



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Power System On-Line Static Security Assessment by Using Multi-Class Support Vector Machines

M. Mohammadi and G.B. Gharehpetian
No. 424, Hafez Ave, Department of Electrical Engineering,
Amirkabir University of Technology, 15914, Tehran, Iran

Abstract: This study presents a multi-class Support Vector Machine (SVM) based method for on-line static security assessment of power systems. To classify the system security status, a group of binary SVMs have been trained. The multi-class Fisher score has been used for feature selection algorithm and the data selection has been done based on the confidence measure, to reduce the problem size and consequently to reduce the training time. The proposed method has been applied to New England 39-bus power system. The simulation results demonstrate the effectiveness and stability of the proposed method for on-line static security assessment procedure of large scale power systems.

Key words: Machine learning, multi-class Support Vector Machines (SVM), power system security

INTRODUCTION

The present trend towards deregulation has forced modern electric utilities to operate their systems under stressed operating conditions closer to their security limits. Under these conditions, any disturbance could endanger the system security. The static security of the power system is defined as the ability of the system, following a contingency, to reach an operating state within the specified safety and supply quality (Kirschen, 2002). As a result, the security evaluation is a major concern in the operation of modern power systems and there is a pressing need to develop fast on-line security monitoring method which could analyze the level of security and forewarn the system operators to take necessary preventive actions.

The evaluation of the present status of the power system and the impact of possible line or generator outage on the system security can be determined by solving the nonlinear load flow equations for all possible contingencies or for those ranked as the most important cases (Kirschen, 2002). These numerical methods are time consuming and therefore are not suitable for on-line applications.

The application of machine learning methods for on-line security assessment has been proposed by many researchers. For fast static security assessment Artificial Neural Networks (ANN) (Neibur and Germond, 1991; Shanti, 2008; Zhou *et al.*, 1994; Wehenkel, 1998), Decision

Trees (DT) (Wehenkel, 1998; Hatziaargyriou *et al.*, 1994) and Bayesian classifiers (Kim and Singh, 2005) have been suggested.

The most popular method is ANN, because of its ability to classify patterns and its good accuracy in comparison with other machine learning methods. Its disadvantages can be listed as follows:

- It requires an extensive training process and a complicated design procedure
- If some components of the pattern vector are strongly correlated, the methods become inaccurate and an accurate feature extraction must be performed to hopefully yield an uncorrelated set of components
- Although neural networks are good in interpolation but not good in extrapolation. Training sets have to represent the different states of the power system. This means that they need to comprise the complete pattern space of the secure and insecure power system operation. A large training set of input data is necessary, to provide the best results in the output.

Recently, Support Vector Machines (SVM), based on statistical learning theory, have been used in the different areas of machine learning, computer vision and pattern recognition, because of their high accuracy and good generalization capability (Vapnik, 1999; Platt, 1998). The main difference between ANN and SVM is in the principle

of risk minimization (Scholkopf, 1998). In the case of SVM, Structural Risk Minimization (SRM) principle is used to minimize an upper bound on the expected risk. But for ANN, traditional Empirical Risk Minimization (ERM) is used to minimize the error on the training data. This difference leads to a better generalization performance for SVM. The SVM has following advantages, too:

- The global optimum can be derived
- The over-fitting problem can be easily controlled by the choice of a suitable data separation margin
- The SVM algorithm is less sensitive to the size of input features and does not need to extract signal features
- SVM only needs a small quantity of samples to train the classifier

The application of SVM for dynamic security assessment has been reported by Moulin *et al.* (2004) and Wahab *et al.* (2007). In these references, the superior performance of SVM over ANN in terms of accuracy, speed and distribution of high-risk cases has been presented for a large scale power system transient stability.

In this study, different status for system security levels such as normal (class A), alert (class B), emergency 1 (class C), emergency 2 or system splitting status (class D); have been considered and a multi-class SVM has been used to classify the power system security.

The feature selection is one of the important steps in the classification problem. In this study, Fisher-like criterion (F-score) has been used for feature selection; also a data preprocessing procedure to select SVs candidates has been used, too.

