Power System Security Assessment Using Neural Networks: Feature Selection Using Fisher Discrimination

Craig A. Jensen, Mohamed A. El-Sharkawi, and Robert J. Marks, II

***Abstract*—One of the most important considerations in applying neural networks to power system security assessment is the proper selection of training features. Modern interconnected power systems often consist of thousands of pieces of equipment each of which may have an affect on the security of the system. Neural networks have shown great promise for their ability to quickly and accurately predict the system security when trained with data collected from a small subset of system variables. This paper investigates the use of Fisher's linear discriminant function, coupled with feature selection techniques as a means for selecting neural network training features for power system security assessment. A case study is performed on the IEEE 50-generator system to illustrate the effectiveness of the proposed techniques.**

***Index Terms*—Dynamic security, intelligent systems, neural network, power system.**

I. Introduction

THE trend toward deregulation has forced modern utilities to operate their systems closer to security boundaries. This has fueled the need for faster and more accurate methods of security assessment. Power system security assessment deals with the system's ability to continue to provide service in the event of an unforeseen contingency. Such contingencies may include the unexpected loss of an important transmission circuit or a sudden change in a large load. Either of which could lead to a disruption of service to part or all of the system. The goal of security assessment is to determine when a disruption of service is likely to occur and to take steps to reduce the risk.

Neural networks have shown great promise as a means of predicting the security of large electric power systems [1]–[3]. Neural networks offer several advantages over traditional techniques including the ability to learn from examples.

The first step in applying neural networks to power system security assessment is the creation of an appropriate training data set. A common approach is to simulate the system in response to various disturbances and then collect a set of pre-disturbance system features along with the corresponding system security index. Possible security indices include the Critical Clearing Time (CCT) [1] and the system Energy Margin (EM) [3]. One of the most important aspects of achieving good neural network performance has proven to be the proper selection of training features. The *curse of dimensionality* states that, as a rule of thumb, the required cardinality of the training set for accurate training increases exponentially with the input dimension [5]. Thus, choosing a small subset of the thousands of possible features, *i.e.* feature *selection*, requires a small fraction of the training samples required if all features are used. Feature selection is the process of identifying those features that contribute most to the discrimination ability of the neural network. Only these features are then used to train the neural network and the rest are discarded. Proposed methods for selecting an appropriate subset of features are numerous [1]–[4].

An alternate to feature selection is feature *extraction*. Here, the dimensionality of a feature set is reduced by combining features while retaining characteristics that allow for accurate classification. Feature extraction is the process of mapping all available features into a composite feature set of lower dimension. Feature extraction techniques applied to power system security assessment have previously been presented by Weerasooriya and El- Sharkawi [6].

Many feature extraction techniques such as the principle components algorithm are based on the assumption that the greater the spread of the data in a particular axis, the greater the effect that will have on the discrimination ability of the neural network. This need not be true. Feature selection methods, on the other hand, generally are based on ranking different combinations of features in accordance to their classification performance and choosing the combination that achieves the highest ranking. Unlike feature extraction, no preprocessing is required once the features are chosen.

The basic problem encountered in all feature selection algorithms is how to deal with the computational complexity involved in searching among the large number of possible solutions. Given a set of features y_i, $i = 1, \ldots, D$, the task is to determine the subset of $d < D$ features that best satisfies some selection criteria. The search for the optimal subset of features is a combinatorial problem that requires selection among N possible solutions, where,

$$N = \frac{D!}{(D-d)!\, d!}$$

N becomes excessive even for relatively small values of D and d.

The simplest method to deal with this complexity, involves determining a *goodness* score for each individual feature, and then ranking the features based on their scores. A commonly used example of a *goodness* score uses the *Fisher discriminant*.

Manuscript received November 29, 1999; revised June 12, 2001. This work was supported by the *National Science Foundation*.

The authors are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195.

Publisher Item Identifier S 0885-8950(01)09421-4.

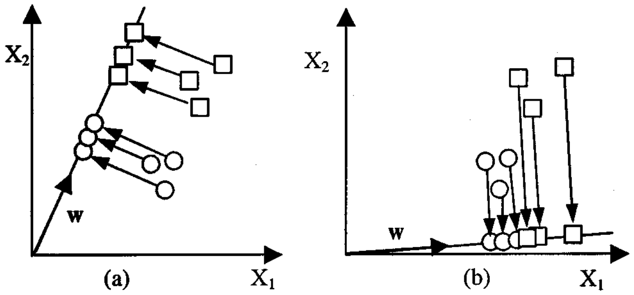

Fig. 1. (a) The Fisher linear discriminant projects data onto a lower dimension that maximally separates the classes. (b) A nonoptimal discriminant function.

