NUKLEONIKA 2011;56(4):323–332

ORIGINAL PAPER

A real-valued genetic algorithm to optimize the parameters of support vector machine for classification of multiple faults in NPP

**Fathy Z. Amer,
Ahmed M. El-Garhy,
Medhat H. Awadalla,
Samia M. Rashad,
Asmaa K. Abdien**

Abstract. Two parameters, regularization parameter c, which determines the trade off cost between minimizing the training error and minimizing the complexity of the model and parameter sigma (σ) of the kernel function which defines the non-linear mapping from the input space to some high-dimensional feature space, which constructs a non-linear decision hyper surface in an input space, must be carefully predetermined in establishing an efficient support vector machine (SVM) model. Therefore, the purpose of this study is to develop a genetic-based SVM (GASVM) model that can automatically determine the optimal parameters, c and σ, of SVM with the highest predictive accuracy and generalization ability simultaneously. The GASVM scheme is applied on observed monitored data of a pressurized water reactor nuclear power plant (PWRNPP) to classify its associated faults. Compared to the standard SVM model, simulation of GASVM indicates its superiority when applied on the dataset with unbalanced classes. GASVM scheme can gain higher classification with accurate and faster learning speed.

Key words: support vector machine (SVM) • fault classification • multi fault classification • genetic algorithm (GA) • machine learning

F. Z. Amer, A. M. El-Garhy, M. H. Awadalla
Faculty of Engineering,
Department of Electronics, Communications and Computers,
Helwan University,
Helwan Governerate,
Helwan, Egypt

S. M. Rashad
Department of Nuclear Law,
Emergency Control Center (ECC),
3 Ahmed el-Zomor, Nasr city, Cairo, Egypt

A. K. Abdien✉
Department of Quality Assurance and Quality Control,
Emergency Control Center (ECC),
3 Ahmed el-Zomor, Nasr city, Cairo, Egypt,
Tel.: 010371 9445, Fax: 2274 0238,
E-mail: asmaak@hotmail.com

Received: 6 April 2010
Accepted: 15 March 2011

Introduction

SVM is a relatively new soft computing method based on statistical learning theory [25]. In SVM, original input space is mapped into a high dimensional dot product space called feature space in which the optimal hyper plane is determined to maximize the generalization ability of the classifier. The optimal hyper plane is found by exploiting a branch of mathematics, called optimization theory, and respecting the insights provided by the statistical learning theory (SLT) [16]. SLT is based on structure risk minimization and has good learning ability even though fewer learning samples are used. Recently, SVM is widely used in many realms, such as face recognition [11], non-linear equalization [24] and spam categorization [6], system reliability forecasting [7, 21]. In fault diagnosis area, some researches also have been done, [29] have found recent applications in data driven modeling, classification and process fault detection [10]. SVM based fault classifiers are claimed to have better generalization properties than artificial neural network (ANN) ones [22]. SVM was originally designed for binary classification, which is not suitable for fault diagnosis, because it has several fault classes in addition to health condition. The main aim of an SVM classifier is obtaining a function $f(x)$, which determines the decision boundary or hyper plane [5]. Kernel-Adatron algorithm was proposed in [5] which could automatically create the decision hyper plane without testing on a validation data. Unfortunately, this algorithm is ineffective if the data have a flat ellipsoid

distribution [17]. Therefore, one possible way to solve the problem is to consider the distribution of the data. Interestingly, various specific functions in SVM, after the learning stage can create the decision hyper plane of the same type [19]. To solve the problem, a systematic method for selecting SVM parameters was provided in [20]. GASVM is introduced in this paper and is exemplified by solving the problem of fault classification in a nuclear power plant (NPP). There are as many as 30 variables in the process comprising various pressures, temperatures, flow, etc. that can be monitored over a period of time. The data reflects a good amount of variability and represents the dynamics of the process. The problem of classifying faults in PWRNPP is tackled in this paper. Multiple faults in a PWRNPP system are generated with the aim to classify them as accurately as possible based on the knowledge of the dataset. This paper investigates the use of genetic algorithms (GAs) [13] to automatically tune the parameters of the binary SVMs contained in common decomposition strategies. Thus, the parameter adjustment problem was formulated as a search for combinations of parameter values able to minimize the error rates obtained in the multi--class problem solution. The GA searches for a set of parameter values which will be common to all classifiers in the decomposition. The paper is organized as follows: section 'Introduction' introduces an overview on the research subject. The monitored states of PWRNPP are given in the section 'Pressurized water reactor nuclear power plant (PWRNPP)'. The mathematical model of the NPP is presented in the section 'Mathematical dynamic model'. The concepts of SVM are reviewed in the section 'Support vector machine (SVM)'. Application of SVM in multi-classification is explained in the section 'Multi-classification SVM'. GA is addressed in the section 'Genetic algorithm (GA)'. Proposed GASVM scheme is clarified in the section 'Proposed GASVM scheme'. Simulation and testing procedures are carried out in the section 'Simulation and testing'. Results of applying GASVM to classify multiple faults in PWRNPP are scrutinized in the section 'Results'. Finally conclusions are given in the section 'Conclusions'.

Pressurized water reactor nuclear power plant (PWRNPP)

The NPP can be subjected to many failures which can affect almost some or all of its state variables. Since the performance of any statistical method depends solely on the datasets on which it is trained, it is advisable that datasets should be created such that all possible dynamics of the state variables are captured. The selected features should be as decorrelated as possible to avoid the problem of multicollinearity. The mathematical model of the PWRNPP [23] based on a neutronic process and thermal hydraulic process with the components: reactor core, pressurizer, pump, steam generator is employed to investigate the effectiveness of the proposed algorithm. The list of PWRNPP state variables is given in Table 1.

