AN EFFICIENT MODEL SELECTION FOR SVM IN REAL–WORLD DATASETS USING BGA AND RGA

Omid Naghash Almasi[*], *Ehsan Akhtarshenas*[†], *Modjtaba Rouhani*[‡]

Abstract: Support vector machine (SVM) has become one of the most popular machine-learning methods during the last years. The design of an efficient model and the proper adjustment of the SVMs parameters are integral to reducing the testing time and enhancing performance. In this paper, a new bipartite objective function consisted of the sparseness property and generalization performance is proposed. Since the proposed objective function is based on selecting fewer numbers of the support vectors, the model complexity is reduced while the performance accuracy remains at an acceptable level. Due to the model complexity reduction, the testing time is decreased and the ability of SVM in practical applications is increased Moreover, to prove the performance of the proposed objective function, a comparative study was carried out on the proposed objective function and the conventional objective function, which is only based on the generalization performance, using the Binary Genetic Algorithm (BGA) and Real-valued vectors GA (RGA). The effectiveness of the proposed cost function is demonstrated based on the results of the comparative study on four real-world datasets of UCI database.

Key words: *Model selection, model complexity, support vector machines, genetic algorithms, classification, real-world datasets*

Received: September 20, 2013
Revised and accepted: October 14, 2014

DOI: 10.14311/NNW.2014.24.029

1. Introduction

Support Vector Machine (SVM) introduced based on statistical learning theory by Vapnik [31]. It is one of the supervised learning methods that have been used for classification, regression and more recently for one-class classifications. The main

[*] Omid Naghash Almasi – Corresponding Author, Young Researchers and Elite Club, Mashhad Branch, Islamic Azad University, Ostad Yousefi St., Emamiye Blvd., Ghasem Abad, Khorasan Razavi, Iran, 9187147578 Tel.: +98 9153068100, E-mail: o.almasi@ieee.org

[†] Ehsan Akhtarshenas, Control and Electrical Engineering, Islamic Azad University, Daneshgah Blvd., Basij Square, Khorasan Razavi, Iran, 96916-29, Tel.: +98 9155020341, E-mail: ehsan_akhtarshenas@yahoo.com

[‡] Modjtaba Rouhani, Control and Electrical Engineering, Islamic Azad University, Gonabad Branch, Daneshgah Blvd., Basij Square, Gonabad, Khorasan Razavi, Iran, 96916-29, Tel: +98 9153139734, E-mail: m.rouhani@ieee.org

idea of the support vector machine is to create a hyper-plane as the decision-making surface that can maximize the margin of separation between two classes of positive and negative data. More specifically, the support vector machine is a discriminate that can minimize the separation risk by maximizing the margin between the two classes of data [32].

Unfortunately, there are two problems with the practical applications of SVM. The first problem concerns the lack of any precise method to adjust SVM parameters, and the other one is related to the prolonged testing time preventing the use of SVM in real-world applications. The generalization ability of SVM depends on the appropriate selection of its parameters, i.e. kernel parameter(s) and regularization parameter. The optimal choice of those parameters is called SVM's model selection problem [6,28,35].

The kernel parameter(s) implicitly describes the geometry of the data in high-dimensional space called feature space. In this space, the data become linearly separable to obtain the highest margin between two classes. The selection of kernel parameter(s) will change the shape of the separating surface in input space. Selecting extremely large and small values for kernel parameter(s) may lead to the over-training and under-training of SVM models, respectively. Consequently, the generalization ability of the SVM is reduced in both cases [17,24,37].

In non-separable problems, the slack parameters are introduced to determine the margin violation by noisy training data. Thus, a penalty factor, C, is assigned to control the amount of margin violations. In other words, C is defined to determine the trade-off between empirical error and structural risk minimization, which ensures the precise output of the classifier in presence of the noisy data. Higher C values cause the margin to be hard and the cost of violation to become too high, so the separating model surface over-fits the training data. In contrast, lower C values allow the margin to be soft, which results in under-fitting separating model surface. In both cases, the generalization performance of classifier is unsatisfactory, so it makes the SVM model useless.

Among all the learning algorithms, SVM is extremely popular for its sparseness feature. This feature enables SVM to allow only some parts of the training data, i.e. support vectors, to participate in construction of the optimal hyper-plane. Hence, the model size will be small and it is expected that SVM consumes less time in the testing phase.

The model selection parameters not only control the generalization performance, but also affect the SVM model size. Large problems generate large data sets and due to these data sets the SVM's model size (number of SV) will increase. On the other hand, the SV's model size is still increased because of the large dataset; however, SVM is a sparse machine learning method. The testing phase is slow due to the large model size, so practical applications of SVM are restricted.

There are several methods for reducing the number of support vectors and reducing the size of the network, which can transform SVM into a practical instrument. Among these algorithms, reference [2] integrates SVM algorithm with fast nearest neighbor condensation algorithm (FNNC) algorithm. A hybrid strategy algorithm that combines PSO and EGA algorithms for simplification of SVM solution proposed in [25]. Reduced support vector machine (RSVM) [2] and its different types [15] and [19] are other ways to achieve this goal. In [3] cross validation algorithm proposed to overcome this problem. Clustering algorithm based on SVM (CB-SVM) introduced in [39]. The selection of adaptive feature vector based on correlation principle and greedy algorithm for simplification of SVM solution is another method introduced in [22]. The strengthening of sparseness by removing support vectors that are linearly related to SVM solution is another method proposed in [9].

Although a variety of appropriate algorithms have been proposed to reduce the number of support vectors, none of them take an action to solve the model selection problem and reduce the SVM model size at one. There are a variety of the methods accounting different criteria including Jaakkola-Haussler Bound [14] Opper-Winther Bound [27], Span Bound [33], Radius/Margin Bound [16] and analysis of Distance Between Two Class in the feature space (DBTC) [3], and generalization performance (cross validation) [12] to select the optimal SVM's model.