SVM: SUPPORT VECTOR MACHINE

SVM has been introduced by Vapnik in the late 1960s on the foundation of statistical learning theory (Vapnik, 1999). However, since the middle of 1990s, the algorithms used for SVM have been started emerging with greater availability of computing power, paving the way for numerous practical applications.

Binary SVM formulation: The basic SVM deals with two-class problems, in which the data are separated by a hyper-plane defined by a number of Support Vectors (SVs). The performance of the SVM can be explained easily in two-dimensional plane, as shown in Fig. 1, without any loss of generality. This Fig. 1 shows a set of points for two different classes of data; circles (class A)

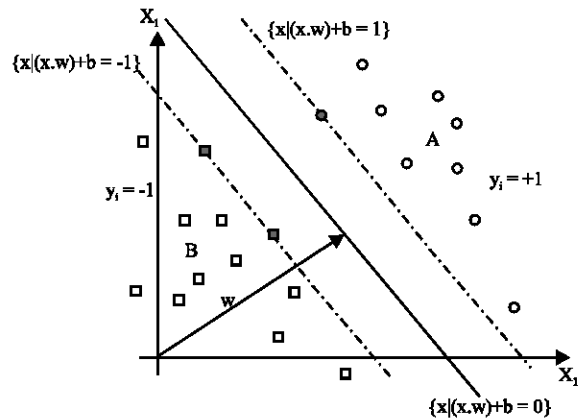


Fig. 1: SVM based classification of data

and squares (class B). The SVM attempts to place a linear boundary (solid line in Fig. 1) between the two different classes and shifts it to maximize the margin (dotted lines in Fig. 1). The distance between the boundary and the nearest data point in each class must be maximized. The boundary is then placed in the middle of this margin. The nearest data points, used to define the margins, are known as Support Vectors (SVs) (gray circle and square in Fig. 1). Once the Support Vectors (SVs) are selected, the rest of the points can be discarded, since the SVs contain all the necessary information for the classifier.

The boundary can be expressed by the following equation:

$$(w \cdot x) + b = 0, \quad w \in R^N, b \in R \tag{1}$$

where, the vector w defines the boundary, x is the input vector of dimension N and b is a scalar threshold. At the margins, where the SVs are located, we have:

$$(w \cdot x) + b = 1 \quad \text{For class A} \tag{2}$$

and

$$(w \cdot x) + b = -1 \quad \text{For class B} \tag{3}$$

As SVs correspond to the extremities of the data for a given class, the following decision function can be used to classify any data point in either class A or class B:

$$f(x) = \text{sign}((w \cdot x) + b) \tag{4}$$

The optimal hyper-plane separating the data can be obtained as a solution to the following optimization problem (Scholkopf, 1998):

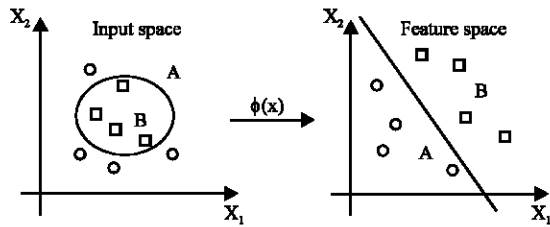


Fig. 2: Non-linear separation of input data and linear separation in feature space

$$\text{Minimize } \tau(w) = \frac{1}{2} \|w\|^2 \quad (5)$$

$$\text{Subject to: } y_i((w \cdot x_i) + b) \geq 1, \quad i = 1, 2, \dots, l \quad (6)$$

where, l is the size of the training set.

The solution of the constrained optimization problem can be obtained as follows (Scholkopf, 1998):

$$w = \sum_{i=1}^l v_i \cdot x_i \quad (7)$$

where, x_i are SVs obtained from training. Substituting Eq. 2 in Eq. 4, the decision function can be obtained as follows:

$$f(x) = \text{sign} \left(\sum_{i=1}^l v_i (x \cdot x_i) + b \right) \quad (8)$$

In cases where the linear boundary can't properly separate two classes, it is possible to create a hyper-plane that allows linear separation in the higher dimension. In SVM, this is achieved by using the transformation $\phi(x)$. This transformation converts the data from an N -dimensional input space (x) to a Q -dimensional feature space as follows:

$$s = \phi(x) \quad (9)$$

where, $x \in R^N$ and $s \in R^Q$. Figure 2 shows the transformation from the input space to the feature space where the non-linear boundary has been transformed into a linear boundary in the feature space.