Table 1. List of PWRNPP state variables

State variable	State symbol	State name
$X1$	T_{pi}	Primary inlet plenum
$X2$	T_{p1}	Parallel flow first primary fluid lump
$X3$	T_{p2}	Parallel flow second primary fluid lump
$X4$	T_{p3}	Counter flow first primary fluid lump
$X5$	T_{p4}	Counter flow second primary fluid lump
$X6$	T_{po}	Primary outlet plenum
$X7$	T_{m1}	Parallel flow first tube metal lump
$X8$	T_{m2}	Parallel flow second tube metal lump
$X9$	T_{m3}	Counter flow second tube metal lump
$X10$	T_{m4}	Counter flow second tube metal lump
$X11$	L_{s1}	Evaporator sub-cooled water lump
$X12$	Ps	Secondary pressure
$X13$	Xe	Steam mass quality
$X14$	L_{Dw}	Drum water level
$X15$	T_{DC}	Down comer avg. temperature
$X16$	T_{HL}	Primary water inlet temperature (hot water)
$X17$	T_{c1}	Inlet temperature to the core
$X18$	T_{CL}	Primary cold leg temperature
$X19$	T_F	Average fuel temperature
$X20$	T_{UP}	Reactor upper plenum temperature
$X21$	T_{LP}	Reactor lower plenum temperature
$X22$	P_N	Normalized power level
$X23$	P_P	Primary pressure
$X24$	Rt	Total reactivity
$X25$	T_{C2}	Coolant temperature at the exit of the core
$X26$	$W1$	Inlet mass flow rate to the evaporator sub-cooled water lump
$X27$	$W2$	Outlet mass flow rate to the evaporator sub-cooled water lump
$X28$	$W3$	Inlet mass flow rate evaporator steam/water mixture lump
$X29$	$W4$	Inlet mass flow rate evaporator steam/water mixture lump
$X30$	T_{Fw}	Feed water avg. temperature

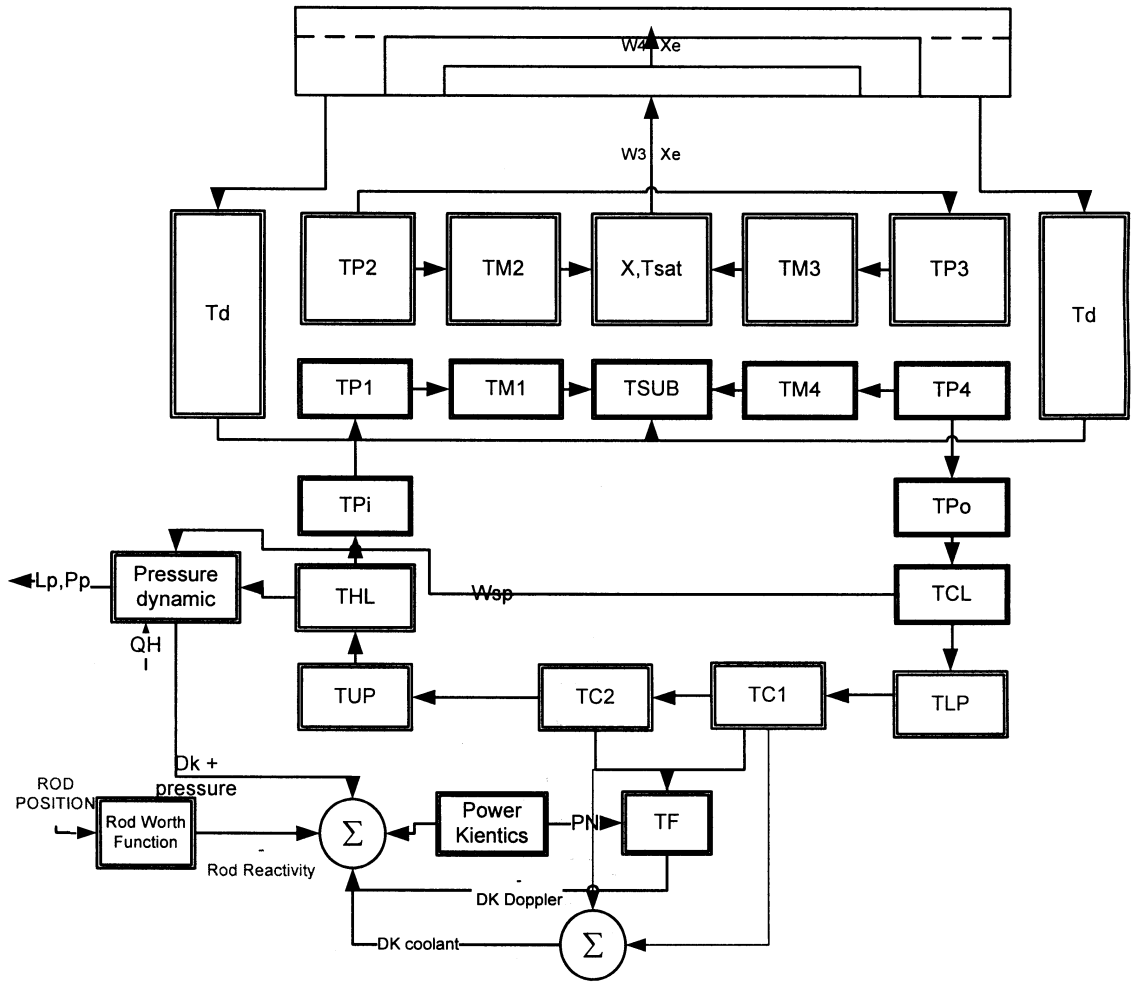

Fig. 1. Simplified block diagram for dynamic model of the primary first loop of the PWRNPP.

The simplified block diagram for dynamic model of the primary first loop of the PWRNPP is shown in Fig. 1.

Mathematical dynamic model

Reactor

The reactor model includes representations for point kinetics and thermal balance to describe how the nuclear heat generated by fission in the fuel elements is transferred to the coolant [23].

Neutron kinetics

A point kinetics model with six delayed neutron groups and reactivity control from external sources (e.g. control rods and boron injection) and feedback from moderator temperature, Doppler and void was formulated.

$$\frac{d\Phi}{dx} = \frac{\rho_{ex} - B}{l}\Phi + \sum_{i=1}^{6} \lambda_i C_i \tag{1}$$

$$\frac{dC_i}{dx} = \frac{B_i}{l}\Phi - \lambda_i C_i \tag{2}$$

where: Φ is the neutron density or fission power; C_i is the concentration precursor; ρ_{ex} is the reactivity; B is the delayed neutron fraction; l is the neutron generation time; λ_i is the decay constant for the i-th delayed neutron group, and t is time.

Feedback reactivity

$$\rho_T = \alpha_p (P_p - P_{po}) \tag{3}$$

where: α_p is coolant pressure coefficient of reactivity; P_p is the primary system pressure; P_{po} is the primary system initial pressure.