In general, the methods for selection of optimal SVM models based on the existing criteria can be divided into two categories: the classic methods and methods based on evolutionary strategies. The classic methods such as [11,16,28,3,33] use the gradient descent technique to optimize the model selection criterion. The optimization based on the gradient methods is rather fast; however, if the object function is uniform, the optimization algorithm will be trapped in the local minimum. Similarly, if the objective function is not uniform, it will be non-differentiable, thus making it impossible to use gradient method. The problems associated with all other gradient-based methods include the memory occupation, inverting Gram Matrix, and solving a quadratic optimization problem [11,17,24,28].

To overcome the problems of the first group, the second group is introduced based on the evolutionary strategy. In this group, the global search methods like PSO [21,34,36], chaos adaptive PSO [7], quantum PSO [26], simulated annealing [40], ant colony [4] and genetic algorithms [10,13,38] are used to find the best solution of the objective function.

SVM is a sparse machine-learning algorithm. However, the sparseness of the solution in SVM is not as good as we expect and making it extremely slows in testing stage. Furthermore, the best generalization performance depends on the appropriate model selection. In this paper, a bipartite objective function composed of sparseness property and generalization performance is proposed. The proposed objective function selects the best efficient model with an acceptable level of performance accuracy. Such model has smaller network size, and thus lower testing speed, which is highly important to real-world applications. The evolutionary strategies of the genetic algorithm of binary representation and real-valued vector are used to evaluate the performance of the proposed objective function in comparison with the conventional objective function, which only consists of generalization performance [6,12,21,35,36,38].

The Section 2 of this paper states the problem and discusses how to use an evolutionary algorithm to select the model and how to encode it. SVM formulation is elaborated on in Section 3. The evolutionary strategies of the genetic algorithm of binary representation and real-valued vector are described in Section 4. The main idea of the proposed objective function is discussed in Section 5 where it is evaluated using real-world data set as well as BGA and RGA algorithms, proving the performance of the objective function. Finally, some conclusions are made in Section 6.

2. Problem Statement

There are two key points in using evolutionary algorithms to solve model selection problem and enhance the sparseness property at once. The first one is how to encode the problem, i.e. how to display the model parameters as chromosomes. The second is how do define the objective function as a way of evaluating the performance of each chromosome. These two factors are as follows:

Encoding: Each parameter of the model selection problem is considered as one of the dimensions of the candidate solution. The final optimal solution of the evolutionary algorithms includes kernel and SVM regularization parameters.

Objective function: The objective function is the sum of generalization performance and sparseness property of SVM.

Although the SVM draws on the minimization principle of structural risk to minimize the upper bound of generalization error, it still suffers from over and under training as well as sizable network in real-world applications. By considering the sparseness as the second part of the objective function, the over and under training problems can be resolve. Also, the sparseness of the solution is strengthened and the testing time is reduced. To evaluate the proposed objective function, it is compared with the conventional objective function. Fig. 1 shows the procedure of comparing both objective functions using the BGA and the RGA.

3. Support Vector Machine

This section reviews the theory behind the support vector machine. We have a limited number of training dataset with D-dimensional feature vector and the maximum size of l with corresponding class label $y_i \in \{-1, +1\}$. The objective is to optimally solve a two-class classification problem. The optimal hyper-plane in SVM is achieved by solving a convex quadratic problem which is formulated as follows:

$$\min \varphi(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \tag{1}$$

s.t.

$$\begin{aligned} &y_i\left(w^{\mathrm{T}}.x_i + b\right) \geq 1 - \xi_i, && i = 1, 2, \ldots, l \\ &\xi_i \geq 0, && i = 1, 2, \ldots, l \end{aligned} \tag{2}$$

where $\xi_i(s)$ are the slack parameters to determine the classification error of each data points. The regularization parameter, C is determined by the user. It controls the trade-off between maximization of separable margin and the error of classification. The data is mapped on the feature space by a function called kernel $k(.,.)$ in which it is inner product of φ in form of $k(x_i, x_j) = \varphi(x_i).\varphi(x_j)$. The functions which comply with the Mercer conditions [5,24] can be used as kernel function.

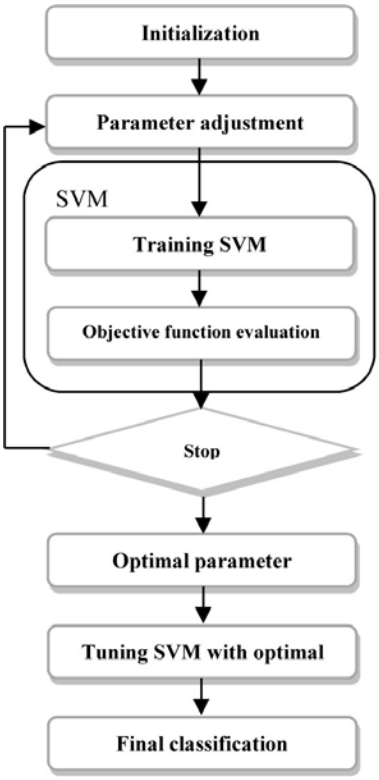


Fig. 1 *The procedure of comparing both objective functions using BGA and RGA.*

By developing the Lagrangian equation and taking the derivative based on KKT optimal conditions, the dual form of SVM problem is obtained as follows:

$$\begin{aligned}
&\max Q(\alpha) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i, x_j) - \sum_{j=1}^{l}\alpha_j \\
&\text{s.t.} \\
&\sum_{j=1}^{l}\alpha_i y_i = 0 \\
&0 \le \alpha_i \le C, \qquad i = 1,\dots,l
\end{aligned} \tag{3}$$

where $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_l)$ are non-negative Lagrangian parameters. The x_i(s) corresponding to non-zero solutions of dual equation are called Support Vectors (SVs). Many of the solutions for dual equations are equal to zero. Thus, they are not included in the construction of the final hyper-plan.

Assume ν as the set that include all indices of support vectors, the final optimal separator hyper-plane is as follows:

$$\sum_{i\in\nu}^{\#sv} \alpha_i y_i k\left(x_i, x_j\right) + b = 0 \tag{4}$$

and the final decision-making function is defined as follows:

$$y\left(x\right) = \operatorname{sign}\left[\sum_{i\in\nu}^{\#sv} \alpha_i y_i k\left(x_i, x_j\right) + b\right] \tag{5}$$

where the bias value can be easily calculated based on KKT conditions.