Substituting the transformation Eq. 9 in Eq. 8 gives the decision function as follows:

$$f(x) = \text{sign} \left(\sum_{i=1}^l v_i (\phi(x) \cdot \phi(x_i)) + b \right) \quad (10)$$

The transformation into higher-dimensional feature space is relatively computation-intensive. A kernel can be used

to perform this transformation and the dot product in a single step provided the transformation can be replaced by an equivalent kernel function. This point helps in reducing the computational load and at the same time retaining the effect of higher-dimensional transformation. The kernel function, $K(x, x')$, is defined by the following equation:

$$K(x, x') = \phi(x) \cdot \phi(x') \quad (11)$$

The basic form of SVM is accordingly obtained after substituting Eq. 11 in the decision function, i.e., Eq. 10. As a result, we have:

$$f(x) = \text{sign} \left(\sum_{i=1}^l v_i K(x, x_i) + b \right) \quad (12)$$

In Eq. 12, the parameter v_i ($0 < v_i < 8$) are used as a weighting factor to determine which of the input vectors actually are SVs. There are different kernel functions. Some of the commonly used kernel functions are polynomial, Gaussian Radial Basis Function (RBF), Exponential Gaussian Radial Basis Function (ERBF) and sigmoid kernels which are presented in Eq. 13-16, respectively (Scholkopf, 1998).

$$K(x, x') = (\gamma \langle x, x' \rangle + r)^d, \quad \gamma > 0, \quad (13)$$

$$K(x, x') = e^{-\gamma \|x - x'\|^2}, \quad \gamma > 0, \quad (14)$$

$$K(x, x') = e^{-\gamma \|x - x'\|}, \quad \gamma > 0, \quad (15)$$

$$K(x, x') = \tanh(\gamma \langle x, x' \rangle + r), \quad \gamma > 0, \quad r < 0, \quad (16)$$

Multi-class support vector machine: SVM has been originally designed for binary classification. There are two approaches for multi-class SVM. The first one is based on the construction and the combination of several binary classifiers while the other one can directly consider all data in one optimization formulation. In general, it is computationally more expensive to solve a multi-class problem than a binary problem with the same number of data. Therefore, methods based on constructing and combining several binary classifiers are preferable to methods based on solving a multi-class problem (Hsu and Lin, 2002).

The most popular algorithms, which are based on several binary classifiers, are: One-Against-All (OAA) (Vapnik, 1998), One-Against-One (OAO) (Platt *et al.*,

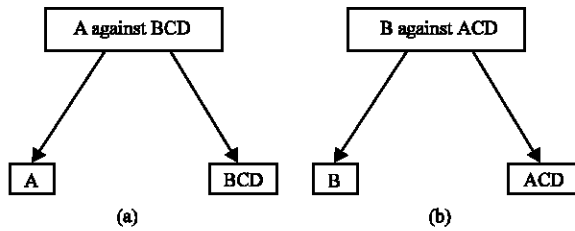


Fig. 3: OAA; Multi-class SVM classifiers (a) class A against others and (b) class B against others

1999), Direct-Acyclic-Graph (DAG) (Krebel, 1999) and Half-Against-Half (HAH) algorithms (Lei and Govindaraju, 2005).

One-against-all algorithm: The earliest used implementation for SVM multi-class classification is probably the One-Against-All (OAA) algorithm (Vapnik, 1998). It constructs k SVM models, where k is the number of classes. The i^{th} SVM is trained with all of the examples in the i^{th} class with positive labels and all other examples with negative labels. Figure 3 shows the multi-class SVM classifier by OAA for a four-class problem.

One-against-one algorithm: The One-Against-One (OAO) algorithm, constructs $k(k-1)/2$ classifiers where each one is trained on data from two classes (Platt *et al.*, 1999).

