International Journal of Computer Science and Information Security (IJCSIS),
Vol. 14, No. 2, February 2016

OPTIMIZATION OF SVM PARAMETERS BASED ON MOPSO ALGORITHM

Samira Shahinifar

***Abstract*— Parameters selection of support vector machine is a very important problem, which has high influence on the performance of support vector machine.This paper presents a Multi-Objective Particle Swarm Optimization Algorithm (MOPSO) approach to optimize the kernel parameters. In this paper, a MOPSO is designed with two conflicting objectives to be optimized simultaneously. These two objectives are based on the error rate and a ratio of number of support vectors to the number of instances of the dataset under evaluation. To evaluate the performance of the proposed method, experiments were executed on the datasets from LibSVM (library for SVM) and the results obtained were compared with NSGAII algorithm for parameters searching. The results obtained show that the proposed approach has less error rates and vector count across some of the datasets as compared to NSGAII algorithm**

Keywords: Support Vector Machine; Multi-Obejevtive Particle Swarm Optimization; Multi-Obejevtive Genetic Algorithm; Parameter Selection.

I. INTRODUCTION

Support Vector Machine (SVM) that is proposed by Vapnik in 1990's is a new method of machine learning.It is based on Structural Risk Minimization and Vapnik Chervonenks dimensions theory of Statistical Learning Theory. In order to obtain the best generalization ability, it searches for the best compromise between complexity of model and learning ability on the basis of limited sample information [1]. SVM has some advantages such as theoretical foundation is complete, global optimization; training time is short and good generalization performance and so on and it has been a hotspot in pattern recognition area. Parameters selection is a very important problem in SVM research area and the learning ability and generalization performance depend on the parameters selection of SVM.

The analogy of PSO with evolutionary algorithms makes evident the notion that using a Pareto ranking scheme could be the straightforward way to extend the approach to handle multiobjective optimization problems. The historical record of best solutions found by a particle (i.e., an individual) could be used to store nondominated solutions generated in the past (this would be similar to the notion of elitism used in evolutionary multiobjective optimization). The use of global attraction mechanisms combined with a historical archive of previously found nondominated vectors would motivate convergence toward globally nondominated solutions.In this paper, we proposed a method for searching for the optimal parameters of SVM based on MOPSO and this is an efficient approach for parameters selection of SVM. The paper is organized as follows. We briefly introduce the basics of SVM classification in Section 2. In Section 3 we give MOPSO algorithm. The experiment conditions are described in Section 4. The main experiments on SVM classification and experiments analysis are shown in Section 5. In Section 6 we have concluding.

II. SUPPORT VECTOR MACHINE

The basic idea of SVM learning algorithm can be summarized two steps. Firstly, the input space is transformed to a higher dimensional linear feature space by a nonlinear transform function φ . Then the optimal linear separating plane can be constructed in this higher dimensional feature space. The nonlinear transformation can be realized by defining proper kernel function. The classification problem can be considered as two-class problem. Given the training data vectors $D = \{(x_1, y_1), . . . , (x_l, y_l)\}$, $x_i \in R_n$ which belongs to a class labeled by $yi \in \{-1, 1\}$, and the goal is to separate the two classes by the hyperplane (1) which is induced from available examples.

$$(w, x) + b = 0. \tag{1}$$

where w is weight vector, b is threshold.

For non-linear problems, the optimization is to minimize the classification error as well as minimizing the bound on the VC dimension to the classifier. The optimal separating hyperplane with the constraints of:

$$y_i[(w, x_i) + b] \geq 1 - \zeta_i , i= 1, . . . , l. \tag{2}$$

minimizes the function

$$\varphi(\omega,\zeta) = \frac{1}{2}||\omega||^2 + C\left(\sum_i^l \zeta i\right). \tag{3}$$

where $\zeta = (\zeta_1, . . . , \zeta_l)$, ζ_i is a measure of the misclassification errors, C is a constant which controls the tradeoff between the complexity of the decision function and the number of training examples misclassified.

The optimization problem (3) under the constraints of Equation (2) can be transformed to its dual problem.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

$$Max: W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j). \quad (4)$$

with constraints (5)(6)

$$ST: \sum_{i=1}^{n} y_i \alpha_i = 0. \quad (5)$$

$$0 \leq \alpha_i \leq C.\ i = 1, \ldots, l. \quad (6)$$

where α_i is Lagrange multiplier.

