

# Support Vector Machines for Transient Stability Analysis of Large-Scale Power Systems

L. S. Moulin, A. P. Alves da Silva, *Senior Member, IEEE*, M. A. El-Sharkawi, *Fellow, IEEE*, and R. J. Marks II, *Fellow, IEEE*

**Abstract**—The pattern recognition approach to transient stability analysis (TSA) has been presented as a promising tool for online application. This paper applies a recently introduced learning-based nonlinear classifier, the support vector machine (SVM), showing its suitability for TSA. It can be seen as a different approach to cope with the problem of high dimensionality. The high dimensionality of power systems has led to the development and implementation of feature selection techniques to make the application feasible in practice. SVMs' theoretical motivation is conceptually explained and they are tested with a 2684-bus Brazilian system. Aspects of model adequacy, training time, classification accuracy, and dimensionality reduction are discussed and compared to stability classifications provided by multilayer perceptrons.

**Index Terms**—Feature selection, neural networks, support vector machine, transient stability analysis.

## I. INTRODUCTION

THE increasing load demand in power systems without accompanying investments in generation and transmission has affected the analysis of stability phenomena, requiring more reliable and faster tools. One of the most challenging problems in real-time operation of power systems is the assessment of transient stability. Its importance has increased due to the reduction of operational safety margins. Analytical techniques alone do not allow to take preventive or corrective actions in due time. A possible solution to overcome this drawback is the application of the pattern recognition approach.

Research efforts in communication and computer processing have enabled the development of online tools for transient stability analysis (TSA) [1]–[3]. A number of pattern recognition methods have been reported as playing important roles in such tools [1], [4], [5]. The integration of automatic learning/pattern recognition techniques with analytical TSA methods can provide more accurate monitoring, improved use of power systems resources (e.g., reduced spinning reserves), flexibility in maintenance scheduling, etc. [2]. Besides avoiding the repetitive burden of analyzing similar operating points, the pattern

recognition approach for online TSA [or even assessment (i.e., including control)] can deal with modeling uncertainties (e.g., dynamic load modeling [6]) and measurement errors.

Analytical methods hardly provide, alone, all functionalities that control center operators would like to have, which are

- current operating point qualitative evaluation;
- stability margins;
- visualization of security regions;
- available transfer capability;
- preventive and/or corrective controls;
- “optimum” load shedding.

The pattern recognition approach for online TSA can fulfill directly the first four operators' desires, and could also help on providing the last two. In particular, ultrafast stability margin estimation can provide a feedback variable with a system-wide view for the controllers. So far, online centralized coordination has not been possible for the control of fast phenomena. The current decentralized approach, based on local measurements, does not produce adequate pre and postcontingency control, reducing stability limits and increasing the need for more stabilizers.

Neural networks (NNs) technology has been reported as an important contributor for reaching the goals of online TSA [1], [2], [7]–[16]. It presents desirable characteristics, such as fast response in simple format (stable/unstable or stability margin), heavy computational burden is paid offline, failure tolerant with respect to data requirement, and it can allow better real-time control. Explanation capability can also be introduced through the extraction of if-then rules from the NN [17]. Recent proposals of NNs' application to online TSA show how these properties can be turned into practical use. In general, these proposals present one of the following ideas:

- a) to rank or screen the contingencies, and after that perform detailed time-domain simulations [2], [12], [13];
- b) to provide a stability evaluation during time-domain simulations, halting the cases clearly evaluated as stable [14];
- c) to provide fast stability evaluations and allow border identification [11], [18].

In most of the NN proposals for online TSA, multilayer perceptrons (MLPs) are used, which present, as a major drawback, the extensive training process. Like other nonlinear learning machines, they lack simple design procedures. In estimating a NN, one is found between two opposing extremes: i) to use lots of data for learning and suffer from long training, or ii) to use less data, and suffer from “insufficient” learning.

Support vector machines (SVMs), a recently introduced learning paradigm, have very interesting theoretical and prac-

Manuscript received August 28, 2003. This work was supported in part by the Brazilian Research Council (CNPq).

L. S. Moulin is with the Electric Power Research Center (CEPEL), Ilha da Cidade Universitária, Rio de Janeiro, RJ 21941-590 Brazil (e-mail: moulin@cepel.br).

A. P. Alves da Silva is with the Federal University of Rio de Janeiro, Rio de Janeiro, RJ 21945-970, Brazil (e-mail: alex@coep.ufjf.br).

M. A. El-Sharkawi and R. J. Marks II are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195-2500 USA (e-mail: elsharkawi@ee.washington.edu; marks@ee.washington.edu).

Digital Object Identifier 10.1109/TPWRS.2004.826018

tical characteristics [19], [20]. They rely on so called support vectors (SVs) to identify the decision boundaries between different classes. The SVs are located near the separation surfaces, which are critical to achieve correct classifications. SVMs can map complex nonlinear input/output relationships, and they are very well suited for TSA because the learning focus is on the security border. SVMs are based on a linear machine in a high dimensional feature space, nonlinearly related to the input space, which has allowed the development of somewhat fast training techniques, even with a large number of input variables and big training sets [21]. Investigations of the application of SVMs to TSA can be found in [7]–[10]. In the present work, it is shown that SVMs cope with the demands of large power systems' TSA, and how they compare to MLPs.

Feature selection techniques have been previously proposed to make the matter of high dimensionalities easier, especially in TSA, where the power system representation leads to a large number of input features [16]. Feature selection reduces the input dimensionality in order to use as few variables as possible, getting a more concise representation of the power system.

