Static Security Evaluation in Power Systems using Multi-Class SVM with Different Parameter Selection Methods

S. Kalyani, Member, *IEEE* and K. S. Swarup, Senior Member, *IEEE*

Abstract— Security evaluation is a major concern in real time operation of electric power systems. Traditional method of security evaluation performed by continuous load flow analysis involves long computer time and generates voluminous results. This paper presents a practical and feasible *Support Vector Machine Based Pattern Classification* (SVMBPC) approach for static security evaluation in power systems. The proposed approach classifies the security status of any given operating condition in one of the four classes - Secure, Critically Secure, Insecure and Highly Insecure based on the computation of a numeric value called security index. The feature selection stage uses a simple and straightforward forward sequential method to select the best feature set from a large set of variables. The static security classifier is designed by a multi-class SVM with different parameter tuning methods. The proposed approach is implemented in New England 39 bus and IEEE 118 bus systems and the results are validated.

***Index Terms*—Parameter Selection, Pattern Classifier, Static Security, Support Vector Machine**

I. Introduction

Security evaluation is the analysis performed to determine whether, and to what extent, the system is reasonably safe from serious interference to its operation. Occurrence of certain severe perturbations may move the system to an undesirable emergency state, if the system security status is not well defined beforehand. Hence, effective control of modern power systems necessitates a quick security evaluation of their operating states. Power System Security is defined as the system's ability to withstand unexpected failures and to remain secure without serious consequences to any pre-selected list of credible contingencies [1].

Security analysis may be broadly classified as Static Security Evaluation (SSE) and Transient Security Evaluation (TSE). The traditional method used for security analysis involves solving full AC load flow and rotor dynamics of machines for each contingency scenario. This procedure is highly time consuming and generates voluminous results, making it inadequate for real time applications [2], [3]. A method is, therefore, required to evaluate and classify system security status using real time data in minimum time and with maximum accuracy.

In recent years, use of many Artificial Intelligence (AI) techniques and expert systems like fuzzy set theory has been proposed for security evaluation problem, overcoming the pitfalls of traditional method. Literatures have reported the use of Artificial Neural Network techniques [4], [5], fuzzy logic combined with neural network [6], genetic based neural network [7] for static security evaluation process. The performance of all these existing techniques are highly problem dependent and hence its suitability cannot be generalized. Nowadays, pattern classification is gaining more importance in solving many power system problems. In this approach, main bulk of work is done off-line to generate sufficient dataset. The classification function, designed based on the train set, helps to access the system security level in a short period of time.

This paper addresses security evaluation as a pattern classification problem with the classifier function designed by Support Vector Machine (SVM). SVM is a new and promising tool for learning separating functions in PR system with the capability of handling non-linear separability. The SVM classifier is designed for multi-classification based on the calculation of a term called *Static Security Index* (SSI), for each specified contingency. In this paper, four class logic is used for the definition of system security viz., secure, critically secure, insecure, highly insecure. An operator likes to know exactly the severity level of disturbances for a given system operating condition. On-line security evaluation allows the operator to know the security status and helps to determine the corrective actions. This paper also addresses different heuristic optimization techniques like Particle Swarm Optimization [8], Real Coded Genetic Algorithm [9] and Differential Evolution [10] used in the selection of SVM parameters globally. The classification approach is implemented in New England (NE) 39 bus system and IEEE 118 bus system and the results are compared.

II. Power System Security Evaluation

The term 'Security' as defined by NERC (1997) is the ability of the electric systems to withstand sudden disturbances such as electric short-circuits or unanticipated loss of system element [11]. Security Evaluation is the process of determining, whether and to what extent, a system is 'reasonably' safe from serious interference to its operation [12]. A set of most probable contingencies is first specified for security evaluation. This set may include outage of a line/generator, sudden increase in load, three phase fault in the system, etc.

Manuscript received March 30, 2010.

S. Kalyani was with Indian Institute of Technology Madras. She is now with Department of EEE, K.L.N. College of Engineering, Pottapalayam – 630611, Sivagangai District, Tamilnadu, India. (Phone: +91-09443726963; Fax: +91-452-2090070; e-mail: kal_yani_79@yahoo.co.in).

K.S. Swarup is with Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai – 600036, Tamilnadu, India (e-mail: swarup@ee.iitm.ac.in).