The fuel temperature reactivity is given by:

$$\rho_D = \alpha_f (T_f - T_{fo}) \tag{4}$$

where: α_f is fuel temperature (Doppler) coefficient of reactivity; T_f is the average fuel temperature; T_{fo} is fuel temperature when the reactor is cold.

The moderator temperature coefficient of reactivity is defined as

$$\rho_T = \alpha_f \sum_{\substack{\text{coolant} \\ \text{lumps}}} F_{Ci}(T_{Ci} - T_{CiO}) \tag{5}$$

Vessel lower plenum

All the plenums in the system were modeled as well mixed volumes the equation is

$$\frac{dT_{LP}}{dt} = \frac{1}{\tau_{Lp}(T_{LC} - T_{LP})} \tag{6}$$

where: T_{LP} is reactor lower plenum temperature; T_{CL} is primary cold leg temperature; τ_{Lp} is fluid residence time in the lower plenum.

Reactor core

Fuel

The fuel element is simulated by employing a lumped parameter model for thermal conduction in the fuel [27]. The transient heat transfer process for the fuel element may be written as

$$\frac{dT_f}{dt} = \frac{1}{[\![(MC]\!]_p)}\varphi - \left(\frac{UA_f}{MC_p}\right)_f (T_f - T_{c1}) \tag{7}$$

where: T_f is the average fuel temperature; T_{c1} is inlet coolant temperature;$(MC_p)_f$ total heat for fuel lump capacity and A_f is the total heat transfer area.

Coolant

$$\frac{dT_{c1}}{dt} = \left(\frac{UA_f}{MC_p}\right)_C (T_f - T_{c1}) - \frac{2}{\tau}(T_{c1} - T_{C_{in}}) \tag{8}$$

$$\frac{dT_{c2}}{dt} = \left(\frac{UA_f}{MC_p}\right)_C (T_f - T_{c1}) - \frac{2}{\tau}(T_{c2} - T_{c1}) \tag{9}$$

where: T_{c2} is outlet coolant temperature; $(MC_p)_c$ is total heat capacity of coolant associated with the fuel lump; τ is residence time of the coolant in the core; T_{Cin} is inlet temperature to the core and U is overall fuel – to coolant heat transfer coefficient.

Vessel upper plenum

The equation for the upper plenum is given by:

$$\frac{dT_{UP}}{dt} = \frac{1}{\tau_{up}(T_{c2} - T_{uP})} \tag{10}$$

where: T_{uP} is the reactor upper plenum temperature; T_{c2} is coolant temperature at the exit of the core; τ_{up} is residence time of the coolant in the upper plenum.

Reactor coolant pump

For predicting a flow transient through the primary loop, the equation for the time rate of the flow rate changes of the reactor coolant pump as the following.

$$\frac{dW_{\text{pump}}}{dt} = \frac{W_{\text{ref}} - W_{\text{pump}}}{\tau_{\text{pump}}} \tag{11}$$

where: W_{ref} is the set point of the primary coolant pump flow rate and τ_{pump} is the time constant for the primary coolant pump.

Pressurizer model

The pressurizer model is based on mass, energy, and volume balances.

Water mass balance

$$\frac{dM_w}{dt} = W_{wi} + W_{sp} \tag{12}$$

Steam mass balance

$$\frac{dM_s}{dt} = W_s \tag{13}$$

Water energy balance

$$\frac{dE_w}{dt} = W_{wi}h_{wi} - W_s h_{wi} - PV_w + q + W_{sp}h_{sp} \tag{14}$$

Steam energy balance

$$\frac{dE_s}{dt} = W_s h_s - PV_s \tag{15}$$

Volume balance

$$V_s + V_w = V_t \tag{16}$$

where: M_w is mass of water in the pressurizer; M_s is mass of steam in the pressurizer; W_{wi} is mass flow of water into (or out) of the pressurizer; E_w is internal energy of water in the pressurizer; E_s is the internal of steam in the pressurizer; h_{wi} is enthalpy of water entering the pressurizer; h_s is enthalpy of steam in the pressurizer; P is pressure in the pressurizer; q is rate of heat addition to the pressurizer water from the electric heaters; T_s is saturation temperature; V_w is volume of water in the pressurizer; V_s is volume of steam in the pressurizer; V_t is internal volume of the pressurizer. And the primary pressure is [15]:

$$\left(\frac{\partial h_w}{\partial P}M_w + \frac{\partial h_s}{\partial P}M_s\right)\frac{dP}{dt} = q + (h_{wi} - h_w)\frac{dM_w}{dt} + (h_{wi} - h_s)\frac{dM_s}{dt} + W_{sp}(h_{sp} - h_{wi}) \tag{17}$$

where: h_{sp} is enthalpy of sub-cooled water calculated at the cold leg temperature; h_{wi} is the enthalpy of sub--cooled water calculated at the hot leg temperature and h_w is the enthalpy of saturated water at pressure P.

U tube steam generator model

Primary side lump

– Primary inlet plenum

$$\frac{d}{dt}(T_{Pi}) = \frac{1}{\tau_{Pi}}(\theta_i - T_{Pi}) \tag{18}$$

– Parallel flow first primary fluid lump

$$\frac{d}{dt}(T_{P1}) = -\left(\frac{1}{\tau_{P1}} + \frac{U_{pm}}{M_{p1}C_{P1}}\right)T_{P1} + \frac{U_{pm}L_{s1}}{M_{p1}C_{P1}}T_{m1} + \frac{1}{\tau P_1}T_{Pi} \tag{19}$$

– Parallel flow second primary fluid lump

$$\frac{d}{dt}(T_{P2}) + \left(\frac{T_{P1} - T_{P2}}{2}\right)\frac{d}{dt}(L_{s1}) = \frac{1}{\tau P_2}T_{p1} - \left(\frac{1}{\tau P_2} + \frac{U_{pm}L_{S2}}{M_{p2}C_{P1}}\right)T_{P2} + \frac{U_{pm}L_{S2}}{M_{p2}C_{P1}}T_{m2} \tag{20}$$