This paper presents the application of MLP and SVM classifiers to the TSA of a real power system, the Brazilian Priba, comprising 2484 buses, 200 generation buses, and 5720 branches. The paper shows how the large input dimensionalities represent a concern in stability classification. The paper also presents a comparison between MLP and SVM models, since the former is used in almost all previous proposals of NN application to TSA. Aspects of model adequacy, training time, classification accuracy, and dimensionality reduction are discussed, may any of the ideas (a), (b), or (c) be pursued.

The structure of the paper is as follows. In Section II, a summarized description of SVM classifiers is sketched considering the conceptual ideas and discussions on advantages and disadvantages. Section III describes the power system used in the tests and how the transient stability data have been collected for the NNs' training. In Section IV, the details about the MLP and SVM training procedures are presented, including the feature selection processing. In Section V, the results of NN stability classifications are presented. Comparisons and discussions about the two models are also carried out. Finally, conclusions are drawn in Section VI.

## II. SUPPORT VECTOR MACHINE CLASSIFIERS

SVMs are nonlinear models based on theoretical results from the statistical learning theory [19]. This theoretical framework formally generalizes the empirical risk minimization principle that is usually applied for NN training (i.e., the minimization of the number of training errors). In traditional NN training, several heuristics are applied in order to estimate a classifier with adequate complexity for the problem at hand.

An SVM classifier minimizes the generalization error by optimizing the tradeoff between the number of training errors and the so-called Vapnik-Chervonenkis (VC) dimension, which is a new concept of complexity measure.

A formal theoretical bound exists for the generalization ability of an SVM, which depends on the number of training errors ( $t$ ), the size of the training set ( $N$ ), the VC dimension

associated to the resulting classifier ( $h$ ), and a chosen confidence measure for the bound itself ( $\eta$ ) [19]

$$R < \frac{t}{N} + \sqrt{\frac{h(\ln(2N/h) + 1) - \ln(\eta/4)}{N}} \quad (1)$$

The risk  $R$  represents the classification error expectation over the entire population of input/output pairs, even though the population is only partially known. This risk is a measure of the actual generalization error and does not require prior knowledge of the data probability density function. Statistical learning theory derives inequality (1) to mean that the generalization ability of an SVM is bound by the right-hand side of (1). This upper limit is valid with probability  $1 - \eta$  ( $0 < \eta < 1$ ). As  $h$  increases, the first summand of the upper bound (1) decreases while the second summand increases, so that there is a balanced compromise between the two terms (i.e., training error and complexity), respectively.

Consider a training set  $T = \{\mathbf{x}_i, y_i\}_{i=1}^N$ , where  $\mathbf{x}_i$  is a real-valued  $n$ -dimensional input vector (i.e.,  $\mathbf{x}_i \in R^n$ ) and  $y_i \in \{+1, -1\}$  is a label that determines the class of  $\mathbf{x}_i$ . The SVMs employed for two-class problems are based on hyperplanes to separate the data, as exemplified by Fig. 1. The hyperplane (indicated by the dotted line in Fig. 1) is determined by an orthogonal vector  $\mathbf{w}$  and a bias  $b$ , which identifies the points that satisfy  $\mathbf{w}^t \cdot \mathbf{x} + b = 0$ . By finding a hyperplane that maximizes the margin of separation  $\rho$ , it is intuitively expected that the classifier will have a better generalization ability. The hyperplane with the largest margin on the training set can be completely determined by the nearest points to the hyperplane. Two such points are  $\mathbf{x}_1$  and  $\mathbf{x}_2$  in Fig. 1(b), and they are called SVs because the hyperplane (i.e., the classifier) depends entirely on them.

Therefore, in their simplest form, SVMs learn linear decision rules as

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^t \cdot \mathbf{x} + b) \quad (2)$$

so that  $(\mathbf{w}, b)$  are determined to classify correctly the training examples and to maximize  $\rho$ .

To show the underlying reason for doing this, consider the fact that it is always possible to scale  $\mathbf{w}$  and  $b$  so that

$$\mathbf{w}^t \cdot \mathbf{x} + b = \pm 1 \quad (3)$$

for the SVs with

$$\mathbf{w}^t \cdot \mathbf{x} + b > +1 \quad \text{and} \quad \mathbf{w}^t \cdot \mathbf{x} + b < -1 \quad (4)$$

for non-SVs. Using the SVs  $\mathbf{x}_1$  and  $\mathbf{x}_2$  of Fig. 1 and (3), the margin  $\rho$  can be calculated as

$$\rho = \frac{\mathbf{w}^t \cdot (\mathbf{x}_2 - \mathbf{x}_1)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}. \quad (5)$$

For linearly separable data, the VC dimension of SVM classifiers can be estimated by [19]

$$h < \min \left\{ n, \frac{4D^2}{\rho^2} \right\} + 1 = \min \{ n, D^2 \|\mathbf{w}\|^2 \} + 1 \quad (6)$$

where  $D$  is the minimum radius of a ball which contains the training points. For linearly separable data, as shown in Fig. 1,

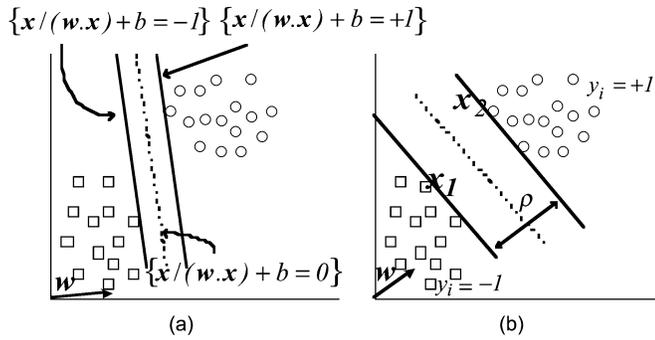


Fig. 1. Maximum margin classifier.

a linear classifier can be found such that the first summand of bound (1) is zero. Therefore, the risk (1) can be reduced by decreasing the complexity of the SVM (i.e., by increasing the margin of separation  $\rho$ , which is equivalent to decreasing  $\|w\|$ ).