A. Static Security Evaluation

Static security is the ability of the system to reach a steady state within the specified secure region (defined by bounding limits) following a contingency [13]. Limit violation of any component may lead to cascading of outages and hence severe ‘blackout’. The violations of thermal limits of transmission lines and bus voltage limits are the main concerns for static security analysis. In conventional practice, static security evaluation is performed by analytically modelling the network and solving the algebraic load flow equations repeatedly for all prescribed outages, one at a time. This traditional approach is not entirely satisfactory because a huge number of simulations need to be carried out.

A given system operating condition is said to be ‘static secure’, if the bus voltage magnitudes and real power generation of generator buses are well within their limits, without any occurrence of line overloads. In this paper, we define a term called *Static Security Index (SSI)* for evaluating static security level for a given system operating condition and a specified contingency. The SSI is defined by calculating the Line Overload Index (LOI) and Voltage Deviation Index (VDI) as given by (1) and (2) respectively.

$$LOI_{km} = \begin{cases} \dfrac{S_{km} - MVA_{km}}{S_{km}} \times 100 & if\ S_{km} > MVA_{km} \\ 0 & if\ S_{km} \le MVA_{km} \end{cases} \tag{1}$$

$$VDI_k = \begin{cases} \dfrac{|V_k^{\min}| - |V_k|}{|V_k^{\min}|} \times 100 & if\ |V_k| < |V_k^{\min}| \\ 0 & if\ |V_k^{\min}| \le |V_k| \le |V_k^{\max}| \\ \dfrac{|V_k| - |V_k^{\max}|}{|V_k^{\max}|} \times 100 & if\ |V_k| > |V_k^{\max}| \end{cases} \tag{2}$$

$$SSI = \frac{W_1 \sum_{i=1}^{N_L} LOI_i + W_2 \sum_{i=1}^{N_B} VDI_i}{N_L + N_B} \tag{3}$$

where S_{km} and MVA_{km} represents the Mega Volt-Ampere (MVA) flow and MVA limit of branch k-m, $|V_k^{min}|$, $|V_k^{max}|$ and $|V_k|$ the minimum voltage limit, maximum voltage limit and bus voltage magnitude of kth bus respectively, N_L and N_B being number of lines and buses respectively.

III. Design of Static Security Classifier

Classification of power system state is the primary stage in security monitoring process of real power system networks. A suitable pattern classifier system is developed for multi-class static security assessment problem addressed herein. The pattern classification approach is applied to reduce on-line computational requirements at the expense of an extensive off-line simulation. The design of pattern recognition system, thus, consists of an off-line simulation process called data generation followed by feature selection and classifier design. The sequence of steps carried out in designing the multi-class static security classifier through off-line process is shown in detail in Fig. 1.

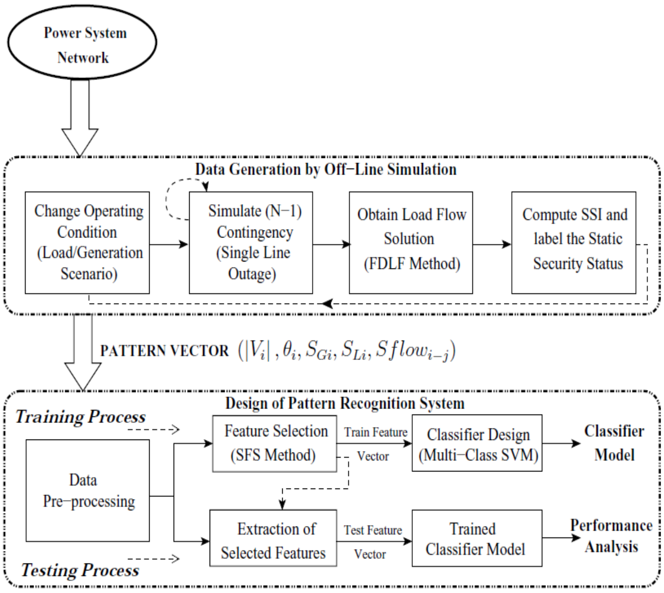
Power System Network
Data Generation by Off-Line Simulation
Change Operating Condition (Load/Generation Scenario)
Simulate (N-1) Contingency (Single Line Outage)
Obtain Load Flow Solution (FDLF Method)
Compute SSI and label the Static Security Status
PATTERN VECTOR $(|V_i|, \theta_i, S_{Gi}, S_{Li}, Sflow_{i-j})$
Design of Pattern Recognition System
Training Process
Data Pre-processing
Feature Selection (SFS Method)
Train Feature Vector
Classifier Design (Multi-Class SVM)
Classifier Model
Extraction of Selected Features
Test Feature Vector
Trained Classifier Model
Performance Analysis
Testing Process