The simple ranking method gives poor performance in cases when many features are highly correlated. In such cases, several features may have high individual discrimination ability, but due to their correlation, they offer no improvement when used together. For this reason, more advanced methods are presented to find d features that work well together.

Section III presents the IEEE 50-genrator transient stability test system. Section IV outlines several feature selection techniques that have been proposed. In Section V, a case study of the proposed techniques is presented to illustrate how neural network training features can be selected that are independent of changes of system topology. Finally, conclusions are presented in Section VI.

II. Fisher's Linear Discriminant

The Fisher linear discriminant function was proposed in 1936 [6],[7]. It seeks to find the optimal linear discriminant function for separating two classes of data. Given a set of n D-dimensional training samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ with n_1 samples in class C_1 and n_2 samples in class C_2, the task is to find the linear mapping, $y = \mathbf{w}^T\mathbf{x}$, that maximizes,

$$F(\mathbf{w}) = \frac{|m_1 - m_2|^2}{\sigma_1^2 + \sigma_2^2}$$

where m_i is the mean of class C_i and σ_i^2 is the variance of C_i. In words, this corresponds to finding the line that when the data are projected onto, provides the maximum separation (see Fig. 1).

The criterion function, F, can be rewritten as an explicit function of $\mathbf{w}$ as

$$F(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}} \tag{1}$$

where S_B is referred to as the *between-class scatter* matrix and S_W is the *within-class scatter* matrix. The *within-class scatter,* S_W, is defined as

$$S_W = S_1 + S_2$$

where S_1 and S_2 are

$$S_i = \sum_{x \in C_i} (\mathbf{x} - m_i)(\mathbf{x} - m_i)^T; i = 1, 2$$

and

$$m_i = \frac{1}{n_i} \sum_{x \in X_i} \mathbf{x}.$$

The *between-class scatter,* S_B, is

$$S_B = (m_1 - m_2)(m_1 - m_2)^T.$$

Solving (1) for the Fisher weights, $\mathbf{w}$, that maximize F can then be written as

$$\mathbf{w} = S_W^{-1}(m_1 - m_2).$$

Fisher's linear discriminant function is a projection from a D -dimensional space onto a line in such a manner that the training data is best separated. Fig. 1(a) shows the optimal (Fisher) projection for a 2-dimensional problem and Fig. 1(b) shows a nonoptimal projection. It can be seen that the Fisher vector, $\mathbf{w}$, defines the line — or, more generally, the hyperplane – that results in the greatest separation between the □'s and ○'s when data are linearly.

The Fisher discriminant can also be used as a means of assessing the linear separability of two classes of data. For example, the magnitude of the Fisher function, F, can be used as a measure of how well a linear classifier will perform on a given problem. The higher the value of F, the more separable the data are, and thus the higher the expected performance of the classifier. Different combinations of features can also be compared by comparing their respective Fisher values. This technique is used in the proceeding sections as a method for comparing different combinations of features in an effort to find the optimal combination.

III. Feature Selection Techniques

If the dimension of the problem is small, a reasonable technique is to simply search through all possible combinations of features and calculate their corresponding Fisher values. The optimal feature set is then the set with the highest Fisher value. This is only practical on small problems as it requires the evaluation of $D!/[(D-d)!\,d!]$ Fisher values.

In problems where the dimension is too high for an exhaustive search, a search technique called the backtrack search [4], also known as the branch-and-bound algorithm, can be employed to ease the computational burden. The backtrack search guarantees the optimal solution if the criterion function satisfies the monotonicity condition. The monotonicity condition requires that the value of the criterion function be nondecreasing when additional features are added. The Fisher linear discriminant function satisfies this condition because increasing the input dimension, D, can never result in a decrease in the Fisher value F.