4. Genetic Algorithms

Genetic algorithm is a population-based iterative algorithm inspired by Darwin's evolution theory [1,23,38]. Given its specific features, genetic algorithm is a popular global search method. In summary, GA algorithm consists of a population of individuals who try to find the optimal solution through the iterations based on the evolutionary rules. In each iteration, which is called a generation, the best parent are chosen by a selection mechanism from all the individuals based on the objective function evaluation. Then the descendants are produced from these parents by crossover and mutation operations respectively. These descendants constitute the next generation population of the GA, and this process is repeated until the stopping condition is met. GA population algorithm has two binary and real-valued representations which are called BGA and RGA. In this study, the BGA and RGA are used to compare the effectiveness of the proposed and the conventional objective functions.

4.1 Model selection using BGA

The binary representation of genetic algorithm consists of a binary vector with the length of l. In BGA, the population of chromosomes is formed by binary vectors of $g_i = (g_{i1,\dots,g_{il}}) \in (0,1)$. Each chromosome represents an individual and each individual is representative of a candidate solution. Fig. 2 shows the form of a bipartite chromosome used in this paper to select optimal SVM model. The binary chromosomes in BGA are transformed into a set of individuals based on equation (8) for objective function evaluation process. This conversion process and its opposite are called encoding/decoding process. It should be noted that the representation accuracy of parameters depends on the length (l) of binary string.

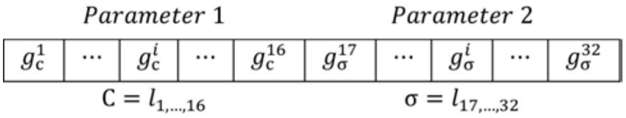


Fig. 2 *Chromosome in BGA.*

$$x = x_{\min} + (x_{\max} - x_{\min}) \times n/(2^{l} - 1) \tag{6}$$

where n, is the value of genotype parameter and x is the value of phenotype parameter within the range of $[x_{\min}\ , x_{\max}]$

4.1.1 Selection mechanism

In each generation, the selection mechanism is essential for choosing the best parents that are likely to produce the best descendants with higher efficiency. In this study, the Roulette wheel selection is used, as it offers higher probability of choosing the individuals with greater efficiency.

4.1.2 Crossover operator

Crossover is one of the evolutionary operators used in GA algorithm to produce descendants from selected parents. In this study, one-point crossover has been used as crossover operator. P_c is the probability of crossover occurrence.

4.1.3 Mutation operator

Mutation is another evolutionary operator in GA. Bitwise mutation is used in this study. Bitwise mutation selects a bit of chromosomes randomly, changing its value from one to zero or vice versa.

4.1.4 Stopping condition

Several criteria can be assumed for stopping condition of GA algorithm. In this study, to establish corresponding conditions, access to the max generation is used as stopping condition in two GA algorithms. The general process for BGA algorithm is shown in Fig. 3.

4.2 Model selection using RGA

The real-valued vector (RGA) is more common than BGA algorithms. These algorithms do not require encoding/decoding process. RGA represents directly all parameters or variables in form of a real-valued chromosome. Thus, the representation of chromosomes in the RGA is straighter than BGA algorithm. A two-dimensional vector encodes SVM parameter in form of a chromosome. Fig. 4 shows the chromosome used in this paper for RGA. In RGA similar to BGA algorithm the selection mechanism of crossover and mutation operators are used to produce descendants and the next generation.

4.2.1 Selection mechanism

Tournament selection was used to determine which chromosome could be transferred to the next generation.

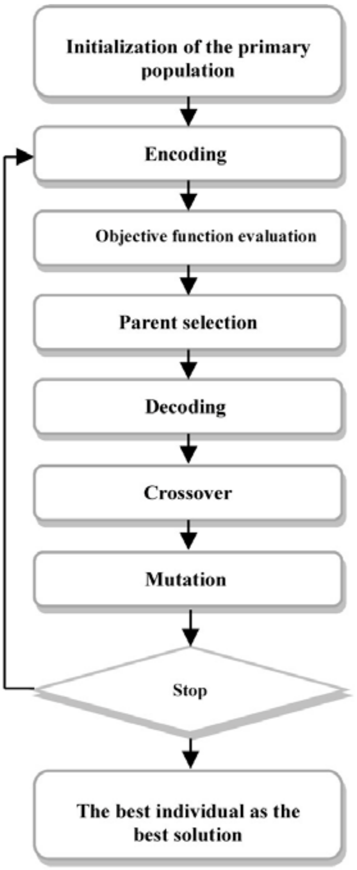
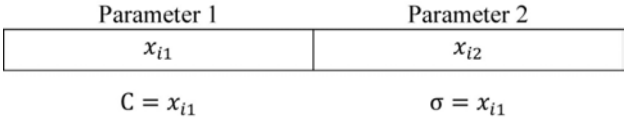

Initialization of the primary population
Encoding
Objective function evaluation
Parent selection
Decoding
Crossover
Mutation
Stop
The best individual as the best solution

Fig. 3 *The working process of BGA algorithm.*



Parameter 1	Parameter 2
x_{i1}	x_{i2}
$\mathrm{C} = x_{i1}$	$\sigma = x_{i1}$

Fig. 4 *Chromosome representation in RGA.*

4.2.2 Crossover operator

In this study, the crossover operator with linear combination was used. P_c is the probability of crossover operator occurrence. Two individuals selected for crossover operator are considered as $x_1^{\mathrm{old}} = (x_{11}, \ldots, x_{1n})$ and $x_2^{\mathrm{old}} = (x_{21}, \ldots, x_{2n})$. By examining the value of P_c, one of the (9) or (10) equations is applied to the individuals for producing the next generation.

$$\begin{array}{l} x_1^{\text{new}} = x_1^{\text{old}} + \sigma(x_1^{\text{old}} - x_2^{\text{old}}) \\ x_2^{\text{new}} = x_2^{\text{old}} - \sigma(x_1^{\text{old}} - x_2^{\text{old}}) \end{array} \tag{7}$$

and

$$\begin{array}{l} x_1^{\text{new}} = x_1^{\text{old}} + \sigma(x_2^{\text{old}} - x_1^{\text{old}}) \\ x_2^{\text{new}} = x_2^{\text{old}} - \sigma(x_2^{\text{old}} - x_1^{\text{old}}) \end{array} \tag{8}$$

x_1^{old} and x_2^{old} are chromosomes prior to the application of the crossover operator and the new chromosome are represented by x_1^{new} and x_2^{new}. To control the variance (σ) of each crossover operator, a random number with micro range is produced. As σ value increases, the descendants lose their similarity to the parents. Therefore, the exploration ability of individuals in the feasible space of the problem is strengthened, and as the σ value decreases, the descendants become more similar to their parents, thus increasing the exploitation ability of population individuals.

