	9
Iteration step	0.05	False dismissals	2
Momentum factor	0.05	Percentage false alarms	3.600
Training patterns	1500	Percentage false dismissals	0.800
Iteration cycles	1000	Percentage false classifications	2.200

6.2 Case 2: outage of a internal transmission line

The test conditions are identical to those for case 1 except that line 12 between buses 5 and 6 within the study system is tripped. Therefore there is no reconfiguring of the generation in the postcontingency system. The postcontingency power flow followed by a security analysis reveals the voltage status of the transmission system. Based on the cost index F, it is found that a NN trained on the seven input features V_6, Q_7, Q_5, P_6, P_9, P_{10} and P_7 — gives the required classification accuracy.

The training and the testing statistics of the NN are given in Table 4. A very low classification error is obtained using the method. As before, equal proportions of secure and insecure data are used in the training. The seven variables are selected by looking at the discontinuities in the F sequence as explained under case 1.

Table 4: Training and testing statistics for the NN in case 2

Network architecture and training information		Testing statistics	
Inputs	7	Testing data	500
Outputs	1	True secure patterns	250
Hidden layers	1	True insecure patterns	245
Hidden neurons	6	False alarms	0
Iteration step	0.10	False dismissals	5
Momentum factor	0.01	Percentage false alarms	0.000
Training patterns	1500	Percentage false dismissals	2.000
Iteration cycles	1000	Percentage false classifications	1.000

7 Conclusions

A NN-based static-security-assessment technique for a large-scale power system is proposed. Multiple neural networks have been successfully trained to assess static voltage security of a study power system interconnected to two external networks under a two specified line contingencies. Feature-selection techniques have been applied to reduce effective problem dimension. The classification assumes the availability of P, Q injections and V magnitude at selected buses. These quantities are directly measurable from the power system and are usually available at the control centre. Therefore one could foresee this technique being used as an approximate fast online static-security estimator. It is approximate because there is no guarantee of achieving a zero classification-error rate. However the computational efficiency of the NN classifier can far outweigh this drawback.

The applicability of this concept in security assessment of large-scale power systems depends on how well the system can be decomposed into multiple subsystems and boundary buses without compromising accuracy. The possibility of handling interactions between subsystems is an area which needs further attention.

The power-system topology undergoes temporal variations due to component switching, faults, scheduled outages, etc. Once the topology changes, the transparent mapping between the feature space and the security

status is bound to change. Hence a NN trained to handle one topology may not necessarily perform well under another. A further understanding of the topological generalisation capabilities of the NN is required to effectively overcome the problem. With emerging neural network hardware, the proposed scheme holds promise as a fast online classifier of static security of large-scale power systems.

8 Acknowledgment

This work was supported by the National Science Foundation and the Washington Technology Center. The financial support of the electric energy industrial consortium (EEIC) at the University of Washington is greatly appreciated.

9 References

1 ALSAC, O., and STOTT, B.: 'Optimal load flow with steady state security', *IEEE Trans.*, 1974, **PAS-94**, pp. 745–751
2 DEBS, A.S.: 'Modern power system control and operation' (Kluwer Academic Publishers, Boston, 1988).
3 DOMMEL, H.W., and TINNEY, T.F.: 'Optimal power flow solutions', *IEEE Trans.*, 1968, **PAS-87**, pp. 1866–1876
4 EPRI: 'Security analysis and software needs: survey results'. Report EL-6753, Project 2473-37, Final report, March 1990
5 PANG, C.K., PRABHAKARA, F.S., EL-ABIAD, A.H., and KOIVO, A.J.: 'Security evaluation in power systems using pattern recognition'. IEEE Power Engineering Society winter meeting, NY, January 1973
6 PRABHAKARA, F.S., and HEYDT, G.T.: 'Review of pattern recognition methods for rapid analysis of transient stability'. IEEE Power Engineering Society technical report, 87TH0169-3-PWR, 1987
7 PAO, Y.: 'Adaptive pattern recognition and neural networks' (Addison Wesley, 1989)
8 McCLELLAND, J.L., RUMELHART, D.E., and the PDP Research Group: 'Parallel distributed processing: exploitation of the microstructure of cognition. Vol. II' (Bradford Books, Cambridge, MA, 1986)
9 SANTOSO, N.I., and TAN, O.T.: 'Neural net based real-time control of capacitors installed on distribution systems'. Paper 89SM768-3 PWRD, IEEE Power Engineering Society summer meeting, California, 1989
10 MORI, H., UEMATSU, H., TSUZUKI, S., SAKURAI, T., KOJIMA, Y., and SUZUKI, K.: 'Identification of harmonic loads in power systems using an artificial neural network'. 2nd symposium on Expert System Applications to Power Systems, Seattle, 1989
11 MORI, H., and TSUZUKI, S.: 'Power system topological observability analysis using a neural network model'. 2nd symposium on Expert System Applications to Power Systems, Seattle, 1989
12 AGGOUNE, M.E., ATLAS, L.E., COHN, D.A., DAMBORG, M.J., EL-SHARKAWI, M.A., and MARKS, R.J.: 'Artificial neural networks for power systems static security assessment'. International symposium on Circuits and Systems, Portland, OR, 1989
13 EL-SHARKAWI, M.A., MARKS, R.J., AGGOUNE, M.E., PARK, D.C., DAMBORG, M.J., and ATLAS, L.E.: 'Dynamic security assessment of power systems using back error propagation artificial neural networks'. Second symposium on Expert System Application to Power Systems, Seattle, 1989
14. SOBAJIC, D.J., and PAO, Y.: 'Artificial neural net based dynamic security assessment for electric power systems', *IEEE Trans.*, 1989, **PWRS-4**, pp. 220–228
15 PATRICK, E.A.: 'Fundamentals of pattern recognition' (Prentice-Hall, 1972)
16 YOUNG, T.Y., and FU, K.S.: 'Handbook of pattern recognition and image processing' (Academic Press, 1986)
17 HWANG, J.N., and KUNG, S.Y.: 'Parallel algorithms/architectures for neural networks', *J. VLSI Processing*, 1989, **1**, pp. 221–251