As practical problems are not likely to be linearly separable, the linear SVM has been extended to a nonlinear version by mapping the training data to an expanded feature space using a nonlinear transformation

$$\Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_m(\mathbf{x}))^t \in \mathbf{R}^m \quad (7)$$

where  $m > n$ . Then, the maximum margin classifier of the data in the new space can be determined. With this procedure, the data that are nonseparable in the original space may become separable in the expanded feature space. The next step is to estimate the SVM by minimizing (i.e., maximizing  $\rho$ )

$$V(\mathbf{w}) = \frac{1}{2} \mathbf{w}^t \cdot \mathbf{w} \quad (8)$$

subject to the constraint that all training patterns are correctly classified, that is

$$y_i \cdot \{\mathbf{w}^t \cdot \Phi(\mathbf{x}_i) + b\} \geq 1, \quad i = 1, \dots, N. \quad (9)$$

However, depending on the type of nonlinear mapping (7), the training points may not happen to be linearly separable, even in the expanded feature space. In this case, it will be impossible to find a linear classifier that fulfills all of the conditions (9). Therefore, a new cost function is used, instead of (8)

$$V(\mathbf{w}, \varepsilon) = \frac{1}{2} \mathbf{w}^t \cdot \mathbf{w} + C \sum_{i=1}^N \varepsilon_i \quad (10)$$

where  $N$  non-negative *slack* variables  $\varepsilon_i$  are introduced to allow for training errors (i.e., training patterns for which  $y_i \cdot \{\mathbf{w}^t \cdot \Phi(\mathbf{x}_i) + b\} \geq 1 - \varepsilon_i$  and  $\varepsilon_i > 1$ ). By minimizing the first summand of (10), the complexity of the SVM is reduced, and by minimizing the second summand of (10), the number of training errors is decreased.  $C$  is a preselected positive penalty factor that acts as a tradeoff between the two terms.

The minimization of the cost function (10) leads to a quadratic optimization problem with a unique solution. In fact, the nonlinear mapping (7) is indirectly obtained by the so-called Mercer Kernel functions, which correspond to inner products of data vectors in the expanded feature space

$K(\mathbf{a}, \mathbf{b}) = \Phi(\mathbf{a})^t \cdot \Phi(\mathbf{b})$ ,  $\mathbf{a}, \mathbf{b} \in \mathbf{R}^m$ . Because the SVM formulation ends up with an inner product format (see the Appendix for more details), the Kernel function can substitute the nonlinear mapping (7) wherever it appears. In order for this equivalence to be valid, a Kernel function must satisfy some requirements called Mercer Conditions [20]. The most commonly used functions are the RBF kernel

$$K(\mathbf{a}, \mathbf{b}) = e^{-\frac{\|\mathbf{a}-\mathbf{b}\|^2}{\sigma^2}} \quad (11)$$

and the polynomial kernel

$$K(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^t \cdot \mathbf{b} + 1)^p \quad (12)$$

where the parameters  $\sigma$  and  $p$  in (11) and (12) must be preset. One important advantage of using a Kernel function instead of the nonlinear mapping (7) is that some of its key aspects like representation, complexity, and generalization capability become highly dependent on a few control parameters, as will be shown later. Another important advantage is related to the computational complexity of the large expanded dimension space  $\mathbf{R}^m$ . For example, the polynomial kernel (12) corresponds to a nonlinear expanded space of dimension  $m = \binom{n+p}{p}$ , and the features  $\phi_j(\mathbf{x})$ ,  $j = 1, \dots, m$ , represent all of the monomials of the original input vector up to and including degree  $p$ . In power systems, where  $n$  is typically large,  $m$  would become computationally intractable. However, by substituting the nonlinear mapping by the Kernel function, all calculations are performed in the original input space dimension.

In summary, a nonlinear mapping (7) can be indirectly defined by a Kernel function [i.e., there is no need for specifying (7)], for example (11) or (12). Overfitting problems in the expanded feature space are overcome by implicit generalization control in the learning process. The parameters  $\sigma$  and  $p$  affect how sparse and easily separable the data are in the expanded feature space, and consequently, they affect the complexity of the resulting SVM classifier and the training error rate. The parameter  $C$  also affects the model complexity. Currently, there is no indication, besides trial and error, on how to set  $C$ , to choose the best Kernel function, and to set the Kernel parameters. In practice, a range of values has to be tried for  $C$  and for the Kernel parameters, and then the performance of the SVM classifier is estimated for each of these values (and Kernel functions). Details on the minimization of (10) and the SVM architecture are shown in the Appendix.

### III. POWER SYSTEM DESCRIPTION AND DATA SET GENERATION

The power system used for TSA tests is a subsystem of the Brazilian southeast grid, which is located in the region with the largest power consumption in the country. The system is basically formed by the hydroelectric plants along Paranaíba and Grande rivers, and by the power grid around these plants. This so-called "Priba System" has 2484 buses, 200 generation buses, and 5730 branches, including 26 major 750-kV, 500-kV, and 345-kV transmission lines and transformers connecting the generation plants to load centers and to other subsystems.

The transient stability studies assume that 14 of the major branches become unavailable due to maintenance scheduling,

one at a time. For each major branch outage for maintenance scheduling, single contingencies are assumed in nine other major branches. The contingencies consist of three-phase short-circuits, which are cleared by tripping the corresponding line. Three load levels (light, medium, and heavy) have been simulated, besides the combinations of different generation dispatches and power exchanges between subsystems. These base cases for TSA have been simulated in the time domain, and each one has been classified as stable or unstable. With this procedure, 994 training patterns and 248 test patterns have been obtained. The TSA data set has a large percentage of stable cases, with the ratio of stable to unstable cases of approximately 10:1.