SVM is a linear maximal margin classifier in a high-dimensional feature space where data are mapped through a non-linear function $\varphi(x_1)\cdot\varphi(x_2) = K(x_1, x_2)$.

In order to get the optimal hyperplane in feature space, kernel function should be used. Usually, Radial Basis Function(RBF) is used as the kernel function

A. Selection of the Kernel Function

The most common kernel functions available include linear,Radial Basis Function (RBF), sigmoid, and polynomial kernels which are given in eq. (7), (8), (9),(10) respectively [14]. In order to improve classification accuracy, the parameters for these kernel functions should be appropriately set.

Linear kernel function:

$$K(x_i, x_j) = x_i * x_j \quad (7)$$

polynomial kernel function:

$$K(x_i, x_j) = (kx_i x_j + c)^d \quad (8)$$

Radial Basis Function:

$$K(x_i, x_j) = exp\{-\gamma^* |x_{i-} x_j|^d\} \quad (9)$$

sigmoid kernel function:

$$K(x_i, x_j) = tanh(kx_i x_j + c) \quad (10)$$

B. Impacts of the Parameters

The generalization ability of SVM algorithm depends on a set of parameters, including the penalty actor C, the estimated accuracy and the RBF kernel parameter [3].

• Impact of the penalty factor C:The aim of the penalty factor C is to modulate the ratio between the space credibility and the experience risk in a certain digital space, so as to attain the best generalization ability for the machine model. Different digital space requires different optimal parameter C. In certain digital space, a small value of C could lead to weak punishment for the experience error, little complexity of learning machine yet large experience risk, or vice versa. The former is called "lesstrained", and the latter is called "over-trained". When C exceeds a certain value, the complexity of SVM achieves the maximum tolerated by the data space, and the experience risks and generalization ability would not change any more. In each digital space there exists at least one suitable C to achieve the best generalization ability.

• Impacts of the RBF kernel parameter: RBF kernel parameter reflects the distribution or scope characteristics of training sample data, which defines the width of local neighborhood. A large mea10ns relatively little variance.

Impacts of the estimated accuracy: The relaxation factor determines the width of the non-sensitive zone, and affects on the number of support vectors. By selecting a small value, the regression estimation becomes greatly accurate. However, in that case, the number of the support vectors and the complexity of SVM algorithm would both increase. By selecting a great value, regression estimation becomes less accurate, but the number of the support vectors could decrease and the complexity of SVM algorithm would be weakened. Similar to the relaxation factor , the estimated accuracy has the same impacts on the system.

Therefore, in the standard support vector machines, parameters C and determine the complexity of the model through different ways [2].

III. Multi-Obejective Particle Swarm Optimization Algorthm MOPSO

The analogy of PSO with evolutionary algorithms makes evident the notion that using a Pareto ranking scheme could be the straightforward way to extend the approach to handle multiobjective optimization problems. The historical record of best solutions found by a particle (i.e., an individual) could be used to store nondominated solutions generated in the past (this would be similar to the notion of elitism used in evolutionary multiobjective optimization). The use of global attraction mechanisms combined with a historical archive of previously found nondominated vectors would motivate convergence toward globally nondominated solutions [4].

A. Main Algorithm

The algorithm of MOPSO is the following [4].

1) Initialize the population :
   (a) FOR I=0 TO MAX /*MAX=number of particles*/
   (b) Initialize POP[i]
2) Initialize the speed of each particle:
   (a) FOR i = 0 TO MAX
   (b) VEL[i] = 0
3) Evaluate each of the particles in POP[i].
4) Store the positions of the particles that represent nondominated vectors in the repository .
5) Generate hypercubes of the search space explored so far, and locate the particles using these hypercubes as coordinate system where each particle's coordinates are defined according to the values of its objective functions.