Fig. 1. Design process of static security classifier for multi-classification

A. Data Generation

The success of any pattern directed inference system relies on a good training set. This set must adequately represent the entire range of power system operating states [14]. The patterns can be generated either from real time measurements or synthesized from off-line simulations. In this paper, a large number of characteristic operating points are generated by offline simulations as shown in the upper part of Fig. 1. Different operating conditions are considered by varying the system load and generation from 50% to 200% of their base case values. The variation in generation is bounded to their min-max generation limits. For each operating scenario considered, N-1 contingency case (single line outage) is simulated and load flow solution by Fast Decoupled Load Flow (FDLF) method is obtained. Each operating condition is termed as a *p*attern [3]. Each pattern is characterized by a number of attributes like load level, bus voltages, power generation, forming the components of a vector called pattern vector *XSSA*, as listed in (4).

$$X_{SSA} = \{ |V|_i, \delta_i, S_{Gi}, S_{Li}, Sflow_{km} \} \tag{4}$$

where,

$|V_i|$ voltage magnitude at *i*th bus

δ_i voltage angle at *i*th bus

S_{Gi} complex power generation at *i*th generator bus

S_{Li} complex power load at *i*th load bus

$Sflow_{km}$ MVA power flow in branch k-m

Evaluating the Static Security Index (SSI) as given by (3), each pattern is labeled as belonging to one of the four classes as shown in Table I. In calculation of the value of SSI, weighting factors for LOI and VDI are assumed as $W_1 = 3$ and $W_2 = 2$ respectively. These weighting factors are fixed based on the order of priority in requirement of system security. SSI is a percentage measure of system security level, taking value in the range of 0 to 100.

TABLE I. TABLE I. CLASS LABELS FOR STATIC SECURITY ANALYSIS

Static Security Index (SSI)	Class Category / Label
$SSI = 0$	Class A : Secure
$SSI > 0$ & $SSI <= 5$	Class B : Critically Secure
$SSI > 5$ & $SSI <= 15$	Class C : Insecure
$SSI > 15$	Class D : Highly Insecure

B. Feature Selection

The number of variables in the pattern vector is normally very large. Therefore, it becomes necessary to determine relatively small number of variables distinctive for classification [15]. Feature Selection is the process of selecting a small optimal set of attributes called *features*, which will give more useful information for classification. The selected features form the components of a vector called feature vector Z. In this work, a simple and quick procedure called *Sequential Forward Selection (SFS)*, wrapper method, is used. The SFS method starts with an empty feature set and iteratively selects one feature at a time, until no further decrease in criterion function is achieved. The criterion function, J, is the minimization of misclassification rate.

C. Classifier Design

After selecting the desired features, the next step is to design a decision function or classifier. The classifier represents the boundary between separating classes. The classifier attempts to assign every data point in the entire feature space to one of the possible classes. The design of the classifier is based on the design (training) set of selected features. The main requirement of any classifier model is that it should provide high classification accuracy and less misclassification rate, when evaluated for unlabeled (unseen) test set samples. Support Vector Machine, a popularly used machine learning tool, has been applied for efficient pattern classifier design.

1) Multi-Class SVM Classifier: The security evaluation problem is focused as a multi-classification problem in this paper. Direct solution of multi-class problem using single SVM formulation is not possible. A better approach is to use a combination of several binary SVM classifiers to solve multi-class problems. Popular methods available are: (i) One-Versus-All (OVA) method and (ii) One-Versus-One (OVO) method. The former method constructs K SVM models, with class i against all other classes, K being number of distinct classes of the problem. The OVA method, although simple, is computationally expensive and not commonly preferred. In this paper, we use the latter method for designing the multi-class static security classifier. The OVO method also called pair-wise SVM, determines the decision functions for all combinations of class pairs. This method constructs K(K – 1)/2 binary classifiers, each being trained from data belonging to the corresponding two classes only, considerably reducing number of train data. The classification in OVO method is performed by a Max-Wins Voting (MWV) strategy. After each of the binary classifiers make its vote, the decision function assigns an instance x to a class having largest number of votes [16]. In case, tie occurs with two classes having identical votes, the one with smallest index is selected.

2) Steps in Design of SVM Classifier

1. Data Scaling or Preprocessing

The input features in train and test sets needs to be scaled properly before applying SVM. Scaling prevents the domination of any feature over the other because of higher numeric values involved and also avoids numerical difficulties during calculation. We recommend each attribute to be linearly scaled to the range of [0, 1].