It is important to clarify that the data used in this work have not been produced for estimating the proposed classifiers. In fact, the simulations were performed during operational studies for an electric utility, without any specific concern regarding the classifiers' design. These studies include the specialists' knowledge about the list of the most important contingencies, the most important variables, typical operating conditions, the required accuracy in power system modeling, etc., which represent the utility's expertise as far as offline TSA is concerned. All of this knowledge, which has been considered and included in the training set, makes up the necessary information to be learned by a NN. This kind of data is usually available in electric utilities, and this paper intends to show that useful classifiers can be obtained from it without any other data requirement.

However, as power systems are planned to operate most of the time under stable conditions, operational studies usually generate highly unbalanced classes. This unbalance between the stable and unstable classes can be troublesome for some classifiers.

Based on the approach developed in [9]–[11], and [16], the following input variables have been chosen to describe the power system operating point and the applied fault: active and reactive power at selected generation buses before fault occurrence; active and reactive power flows on the 26 major branches before fault occurrence; and a binary coding for the nine faults under analysis. The output variable has been chosen to be the two classes of interest: stable (+1) or unstable (−1).

Taking into account the generation buses that yielded relevant information (i.e., significant variation on the generated power), 224 input variables have been preselected. Topology information is implicitly informed by the branches' power flow variables ("no flow" means an open line).

#### IV. NEURAL NETWORKS TRAINING

This section presents details about the MLP and SVM training procedures, including the feature selection processing.

##### A. Multilayer Perceptron Training

The MLPs have been trained by the Stuttgart NNs Simulator [22], which is a free software developed in C. The back-propagation training with adaptive learning and momentum rates, and cross-validation have been used. Cross-validation has been performed by randomly splitting the original training set and

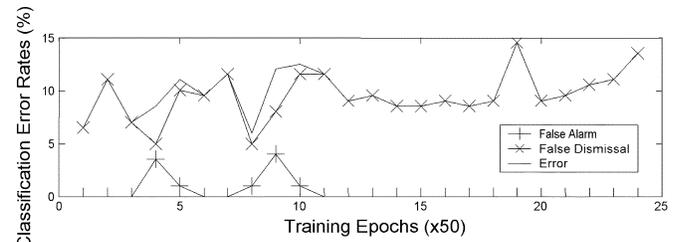


Fig. 2. MLP training on TSA data set.

reserving 20% of its patterns for validation. The random splitting was repeated at every 50 epochs (one epoch is one training cycle for which all training patterns have been presented to the NN once), when the training and validation errors were also calculated in order to monitor their behavior and to stop training earlier (i.e., before overfitting).

It has been noticed that during the entire training process, the false dismissal rate (rate of unstable cases assigned to the stable class) is high and the false alarm rate (rate of stable cases assigned to the unstable class) is close to zero, which is highly undesirable. The training set unbalance between the two classes of interest damages the MLP classifier estimation, because it overfits the stable data. An example of this overfitting is in Fig. 2, which presents the classification performance during the MLP training process. When training is terminated, after 1000 epochs (20 × 50), all of the errors are false dismissals. The error rate is the sum of the false dismissal and false alarm rates.

To try to avoid overfitting the stable data and to decrease the false dismissal rates, a new training procedure has been performed. An augmented training set has been devised with the ratio between stable and unstable patterns artificially modified. A 1:1 ratio has been set by adding copies of unstable training patterns to the original training set and the results on the test set will be presented in Section V.

##### B. SVM Training

The SVM classifier is based on a subset of the training patterns, the support vectors, located at the separation region between the two classes. The SVs define the largest possible margin of separation. Two different kernel functions have been used, the RBF kernel (11) and the Polynomial kernel (12). The parameters  $C$  (10),  $\sigma$  (11), and  $p$  (12) have been searched heuristically, trying to achieve the best generalization capacity. The software SVM<sup>light</sup> [21], developed in C, has been used for training and testing the SVM models.

The SVM training process consists of a quadratic optimization problem in which the support vectors represent the minimum solution. The use of an augmented training set as in the MLP training is not appropriate because of linear dependencies in the constraints. Instead, to account for the training set unbalance, different values for  $C$  can be used. A large value of  $C$  for the unstable patterns and a small value for the stable ones have been adopted during the training process (the corresponding values of  $C$  have been multiplied by 0.1 for the stable training patterns). In this way, the optimization process emphasizes the minimization of the unstable patterns training errors. Different values for  $C$  and for the parameter  $p$  have been tried.

The RBF kernel SVMs have not shown satisfactory results, because in the test set they have maximum false dismissal rate and 0% of false alarm rate, no matter the values of the parameters. On the other hand, polynomial kernel SVMs have been trained successfully, and the results of their performance on the test set will be presented in Section V.

### C. Feature Selection

Because of the high dimensionality of the input space, feature selection techniques have also been applied to achieve a more concise representation of the power system and overcome the curse of dimensionality.

According to the notation introduced in Section II, a classification task in two groups is represented by ordered pairs  $(\mathbf{x}_i, y_i)$  in the training set  $T = \{\mathbf{x}_i, y_i\}_{i=1}^N$ , where  $\mathbf{x}_i$  represents the operating point and  $y_i \in \{+1, -1\}$  denotes the security index of that point. The main objective of a feature selection technique is to generate a  $d$ -dimensional feature vector  $\mathbf{f}$  where  $d < n$ . The “ $d$ ” selected features represent the original data in a new training set  $T' = \{\mathbf{f}_i, y_i\}_{i=1}^N$ .