– Counter flow first primary fluid lump

$$\frac{dT_{P1}}{dt} = \frac{1}{\tau P_S}T_{P2} - \left(\frac{1}{\tau P_S} + \frac{U_{pm}L_{S2}}{C_{p1}M_{ps}}\right)T_{P3} + \frac{U_{pm}L_{S2}}{C_{p1}M_{ps}}T_{m3} \tag{21}$$

– Counter flow second primary lump

$$\frac{dT_{P4}}{dt} + \left(\frac{T_{P4} - T_{P3}}{2}\right)\frac{dL_{S1}}{dt} = \frac{1}{\tau_{P4}}T_{P4} - \left(\frac{1}{\tau_{P4}} + \frac{U_{pm}L_{S2}}{M_{p1}C_{P1}}\right)T_{P3} + \frac{U_{pm}L_{S1}}{M_{p4}C_{P1}}T_{m3} \tag{22}$$

– Primary outlet plenum

$$\frac{d}{dt}(T_{Po}) = \frac{1}{\tau_{Po}}(T_{P4} - T_{Po}) \tag{23}$$

– Metal tube lumps

$$\frac{d}{dt}(T_{m1}) + \frac{(T_{m1} - T_{m2})}{2L_{s1}}\frac{dL_{s1}}{dt} = \frac{1}{M_{m1}C_{m1}}\left[U_{pm}L_{s1}(T_{P1} - T_{m1}) - U_{ms1}L_{s1}(T_{m1} - T_{s1})\right] \tag{24}$$

$$\frac{d}{dt}(T_{m2}) + \frac{(T_{m1} - T_{m2})}{2L_{s2}}\frac{dL_{s1}}{dt} = \frac{U_{pm}L_{S2}}{M_{m2}C_m}T_{P2} - \left(\frac{U_{pm}L_{S2} + U_{mS2}L_{S2}}{M_{m2}C_m}\right) + T_{m2} + \frac{U_{ms}L_{S2}}{M_{m2}C_m}T_{sat} \tag{25}$$

$$\frac{d}{dt}(T_{m3}) + \frac{(T_{m3} - T_{m4})}{2L_{S2}}\frac{dL_{S2}}{dt} = \frac{U_{pm}L_{S2}}{M_{m3}C_m}T_{P3} - \left(\frac{U_{pm}L_{S2} + U_{mS2}L_{S2}}{M_{m3}C_m}\right) + T_{m3} + \frac{U_{ms2}L_{S2}}{M_{m3}C_m}T_{sat} \tag{26}$$

$$\frac{d}{dt}(T_{m4}) + \frac{(T_{m3} - T_{m4})}{2L_{S1}}\frac{dL_{S1}}{dt} = \frac{U_{pm}L_{S1}}{M_{m4}C_m}T_{P4} - \left(\frac{U_{pm}L_{S1} + U_{mS4}L_{S1}}{M_{m4}C_m}\right) + T_{m4} + \frac{U_{ms1}L_{S1}}{2M_{m4}C_m}T_d + \frac{U_{mS1}L_{S1}}{2M_{m4}C_m}T_{sat} \tag{27}$$

– Secondary side lump

$$\frac{d}{dt}(L_{S1}) = \left(A - A41\frac{dp}{dt}\right)\int AA_4 \tag{28}$$

$$A = U_{ms1}L_{s1}(T_{m1} + T_{m4} - T_d - T_{sat}) + W_1C_{p2}T_d - W_2C_{p2}T_{sat} - \rho_s A_s L_{s1}C_{p2}\left(\frac{1}{2\tau_d}(T_{dw} - T_d)\right) \tag{29}$$

– Evaporator steam/water mixture lump

$$\frac{dL_{s2}}{dt} = \frac{w_2 - w_3 - A_s L_{s2}\left(\frac{-\left(\frac{\partial V_f}{\partial p} + \overline{X}\frac{\partial V_{fg}}{\partial p}\right)}{(V_f + \overline{XV}_{fg})^2}\frac{dp}{dt} + \frac{-(V_{fg}/2)}{(V_f + \overline{XV}_{fg})^2}\frac{dX_e}{dt}\right)}{\rho_b A_s} \tag{30}$$

Support vector machine (SVM)

SVM is an effective approach for pattern recognition. The basics of the SVM are found in [5, 16]. For the two classes problem, we assume that we have a dataset, which given m as the amount of the labeled training samples, x_i are the training samples while y_i are the targets or labels in N dimensional space as:

$$\{x_i, y_i, x_i \in R^n,\ y_i \in \{-1,1\},\quad i = 1,...,m\} \tag{31}$$

In SVM, the results in a linearly separable problem correspond to a decision function:

$$f(x) = \operatorname{sgn}\big((w \cdot x) + b\big) \tag{32}$$

The set of samples is said to be optimally separated by the hyperplane if it is separated without error and the margin is maximal, this hyperplane bisects the shortest line between the convex hull of the two classes, thus, it must satisfy the following constrained minimization as:

$$\min \frac{1}{2}w^T \cdot w,\quad y_i(w \cdot x) + b \geq 1 \tag{33}$$

This hyperplane can be constructed by solving quadratic optimization problem which is the solution of w and expand with $w = \sum_i \alpha_i y_i x_i$ in terms of a subset of training pattern that lie on the margin. These training patterns x_i are called support vectors, which provide the important information of classification problems. Then the decision function can be formulated as:

Table 2. Types of kernel functions

Kernel function	Classifier
$k(x,x_i) = \exp(-\|\|x - x_i\|\|^2/2\sigma^2)$	RBF
$k(x,x_i) = (x^T x_i + 1)^d$	Polynomial
$k(x,x_i) = (x^T x_i)$	Linear

$$f(x) = \operatorname{sgn}\left(\sum_i \alpha_i y_i (x.x_i) + b\right) \tag{34}$$

For the linearly non separable case, a modification on previous minimization problem needs to be done to recover the misclassification data points. A new penalizing error variable is introduced; ε as the measurement of violation on the constraints:

$$\min \frac{1}{2} w^T \cdot w + C\sum_i^l \varepsilon_i, \quad y_i (w \cdot x) + b \geq 1 - \varepsilon_i \tag{35}$$

C is used to weight the penalizing parameter ε_i. The SVM separate a non-linear separable classification problem by mapping the data into a feature space via a non-linear map. This solution is done by using kernel (K). By using kernel, the non-linear separated samples input space will be turn out to be linearly separated after being mapped in feature space. The decision boundaries function for non-linear problems can be formulated as:

$$f(x) = \sum_i \alpha_i y_i K(x, x_i) + b \tag{36}$$

There are a lot of kernel functions. Some of them are shown in Table 2. Any type of kernel can chose according to experiments as it is dependent on the sample data.