The mechanics of the backtrack search are best illustrated, as in Fig. 2, in the form of a search tree. The root of the tree contains all of the features to be considered. Each node under the root corresponds to the removal of a single feature. The tree has depth $D - d$, and the leaves of the tree contain all possible combinations of d features out of the original D. The algorithm is initialized by calculating the Fisher value at any one of the leaves and the value is stored as the current best solution. The tree is traversed starting at the root and calculating the Fisher value, F, at each node along the way. Since the monotonicity condition applies, the value of F is guaranteed to be nonincreasing as the tree is traversed Therefore, if at any point along the traversal,

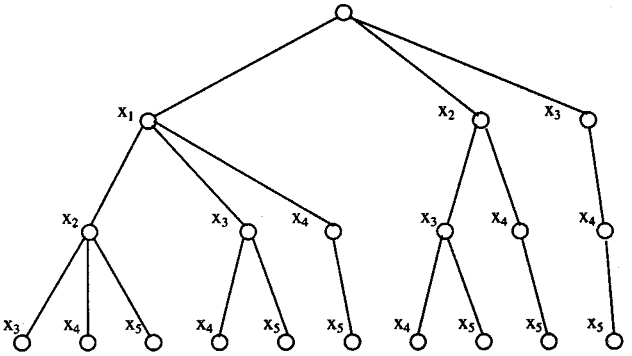

Fig. 2. Search tree for the backtrack algorithm for choosing the 2 best features out of 5.

the value of F falls below current best value found at any of the leaves, the rest of that branch need not be evaluated.

An example illustrating the backtrack search when applied to the selection of the two best features out of five is shown in Fig. 2. Initially, the branch at the far right side is evaluated and the Fisher value of the subset $\{x_1, x_2\}$ is determined and saved in the variable F_0. Next, the algorithm backtracks and proceeds down the next unexplored branch comparing the Fisher value, F, at each node to F_0. If the value of F exceeds F_0, there is still a chance that a better feature set will be discovered, so the algorithm must continue to search along the rightmost unexplored branch. When the bottom of the tree is reached, if the value of F is still greater than F_0, then F_0 is updated accordingly. If on the other hand, a node is found where the value of F is less than F_0, the rest of that branch need not be traversed due to the monotonicity condition. After the entire tree has been evaluated the optimal feature set is the one corresponding to F_0.

The backtrack method is particularly efficient when the features are preordered and placed in the tree in order of the magnitude of their individual Fisher values from left to right. This tends to put the better solutions toward the right side of the tree, which, in turn, tends to allow the algorithm to eliminate the maximum number of branches from the search.

The backtrack method should be used with caution because its worst-case run time can be much worse than exhaustive search. In worst case, the backtrack algorithm must evaluate every node in the tree. The number of evaluations required in the worst case is therefore,

$$\sum_{h=0}^{D-d} \frac{(D-h)!}{(D-h-d)!d!}$$

where

D total number of raw features;

d number of features to be selected;

h height of the nodes in this level of the tree.

In cases where the backtrack method proves computationally infeasible, a sub-optimal solution needs to suffice. Sequential ranking methods are relatively simple methods utilizing a heuristic that involves building a feature set by choosing the next feature that works best with the previously chosen features. There are several variations including the forward sequential, backward sequential and the *plus-l take away-r* algorithm.

The forward sequential method is a bottom-up algorithm that starts by choosing the best individual feature. Then the feature set is built from the ground up, by repeatedly adding the next feature that works best with the previously chosen features. This algorithm requires $D! - (D-d)!$ evaluations. The backward sequential method is very similar to the forward sequential except that it is a top-down algorithm. Instead of starting with an empty set and building a set of features, the backward sequential method starts with the complete set of features and iteratively removes them one at a time until only the desired number remain. This is done by calculating the Fisher value for all subsets of $n-1$ features where n is the current number of features remaining. The feature that results in the smallest decrease in Fisher value at each iteration is then removed. The backward sequential method requires the same number of evaluations as the forward method, but is more computationally demanding because the calculation of the Fisher discriminant is in a higher dimensional space at each iteration.

It should be noted that while the Fisher linear discriminant function has been shown to produce the optimal classification boundary for a linear classifier, its performance on a nonlinear neural network classifier is not known. The nonlinear neural network is commonly accepted as a more general model than a linear classifier. The *plus-l take away-r* algorithm is essentially a combination of the forward sequential and backward sequential methods. In this algorithm, l features are added via the forward method followed by the removal of r features by the backward method. This is advantageous in that it provides the algorithm the ability to remove features that are no longer needed due to combinations of features subsequently added. The algorithm works in either the top-down or bottom-up direction, depending on if $l > r$ or $r > l$. For the case where, $r > l$, the order of the forward-backward methods is be reversed, *i.e.,* the algorithm starts with all the features and r features are removed followed by the addition of l features.