If the feature selection is successful, a point in  $R^d$  can be assigned to one of the two classes with minimum error. Two feature selection techniques are used in this work, as presented in [16]: sequential search and genetic algorithms (GAs). Reductions on the data set dimensionality from  $n = 224$  to  $d = 150, 100$ , and  $50$  have been tested.

## V. RESULTS AND DISCUSSIONS

The following comparison between MLP and SVM models results from an extensive search over their parameters and the number of input variables ( $n$  and  $d$ ). The classifiers have been initially trained to achieve low classification error rates. After that, the ones having the lowest false dismissal rates have been picked.

Taking  $\bar{y}$  as the NN output (stability condition estimated by the NN), notice that the output range of the MLP model is  $[-1, +1]$ , whereas the output range of the SVM model is  $[-\infty, +\infty]$ . Operating points close to the stability border (outputs close to zero) indicate a dangerous situation, even if they are on the stable side, and deserve similar treatment as the unstable points. Therefore, besides stable and unstable, it is useful to take a third stability classification, according to the classifiers' outputs  $\bar{y}$ . A high risk range is devised near the 0 classification threshold, and if  $\bar{y}$  falls within this range, the point is classified as “high risk” (to mean it is not considered stable nor unstable). For the SVM classifier, points with output values in the range  $[0, +1]$  are natural candidates for high risk cases, because they are located between the support vectors of different classes and near the classification border.

Taking these considerations into account, the results of the two classifiers on the test set are presented in Table I. Column (2) shows the classification results with the high risk range  $[0, +1]$  for the SVM with 224 inputs,  $C = 1000$  and  $p = 2$ . The false dismissal cases occur when  $\bar{y} > +1$  and  $y \leq 0$ . The false alarm cases occur when  $\bar{y} \leq 0$  and  $y > +1$ . The high risk cases occur whenever  $0 < \bar{y} \leq +1$ . The error rate is the sum of the false dismissal and false alarm rates. The total error, false dismissal,

TABLE I  
SVMs AND MLPs PERFORMANCES ON THE TEST SET

(1)	(2)	(3)	(4)	(5)	(6)
<b>Model</b>	SVM	SVM	MLP	MLP	MLP
<b>Feature Selection</b>	--	--	GA	GA	GA
<b>No. of Inputs</b>	224	224	150	150	150
<b>C</b>	1000	1000	--	--	--
<b>p</b>	2	2	--	--	--
<b>SVs</b>	168	168	--	--	--
<b>Hidden Neurons</b>	--	--	15	15	15
<b>Training Time</b>	18.6	18.6	46.7	46.7	46.7
	cpu s	cpu s	cpu min	cpu min	cpu min
<b>Error Rate (%)</b>	5.6	4.8	14.9	14.1	14.1
<b>F. Dismissal Rate (%)</b>	0.8	0	3.2	2.4	2.4
<b>F. Alarm Rate (%)</b>	4.8	4.8	11.7	11.7	11.7
<b>High Risk Rate (%)</b>	14.1	27.0	3.23	35.5	36.3
<b>High Risk Range</b>	[0, 1]	[0, 1.6]	[0, 0.75]	[0, 0.85]	[0, 0.95]

and false alarm rates are calculated as a percentage of 248 test patterns (225 stable and 23 unstable). Results with a more conservative high risk range  $[0, +1.6]$  are shown in column (3) of Table I for the same SVM classifier, where the performance rates are calculated like the case represented by column (2).

For the MLP classifier with 150 inputs selected by the GA, arbitrarily large high risk ranges  $[0, +0.75]$ ,  $[0, +0.85]$ , and  $[0, +0.95]$  have been defined, and the results are shown in columns (4), (5), and (6) of Table I, respectively. The performance rates have been calculated according to each high risk range, just as explained for the SVM model.

The use of such high risk ranges is well known in NN applications to TSA, where false dismissal rates must be very close to 0%. There is a clear compromise between the high risk ranges and the false dismissal rates, because one increases as the other decreases. In a contingency screening application, by taking the SVM classifier criterion of column (3) instead of the one in column (2), one would favor reliability and would conservatively accept  $(27\% - 14.1\%) = 12.9\%$  more high risk cases. Although most of these would be stable cases, they could start up preventive actions or detailed stability simulations, depending on the specific approach taken. As the high risk range is increased, the classifier becomes more reliable (i.e., with lower false dismissal rate), but less effective in screening out true stable cases. The false dismissal rate would decrease from 0.8% to 0% and both classifiers of columns (2) and (3) have good performances. The decision about which one to take is a project decision. The conceptual description of high risk cases for the SVMs leads to a comprehensive definition of high risk ranges, whose validity is confirmed in practice by the results presented in Table I.

By taking the MLP of column (5) instead of the one of column (4), the false dismissal rate would decrease from 3.2% to 2.4%. However, it cannot be lowered further, even by setting a very large upper limit for the high risk range, as it is shown in column (6) of Table I.

A false dismissal rate of 0.8% (2/248) on the test set means that 8.7% (2/23) of the unstable patterns in the test set have been misclassified. A false dismissal rate of 2.4% (6/248) means that 26.1% (6/23) of the unstable patterns in the test set have been misclassified.

For the MLP, feature selection allows a gain in generalization resulting from the dimensionality reduction, despite the loss of information due to the discarded variables. However, the best MLPs offer no competition to the best SVMs. Surprisingly, the dimensionality reduction does not provide any improvement to the overall performance of the SVMs, if compared to the ones estimated with the original inputs.