Multi-classification SVM

The original SVM is binary classification method, i.e., it classifies the two classes problems. However, fault classification requires multi-class classification. There are three methods for multi-classification; namely: one to one, one to all and one to others. One to one algorithm [18, 27] constructs all two class SVM classifiers between any two classes so k $(k - 1)/2$ two class SVM classifiers can be constructed in all for a case of k classes. In the recognition stage, a new sample x is input to the classifier between the m class and the n class with classifying function:

$$f^{mn}(x) = \operatorname{sgn}\left\{\sum_{i=1}^{l} \alpha^{mn} y^{mn} K(x, x_i) + b^{mn}\right\} \tag{37}$$

If the classifier shows that x belongs to the m class, then a ballot is cast for the m class. After recognized by all these k $(k-1)/2$ classifiers, x can be judged to belong to the class that has the most ballots. Some disadvantages of this algorithm are: (1) the number k $(k - 1)/2$ of two class SVM classifiers increases greatly with the class number k, calculation increases greatly as well, and the rate of training and recognition is very slow; (2) when two or more classes have the same ballots, it is hard to judge which class the new sample x belongs to; (3) there is at least a class whose ballots are the most, so a new sample x that does not really belong to any of these k classes would be misjudged to belong to them and the classifying error appears.

One to all algorithm [18, 27] takes any one class m of these k classes as a category, takes the rest $k - 1$ classes as another category, constructs a two class SVM classifier and names it SVM m. In this way, k two class SVM classifiers can be constructed in all for a case of k classes. The classifying function of SVM m is:

$$f^m(x) = \operatorname{sgn}\left\{\sum_{i=1}^{l} \alpha^m y^m K(x, x_i) + b^m\right\} \tag{38}$$

In the recognition stage, the new sample x is input to all these k classifiers. There are k outputs in all. In this classifier, 1 class represents (class 1) and 0 class represents (classes 2 and 3). Unlike one-vs.-one classifier, all data are used in each classifier. To classify an observation, y pred is determined for all classifiers. The test point is identified as the class with maximum y pred, which is class 1 in this example. The new sample x is judged to belong to the class whose classifier has the largest output. Some disadvantages of this algorithm are: (1) all samples have to be taken into the training of all these k classifiers; in the recognition stage, only after all these k classifiers have recognized the new sample x can the classification result be obtained. Therefore the calculation is very time-consuming and the rate of training and recognition is slow as well; (2) there is at least a two-class SVM classifier whose output is the largest, so a new sample x that does not really belong to any of these k classes would be misjudged to belong to them and the classifying error appears; (3) when two or more classifiers have the same outputs, it is hard to judge which class the new sample x belongs to.

In 'one to others' algorithm [12, 27] for a data set of k classes, the classifying function is given by:

$$(x_i, y_i), \quad i = 1, 2, ..., l, \quad x_i \in R^d, \quad y_i \in \{1, 2, ..., k\} \tag{39}$$

where: l is sample number; d is dimension of sample vector; k is class number. First, we have a case of k classes, take a top-priority class from these k classes as a category, and take the rest $(k - 1)$ classes as another category, construct a two-class SVM classifier and name it SVM 1. Next, this top-priority class is excluded, and then we have a case of $(k - 1)$ classes, take a top-priority class from those $(k - 1)$ classes as a category, and take the rest $(k - 1 - 1)$ classes as another category, and construct a second two-class SVM classifier and name it SVM 2, and so on and so forth till the last two-class SVM classifier is constructed and named SVM $(k - 1)$. In this way, $(k - 1)$ two-class SVM classifiers can be constructed in all for a case of k classes. In the fault diagnosis, the most common or the most dangerous fault can be given top priority. The 'one to others' multi-class SVM classifier is a binary tree composed of several two-class SVM classifiers organized by fault priority as shown is Fig. 2.

Genetic algorithm (GA)

GA is a stochastic search algorithm modeled on the process of natural selection, which underlies biological evolution [26]. GA has been successfully applied in

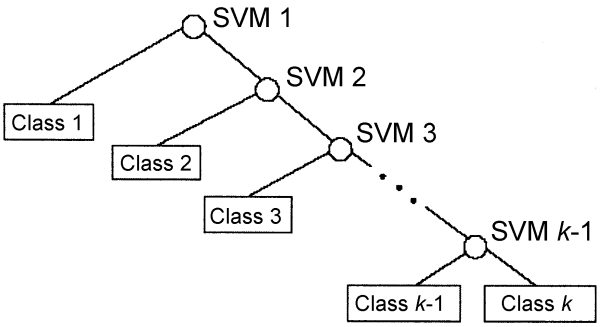

Fig. 2. Framework of the 'one to others' SVM algorithm.

optimization and machine learning problems [3, 17]. GA evolves a population of chromosomes as potential solutions to an optimization problem. There are three major design decisions to consider when implementing a GA to solve a particular problem. A representation for candidate solutions must be chosen and encoded on the GA chromosome, fitness function must be specified to evaluate the quality of each candidate solution, and finally the GA run parameters must be specified including which genetic operators to use.

Initial population

In general, the initial population is generated randomly. In this way, however, we will end up with a population where each individual contains the same number of 1's and 0's on the average. To explore subsets of different numbers of features, the number of 1's for each individual is generated randomly. Then, the 1's are randomly scattered in the chromosome.

Mutation

Mutation is the genetic operator responsible for maintaining diversity in the population [15]. Mutation operates by randomly "flipping" bits of the chromosome, based on some probability. This probability should usually be set fairly low. If it is set to high, the search will turn into a primitive random search.

Crossover

Crossover is a genetic operator that allows new solution regions in the search space to be explored. It is a random mechanism for exchanging genes between two chromosomes using the one point crossover, two point crossovers, or homologue crossover. Offspring replaces the old population using the elitism or diversity replacement strategy and forms a new population in the next generation. Genetic crossover and mutation operation is shown in Fig. 3.