IV. IEEE 50 Generator System

To assess the applicability of the Fisher linear discriminant function as a means of selecting neural network training features, an empirical study was conducted on the IEEE 50-generator transient stability test system [13]. The system consists of 50 generators, 145 buses and 453 transmission lines and transformers. A training database was collected by simulating 3-phase faults at nine different buses throughout the system. Of the nine faults, 4 were on the 500 KV subsystem, 1 was on the 220 KV system and 4 were on the 100 KV subsystem. The locations were chosen to cover a wide cross-section of the system. The critical clearing time for each fault was determined by simulating the system response using the EPRI Electric Transient Mid-Term Stability Program [12] (ETMSP) software program. The system was simulated for 651 different system operating conditions. The real and reactive power output of each of the 50 generators as well as the total system real and reactive load level were recorded along with the system critical clearing time for each of the 651 simulations.

TABLE I
STATISTICAL ANALYSIS OF FISHER'S LINEAR DISCRIMINANT APPLIED TO FEATURE SELECTION FOR SECURITY ASSESSMENT.

Fault	Random selection of features				Fisher Feature Selection		$P\left[\lvert X - x\rvert \geq F_{\min}\right]$
	Samples	Mean	Std.	Min error	Features Used	RMS Error (F_{min})	Chebyshev inequality
1	1000	0.1135	0.0210	0.0692	51, 2, 65, 80	0.0668	0.2022
2	1019	0.1676	0.0423	0.0428	53, 39, 50, 2	0.0471	0.1233
3	1005	0.1039	0.0202	0.0657	51, 101, 65, 2	0.0628	0.2421
4	1036	0.1069	0.0201	0.0627	51, 65, 2, 100	0.0641	0.2202
5	1208	0.1179	0.0202	0.0779	51, 101, 2, 65	0.0841	0.3584
6	1115	0.1077	0.0119	0.0824	9, 68, 3, 36	0.0853	0.2818
7	1131	0.1767	0.0135	0.1156	54, 90, 101, 57	0.1057	0.0364
8	1036	0.1152	0.005	0.0873	74, 63, 99, 91	0.0801	0.0200
9	1000	0.1756	0.0099	0.1018	90, 101, 99, 93	0.1027	0.0184