4.2.3 Mutation operator

Uniform mutation is one of the most common forms of mutation operator in RGA, which is used in this study based on [1,38]. P_m is the probability of the mutation operator occurrence. By considering each individuals $x^{\text{old}} = (x_1, \ldots, x_n)$, the uniformed mutation is defined as follows:

$$x_k^{\text{old}} = LB_k + r(UB_k - LB_k) \tag{9}$$

$$x^{\text{new}} = (x_1, x_2, \ldots, x_k^{\text{new}}, \ldots, x_n) \tag{10}$$

where r is a random number with uniform distribution between 0 and 1 ($r \in [0, 1]$). K is the position of mutation operator occurrence in the chromosome and n represents the number of dimension in the chromosomes. LB_k and UB_k are respectively the upper and lower bounds of the parameters in k position. The general process of the RGA is shown in Fig. 5.

5. Experiments and simulation

5.1 Experimental and simulation conditions

To evaluate the performance of the proposed objective function, a computer set with the hardware specifications listed in Tab. I was used. The implementation of the proposed objective function and evolutionary strategies was undertaken in MATLAB R2008b. Tab. II shows the characteristics of the dataset. Four different sets of real-world data that are frequently used in the literature are used to challenge the proposed objective function in comparison with the conventional objective function in terms of selecting optimal SVM model. The proposed objective function is a combination of generalization performance and sparseness property. There are several methods for evaluating the performance of SVM. In this study, K-fold cross validation method is used K-fold cross validation is an iterative and robust evaluation method for measuring the performance. It divides the training data set randomly into K equal subsets. In each iteration, K-1 subset is selected

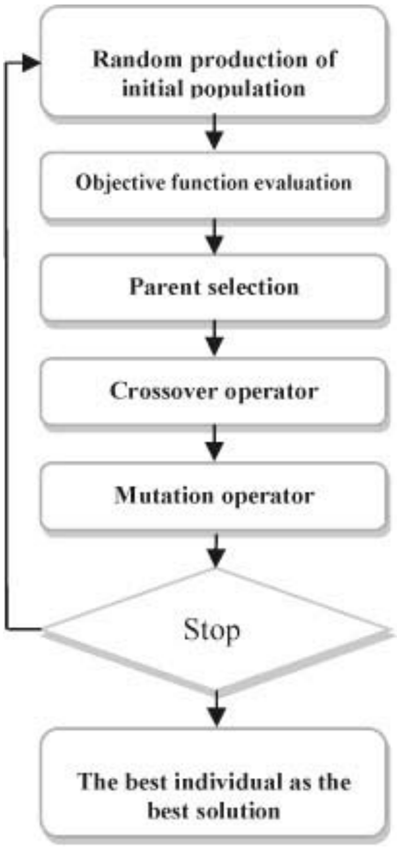

Random production of initial population
Objective function evaluation
Parent selection
Crossover operator
Mutation operator
Stop
The best individual as the best solution

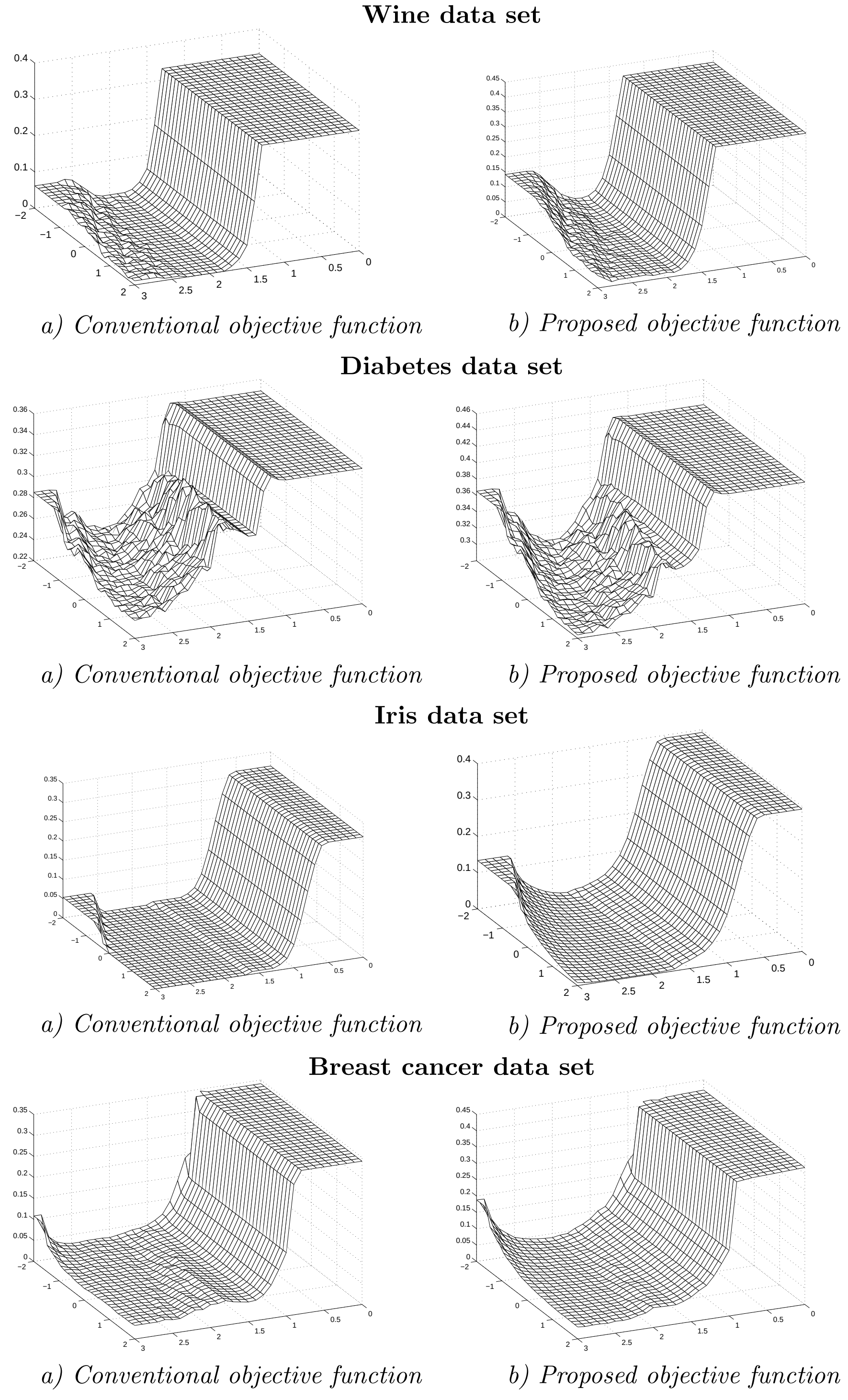
Fig. 5 *The working process of RGA algorithm.*