Replacement

Replacement schemes determine how a new population is generated. The concept of overlapping populations is used in this work, where parents and offspring are merged, and the best individuals from this union will form the next population.

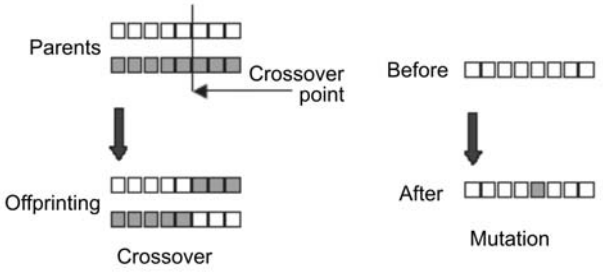

Fig. 3. Genetic crossover and mutation operation.

Selection

This is the process of choosing parents for reproduction. Usually, it emphasizes the best solutions in the population, but since the replacement scheme employed here already offers enough evolutionary pressure, a random selection approach was chosen.

Random immigrant

This is a method that helps to keep diversity in the population, minimizing the risk of premature convergence works by replacing the individuals whose fitness is under the mean by recently initialized individuals [1]. Random immigrant is invoked when the best individual does not change for a certain number of generations.

Fitness function

The main goal of feature selection is to use fewer features to obtain the same or better performance [2]. Fitness function is one of the most important parts in genetic search. This function is used to evaluate the effectiveness of each individual in a population, so it has an individual as an input and it returns a numerical evaluation that must represent the goodness of the feature subset. The search strategy's goal is to find a feature subset minimizing this function. The crossover and mutation functions are the main operators that randomly impact the fitness value. The evolutionary cycle of GA is shown in Fig. 4.

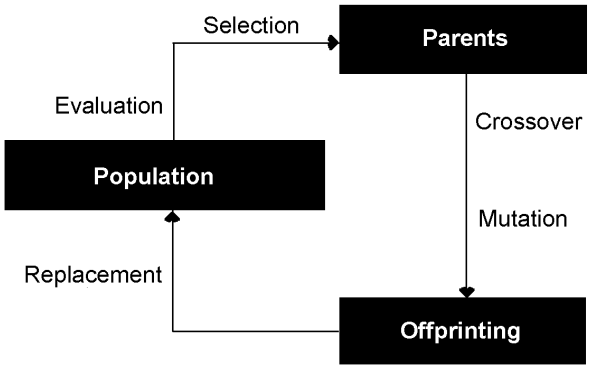

Fig. 4. Evolutionary cycle.

Proposed GASVM scheme

SVM with radial basis function (RBF) is used in this work. The selection of the SVM parameters values plays an important role in the performance of SVM [4]. To design an effective SVM model, values of parameters in SVM have to be chosen carefully in advance [8, 14]. The SVM with RBF includes the following parameters:

a) Regularization parameter C, which determines the tradeoff cost between minimizing the training error and minimizing the complexity of the model [9].
b) Parameter σ of the kernel function which defines the non-linear mapping from the input space to some high-dimensional feature space, which constructs a non-linear decision hyper surface in an input space [28].

In this study, GA is used to determine the optimal values of C and σ that assure highest predictive accuracy and generalization ability simultaneously. The proposed model is named GASVM. Since two parameters should be optimized, the chromosome should comprises three parts, C and σ as shown in Fig. 5, where the binary coding system was used to represent it.

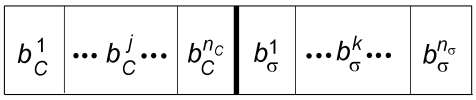

Fig. 5. The chromosome comprises three parts C and σ.

The main parameters of the GA are given hereafter: population size = 100, maximum number of generations = 1000, length of binary string of chromosome = 8, probability of crossover = 0.7, probability of mutation = 0.3, adjustable range of $C = 1 \rightarrow 1000$ and adjustable range of $\sigma = 1 \rightarrow 100$. Fitness function, assessing the performance of each chromosome, should be defined. Many forecasting performance indices can be used as a fitness function, such as MSE (mean-squared error), RMSE (root-mean-squared error), MAE (mean absolute error), and MAPE (mean-absolute percent error). In this research, the aim is to maximize the ROCC (rate of correct class) which expresses the SVM performance as described in Eq. (10).

$$\text{(40)}\quad \text{ROCC} = \text{SVM performance} = \frac{1}{l}\left(\sum_1^l |y_i - y_t|\right) * 100\%$$

where: l is the number of training data; y_i is the actual output; y_t is the expected output. The fitness function is selected as:

$$\text{(41)}\quad \text{Fitness function} = -\text{ROCC} = -\frac{1}{l}\left(\sum_1^l |y_i - y_t|\right) * 100\%$$

GA is used to estimate the values of c and σ that minimize the fitness function (maximize ROCC). The scheme of the proposed GASVM used to classify PWRNPP faults is shown in Fig. 6.