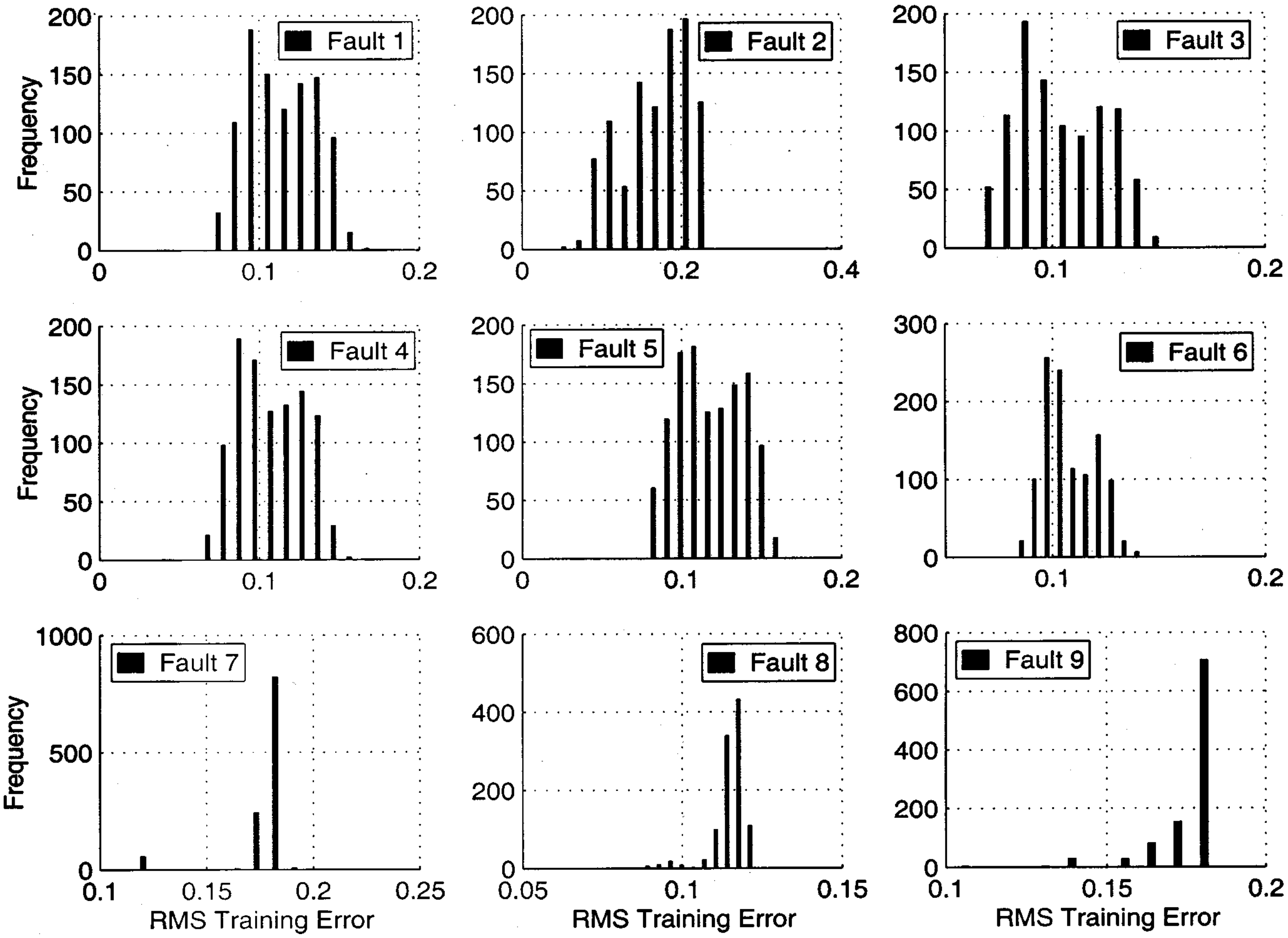

Fig. 3. Histograms showing the distribution of the neural network training error for various combinations of 4 randomly chosen features.

V. JUSTIFICATION OF FISHER FEATURE SELECTION

The Fisher forward sequential method was used to select the best 4 features for each of the nine faults considered in the study. Then a neural network was trained for each of the faults using the 4 best features found from the forward sequential method. The neural network training results are summarized in Table I. A commercial neural network simulation package called QwikNet [11] was used to train and test the neural networks.

Therefore, its performance should be *at least* as good as that of the linear classifier. The question then arises as to how well the neural network performs given the optimal set of features for a linear classifier? This question is not easy to answer theoretically since, short of exhaustive search, there is no known technique for determining the optimal feature set for a multilayer perceptron neural network. What can be done, however, is to perform a series of computer simulations to assess the discrimination ability of a given feature set on a given neural network model. These simulations can then be used to compare different feature sets and a statistical framework can be developed to quantify the performance of the Fisher linear discriminant.

Many statistical techniques exist for defining confidence intervals based on known probability distribution functions [9]. In this case, the probability distribution function of the neural network classification error for various combinations of input features can be estimated by randomly generating different feature combinations and determining their corresponding neural network classification error. The neural network model used in this study consisted of 3 layers with 4 inputs, 10 hidden neurons and a single output. For each of the nine faults, 1000 random combinations of 4 features were generated. A neural network was then trained using each of the 1000 feature combinations. Fig. 3 is a histogram of the neural network training error based on randomly selecting 4 power system input features out of a total of 102. The training error for each network was based on the average training error over 5 training sessions with different random initial weights. The results are summarized numerically in Table I.

Several techniques exist for calculating tail probabilities of the form $P\left[|X| \geq t\right]$, where X is an unknown random variable and t is a given distance from the mean. If the variance, σ^2, and the mean, m, of the underlying process are known, the Chebyshev inequality [9] can be used to establish bounds on the probability of a sample occurring greater than a given distance, a, from the mean. The Chebyshev inequality states that,

$$P\left[|X - m| \geq a\right] \leq \frac{\sigma^2}{a^2}.$$

The Chebyshev inequality makes no assumptions regarding the distribution of the underlying process, other than its mean and variance. Because of this, the established bounds are often very loose. Nevertheless, the Chebyshev inequality is still useful for establishing a theoretical upper bound on the probability that a feature combination exists that is superior to the combination determined by the Fisher feature selection method. The right column of Table I shows the probability of the existence of a combination of features superior to that found by the Fisher forward sequential technique.