Table I also shows the training times in cpu seconds and cpu minutes, run in an 850-MHz PC. The SVM training time is lower than the MLP training by one order of magnitude. It is worth noting that the training set size is not large compared to the number of MLP parameters. In fact, larger training sets would be recommended for a better representation of the power system behavior, though the relationship between the computational burden and the training set cardinality is inconvenient when employing backpropagation learning.

The number of SVs (168) determines the number of free parameters of the SVM classifier [see Fig. 4 and (15) in the Appendix, and recall that  $\alpha_i > 0$  only for the SVs]. That helps to understand why a training set with 994 patterns only is enough to estimate an SVM with good performance, despite the large number of input variables. The results show how the SVM models can take advantage of a stability study database already available in a control center. Their performance could continuously improve during the online TSA process by adding new operating conditions to the training set.

The SVM performance stems from its capacity to generalize well from the available training data, which is related to an “implicit” feature selection ability. In order to make that clear, take the receiver operating characteristic (ROC) curves, shown in Fig. 3. This graph presents false dismissal rates on the horizontal axis and detection rates on the vertical axis. The detection rate is just an indirect measure of the false alarm rate, calculated by subtracting it from 100% (without considering the high risk range, that is, when the classification threshold is zero). Each ROC curve in Fig. 3 corresponds to SVMs trained with the indicated input variables and different values of  $p$ . For each curve, a specific value of  $C$  has been chosen to the best possible ROC, which is the one having the points with the lowest false dismissal rates coupled with the highest detection rates. ROC curves cannot be drawn for the MLPs, because there is no parameter to control the relationship between false alarm and false dismissal rates. The number of factors that affect an MLP’s performance are large and inter-related.

The “best” curve in Fig. 3 is from 224 inputs. As the number of input variables decreases, the curves get “worse,” which is an indication of loss of discrimination capability as variables are discarded from the original input set. On the other hand, it is also an indication of how good the SVMs can be on the high dimensional original space. That means, embedded in the learning process of an SVM there is an automatic feature selection, which prevents it from being trapped by the curse of dimensionality on the expanded feature space.

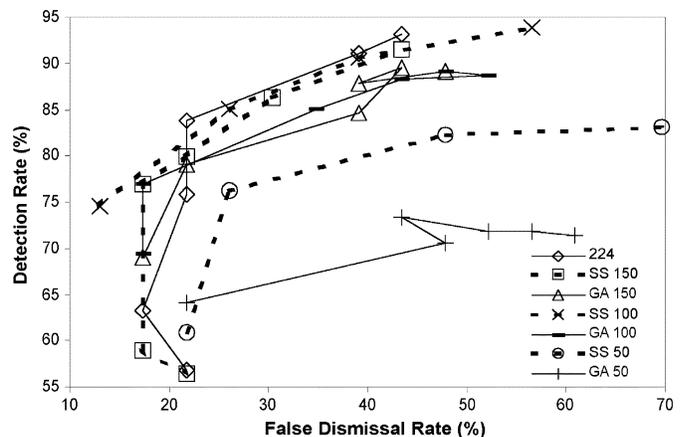


Fig. 3. Polynomial SVM ROC curves for different input sets.

Fig. 3 also shows that the parameters  $C$  and  $p$  can be used to control the relationship between the number of false dismissals and false alarms. For a specific value of  $C$ , it is possible to move the SVM classifier to the left of the ROC curve as much as desired to achieve lower false dismissal rates, at the expense of larger false alarm rates. This procedure gives some inspiration on a structured way to design an SVM classifier for TSA. For a selected Kernel function:

- perform a fine search over the values of  $C$  and of the Kernel parameter, and draw ROC curves for them;
- choose the curve with the “best” RO characteristic (the one having points with the largest ratios between detection rates and false dismissal rates);
- choose the point in the “best” ROC curve with the largest ratio between detection and false dismissal rates;
- pick the values of  $C$  and of the Kernel parameter that correspond to the point chosen in (c);
- choose a high risk range that provides acceptable false dismissal rate for the classifier estimated from (d).

## VI. CONCLUSION

This paper shows that SVMs fit the TSA task for large power systems. They provide a different strategy to tackle the curse of dimensionality. The SVMs performed better when the complete set of input variables was used, which confirms, in practice, their implicit feature selection capability and the validity of the theoretical developments on generalization control. The SVM learning machine allows a deep understanding of its practical implications, which can be used to devise structured design practices for the model.

The sparsity reduction of the data has turned the training process into an easier task for MLPs. However, the MLPs performance (3.2% of false dismissal rate, 11.7% of false alarm rate, and 3.23% of high risk rate) is not as good as the SVMs (0.8% of false dismissal rate, 4.8% of false alarm rate, and 14.1% of high risk rate). It has been shown that stability studies databases already available in electric utilities, containing specialists’ knowledge, can be used in NN-based TSA as a good starting point.

Future work will focus on dynamic features. The majority of work about TSA, based on the pattern recognition approach,

has focused the analysis on pre-fault static features. With the popularization of synchronized phasor measurements acquisition systems (phase angle monitors), loss of synchronization can be predicted, in real time, based on post-fault phasor measurements (i.e., speed and acceleration at each generation bus are calculated from the synchronized phasor measurements) [23]. Therefore, the next generation of transient stability assessment tools will be allowed to move from preventive countermeasures (contingency analysis on pre-fault operating points) to corrective control.

Another promising idea for online TSA of large-scale power systems is the hybrid approach based on direct-type methods coupled with detailed time simulation [24]. In this approach, NNs can be used as filters to discard stable contingencies in a very fast way.