There are different methods for doing the future testing after that all $k(k-1)/2$ classifiers are constructed. Most popular method is Max Wins strategy. In this strategy, if

$$\text{sign}\left(\sum_{i=1}^j v_i K(x, x_i) + b\right)$$

says x is in the i^{th} class, then the vote for i^{th} class is added by one. Otherwise, the j^{th} class vote is added by one. Finally we predicate that x is in the class with the largest vote.

Direct-acyclic-graph algorithm: The third algorithm, Direct-Acyclic-Graph (DAG), has been proposed by Krebel (1999). Its training phase is the same as the OAO algorithm. However, in the testing phase, it uses a root binary DAG which has $k(k-1)/2$ internal nodes and k levels. Each node is a binary SVM of i^{th} and j^{th} classes. Given a test sample x , starting at root node, the binary decision function is evaluated. Then it moves to either left or right depending on the output value. Figure 4 shows the DAG-SVM structure of a four-class problem.

Half-against-half algorithm: The latest algorithm is Half-Against-Half (HAH), which is built via recursively

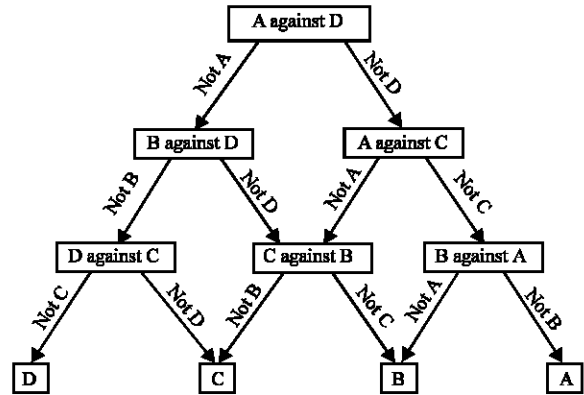


Fig. 4: DAG structure for a four-class classification

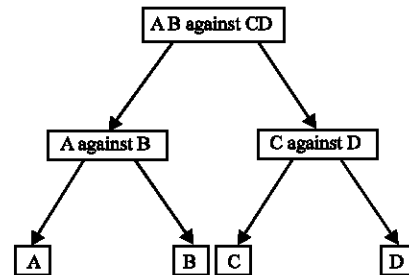


Fig. 5: HAH classifier structure for a four-class problem

dividing the training data set of k classes into two subsets of classes. The structure of HAH is the same as a decision tree which at each node it has a binary SVM classifier that determines the class of a testing sample.

The trained HAH classifier model consists of at most k binary SVM. If k is an even number it needs k binary SVM and if k is odd it needs $(k-1)$ binary SVM. For each classification testing, HAH requires at most \log_2^k binary SVM evaluations. Figure 5 shows the HAH classification structure of a four-class problem.

Both theoretical estimation and experimental results show that HAH has advantages over OAA, OAO and DAG based algorithms in the testing speed, the size of the classifier model and accuracy (Lei and Govindaraju, 2005). Considering these merits the HAH algorithm has been used in this study.

PROPOSED ALGORITHM

The application of machine learning methods is based on our knowledge about the behavior of the system, obtained from a large number of off-line simulations. These simulations define two data sets; the training set and the testing set. The training data set is used to derive the security evaluation structure and the