Confidence intervals [10] are a well-known metric for assessing the reliability of experimental results. The equations for the upper and lower bounds on experimental results are,

$$P_L = \frac{\bar{X} + \frac{z^2}{2n} - \left(\frac{z}{\sqrt{n}}\right)\sqrt{\bar{X}(1-\bar{X}) + \frac{z^2}{4n}}}{1 + \frac{z^2}{n}}$$

$$P_H = \frac{\bar{X} + \frac{z^2}{2n} + \left(\frac{z}{\sqrt{n}}\right)\sqrt{\bar{X}(1-\bar{X}) + \frac{z^2}{4n}}}{1 + \frac{z^2}{n}} \qquad (2)$$

where

$\bar{X}$ point estimate of the trial;

n number of samples;

z $1 - \alpha/2$ % of the normal distribution.

(2) can be used to place a bound on the observed results with α percent certainty. Experimental results show that the features selected by the Fisher technique were found to be better than all but 46 out of 9550, or 99.54% of all randomly generated feature combinations. The 95% confidence interval is (99.38, 99.65). Table II shows the experimental 95% confidence intervals for each of the faults under study.

TABLE II
COMPARISON OF THE PROBABILITY OF THE EXISTENCE OF A FEATURE SET BETTER THAN THE SET SELECTED BY THE FISHER FEATURE SELECTION ALGORITHM

Faul t	# of random trials	# of improve-ments	$P[Err(X) \leq E_{Fisher}]$	
			Observed	95% confidence
1	1000	0	0.0000	-
2	1019	1	0.0010	(0.0002, 0.0055)
3	1005	0	0.0000	-
4	1036	1	0.0010	(0.0002, 0.0054)
5	1208	34	0.0281	(0.0202, 0.0391)
6	1115	9	0.0081	(0.0043, 0.0153)
7	1131	0	0.0000	-
8	1036	0	0.0000	-
9	1000	1	0.0010	(0.0002, 0.0056)
All	9550	46	0.0048	(0.0035, 0.0062)

TABLE III
TOPOLOGY CHANGES

Line Outage		Voltage Level
From Bus	To Bus	
1	2	500
1	6	500
6	12	500
12	14	500
17	22	500
25	27	500
53	55	100
59	92	100
67	97	100
119	130	100
141	142	100

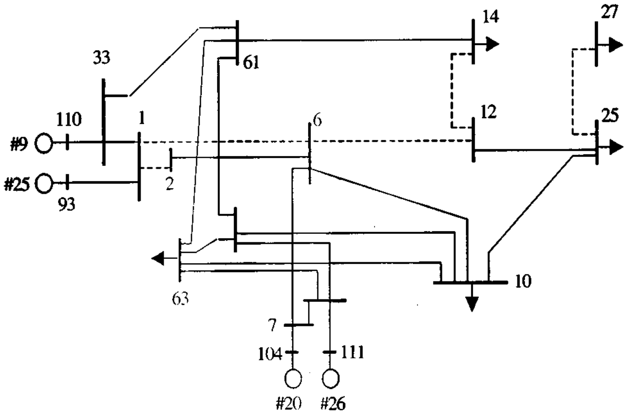

Fig. 4. A portion of the high voltage transmission system surrounding the fault. The fault is on bus #7 and cleared by removing the line between buses 6 and 7. Dashed lines indicate several of the transmission lines that were removed to simulate topology changes.

VI. CASE STUDY – APPLICATION TO TOPOLOGY CHANGES

In this section, a case study is performed on the IEEE 50-generator system to show how the Fisher feature selection

TABLE IV
COMPARISON OF NEURAL NETWORK PERFORMANCE BASED ON FEATURES SELECTED FROM A SINGLE SYSTEM TOPOLOGY AND MULTIPLE SYSTEM TOPOLOGIES

	Features selected based on a single topology (112-Q, 104-P, 111-P)		**Features selected based on multiple topologies (111-Q, 104-P, 110-Q)**	
Trial #	Training Error	Testing Error	**Training Error**	**Testing Error**
1	0.0122673	0.114222	**0.0159126**	**0.0156532**
2	0.0123515	0.131822	**0.0159686**	**0.0156648**
3	0.0124859	0.133681	**0.0154527**	**0.0158434**
4	0.0123491	0.132743	**0.0152939**	**0.0151090**
5	0.0130339	0.092719	**0.0155665**	**0.0155469**
Average	0.0124975	0.121077	**0.0155989**	**0.0155635**

technique can be used to select features that are independent of system topology. The system topology refers to the operating status of the numerous devices connected to the system. Selecting features that are independent of topology is important since the topology of modern power systems is continuously changing. These changes are due to many factors including maintenance, repair, and the addition of new equipment.