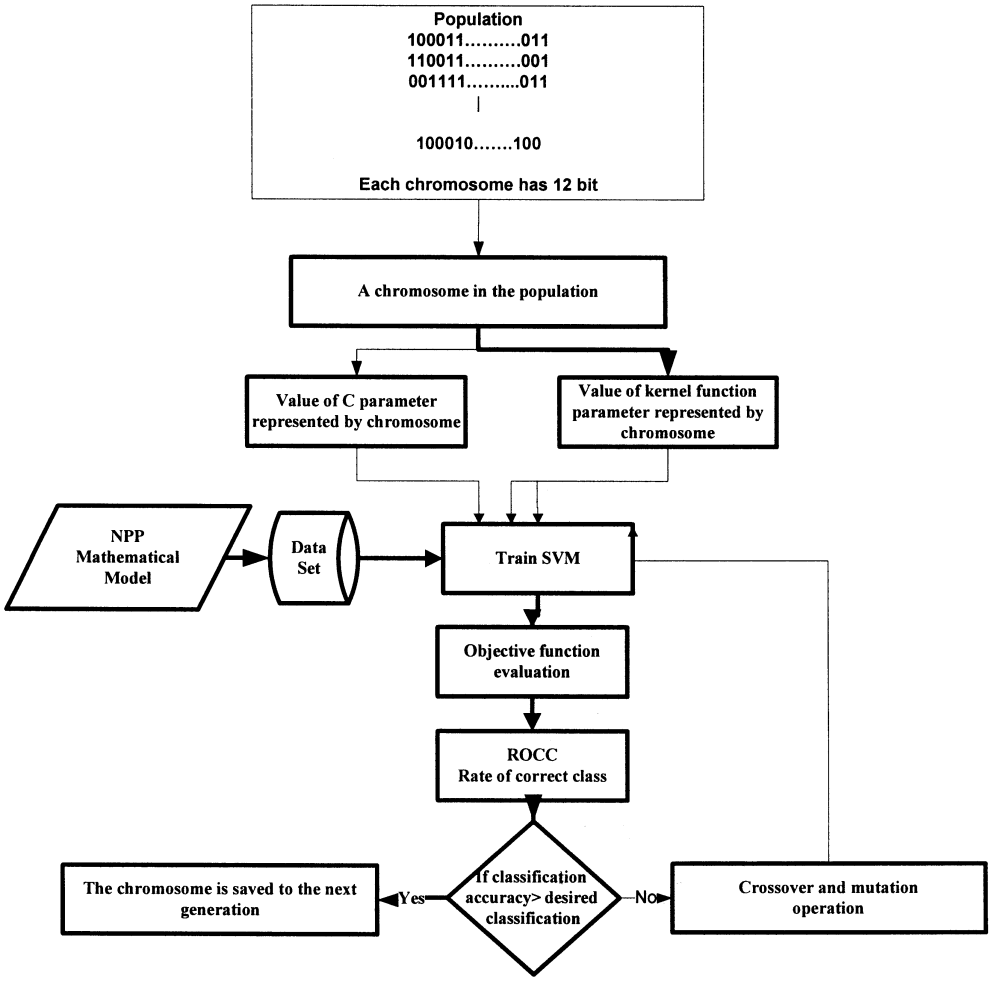

Fig. 6. The scheme of the proposed GASVM.

Table 3. Description of considered faults in PWRNPP

Fault	Type of fault
Fault 1	Step change in reactivity 20*c*
Fault 2	Step change in inlet temperature 4°F
Fault 3	Step change in feedwater temperature 10°F

Simulation and testing

To investigate the proficiency of the proposed GASVM in classifying overlapped multiple failure classes, PWRNPP simulator is used to generate three classes of faulty data. The simulated faults are given in Table 3.

Fault 1 is associated with reactivity of the reactor which affects the reactor temperature. Faults 2 and 3 are associated with reactor cooling water inlet temperature. The three faults are good representation of overlapping data. For each faulty case, two sets of data are generated; namely: training and testing data. Training data is used to build the model and the testing data are used to validate the model. The training and testing data contain 192 and 110 observations, respectively. Each observation includes 30 process monitored states (i.e. $X1, X2, \ldots, X30$). Two variables $X7$ (parallel flow in the first tube metal lump) and $X15$ (down comer average temperature) are clearly affected by faults 1, 2 and 3 while; the remaining 28 variables are not remarkably affected. The proposed GASVM is simulated and applied on different multi-classification methods to estimate optimal values of kernel parameters. Using estimated optimal values, failure classification is performed on the observed data of monitored states $X7$ and $X15$ to classify faults 1, 2 and 3.

Results

The values of kernel parameters were estimated by trial and error [1] and were found 10, 2 for c and σ, respectively. Values of ROCCs corresponding to estimated kernel values are calculated and found 82.29, 84.50 and 90.63% for one to one, one to all, and one to other classification methods, respectively. Proposed GASVM scheme is applied to get the optimal kernel values that minimize the fitness function. The evolution of fitness function for each multi-classification method is shown in Figs. 7–9.

The estimated optimal values of c and σ with the corresponding ROCCs for different classification algorithms are listed in Table 4.

Different multi-classification methods are performed on the observed data of monitored states $X7$ and $X15$ to classify faults 1, 2 and 3. ROCCs based on trial

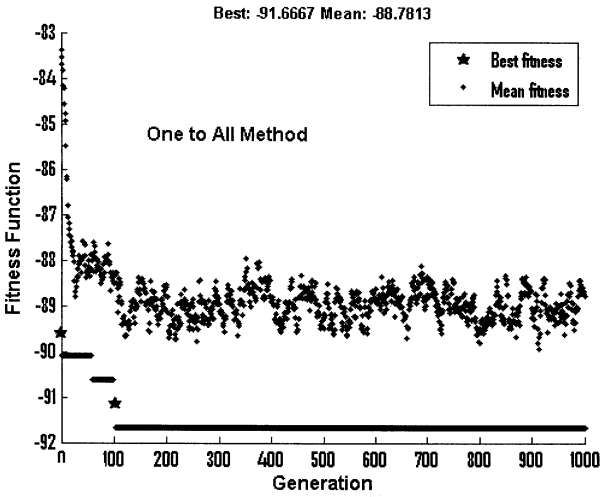

Fig. 8. The evolution of fitness function for one to all multi-classification.

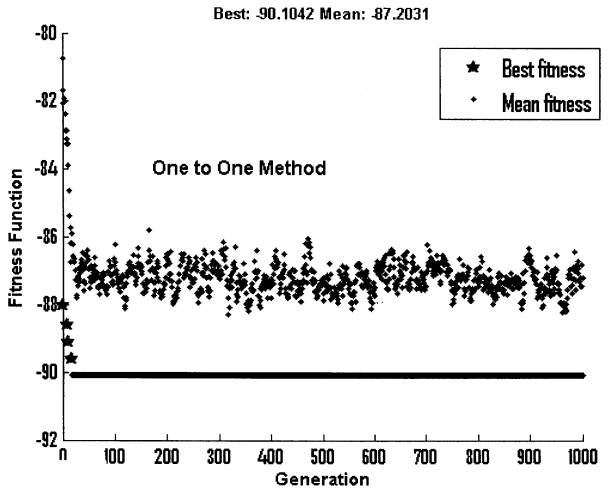

Fig. 7. The evolution of fitness function for one to one multi-classification.