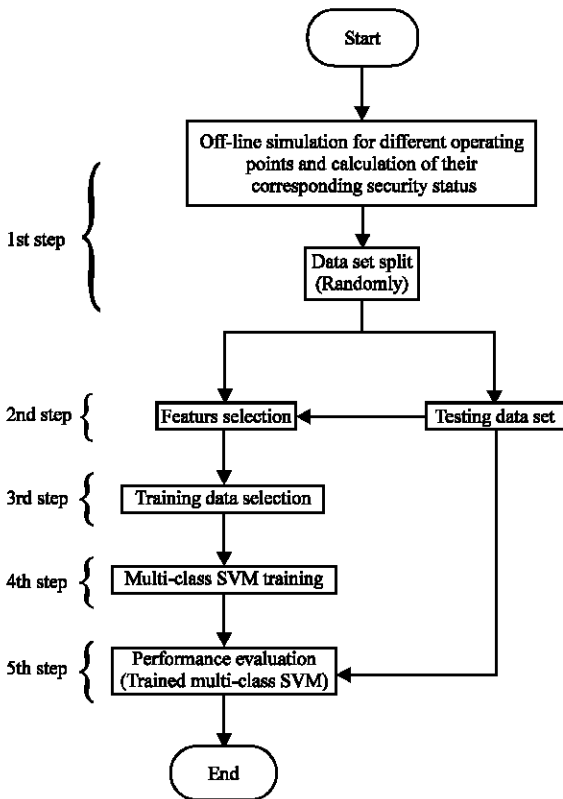


Fig. 6: Flowchart of proposed algorithm

testing data set is used for testing the developed structures. The training data set should cover the entire demand space including hourly, daily and weekly variations of the system demand. Considering these points, the flowchart, shown in Fig. 6, has been proposed for static security assessment in this research.

Creation of knowledge base: The First step in the proposed algorithm shown in Fig. 6, is off-line simulations for different operating points of the system. In this step, the static security index will be calculated based on load flow results. These simulations must be carried out for different operating points at any contingency in the set of possible contingency set. In this study, the following Performance Index (PI) has been chosen to quantify the severity of a contingency and to classify the system security status. PI has been defined as follows (Kirschen, 2002):

$$PI = \sum_{l=1}^{N_l} \frac{w_l}{2n} \left[\frac{S_l}{S_l^{MAX}} \right]^{2n} \quad (17)$$

where, S_l is the apparent power flow (MVA) of the line l , S_l^{MAX} is the maximum value of S_l , N_l is the number of lines

in the system, n is the specified exponent of penalty function and w_l is the real non-negative weighting coefficient (may be used to reflect the importance of some lines). In this study, $n = 2$ and $w_l = 1$ (for all lines) have been selected.

In order to classify the system static security status, four different security levels have been considered; normal (class A), alert (class B), emergency 1(class C, correctable contingencies) and emergency 2 (class D, non-correctable contingencies or system splitting status). Based on the calculated PI index, the system security status can be determined.

Feature selection: Considering the following points, it can be said that one of the main aspects of the successful use of machine learning methods is the feature selection:

- Removing unnecessary or bad features can improve the accuracy of most machine learning algorithms
- The feature selection does provide insights into the quality and productivity of each feature

Several feature selection algorithms, such as Pearson’s correlation coefficient, Fisher-like criterion and Relief family, have been design for the feature selection (Guyon and Elisseeff, 2003; Jensen *et al.*, 2001; Yang and Honavar, 1998).

In this study, Fisher-like criterion (F-score) has been used for feature selection considering its simplicity and accuracy. The Fisher score for a multi-class problem is defined as follows:

$$F(r) = \frac{\sum_{i=1}^C n_i (\mu_i - \mu)^2}{\sum_{i=1}^C n_i \sigma_i^2} \quad (18)$$

where, n_i is the number of data points in class i , μ_i and σ_i is the mean and variance of class i corresponding to the r^{th} feature, respectively, C is the number of classes and μ is the mean value of the whole data set of the r^{th} feature.

Training data selection: The next step in Fig. 6, is the training data selection. The data selection can be used for fast multi-class SVM training. The main point is the selection of the important data for SVM training. The SVM decision function depends only on a small subset of training data, called SVs. Therefore if one knows in advance which patterns correspond to SVs, the same solution can be obtained by solving a much smaller problem that involves only the SVs. According to Huang *et al.* (2007) k-means clustering has been used for

selecting patterns from the training set. According to Yeung *et al.* (2007). Mahalanobis distance has been used to identify SVs and boundary points. According to Wang *et al.* (2007) two data selecting algorithms have been proposed. In the first algorithm, the training data selection is based on a statistical confirm measure. In the second algorithm, the minimum distance from a training example to the training examples of a different class has been used as a criterion to select the important data.