2. Design of SVM Model

Choice of Kernel

The Radial Basis Function (RBF) kernel is chosen as a first choice because of its wide known accuracy. Further, it is capable of handling non-linear relation existing between the class labels and input attributes. The second reason is that RBF kernel, unlike other kernels, has only one kernel parameter, thereby reducing the complexity of the model.

Adjusting the Kernel Parameters

There are two parameters associated with SVM model designed with RBF kernel - Penalty parameter, C and RBF Kernel parameter, γ. The goal is to identify optimal (C, γ) for the classifier to accurately predict the unknown data (test data). This can be achieved by different techniques, description of which follows in the next subsection.

3. Training and Testing the SVM Model

After designing the SVM model with the chosen kernel and optimal parameters, it is trained with the scaled input output train set samples. Once the performance of the SVM classifier is found satisfactory in training phase, the model is validated with test samples to access its overall performance.

D. Selection of SVM Parameters

1) *Grid Search (GS):* Grid search is the most common and simplest method. Grid search method adopts v-fold Cross Validation technique. In a v-fold cross validation, we divide the whole training set into v subsets of equal size. Sequentially one subset is tested using the SVM classifier trained on the remaining (v-1) subsets. Thus, each instance of the train set is predicted once and the cross-validation accuracy is the percentage of data samples that are correctly classified [17]. In this work, Grid Search using 5-fold cross validation is used.

2) *Particle Swarm Optimization (PSO):* Particle Swarm Optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995. In PSO, each single solution is called as particle. To discover the optimal solution, each particle is updated by two ‘best’ values in each iteration. After finding these two best values, each particle changes its velocity and position according to the cognition part (P_{best}) and social part (G_{best}). The update equations for particle’s velocity and position are given by (5) and (6).

$$V_{id}^{k+1} = w \times V_{id}^{k} + c_1\, rand_1\, (Pbest_{id}^{k} - X_{id}^{k}) + c_2\, rand_1\, (Gbest_{d}^{k} - X_{id}^{k}) \tag{5}$$

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k} \tag{6}$$

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{Max.\ Iterations} xCurrent\ \ Iteration \tag{7}$$

where w is the inertia weight calculated by (7), V_{id} is the particle velocity, X_{id} is the current particle position (solution), *rand*1 is a random number between (0,1), c_1 and c_2 indicates cognition and social learning factors respectively.

PSO Algorithm for SVM Parameter Selection

Step 1) Randomly initialize a population of particles with positions X_{id} (C,γ) and velocities V_{id} of the i^{th} particle in *dth* dimension.

Step 2) Set PSO parameters, $C_1=C_2= 2$, $w_{max}=0.9, w_{min}=0.5$.

Step 3) Evaluate the fitness of each particle in the population. The SVM model is built with each particle's position (SVM parameters) and trained with 90% of samples in train set feature vector. This SVM model is validated using the remaining 10% samples and misclassification (error) rate as given by (8), called fitness, is computed for each particle.

$$Fitness = \frac{No.\ of\ samples\ misclassified}{Total\ No. of\ Samples} x100 \tag{8}$$

Step 4) Compare the current position with particle's previous best experience, *Pbest*, in terms of fitness value and hence update Pbest for each particle in the population.

Step 5) After updating the *Pbest*, choose the best value (with less misclassification rate) among all the particles in *Pbest* and call it as Global best, *Gbest*.

Step 6) Update the particle's velocity using (5) and clamp to its minimum (V_{min}) and maximum (V_{max}) limit, whichever violates.

Step 7) Move to the next position of the particle using (6) bounded to its upper and lower limits.

Step 8) Stop the algorithm and print the optimal solution (Final *Gbest*) if termination criterion, maximum iterations, is reached; otherwise loop to Step 3.

3) *Real Coded Genetic Algorithm (RCGA):* Genetic Algorithm (GA) belongs to the class of randomized heuristic search techniques. GA is a general purpose search procedure that uses the principles inspired by natural genetic populations to evolve solution. The traditional GA uses binary representation of strings, which is not preferred in the continuous search space domain. The problem of optimal selection of SVM parameters is an optimization problem in continuous domain. Real Coded Genetic algorithm (RCGA) gives a straightforward representation of chromosomes by directly coding all variables. The chromosome X is represented as X={p_1 , p_2}, where p_1 denotes penalty the parameter C and p_2 the kernel parameter γ.