Since the system topology is not fixed, special care needs be taken to minimize the effect of topology changes on the performance of the neural network. One approach is to train different neural networks for each change in topology, and then use the specific neural network that reflects the current topology of the system. This approach is only applicable to problems that involve relatively few changes in topology. If a large number of topology changes need to be considered, the number of neural networks required becomes very large, and the problem quickly becomes impractical.

Another approach is to choose features for training the neural network that are independent of changes in topology. This allows a single neural network to learn the security of the system with respect to various different system topologies. Features such as the aggregate generation in a specific area or the flow on important transmission circuits have been suggested [3]. The Fisher feature selection technique can be used to select features that reflect changes in topology, and therefore, provide good performance even in the event of unexpected topology changes. This can be done by applying the Fisher selection technique to a training database that contains examples of as many different system topologies as possible. The Fisher selection criterion then selects the features that work best with regard to the various topologies.

An experiment was performed to test the performance of the Fisher technique for selecting topology invariant features for neural network training. The experiment consisted of training two neural networks with features selected by the Fisher forward sequential technique. The networks were trained with data generated from a single 3-phase fault located at bus #7 and cleared by removing the line between buses 7 and 6. The features for the first neural network were selected based on a training database generated with a fixed system topology. The features for the second neural network were selected from a training database containing 11 different pre-fault system topologies. The topology changes involved removing various transmission circuits from the system and then simulating the fault at bus #7 and calculating the system security. The topology changes were spread across a wide area of the system in an attempt to determine the significance of the location of the topology change with respect to the fault. Some topology changes consisted of removing 500 KV lines close to the fault while others were 100 KV lines far removed from the fault. Table III shows the location and voltage level of each of the 11 topology changes.

Fig. 4 shows a one-line diagram of a portion of the high voltage transmission system surrounding the fault at bus #7. Several of the transmission lines that were removed as a result of the topology changes are shown as dashed lines, the rest occur further from the fault and are not shown.

The training data files for each neural network were created from the same raw data file, which included a fixed pre-fault system topology. The first neural network was trained with features 112-Q, 104-P and 111-P, while the second neural network used features 111-Q, 104-P and 110-Q. The features correspond to the either the real (P) or reactive (Q) generator power outputs at the given buses. As previously mentioned, the features for the first neural network were selected by the Fisher method applied to a data file consisting of a fixed system topology, while the features for the second neural network were derived from a data file with multiple system topologies. Both neural networks were tested with a data file composed of 641 patterns with 11 different system topologies. Five training runs were performed for each network to assess their performance on the testing file. The results are shown in Table IV.

It can be seen from Table IV that the neural network trained with features selected from data containing multiple system topologies resulted in nearly a full order of magnitude reduction in testing error. This is a remarkable result considering the neural network was trained with data consisting of only a single system topology! This test clearly shows the importance of selecting features that are independent of changes in system topology and shows that the Fisher selection technique can be effectively used to select such features.

VII. CONCLUSION

The Fisher linear discriminant function coupled with the sequential feature selection technique has been proposed as a means for selecting neural network training features for power system security assessment. Through a statistical analysis, it was shown that the Fisher technique was able to select a feature set in the top one percentile of all possible feature sets. A case study was performed to illustrate how the Fisher technique can be used to select neural network training features that are independent of changes in power system topology. The proposed methods were tested on the IEEE 50-generator transient stability test system and excellent results were demonstrated.

REFERENCES

[1] D. J. Sobajic and Y. H. Pao, “Artificial neural-net based dynamic security assessment for electric power systems,” *IEEE Trans. Power Systems*, vol. 4, no. 1, pp. 220–228, Feb. 1989.

[2] M. A. El-Sharkawi, R. J. Marks II, M. E. Aggoune, D. C. Park, M. J. Damborg, and L. E. Atlas, “Dynamic security assessment of power systems using back error propagation artificial neural networks,” in *Second Symposium on Expert System Application to Power Systems*, Seattle, WA, 1989.