#### APPENDIX

The computation of the decision boundary of an SVM  $f(\mathbf{x}) = \text{sign}(\mathbf{w}^t \cdot \Phi(\mathbf{x}) + b)$  for the nonseparable case consists in solving the following optimization problem:

$$\begin{aligned} \text{minimize: } V(\mathbf{w}, \varepsilon) &= \frac{1}{2} \mathbf{w}^t \cdot \mathbf{w} + C \sum_{i=1}^N \varepsilon_i \\ \text{subject to: } & y_i \cdot \{\mathbf{w}^t \cdot \Phi(\mathbf{x}_i) + b\} \geq 1 - \varepsilon_i, \quad i = 1, \dots, N \\ & \varepsilon_i \geq 0, \quad i = 1, \dots, N. \end{aligned} \quad (13)$$

Instead of solving (13) directly, it is much easier to solve the dual problem (14), in terms of the Lagrange multipliers  $\alpha_i$

$$\begin{aligned} \text{minimize } W(\alpha) & \\ &= - \sum_{i=1}^N \alpha_i + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \Phi(\mathbf{x}_i)^t \cdot \Phi(\mathbf{x}_j) \\ &= - \sum_{i=1}^N \alpha_i + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (14) \\ \text{subject to: } & \sum_{i=1}^N y_i \alpha_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \end{aligned}$$

which is a quadratic optimization problem. From the solution,  $\alpha_i, i = 1, \dots, N$  of (14), the decision rule  $f(\mathbf{x})$  can be computed as

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{w}^t \cdot \Phi(\mathbf{x}) + b = \sum_{i=1}^N \alpha_i y_i \Phi(\mathbf{x}_i)^t \cdot \Phi(\mathbf{x}) + b \\ &= \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \end{aligned} \quad (15)$$

The training points with  $\alpha_i > 0$  are the SVs, and (15) depends entirely on them. The threshold  $b$  can be calculated using (3), which is valid for any SV

$$b = y_{SV} - \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_{SV}). \quad (16)$$

An SVM can be represented as in Fig. 4, where the number of units  $K(\mathbf{x}, \mathbf{x}_i)$  is determined by the number of SVs.

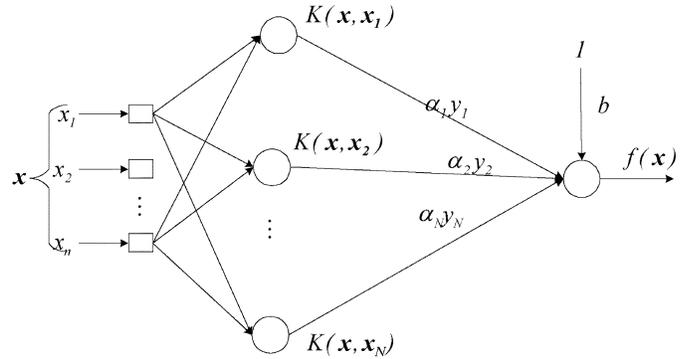


Fig. 4. SVM architecture.

#### ACKNOWLEDGMENT

The authors would like to thank the researchers at GESIS Lab, UNIFEI (Federal University at Itajubá, Brazil), for making the Transient Stability studies used in this paper available. They would also like to thank the reviewers for their valuable comments and questions, which helped to improve this work.

#### REFERENCES

- [1] L. A. Wehenkel, *Automatic Learning Techniques in Power Systems*. Norwell, MA: Kluwer, 1998.
- [2] J. L. Jardim, C. A. da S. Neto, A. P. A. da Silva, A. C. Zambroni de Souza, D. M. Falcão, C. L. T. Borges, and G. N. Taranto, "A unified online security assessment system," in *Proc. CIGRÉ*, Paris, France, Aug. 2000.
- [3] Y. Mansour, E. Vaahedi, A. Y. Chang, B. R. Corns, B. W. Garrett, K. Demaree, T. Athay, and K. Cheung, "B. C. Hydro's on-line transient stability assessment (TSA) model development, analysis, and post-processing," *IEEE Trans. Power Syst.*, vol. 10, pp. 241–250, Feb. 1995.
- [4] D. J. Sobajic, Y.-H. Pao, and M. Djukanovic, "Neural networks for assessing the transient stability of electric power systems," *Neural Networks Applications in Power Systems*, pp. 255–294, 1996.
- [5] R. Fischl, D. Niebur, and M. A. El-Sharkawi, "Security assessment and enhancement," in *Artificial Neural Networks with Applications to Power Systems*, M. A. El-Sharkawi and D. Niebur, Eds., 1996, ch. 9, pp. 104–127. IEEE Catalog no. 96TP112-0.
- [6] A. P. A. da Silva, C. Ferreira, G. L. Torres, and A. C. Z. de Souza, "A new constructive ANN and its application to electric load representation," *IEEE Trans. Power Syst.*, vol. 12, pp. 1569–1575, Nov. 1997.
- [7] A. E. Gavoyiannis, D. G. Vogiatzis, and N. D. Hatzigryriou, "Dynamic security classification using support vector machines," in *Proc. IEEE Int. Conf. Intell. Syst. Applicat. Power Syst.*, Budapest, Hungary, June 2001, pp. 271–275.
- [8] A. E. Gavoyiannis, D. G. Vogiatzis, D. P. Georgiadis, and N. D. Hatzigryriou, "Combined support vector classifiers using fuzzy clustering for dynamic security assessment," in *Proc. Power Eng. Soc. Summer Meeting*, vol. 2, 2001, pp. 1281–1286. 2001.
- [9] L. S. Moulin, A. P. A. da Silva, M. A. El-Sharkawi, and R. J. Marks II, "Support vector and multilayer perceptron neural networks applied to power systems transient stability analysis with input dimensionality reduction," in *Proc. IEEE Power Eng. Soc. Summer Meeting*, Chicago, IL, July 2002.
- [10] —, "Neural networks and support vector machines applied to power systems transient stability analysis," *Int. J. Eng. Intell. Syst.*, vol. 9, no. 4, pp. 205–211, Dec. 2001.
- [11] I. N. Kassabalidis, M. A. El-Sharkawi, R. J. Marks II, L. S. Moulin, and A. P. A. da Silva, "Dynamic security border identification using enhanced particle swarm optimization," *IEEE Trans. Power Syst.*, vol. 17, pp. 723–729, Aug. 2002.
- [12] Y. Mansour, E. Vaahedi, and M. A. El-Sharkawi, "Dynamic security contingency screening and ranking using neural networks," *IEEE Trans. Neural Networks*, vol. 8, pp. 942–950, July 1997.
- [13] Y. Mansour, A. Y. Chang, J. Tamby, E. Vaahedi, and M. A. El-Sharkawi, "Large scale dynamic security screening and ranking using neural networks," *IEEE Trans. Power Syst.*, vol. 12, pp. 954–960, May 1997.