Unlike traditional binary coded GA, decision variables can be directly used to compute the cross validation accuracy called fitness, same as that of the previous algorithm. The RCGA uses selection, crossover and mutation operators to reproduce offspring for the existing population [9]. The RCGA-SVM model incorporates Roulette Wheel selection to decide chromosomes for the next generation. The selected chromosomes are placed in a matting pool for crossover and mutation operations. The crossover operation enhances the global search property of GA and mutation operation prevents the permanent loss of any gene value. In this work, Arithmetic Crossover and Polynomial Mutation, described by [18], has been used to perform crossover and mutation respectively. The detailed procedure of RCGA applied for the problem of SVM parameter selection is shown in the form of a flowchart in Fig. 2.

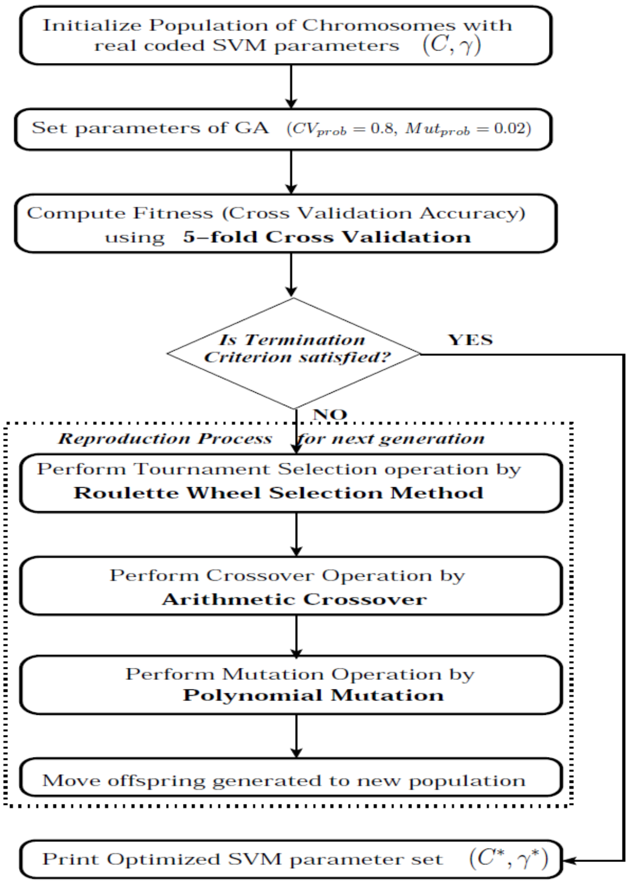
Initialize Population of Chromosomes with real coded SVM parameters (C, γ)
Set parameters of GA $(CV_{prob} = 0.8, Mut_{prob} = 0.02)$
Compute Fitness (Cross Validation Accuracy) using 5−fold Cross Validation
Is Termination Criterion satisfied?
YES
NO
Reproduction Process for next generation
Perform Tournament Selection operation by Roulette Wheel Selection Method
Perform Crossover Operation by Arithmetic Crossover
Perform Mutation Operation by Polynomial Mutation
Move offspring generated to new population
Print Optimized SVM parameter set (C^*, γ^*)

Fig. 2. RCGA algorithm for SVM paramter selection

4) *Differential Evolution (DE):* Differential Evolution, one of the evolutionary optimization technique, was introduced by R. Storn and K. Price in 1995. In this paper, we have used a commonly used strategy denoted as 'DE/rand/1/bin'. In this representation, 'rand' indicates a random mutant vector to be chosen; '1' the number of difference vectors and 'bin' denotes the crossover scheme.

DE Algorithm for SVM Parameter Selection

Step 1) Randomly initialize a population of individuals X_{id} denoting the i^{th} individual in d dimension.

Step 2) Specify the DE parameters; difference vector scale factor F=0.05, minimum and maximum crossover probability CR_{min}= 0.1 and CR_{max}= 0.9.

Step 3) Evaluate the fitness value of each individual in the population. The fitness value is error rate, given by (8), obtained by validating the trained SVM model.

Step 4) Generate mutant vector for each individual x_i according to (9)

$$v_i = x_{s1} + F \times (x_{s2} - x_{s3}) \tag{9}$$

The indices *s*1, *s*2 and *s*3 are randomly chosen from population size. It is important to ensure that these indices are different from each other and also from the running index *i*.