AMDTurion^{TM}64	CPU
2GB	RAM
Vista32bit	OS

Tab. I *The hardware specifications of the computer.*

Data Set Name	# Data	#Feature
Diabetes	768	8
Breast Cancer	699	10
Wine	178	13
Iris	150	4

Tab. II *Characteristics of the data set.*



a) Conventional objective function *b) Proposed objective function*

a) Conventional objective function *b) Proposed objective function*

a) Conventional objective function *b) Proposed objective function*

a) Conventional objective function *b) Proposed objective function*

Fig. 6 *Surfaces of the objective functions for experiment datasets, a) Conventional objective function, b) Proposed objective function.*

as training set and the remaining subset is chosen as the test set. Cross validation method is repeated for K times, thus all subsets are once selected as a test subset.

After K iterations, Kperformance is achieved with the overall generalization performance being the average of all performances. K-fold validation is a method for evaluating the robust generalization performance, for each subset of data is not only dependent from other subsets, but also dependent from all training sets. In all experiments, K is equal to 10. In conventional objective function, only the generalization performance that is achieved by K-fold cross validation is considered as the objective function. This form of the objective function not only creates under/over training in the modeling, but also increases the size of model in real-world problems. The under/over problem reduces the generalization ability of SVM on the testing data set.

SVM has data-driven structure (numbers of SVs) that is automatically organized based on the training data through the training stage. Thus, the way the optimal model is selected plays an important role in navigation (formation) of the structure of SVM. A large dataset generates a complex model. As such, testing SVM on complex dataset will be a time-consuming process that poses a major challenge to use in the practical application. These evidences limit the use of SVM in the real-world problems.

In general, SVM sparseness property means that only a part of the data called SVs is involved in producing decision-making surface. In essence, SVM is a sparse machine-learning algorithm compared to other learning methods such as neural network. However, the sparseness of the solution in SVM is not as good we imagine. Thus, it is necessary to improve this property in the model. Despite the fact that the accuracy of the performance achieved from K-fold cross validation method guarantees a model with high performance accuracy, it neither prevents the complexity of the models and over/under training problem nor tries to boost the sparseness feature in SVM.

Accordingly, the second part of the proposed objective function plays an important role in the simplification of the model and solving the over training problem that occurs in large data due to the selection of many SV.

If the number of SVs is shown by #SV, then the SVM sparseness property is defined as the ratio of SVs to l.

$$Sparseness = \frac{\#N_{sv}}{1} \tag{11}$$

where l is the total number of data. Finally, the bipartite objective function is formulated in Equation (11)

$$Objective\ Function\,(i) = \alpha_1 K\text{-}fold\ Error + \alpha_2 Sparseness \tag{12}$$

where $\alpha_1 = 0.8$ and $\alpha_2 = 0.2$ are the significant coefficients of *K-fold Cross Validation Error* and *Sparseness* in the objective function, respectively. The range of model selection search for C and Sigma is respectively [0.01, 100] and [1, 1000], respectively. The performance of SVM model test is achieved by averaging 1000 optimal models made out of optimal parameters.

All experiments were conducted using RBF kernel due to the following reasons: First, the RBF kernel maps dataset on the feature space nonlinearly. Thus, when the relationship between the optimal output and their input features is nonlinear, the data set can be used. Secondly, the number of parameters in the model selection affects the complexity of the model. A polynomial kernel has more parameters than the RBF kernel does. Finally, the RBF kernel has less numerical problems [5,17,24,30]. In Appendix 1, all parameters needed to implement BGA and RGA algorithms are shown in Tabs. VII and VIII, respectively.

5.2 Analysis of experiment and simulation results

The parameters of the optimal model obtained from BGA and RGA for all data sets are presented in Tabs. IX to XII of Appendix 2, respectively. Tabs. III to VI present the performance of the tested SVM with the obtained optimal model applied to the dataset of this study. In Fig. 6 the surfaces of objective functions for the set of experiment data is shown.

Iris Dataset

According to Tab. III and the comparative study, the evolutionary algorithms have the same performance accuracy, i.e. 100, in both models using the conventional and proposed objective functions. The model achieved using the proposed objective function was an efficient one compared to the other model. Fig. 7 draws a comparison between the accuracy of the best model obtained from the conventional objective function and the best efficient model achieved from the proposed objective function. Iris dataset has a simple model, thus the advantage of the efficient model in reducing the model complexity and testing time is not remarkable.

Model Selection Based on	**Method**	**Accuracy (%)**	**Model Size (#SV)**	**Testing Time (Sec.)**
Conventional objective function	BGA-SVM	100	8.778	2.37788
	RGA-SVM	**100**	**6.562**	**2.32474**
Proposed objective function	BGA-SVM	100	7.360	2.34089
	RGA-SVM	**100**	**5.431**	**2.32314**

Tab. III *Comparison of the model achieved form the proposed objective function with the conventional objective function in Iris dataset.*

Wine Dataset

Fig. 8 draws a comparison between the performance accuracy of the best model obtained from the conventional objective function and the performance accuracy of the best efficient model using the proposed objective function.