The comparison of different data selection schemes with the scheme based on the decision SVM outputs, shows that the confidence measure provides a criterion for training data selection that is almost as good as the optimal criterion based on the desired SVM outputs. Also, it has been shown that the random sampling performance is not good in comparison with the confidence measure-based algorithm at low reduction rates (Wang *et al.*, 2007). In the confidence measure-based algorithm, a sphere centered at each training example x_i , has been considered (Wang *et al.*, 2007). This sphere should be as large as possible without covering a training example of a different class. The number of training examples that falls inside this sphere has been denoted by $N(x_i)$. Obviously, the larger, the number $N(x_i)$, the more training examples (of the same class as x_i) will be scattered around x_i , the less likely x_i will be close to the decision boundary and the less likely x_i will be a SV. Hence, this number can be used as a criterion to decide which training examples should belong to the reduced training set.

Based on this algorithm, $N(x_i)$ has been computed for each training point and then the training data have been sorted according to the corresponding value of $N(x_i)$. Finally, a subset of data with the smallest numbers $N(x_i)$, has been chosen as the reduced training set. In this algorithm, $N(x_i)$ should be computed for each training point.

Performance evaluation: The performance of the proposed method has been evaluated by using Sensitivity index of class i , (SNc_i), Precision Rate index of class i , (PRc_i), for each class and Total Accuracy index (TA), as follows:

$$SNc_i = \frac{Tc_i}{Tc_i + Fc_i} \tag{19}$$

$$PRc_i = \frac{Tc_i}{Tc_i + Fc_{oi}} \tag{20}$$

$$TA = \frac{\sum_{i=1}^c Tc_i}{\sum_{i=1}^c Tc_i + \sum_{i=1}^c Fc_i} \tag{21}$$

Where:

C_i = Index of classes

Tc_i = No. of cases which are actually in class C_i and classified correctly by the trained SVM

Fc_i = No. of cases which are actually in class C_i and the trained SVM classified in other classes by the trained SVM

Fc_{oi} = No. of the cases of other classes classified in class C_i , by the trained SVM

C = No. of classes

In the problem presented in this paper, the numbers of cases in different classes are not equal; therefore TA is not enough as a total performance evaluation index. For example, assume a test data set contains 40 class A, 30 class B, 20 class C and 10 class D cases. If all cases in class D have not been predicated correctly and all of the other cases have been predicated correctly, then TA is equal to 90%. It seems that TA doesn't have enough information and a new evolution index for multi-class classification should be presented. In this paper, this new index is Balanced Error Rate (BER) as follows:

$$BER = \frac{1}{C} \left(\sum_{i=1}^c \frac{Fc_i}{Tc_i + Fc_i} \right) \tag{22}$$

CASE STUDY

In order to demonstrate the effectiveness of proposed technique, results of the application of the method to New England 39-bus power system have been presented. A contingency list consists of 8 single line outages has been considered. The load and generation of buses have been randomly changed between 30% up to 130% of their base case, resulting in 2125 different operating points. Among these operating points, 1500 points have been randomly chosen as the training data set and the remaining 625 cases have been used as the testing data set. In this study, the base case load is 6097 MW and system load has been changed from 1750 to 8250 MW.

All contingencies have been simulated one by one and corresponding PI have been computed for each operating point. Based on calculated PI, the security status of each operating point for each contingency has been determined. The number of each class members in the training and testing data set is given in Table 1.

All contingencies have been simulated one by one and corresponding PI has been computed for each operating point. Based on calculated PI, the security status of each operating point for each contingency has

Table 1: No. of each class members

Class	Training data set	Testing data set
A	9268	430
B	1795	470
C	630	152
D	307	72
Total	12000	500

Table 2: Results of fisher criteria feature selection

Order	Feature description	Value
1	Reactive power of generator G9	0.382
2	Apparent power of generator G10	0.382
3	Apparent power of generator G9	0.372
4	Reactive power of generator G3	0.372
5	Apparent power of generator G5	0.367
6	Voltage at bus 32	0.365
7	Apparent power of generator G3	0.356
8	Active power of generator G9	0.345

been determined. The number of each class members in the training data set and testing data set is given in Table 1.