Step 5) Perform crossover by combining mutant vector *v* with target vector *x* using (10).

$$u_{ij} = \begin{cases} v_{ij} & rand(j) \le CR \; or \; j = randn(i) \\ x_{ij} & rand(j) \le CR \; or \; j \ne randn(i) \end{cases} \tag{10}$$

where rand(j) ∈ [0, 1] is the jth evaluation of a uniform random generator number. Randn (i) ∈{1, 2...D} is a randomly chosen index ensuring that ui gets atleast one element form mutant vector, v_i. CR is the time-varying crossover probability constant determined using (11).

$$CR = CR_{\min} + \frac{(CR_{\max} - CR_{\min})}{Max.Iterations} x \; Iter. \tag{11}$$

Step 6) Perform selection operations based on fitness value and generate new population. If the trial vector u_i yields a better fitness, then x_i is replaced by u_i, else x_i is retained at its old value.

Step 7) If stopping criterion (max. iterations) is reached, stop and print the optimized parameter set (C*, γ*); else increase iteration count and loop to Step 3.

IV. Results and Discussion

The proposed SVM based Pattern Classification approach for the static security evaluation problem is implemented in New England 39 bus and IEEE 118 bus power system networks. The security limit for bus voltage magnitude is assumed in the range of 0.90pu to 1.10pu for all test case systems. MVA limit of system branches is assumed as 130% of base case values. The results of data generation and feature selection are shown in Table II. As seen from Table II, the number of input features for classifier design is reduced many folds, making the application of pattern analysis to security evaluation more attractive. This is clearly evident from the figure of dimensionality reduction, which gives a percentage measure of selected feature variables with respect to total number of pattern attributes.

TABLE II. RESULTS OF DATA GENERATION AND FEATURE SELECTION

	NE 39 Bus	**IEEE 118 Bus**
Operating Scenarios	548	3537
Class A : Secure (S) Class B : Critically Secure (CS) Class C : Insecure (I) Class D : Highly Insecure (HI)	87 275 158 28	174 2391 344 628
No. of Pattern Variables No. of Features selected	153 17	568 52
Dimensionally Reduction	**11.111%**	**9.515%**

The SVM parameters are selected by different evolutionary optimization techniques as described in the previous section. The results of different parameter selection methods adopted for the design of SVM model are shown in Table III. All evolutionary algorithms (PSO, RCGA, DE) described for SVM parameter selection use a population size of 40 and search space boundary of $C = [2^{-5}, 2^{15}]$, $\gamma = [2^{-15}, 2^{5}]$ in the simulation. About 50 independent trials are performed for each parameter selection algorithm. The mean and standard deviation obtained from the global solution of these trials and the best parameter values obtained for the trial yielding best fitness value are pictured in Table II for NE 39 Bus and IEEE 118 Bus systems. It can be observed from Table III that DE algorithm gives a better optimal solution for SVM parameters with less standard deviation, especially in large size systems.

TABLE III. RESULTS OF SVM PARAMETERS BY DIFFERENT METHODS

Parameter Selection Method →			GS	PSO	RCGA	DE
NE 39 Bus	$\log_2 C$	Best Trial	5.00	14.70	14.37	13.97
		Mean (μ)	-	13.83	14.06	13.89
		Std. Dev (σ)	-	1.07	0.68	1.08
	$\log_2 \gamma$	Best Trial	1.00	3.60	3.65	2.46
		Mean (μ)	-	3.13	2.99	3.23
		Std. Dev (σ)	-	0.64	2.05	0.59
IEEE 118 Bus	$\log_2 C$	Best Trial	15.00	14.53	10.92	14.36
		Mean (μ)	-	14.56	13.64	14.37
		Std. Dev (σ)	-	0.04	1.23	0.01
	$\log_2 \gamma$	Best Trial	-6.00	-0.51	4.99	-0.47
		Mean (μ)	-	-0.51	4.03	-0.47
		Std. Dev (σ)	-	0.01	0.91	0.00

Fig. 3 shows the 5-fold cross validation plot of the trained SVM classifier for IEEE 118 bus system using Grid Search parameter selection method. The best values of SVM parameters obtained for a maximum cross validation accuracy of 96%, as seen in Figure 3, are penalty parameter, $2^C = 2^{15}$ and the RBF kernel parameter, $2^\gamma = 2^{-6}$.