In addition to the reduction of the model complexity, a significant reduction in testing time is also observed. Given the importance of the model simplicity in the practical applications, a slight decrease in the performance accuracy is negligible (See Tab. IV).

Model Selection Based on	Method	Accuracy (%)	Model Size (#SV)	Testing Time (Sec.)
Conventional objective function	BGA-SVM	98.4500	15.425	2.77984
	RGA-SVM	**98.9841**	**12.464**	**2.51253**
Proposed objective function	BGA-SVM	98.3210	13.242	2.65673
	RGA-SVM	**98.5830**	**11.162**	**2.47043**

Tab. IV *Comparison of the model achieved form the proposed objective function with the conventional objective function in Wine dataset.*

Breast CancerDatasets
According to Tab. V and the results of the comparative study, a significant simplification was achieved in the model complexity as well as a considerable reduction in the testing time (See Fig. 9).

Diabetes Dataset
According to Tab. VI and the results of comparative study, this dataset has a more complex model than other data set models. Similar to the result of the other data sets, the proposed cost function achieves an efficient model. This model has a lower model size and testing time in comparison to the model which is obtained by using conventional objective function (See Fig. 10).

In general, the results demonstrate that considering the sparseness property in the model selection problem, can effectively reduce the model size, improve the

Model Selection Based on	Method	Accuracy (%)	Model Size (#SV)	Testing Time (Sec.)
Conventional objective function	BGA-SVM	96.9977	32.008	4.32661
	RGA-SVM	**97.0355**	**37.337**	**4.4193**
Proposed objective function	BGA-SVM	**96.7721**	**27.509**	**4.1399**
	RGA-SVM	96.9569	30.563	4.2874

Tab. V *Comparing the model achieved from the proposed objective function and the conventional objective function in Breast Cancer dataset.*

Model Selection Based on	Method	Accuracy (%)	Model Size (#SV)	Testing Time (Sec.)
Conventional objective function	BGA-SVM	75.0247	260.835	17.1451
	RGA-SVM	**75.1185**	**237.261**	**14.5767**
Proposed objective function	BGA-SVM	74.9503	220.896	13.5882
	RGA-SVM	**75.0732**	**221.961**	**13.6187**

Tab. VI *Comparing the model achieved from the proposed objective function and the conventional objective function in Diabetes dataset.*

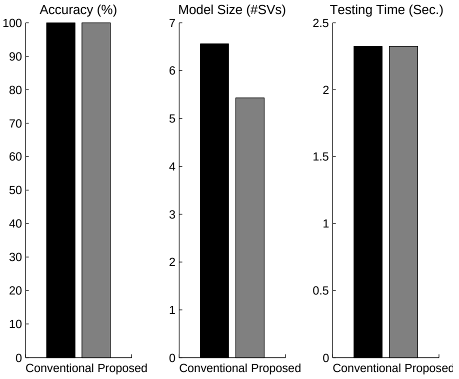


Fig. 7 *Comparing the best model achieved from conventional objective and the proposed objective functions using BGA and RGA in Breast Cancer dataset.*

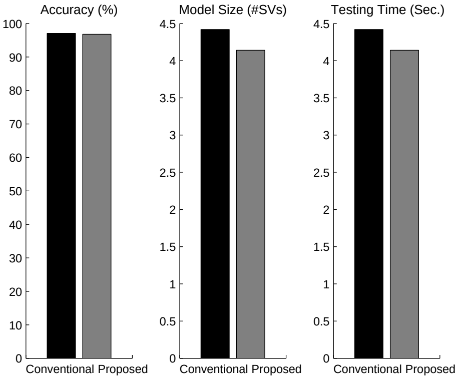


Fig. 8 *Comparing the best model achieved from conventional objective and the proposed objective functions using BGA and RGA in Iris dataset.*

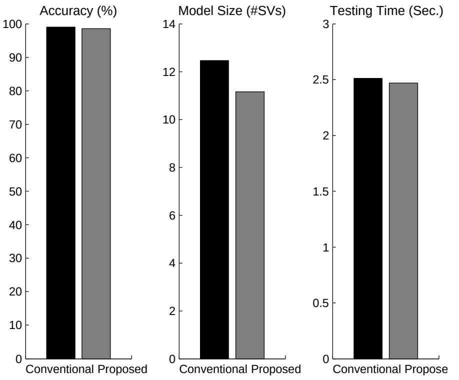


Fig. 9 *Comparing the best model achieved from conventional objective and the proposed objective functions using BGA and RGA in Diabetes dataset.*

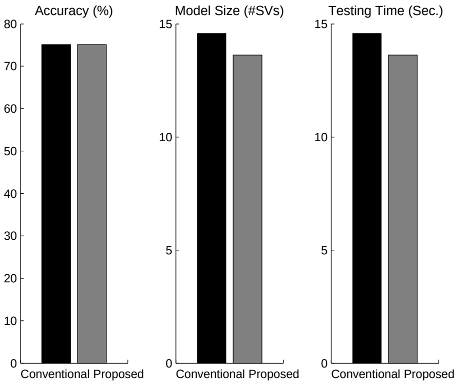


Fig. 10 *Comparing the best model achieved from conventional objective and the proposed objective functions using BGA and RGA in Wine dataset.*

model exploitation in practical applications, and decrease the testing time. Given the effect of model simplicity in reducing the training time, a slight decline in performance accuracy is negligible. It should be noted that by increasing the size of training data in real-world applications, the efficient model had a significant impact on reducing the complexity and testing time of the SVM model.

Although the optimization time of the evolutionary algorithms has not been included in GA model, the RGA algorithm has shorter optimal time and faster convergence compared to BGA.