Feature selection: Line flows, currents, bus voltages and system other parameters can be selected as a system features. In this study, 46 active power flows on branches, 46 reactive power flows on branches, 39 bus voltages, 39 active power injections on buses, 39 reactive power injections on buses and a contingency identification number (an integer number among 1 to 8) have been considered as the input of the proposed feature selection algorithm.

The results of the first eight dominate features have been shown in Table 2.

Training data selection: Here, $N(x_i)$ has been computed for. The points with $N(x_i) \leq 2$ have been selected for reduced training data set. Based on these assumptions, 4823 points have been selected for multi-class SVM training process.

Parameter tuning: The first step in the training of SVM classifier is to choose the kernel and its parameters. Four kernel functions have been mentioned in the section II. For the polynomial and sigmoid kernel functions, $r = 0$ and $\gamma = 1/k$ have been chosen. Where k is the number of features (as a result, $\gamma = 1/8$). A grid search has been performed over the values of the penalty parameter, P for each kernel, the parameter d for polynomial kernel and parameter γ for RBF and ERBF kernels. Figure 7 and 8 give an example for the influence of penalty parameter P upon the prediction performance on training set. These search results show that the best result has been obtained by the ERBF kernel function. The parameters for each binary SVM have been presented in Table 3.

Table 3: ERBF kernel function parameters for each binary SVM

Binary SVM	Penalty parameter (P)	γ
AB against CD	60	5.00
A against B	60	5.00
C against D	30	5.25

Table 4: No. of SVs for each binary SVM

Binary SVM	No. of SVs
AB against CD	2789
A against B	3579
C against D	805

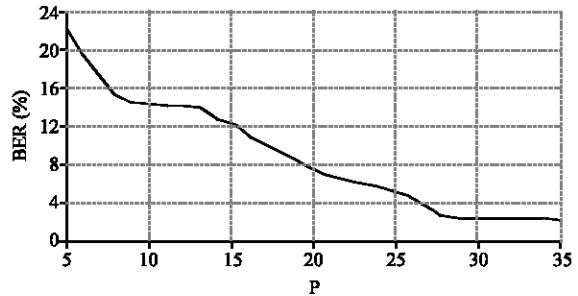


Fig. 7: Influence of penalty parameter P upon the BER

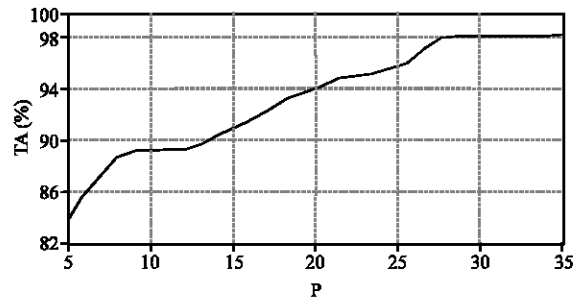


Fig. 8: Influence of penalty parameter P upon the TA

SVM training: Three binary SVM according to the HAH classifier structure have been trained. The number of the SVs for each binary SVM has been presented in Table 4.

Performance evaluation: The SVM prediction results for the testing data set and all training data set have been shown in Table 5.

TA and BER for the testing data set are 98.71 and 0.63% and for the all training data set 96.51 and 2.14%, respectively.

In order to show the effectiveness of the training data selection algorithm, the multi-class SVM has been trained by using all of the training data set (1500 operating points). In this case, the training time has been approximately increased by ten times. Table 6 compares the number of the SVs for reduced training data set and all training data set. Table 6 shows that the proposed data selection algorithm can remove a significant amount of data while the trained binary SVMs are same.

Table 5: Multi-class SVM prediction results

Class	Sensitivity (SN) (%)		Precision rate (PR) (%)	
	Train data set	Test data set	Train data set	Test data set
A	97.53	96.91	99.43	98.32
B	97.43	95.53	96.22	97.18
C	95.68	94.60	96.21	95.11
D	96.32	96.68	97.14	96.32

Table 6: No. of SVs for reduced training data set and all training data set

Binary SVM	Reduced training data	All training data set
AB against CD	2789	2791
A against B	3579	3620
C against D	805	812

CONCLUSION

A multi-class SVM based static security assessment algorithm for power system has been proposed. The proposed algorithm has been applied to New England 39-bus power system. It is shown that the proposed algorithm is less sensitive to the size of input features and it does not need to extract signal features. This property is very effective on power system monitoring and its measurements for the system security assessment.