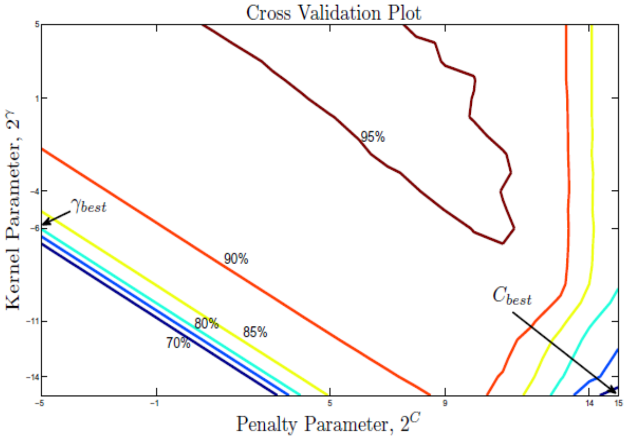

Fig. 3. SVM parameter selection by Gird Search – IEEE 118 Bus

Table IV shows the performance evaluation of various classifiers algorithms obtained during the testing phase. The SVM classifier is trained with the optimal parameters selected by different parameter selection methods as discussed and validated for randomly generated test set samples. About 75% of the data samples generated are randomly chosen for training and remaining 25% for testing processes.

The performance measures of different SVM classifiers are compared with the other equivalent classifiers, viz., Method of Least Squares (MLS) and Probabilistic Neural Network (PNN) classifiers. The LIBSVM software developed by C.C. Chang and C.J. Lin has been used for the design and testing of SVM model [19]. MLS and PNN classifiers are designed using the Statistical toolbox and Neural Network toolbox in Matlab 7.6 respectively.

It can be observed from Table IV that SVM pattern classifier gives a better performance in terms of high classification accuracy and less misclassification rate compared to conventional and neural network pattern classifiers. It is important for the power system security classification problem to minimize the misclassification corresponding to class C and class D. This indicates wrong classification of insecure states, which may be lead to severe blackout. It is well seen that SVM classifiers shows a great reduction in the class C and class D misclassification rate. The high classification accuracy and less misclassification rate makes the SVM classifier suitable for application in online security monitoring system. Furthermore, the SVM+DE is traced to be a more suitable SVM classifier technique, showing an increase in classification accuracy and decrease in Class C and Class D misclassification, as shown highlighted in Table IV.

TABLE IV. PERFORMANCE ANALYSIS OF STATIC SECURITY CLASSIFIERS

		CA	Misclassification (%)			
		(%)	A (S)	B CS)	C (I)	D (HI)
NE 39 Bus	SVM+GS	86.232	4.2860	4.4776	21.951	44.444
	SVM+PSO	86.957	4.7619	7.4627	14.634	66.667
	SVM+RCGA	86.957	4.7619	7.4627	14.634	66.667
	SVM+DE	**89.855**	**9.5238**	**7.4627**	**9.7561**	**33.333**
	MLS	75.363	80.952	2.9960	34.152	11.111
	PNN	85.515	19.052	5.9761	21.952	33.333
IEEE 118 Bus	SVM+GS	95.819	29.545	2.8619	6.6667	0.6369
	SVM+PSO	94.237	25.000	3.8723	12.222	3.8211
	SVM+RCGA	79.887	47.727	3.0303	65.556	50.955
	SVM+DE	**97.062**	**13.636**	**2.1885**	**6.6667**	**0.6369**
	MLS	91.528	100.00	2.5341	12.222	3.1893
	PNN	92.089	50.000	4.5572	17.784	3.1893

V. CONCLUSIONS

This paper presented the pattern analysis method of security evaluation, addressed as a classification task in multi-class labelling environment. The classification of the system static security status in multi-class domain gives an indication of security level to the system operator and helps to initiate necessary control actions at the appropriate time, preventing system collapse. Simulation results have proven that high accuracy classifiers are realizable with SVM algorithm. Furthermore, it has been identified that Differential Evolution method can be applied to fine tune the SVM parameters in the design process in order to get an enhanced performance in the SVM model.

ACKNOWLEDGMENT

The first author would like to thank the Management and Principal of K.L.N. College of Engineering for sponsoring to undergo Ph.D programme in IIT Madras. Authors would also like to thank IIT Madras for providing the necessary facilities to carry out this research work.

REFERENCES

[1] Daniel Kirschen, "Power system security", Power Engineering Journal, vol. 16, no. 5, pp. 241–248, 2002.