6. Conclusion

SVM is a popular algorithm because of its high generalization performance. Despite this significant feature, the practical applications of SVM depend on the generalization performance and the simplicity of the model. In this paper, a new objective function based on a generalization performance and sparseness property of SVM was proposed to select an efficient model. The edge of the proposed objective function over the conventional objective function, which is only based on generalization performance, was proved in selection of the optimal SVM model. The testing time and prolonged exploitation of the complex models in the real-world dataset is one of the practical limitations of SVM. Thus, an efficient model is useful for practical applications and can enhance the sparseness features of SVM. To evaluate the proposed objective function in comparison with the conventional objective function, two well-known GA algorithms namely BGA and RGA were used. The results confirm the superior performance of the proposed objective function in selecting the efficient model of SVM. Consequently, the size of the model and the test speed were decreased with a slight drop in the performance accuracy of the model, though this reduced performance accuracy was negligible compared to the increased test speed and model size of SVM model.

Appendix 1

In this appendix, the parameters of the genetic algorithms with binary and the real-valued vector representations are presented in Tabs. VII and VIII.

Population Size	30
Max Generation	50
Chromosome length	16
Selection Type	Roulette wheel
Crossover Type	One point
Crossover Rate	0.8
Mutation Type	Bitwise
Mutation Rate	0.05

Tab. VII *The parameters of genetic algorithms with binary representation.*

Population Size	30
Max Generation	50
Selection Type	Tournament
Crossover Type	Linear combination
Crossover Rate	0.8
Mutation Type	Uniform
Mutation Rate	0.05

Tab. VIII *The parameter of the genetic algorithm of the real-valued vector.*

Appendix 2

The optimal parameters of the efficient model selection derived from BGA and RGA algorithms for each dataset is presented in Tab. II.

Iris		C	Sigma
Conventional objective function	BGA-SVM	485.0657	60.7237
	RGA-SVM	828.5386	63.1849
Proposed objective function	BGA-SVM	477.0017	52.1726
	RGA-SVM	599.0141	44.1571

Tab. IX *The optimal parameters of model selection for Iris dataset.*

Wine		C	Sigma
Conventional objective function	BGA-SVM	358.3165	42.5870
	RGA-SVM	441.6113	38.0516
Proposed objective function	BGA-SVM	802.0445	34.4128
	RGA-SVM	709.0551	9.6558

Tab. X *The optimal parameters of model selection for Wine dataset.*

Breast Cancer		C	Sigma
Conventional objective function	BGA-SVM	638.9265	58.8300
	RGA-SVM	408.4685	69.9766
Proposed objective function	BGA-SVM	855.4120	36.6086
	RGA-SVM	765.7207	54.2733

Tab. XI *The optimal parameters of model selection for Breast Cancer dataset.*

Diabetes		C	Sigma
Conventional objective function	BGA-SVM	176.4862	93.7179
	RGA-SVM	427.6439	86.9857
Proposed objective function	BGA-SVM	779.7038	41.0568
	RGA-SVM	829.6394	65.6860

Tab. XII *The optimal parameters of model selection for Diabetes dataset.*

References

[1] ADEWUYA A.A. *New methods in genetic search with real–valued chromosomes* [online]. United States, 1996. PhD thesis, Massachusetts Institute of Technology in United States [viewed 2014-10-06]. Available from: http://dspace.mit.edu/bitstream/handle/1721.1/10930/35651641.pdf

[2] ANGIULLI F., ASTORINO A. Scaling up support vector machines using nearest neighbor condensation. *IEEE Transaction on Neural Networks.* 2010, 21(2), pp. 351–357, doi: 10.1109/TNN.2009.2039227.

[3] BAKUR G.H., BOTTOU L., WESTON J. Breaking SVM complexity with cross–training. In: *Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS 2004),* Vancouver, Canada. USA: MIT Press, 2004, pp. 81–88.

[4] BLONDIN J., SAAD A. Metaheuristic techniques for support vector machine model selection. In: *Proceedings of the 10th International Conference on Hybrid Intelligent Systems (HIS 2010)*, Atlanta, USA. USA: IEEE, 2010, pp. 197–200, doi: 10.1109/HIS.2010.5600086.

[5] BORDES A. et al. Fast kernel classifiers with online and active learning.*The Journal of Machine Learning Research.* 2005, 6(1), pp. 1579–1619.

[6] CHAPELLE O. et al. Choosing multiple parameters for support vector machines. *Machine Learning.* 2002, 46(1–3), pp. 131–159, doi: 10.1023/A:1012450327387.

[7] CHENG W. et al. An adaptive chaotic PSO for parameter optimization and feature extraction of LS–SVM based modelling. In: *Proceedings of the 1st American Control Conference (ACC 2011)*, San Francisco, USA. USA: IEEE, 2011, pp. 3263–3268, doi: 10.1109/ACC.2011.5991217.

[8] DING S., LIU X. Evolutionary computing optimization for parameter determination and feature selection of support vector machines. In: *Proceedings of the 1st International Conference on Computational Intelligence and Software Engineering*, Wuhan, China. USA: IEEE, 2009, pp. 1– 5, doi: 10.1109/CISE.2009.5366095.

[9] DOWNS T., GATES K.E. Exact simplification of support vector solutions. *Journal of Machine Learning Research.* 2001, 2(1), pp. 293–297.

[10] FROHLICH H., CHAPELLE O., SCHOLKOPF B. Feature selection for support vector machines by means of genetic algorithms. In: *Proceedings of the 15th International Conference on Tools with Artificial Intelligence.* Sacramento, USA. USA: IEEE, 2003, pp. 142–148, doi: 10.1109/TAI.2003.1250182.

[11] GLASMACHERS T., IGEL C. Gradient–based adaptation of general Gaussian kernels. *MIT Press Neural Computation.* 2005, 17(10), pp. 2099–2105, doi: doi:10.1162/0899766054615635.

[12] GUO X.C. et al. A novel LS–SVMs hyper–parameter selection based on particle swarm optimization. *Neurocomputing.* 2008, 71(16–18), pp. 3211–3215, doi: 10.1016/j.neucom.2008.04.027.

[13] HUANG C.L., WANG C.J. A GA–based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications.* 2006, 31(2), pp. 231–240, doi: 10.1016/j.eswa.2005.09.024.