[3] Y. Mansour, E. Vaahedi, and M. A. El-Sharkawi, “Dynamic security contingency screening and ranking using neural networks,” *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 942–950, July 1997.

[4] J. Kittler, A. Etemadi, and N. Choakjarernwanit, “Feature selection and extraction in pattern recognition,” in *Pattern recognition and Image processing in physics*, R. A Vaughan, Ed., 1990, Proceedings of the 37th Scottish University summer school in physics.

[5] R. D. Reed and R. J. Marks II, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: MIT Press, 1999..

[6] S. Weerasooriya and M. A. El-Sharkawi, “Use of Karhunen-Loe’ve expansion in training neural networks for static security assessment,” in *Proceedings of the First International Forum on Applications of Neural Networks to Power Systems*, Seattle, WA, July 1991, pp. 59–64.

[7] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, pp. 179–188, 1936.

[8] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*: John Wiley and Sons, 1973.

[9] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd ed: Addison-Wesley, 1994.

[10] H. J. Larson, *Introduction to Probability Theory and Statistical Inference*: John Wiley and Sons, 1969.

[11] C. A. Jensen. (1999) *Qwiknet Neural Network Design Software Version 2.22* [Online]. Available: http://www.kagi.com/cjensen

[12] Extended Transient Midterm Stability Program, Version 3.0, Palo Alto, California, vol. 1–6, 1993.

[13] “Transient stability test systems for direct stability methods,” *IEEE Trans. Power Systems*, vol. 7, no. 1, pp. 37–44, Feb. 1992.

Craig A. Jensen received the BSEE and MSEE degrees from the University of North Dakota in 1993 and 1995, respectively, and the Ph.D. degree from the University of Washington in 1999. He is currently working for Microsoft Company as a software engineer. He was employed at Otter Tail Power Company in Fergus Falls, MN as an Engineering intern in the System Planning and System Engineering departments. His research interests are neural networks, computational intelligence, intelligent systems applications to power systems and computer programming.

Mohamed A. El-Sharkawi is a Fellow of IEEE. He received the B.Sc. degree in Electrical Engineering in 1971 from Cairo High Institute of Technology, Egypt. He received the M.A.Sc and Ph.D. degrees in Electrical Engineering were from the University of British Columbia in 1977 and 1980, respectively. In 1980, he joined the University of Washington and is presently a Professor of Electrical Engineering. Dr. El-Sharkawi is the founder of the international conference on the Application of Neural Networks to Power Systems (ANNPS), and a co-founder of the international conference on Intelligent Systems Applications to Power (ISAP). He is a member of the administrative committee of the IEEE Neural Networks Council representing the Power Engineering Society, and the multi-media Tutorial Chair of the IEEE Neural Networks Council. He is a founding Chairman of several IEEE task forces and working groups and subcommittees, including the task force on Application of Neural Networks to Power Systems, the working group on Advanced Control Strategies for dc-type Machines, and the task force on Intelligent Systems Application to Dynamic Security Assessment. He is a co-founder of the IEEE Subcommittee on Intelligent Systems. He is a current or past member of the editorial board or associate editor of several journals including the IEEE Transactions on Neural Networks, the Engineering Intelligent Systems, and the International Journal of Neurocomputing. Dr. El-Sharkawi is the co-editor of the IEEE tutorial book on the applications of NN to power systems, and the author of a textbook on “fundamentals of Electric Drives” published by Brooks/Cole in 2000. Dr. El-Sharkawi was the chairman of the IEEE International Electric Machines and Drives held in Seattle, May 1999.

He organized and taught several international tutorials on intelligent systems applications, power quality and power systems, and organized and chaired numerous panel and special sessions in IEEE and other international conferences.

He has published over 140 papers and book chapters in these areas and holds 5 licensed patents:

Robert J. Marks, II is a Professor and Graduate Program Coordinator in the Department of Electrical Engineering in the College of Engineering at the University of Washington, Seattle. He is Fellow of both IEEE and The Optical Society of America.

Dr. Marks served as the first President of the IEEE Neural Networks Council. In 1992, he was given the honorary title of Charter President. He served as the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS and as a topical editor for Optical Signal Processing and Image Science for the *Journal of the Optical Society on America A*. Space limitations and a keen sense of modesty prohibit listing the rest of his remarkable professional achievements.

Dr. Marks is the co-author of the book entitled “Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks” (MIT Press, 1999)