[2] C. Pang, F. Prabhakara, A. El-Abiad, and A. Koivo, "Security evaluation in power systems using pattern recognition", IEEE Trans. on PAS, vol. PAS-93, no. 3, pp. 969–976, May 1974.

[3] C. Pang, A. Koivo, and A. El-Abiad, "Application of pattern recognition to steady-state security evaluation in a power system", IEEE Trans. on SMC, vol. 3, no. 6, pp. 622–631, 1973.

[4] I. Saeh and A. Khairuddin, "Static security assessment using artificial neural network", in IEEE 2nd International Power and Energy Conference, 2008. PECon 2008, 2008, pp. 1172–1178.

[5] S. Tso, X. Gu, Q. Zeng, and K. Lo, "Deriving a transient stability index by neural networks for power-system security assessment", Engineering Applications of Artificial Intelligence, vol. 11, no. 6, pp. 771–779, 1998.

[6] C. Liu, M. Su, S. Tsay, and Y. Wang, "Application of a novel fuzzy neural network to real-time transient stability swings prediction based on synchronized phasor measurements" IEEE Transactions on Power Systems, vol. 14, no. 2, pp. 685–692, 1999.

[7] A. Mohamed, S. Maniruzzaman, and A. Hussain, "Static Security Assessment of a Power System Using Genetic-Based Neural Network" Electric Power Components and Systems, vol. 29, no. 12, pp. 1111–1121, 2001.

[8] S. Lin, K. Ying, S. Chen, and Z. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," Expert Systems with Applications, vol. 35, no. 4, pp. 1817–1824, 2008.

[9] C. Wu, G. Tzeng, Y. Goo, and W. Fang, "A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy", Expert systems with applications, vol. 32, no. 2, pp. 397–408, Feb. 2007.

[10] S. Zhou, L. Wu, X. Yuan, and W. Tan, "Parameters Selection of SVM for Function Approximation Based on Differential Evolution", in Proceedings of the international conference on intelligent systems and knowledge engineering (ISKE 2007), Chengdu, China, 2007.

[11] L. L. Grigsby, Power System Stability and Control. CRC Press, 2007.

[12] W. Luan, K. Lo, and Y. Yu, "Ann-based pattern recognition technique for power system security assessment", in Electric Utility Deregulation and Restructuring and Power Technologies, 2000. Proceedings. DRPT 2000. Inter. Conference on, 2000, pp. 197–202.

[13] M. Shahidehpour and Y. Wang, Communication and Control in Electric Power Systems. Wiley-IEEE, 2003.

[14] S. Oh, "A pattern recognition and associative memory approach to power system security assessment", IEEE Transactions on Systems, Man and Cybernetics, vol. 16, no. 1, pp. 62–72, 1986.

[15] S. Weerasooriya and M. El-Sharkawi, "Feature selection for static security assessment using neural networks," in 1992 IEEE International Symposium on Circuits and Systems, 1992. ISCAS'92. Proceedings, vol. 4, 1992, pp. 1693–1696.

[16] C. Hsu and C. Lin, "A comparison of methods for multi-class support vector machines", IEEE transactions on Neural Networks, vol. 13, no. 2, pp. 415–425, 2002.

[17] J. Min and Y. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameter", Expert Systems with Applications, vol. 28, no. 4, pp. 603–614, 2005.

[18] Kalyan Moy Deb, ""Multi-Objective Optimization using Evolutionary Algorithms", John Wiley & Sons, 2001.

[19] C. Chang and C. Lin, LIBSVM: A Library for Support Vector Machines, 2001, software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.

S. Kalyani received her Bachelors Degree in Electrical and Electronics Engineering from A.C. College of Engg. & Tech., Karaikudi, in the year 2000 and Masters in Power Systems Engineering from Thiagarajar College of Engg., Madurai in December 2002. She has obtained her Ph.D degree from IIT Madras in the March 2011. She is currently working as Associate Professor/EEE in K.L.N. College of Engineering, Madurai. Her research interests are power system stability, pattern recognition, neural networks and fuzzy logic applications to power system studies.

K. Shanti Swarup (S'87–M'92–SM'03) is currently a Professor in the Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India. Prior to his current position, he held positions at Mitsubishi Electric Corporation, Osaka, Japan, and Kitami Institute of Technology, Hokkaido, Japan, as a Visiting Research Scientist and a Visiting Professor, respectively, during 1992 to 1999. His areas of research are artificial intelligence, knowledge-based systems, computational intelligence, soft computing, and object modeling and design of electric power systems. He is a Senior Member of IEEE since 2003.