[14] JAAKKOLA D., HAUSSLER T.S. Probabilistic kernel regression models. In: *Proceedings of the 1999 Conference on AI and Statistics*, Stockholm, Sweden. San Francisco: Morgan Kaufmann, 1999, pp. 00–04.

[15] JEN L.R., LEE Y.J. Clustering model selection for reduced support vector machines. In: *Proceedings of the 5th International Conference on Intelligent Data Engineering and Automated Learning–IDEAL*, Exeter, UK. Berlin: Springer-Verlag, 2004, pp. 714–719, doi: 10.1007/978-3-540-28651-6_106.

[16] KEERTHI S.S. Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transaction on Neural Networks.* 2002, 13(5), pp. 1225–1229, doi: 10.1109/TNN.2002.1031955.

[17] KEERTHI S.S., LIN C. J. Asymptotic behavior of support vector machines with Gaussian kernel. *Neural Computation.* 2003, 15(7), pp. 1667–1689, doi:10.1162/089976603321891855.

[18] KENNEDY J., EBERHART R.C. *Swarm intelligence.* USA: Academic Press, 2001.

[19] LEE Y.J., LO H.Y., HUANG S.Y. Incremental reduced support vector machines. In: *Proceedings of the 1st International Conference on Informatics, Cybernetics, and Systems* (ICICS 2003), Kaohsiung, Taiwan. 2003, pp. 1–6.

[20] LEE Y.J., MANGASARIAN O.L. RSVM: reduced support vector machines. In: *Proceedings of the 2001 SIAM International Conference on Data Mining*, Chicago, USA. USA: SIAM, 2001, doi: 0.1137/1.9781611972719.13.

[21] LEI P., YI L. Parameter selection of support vector machine using an improved PSO algorithm. In: *Proceedings of the 2nd International Conference on Intelligent Human–Machine Systems and Cybernetics* (*IHMSC 2010).* Nanjing, China. USA: IEEE, 2010, pp. 221–225, doi: 10.1109/IHMSC.2010.149.

[22] LI Q., JIAO L., HAO Y. Adaptive simplification of solution for support vector machine. *Pattern Recognition.* 2007, 40(3), pp. 972–980, doi: 10.1016/j.patcog.2006.07.005.

[23] LIHU A., HOLBAN S. Real–valued genetic algorithms with disagreements. *Memetic Computing.* 2011, 4(4), pp. 317-325, doi: 10.1007/s12293-012-0098-7.

[24] LIN H.T., LIN C.J. *A study on sigmoid kernels for SVM and the training of non–PSD kernels by SMO–type methods* [online]. Taipei: National Taiwan University, 2003 [viewed 2014-10-06]. Technical report. Available from: http://www.csie.ntu.edu.tw/˜cjlin/papers/tanh.pdf

[25] LIN H.J., YEH J.P. A hybrid optimization strategy for simplifying the solutions of support vector machines. *Pattern Recognition Letters.* 2010, 31(7), pp. 563–571, doi: 10.1016/j.patrec.2009.12.020.

[26] LUOL Z. et al. SVM parameters tuning with quantum particles swarm optimization. In: 2008 *IEEE Conference on Cybernetics and Intelligent Systems*, Chengdu, China. USA: IEEE, 2008, pp. 324–329, doi: 10.1109/ICCIS.2008.4670970.

[27] OPPER O., WINTHER M. *Advances in large margin classifiers.* Cambridge, MA: MIT Press, 2000.

[28] PENG X., WANG Y. A geometric method for model selection in support vector machine. *Expert Systems with Applications.* 2009, 36(3), pp. 5745–5749, doi: 10.1016/j.eswa.2008.06.096.

[29] SHI Y.H., EBERHART R. A modified particle swarm optimizer. In: *Proceeding of the 1998 International Conference on Evolutionary Computation.* Anchorage, USA. USA: IEEE, 1998, pp. 69–73, doi: 10.1109/ICEC.1998.699146.

[30] SUN J. et al. Analysis of the distance between two classes for tuning SVM hyperparameters. *IEEE Transaction on Neural Networks.* 2010, 21(2) pp. 305–318, doi: 10.1109/TNN.2009.2036999.

[31] VAPNIK V. *Statistical learning theory.* New York: Wiley, 1998.

[32] VAPNIK V. An overview of statistical learning theory. *IEEE Transaction on Neural Networks.* 1999, 10(5), pp. 988-999, doi: 10.1109/72.788640.

[33] VAPNIK V., CHAPELLE O. Bounds on error expectation for support vector machines. *Neural Computation.* 2000, 12(9), pp. 2013–2036, doi: 10.1162/089976600300015042.

[34] WANG S., MENG B. PSO algorithm for support vector machine. In: *2010 Third International Symposium on Electronic Commerce and Security (ISECS 2010)*, Guangzhou, China. USA: IEEE, 2010, pp. 163–167, doi: 10.1109/ISECS.2010.92.

[35] WANG S., MENG B. Parameter selection algorithm for support vector machine. *Procedia Environmental Sciences.* 2011, 11(Part B), pp. 538–544, doi: 10.1016/j.proenv.2011.12.085.

[36] WEI LIN S. et al. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications.* 2008, 35(4), pp. 1817–1824, doi: 10.1016/j.eswa.2007.08.088.

[37] WILLIAMS P. et al. A geometrical method to improve performance of the support vector machine. *IEEE Transaction on Neural Networks.* 2007, 18(3), pp. 942–947, doi: 10.1109/TNN.2007.891625.

[38] WU C.H. et al. A real–valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert Systems with Applications.* 2007, 32(2), pp. 397–408, doi: 10.1016/j.eswa.2005.12.008.

[39] YU H., YANG J., HAN J. Classifying large data sets using SVMs with hierarchical clusters. In: *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, USA. ACM: New York, 2003, pp. 306–315, doi: 10.1145/956750.956786.

[40] ZHANG W., NIU P. LS–SVM based on chaotic particle swarm optimization with simulated annealing and application. In: *2nd International Conference on Intelligent Control and Information Processing (ICICIP)*, Harbin, China. USA: IEEE, 2011, pp. 931–935, doi: 10.1109/ICICIP.2011.6008387.