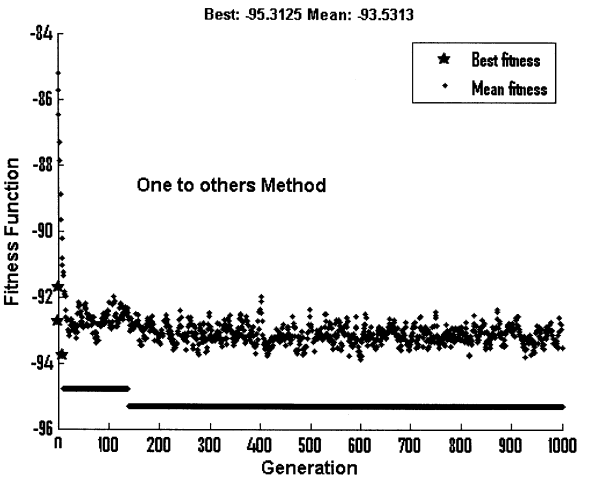

Fig. 9. The evolution of fitness function for one to others multi-classification.

Table 4. Optimal values of kernel parameters and corresponding ROCCs

Multi-classification method	Optimal value of kernel parameter		ROCC values (%)
	c	σ	
One to one	880.9783	1.1045	90.1024
One to all	291.8880	1.1447	91.6667
One to others	789.6410	1.2179	95.5313

Table 5. Improvements in ROCC values

Multi-classification method	ROCC values		Percentage of improvement in ROCC (%)
	Trial and terror (%)	GASVM (%)	
One to one	82.29	90.10	9.49
One to all	84.50	91.67	8.49
One to others	90.63	95.53	5.41

and error kernel values are compared with the GASVM optimal estimated values. Remarkable improvements in ROCC values are recorded when GASVM is applied. Results of comparisons are summarized in Table 5.

Conclusions

In this paper the concepts of SVM are reviewed, the use of SVM in multi-classification is surveyed, the kernel parameters of SVM are optimized using GA with the aim of improving ROCCs, and a new GASVM is proposed to classify multiple faults in PWRNPP. Simulation of the proposed GASVM proves remarkable improvements in ROCCs for different multi-classification algorithms compared to traditional SVM with trial and error kernel values.

References

1. Avci E (2009) Selecting of the optimal feature subset and kernel parameters in digital modulation classification by using hybrid genetic algorithm – support vector machines: HGASVM. Expert Syst Appl 36:1391–1402
2. Babaoglu I, Findik O, Ulker E (2010) A comparison of feature selection models utilizing binary particle swarm optimization and genetic algorithm in determining coronary artery disease using support vector machine. Expert Syst Appl 37:3177–3183
3. Chen LH, Hsiao HD (2008) Feature selection to diagnose a business crisis by using a real GA-based support vector machine: an empirical study. Expert Syst Appl 35:1145–1155
4. Chou PH, Wu MJ, Chen KK (2010) Integrating support vector machine and genetic algorithm to implement dynamic wafer quality prediction system. Expert Syst Appl 37:4413–4424
5. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines. Cambridge University Press, Cambridge
6. Drucker H, Wu D, Joksons DW (1999) Support vector machine for spam categorization. IEEE Trans Neural Networks 10:1048–1054
7. Fei SW, Liu CL, Miao YB (2009) Support vector machine with genetic algorithm for forecasting of key-gas ratios in oil-immersed transformer. Expert Syst Appl 36:6326–6331
8. Fei SW, Sun Y (2008) Forecasting dissolved gases content in power transformer oil based on support vector machine with genetic algorithm. Electr Power Syst Res 78:507–514
9. Fei SW, Zhang XB (2009) Fault diagnosis of power transformer based on support vector machine with genetic algorithm. Expert Syst Appl 36:11352–11357
10. Gao JF, Shi WG, Tan JX, Zhong FJ (2002) Support vector machines based approach for fault diagnosis of valves in reciprocating pumps. IEEE Can Conf Electr Comput Eng 3:1622–1627
11. Guo GD, Li SZ, Chan KL (2001) Support vector machine for face recognition. Image Vision Comput 19:631–638
12. Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Networks 13:415–425
13. Huang C, Wang C (2006) A GA-based feature selection and parameters optimization for support vector machines. Expert Syst Appl 31;2:231–240
14. Huanga HL, Chang FL (2007) ESVM: Evolutionary support vector machine for automatic feature selection and classification of microarray data. BioSystems 90:516–528
15. Huerta EB, Duval B, Hao JK (2006) A hybrid GA/SVM approach for gene selection and classification of microarray data. Springer-Verlag, Berlin
16. Kecman V (2001) Learning and soft computing. Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, US
17. Lahiri SK, Ghanta KC (2008) Prediction of pressure drop of slurry flow in pipeline by hybrid support vector regression and genetic algorithm model. Chin J Chem Eng 16;6:841–848
18. Lorena AC, de Carvalho ACPLF (2008) Evolutionary tuning of SVM parameter values in multiclass problems. Neurocomputing 71:3326–3334
19. Madeira MM, Tokhi MO, Ruano MG (2000) Real-time implementation of a Doppler signal spectral estimator using sequential and parallel processing techniques. Microprocess Microsyst 24:153–167
20. Merler S, Jurman G (2006) Terminated ramp – support vector machines: a nonparametric data dependent kernel. Neural Networks 19:1597–1611
21. Pai PF (2006) System reliability forecasting by support vector machines with genetic algorithms. Math Comput Modell 43:262–274
22. Samanta B, Al-Balushi KR, Al-Araimi SA (2003) Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. Eng Appl Artif Intell 16:657–665
23. Samia MR (1979) Dynamic energy balance of developing power systems with special reference to nuclear units performance. PhD thesis, Faculty of Engineering, Cairo University, Egypt
24. Sebald DJ, Bucklew JA (2000) Support vector machine techniques for nonlinear equalization. IEEE Trans Signal Process 48:3217–3226
25. Vapnik VN (1995) The nature of statistical learning theory. Springer-Verlag, New York, pp 157–173
26. Wahed ME, Ibrahim WZ (2010) Neural network and genetic algorithms for optimizing the plate element of Egyptian research reactor problems. Nucl Eng Des 240:191–197
27. Widodo A, Kim EY, Son JD *et al*. (2007) Support vector machine in machine condition monitoring and fault diagnosis. Mech Syst Sig Process 21:2560–2574
28. Wu CH, Tzeng GH, Goo YJ, Fang WC (2007) A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. Expert Syst Appl 32:397–408
29. Wu KP, Wan SD (2009) Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. Pattern Recogn 42;5:710–717