- [14] I. Kamwa, R. Grondin, and L. Loud, "Time-varying contingency screening for dynamic security assessment using intelligent-systems techniques," *IEEE Trans. Power Syst.*, vol. 16, pp. 526–536, Aug. 2001.
- [15] Y. M. Park, G.-W. Kim, H.-S. Cho, and K. Y. Lee, "A new algorithm for Kohonen layer learning with application to power system stability analysis," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 1030–1034, Dec. 1997.
- [16] L. S. Moulin, M. A. El-Sharkawi, R. J. Marks II, and A. P. A. da Silva, "Automatic feature extraction for neural network based power systems dynamic security evaluation," in *Proc. IEEE Int. Conf. Intell. Syst. Applicat. to Power Syst.*, Budapest, Hungary, June 2001, pp. 41–46.
- [17] P. J. Abrão, A. P. A. da Silva, and A. C. Z. de Souza, "Rule extraction from artificial neural networks for voltage security analysis," in *Proc. Int. Joint Conf. Neural Networks*, Honolulu, HI, May 2002.
- [18] J. D. McCalley, Q. Zhao, S. Wang, G. Zhou, R. T. Treinen, and A. D. Papalexopoulos, "Security boundary visualization for systems operation," *IEEE Trans. Power Syst.*, vol. 12, pp. 940–947, May 1997.
- [19] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
- [20] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [21] T. Joachims, *Advances in Kernel Methods—Support Vector Learning*, B. Scholkopf, J. C. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1998. Making large-scale SVM learning practical.
- [22] Stuttgart Neural Network Simulator [Online]. Available: <http://www-ra.informatik.uni-tuebingen.de/SNNS/>
- [23] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, "Decision trees for real-time transient stability prediction," *IEEE Trans. Power Syst.*, vol. 9, pp. 1417–1426, Aug. 1994.
- [24] D. Ernst, D. R. Vega, M. Pavella, P. M. Hirsch, and D. Sobajic, "A unified approach to transient stability contingency filtering, ranking and assessment," *IEEE Trans. Power Syst.*, vol. 16, pp. 435–443, Aug. 2001.

**Luciano S. Moulin** was born in Nanuque, Brazil, in 1972. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Federal Engineering School at Itajubá (EFEI), Itajubá, Brazil, in 1995 and 1998, respectively. He received the D.Sc. degree in electrical engineering from the Federal University at Itajubá (UNIFEI) [previously EFEI] in 2002.

Currently, he is a Researcher in Electrical Engineering with the Electric Power Research Center (CEPEL). During 2000, he was a Visiting Student in the Department of Electrical Engineering at the University of Washington, Seattle.

**Alexandre P. Alves da Silva (SM'00)** was born in Rio de Janeiro, Brazil, in 1962. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil, in 1984 and 1987, respectively, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 1992.

Currently, he is a Professor in Electrical Engineering at the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, where he and his group have developed intelligent forecasting and security assessment systems that are in operation at the control centers of Brazilian electric utilities. From 1993 to 2002, he was with the Federal Engineering School at Itajuba, Itajuba, Brazil. He was also with the Electric Energy Research Center (CEPEL), Rio de Janeiro, Brazil, from 1987 to 1988. During 1999, he was a Visiting Professor in the Department of Electrical Engineering at the University of Washington, Seattle. He has authored and co-authored many papers on intelligent systems application to power systems.

Dr. Alves da Silva was the Technical Program Committee Chairman of the First Brazilian Conference on Neural Networks in 1994, and of the International Conference on Intelligent System Applications to Power Systems in 1999.

**Mohamed A. El-Sharkawi (F'95)** received the B.Sc. degree in electrical engineering from Cairo High Institute of Technology, Cairo, Egypt, in 1971, and the M.Sc. and Ph.D. degrees from the University of British Columbia, Vancouver, BC, Canada, in 1977 and 1980, respectively.

Currently, he is a Professor of Electrical Engineering at the University of Washington, Seattle. He is the founder of the International Conference on the Application of Neural Networks to Power Systems (ANNPS), which was later merged with the *Expert Systems Conference* and renamed *Intelligent Systems Applications to Power (ISAP)*. He is the co-editor of the IEEE tutorial book on the applications of neural networks to power systems. He has published many papers and book chapters. He holds five patents: three on *Adaptive Var Controller* for distribution systems and two on *Adaptive Sequential Controller* for circuit breakers.

**Robert J. Marks, II (F'94)** is a Professor and Graduate Program Coordinator with the Department of Electrical Engineering at the College of Engineering, University of Washington, Seattle. He is the author of numerous papers and is co-author of the book *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. 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 of America*.

Dr. Marks is a Fellow of the Optical Society of America. He served as the first President of the IEEE Neural Networks Council. In 1992, he was given the honorary title of Charter President.