Also, it needs only a small number of operating points to train the classifier (only support vectors). This capability is very useful to reduce the training data while maintaining the generalization performance of the resulting SVM classifiers and therefore the training time is reduced significantly.

REFERENCES

Guyon, I. and A. Elisseeff, 2003. An introduction to variable and feature selection. *J. Mach. Learning Res.*, 3: 1157-1182.

Hatzigargyriou, N., G.C. Contaxis and N.C. Sideris, 1994. A decision tree method for on-line steady state security assessment. *IEEE Trans. Power Syst.*, 9: 1052-1061.

Hsu, C.W. and C.J. Lin, 2002. A comparison of methods for multi-class support vector machines. *IEEE Trans. Neural Networks*, 13: 415-425.

Huanga, J.J., G.H. Tzeng and C.S. Ong, 2007. Marketing segmentation using support vector clustering. *Expert Syst. Appl.*, 32: 313-317.

Jensen, C.A., M.A. El-Sharkawi and R.J. Marks, 2001. Power system security assessment using neural networks: Feature selection using fisher discrimination. *IEEE Trans. Power Syst.*, 16: 757-763.

Kim, H. and C. Singh, 2005. Power system probabilistic security assessment using bayes classifier. *Elect. Power Syst. Res.*, 74: 157-165.

Kirschen, D.S., 2002. Power system security. *Power Eng.*, 16: 241-248.

Krebel, U., 1999. *Pairwise Classification and Support Vector Machines*. 1st Edn. MIT Press, Cambridge .

Lei, H.S. and V. Govindaraju, 2005. Half-against-half multi-class support vector machines. *Proceeding of Multiple Classifier Systems'2005*, 13-15 June, California, USA., pp: 156-164.

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

Neibur, D. and A. Germond, 1991. Power system static security assessment using kohonen neural network classifier. *IEEE Trans. Power Syst.*, 7: 865-872.

Platt, J., 1998. How to implement SVMs. *IEEE Intell. Syst.*, 13: 26-28.

Platt, J., N. Cristianini and J. Shawe-Taylor, 1999. Large margin DAGs for multiclass classification. In: *Proceeding of Neural Information Processing Systems*, 12: 547-553.

Scholkopf, B., 1998. SVMs-a practical consequence of learning theory. *IEEE Intell. Syst.*, 13: 18-19.

Shanti, K.S., 2008. Artificial neural network using pattern recognition for security assessment and analysis. *Neurocomputing*, 71: 983-998.

Vapnik, V.N., 1998. *Statistical Learning Theory*. 1st Edn. John Wiley and Sons, New York .

Vapnik, V.N., 1999. An overview of statistical learning theory. *IEEE Trans. Neural Networks*, 10: 988-1000.

Wahab, N.I.A., A. Mohamed and A. Hussain, 2007. Transient stability assessment of a power system using PNN and LS-SVM methods. *J. Applied Sci.*, 7: 3208-3216.

Wang, J., P. Neskovic and L.N. Cooper, 2007. Selecting data for fast support vector machine training. *Stud. Comput. Intell.*, 35: 61-84.

Wehenkel, L.A., 1998. *Automatic learning techniques in power systems*. 1st Edn. Kluwer Academic, Boston.

Yang, J. and V. Honavar, 1998. Feature subset selection using a genetic algorithm. *IEEE Intell. Syst.*, 13: 44-49.

Yeung, D.S., D. Wang and E.C. Tsang, 2007. Weighted mahalanobis distance kernels for support vector machines. *IEEE Trans. Neural Networks*, 18: 1453-1462.

Zhou, Q., J. Davidson and A.A. Fouad, 1994. Application of artificial neural networks in power system security and vulnerability assessment. *IEEE Trans. Power Syst.*, 9: 525-532.