6) Initialize the memory of each particle (this memory serves as a guide to travel through the search space. This memory is also stored in the repository):
   (a) FOR i = 0 TO MAX
   (b) PBEST[i] = POP[i]
7) WHILE maximum number of cycles has not been reached DO

   a) Compute the speed of each particle1 using the following expression: where (inertia weight) takes a value of 0.4; and are random numbers in the range ; is the best position that the particle has had;2 is a value that is taken from the repository; the index is selected in the following way: those hypercubes containing more than one particle are assigned a fitness equal to the result of dividing any number (we used in our experiments) by the number of particles that they contain. This aims to decrease the fitness of those hypercubes that contain more particles and it can be seen as a form of fitness sharing. Then, we apply roulette-wheel selection using these fitness values to select the hypercube from which we will take the corresponding particle. Once the hypercube has been selected, we select randomly a particle within such hypercube. is the current value of the particle .

   b) Compute the new positions of the particles adding the speed produced from the previous step

   c) Maintain the particles within the search space in case they go beyond their boundaries (avoid generating solutions that do not lie on valid search space). When a decision variable goes beyonds its boundaries, then we do two things: 1) the decision variable takes the value of its corresponding boundary (either the lower or the upper boundary) and 2) its velocity is multiplied by (1) so that it searches in the opposite direction.

   d) Evaluate each of the particles in .

   e) Update the contents of together with the geographical representation of the particles within the hypercubes. This update consists of inserting all the currently nondominated locations into the repository.Any dominated locations from the repository are eliminated in the process. Since the size of the repository is limited, whenever it gets full, we apply a secondary criterion for retention: those particles located in less populated areas of objective space are given priority over those lying in highly populated regions.

   f) When the current position of the particle is better than the position contained in its memory, the particle's position is updated using The criterion to decide what position from memory should be retained is simply to apply Pareto dominance (i.e., if the current position is dominated by the position in memory, then the position in memory is kept; otherwise, the current position replaces the one in memory; if neither of them isdominated by the other, then we select one of them randomly).

   g) Increment the loop counter.

8) END WHILE

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors).*

IV. SVM Parameters Optimization Algorithm Based on MOPSO

Model selection is an optimization process which requires the choice of several efficient criteria to be optimized [5] . The accuracy of classification and risk of classifier are often used to evaluate the performance of SVM. Therefore, we consider the following two objectives to be optimized simultaneously. The first one is error rate which needs to be minimized and is given by:

A. F (1) = ErrorRate/ Nm (11)

The risk can be estimated by VC dimension. But the VC dimension is difficult to estimate. So we have used a simple bound T for the leave-one-out error given in [1] as our second objective:

B. F (2) = Nsv/Nm (12)

Where Nsv is the number of Support Vectors and Nm are the of training examples. The whole process of the MOGA approach is shown in Fig. 1 below:

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

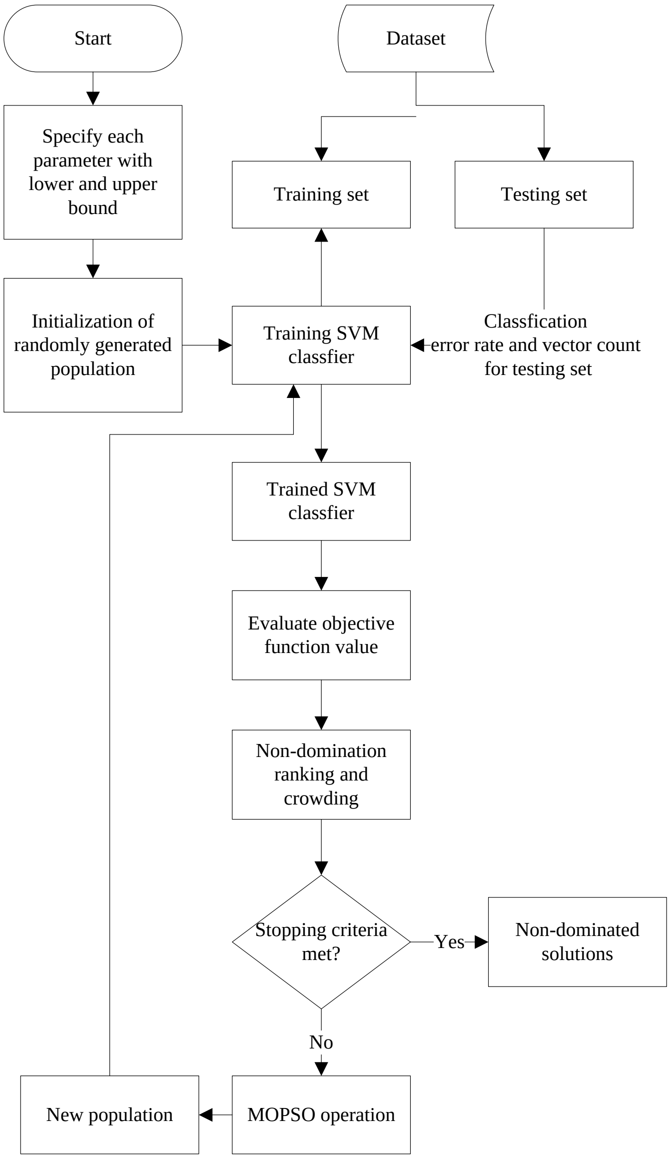

Fig. 1 The MOPSO based approach

V. Experimental Results and Analysis

We have executed our experiments on the MATLAB 2010 using LibSvm. LibSvm is a library for support vector classification. A general use of LIBSVM involves two steps: first, a training data set is used to obtain a model and subsequently the model is validated on a test data set for predictive power. The empirical evaluation was performed on Intel Corei7 CPU running at 1.73 GHz, 4GB of RAM and windows 7 professional Operating System. The proposed MOPSO based model selection is validated on five datasets taken from LibSvm webpage [6]. Table I describes these datasets in terms of number of attributes, instances and classes.

Table I: Datasets Information

Database	Sample
Australian	690
Breast Cancer	683
Diabetes	768
Ionosphere	228
Liver Disorder	345

Table II: Parameters of MOPSO used for classification

Features	Parameters
Population size	100
Repository Size	100
Personal Learning Coefficient	-1.026
Global Learning Coefficient	-1.026
Grid Inflation	0.1
Number of Grids	10
Leader Selection Pressure	4
Repository Member Selection Pressure	2
Termination criterion	Equation(11,12)
Maximum number of generation	500

For each test database in the benchmark we picked, the following test procedures are performed:

1. Adjust the parameters of the SVM classifier with training sets in the database and test the classification performance with the testing sets in the database.
2. Adjust the parameters of the SVM classifier with one training set randomly selected from the training sets in the database and test the classification performance with all testing sets in the database as follows.In order to demonstrate the feasibility of our method, the experiments consist of following two parts corresponding to different test databases.

(a) The experiments are performed on MOPSO with objective function (11,12) and parameters in Table II to run experiments on different databases to validate the method respectively to select parameters for SVM classification and are stopped if the classification erorr rate and vector count do not change in 500 iteration runs.

(b) The experiments are performed on NSGA-II [7] with objective function (11,12).

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Table III: Comparison of Error Rate

Database	Itr 100	Itr 200	Itr 300	Itr 400	Itr 500
Australian	1.3843	1.3899	1.3861	1.3861	1.3861
Breast Cancer	0.8366	0.7321	0.7300	0.7300	0.7300
Diabetes	1.2172	1.2167	1.2167	1.2167	0.2187
Ionosphere	0.6588	0.6667	0.6605	0.6772	0.6605
Liver Disorder	1.7125	1.3588	1.4354	1.3542	1.5426

Table IV: Comparison of Error Rate

Database	Itr 100	Itr 200	Itr 300	Itr 400	Itr 500
Australian	0.3229	0.3124	0.3120	0.3120	0.3120
Breast Cancer	0.1063	0.1050	0.1099	0.1099	0.1099
Diabetes	0.2182	0.2204	0.2204	0.2204	0.2187
Ionosphere	0.0990	0.0920	0.0942	0.0852	0.0942
Liver Disorder	0.3723	0.3458	0.3408	0.3448	0.3447

Table V: Comparison of Vector Count

Database	Itr 100	Itr 200	Itr 300	Itr 400	Itr 500
Australian	0.3145	0.3160	0.3160	0.3152	0.3076
Breast Cancer	0.1100	0.1100	0.1057	0.1057	0.1006
Diabetes	0.2241	0.2206	0.2206	0.2186	0.2186
Ionosphere	0.0978	0.0955	0.0856	0.0955	0.0830
Liver Disorder	0.3599	0.3574	0.3492	0.3509	0.3509

Table VI: Comparison of Vector Count

Database	Itr 100	Itr 200	Itr 300	Itr 400	Itr 500
Australian	1.4261	1.4058	1.4058	1.4029	1.4093
Breast Cancer	0.7701	0.7701	0.7514	0.7514	0.7338
Diabetes	1.0794	1.2117	1.2117	1.1885	1.1885
Ionosphere	0.6579	0.7351	0.6588	0.7351	0.6623
Liver Disorder	1.5391	1.4945	1.3965	1.5281	1.5281

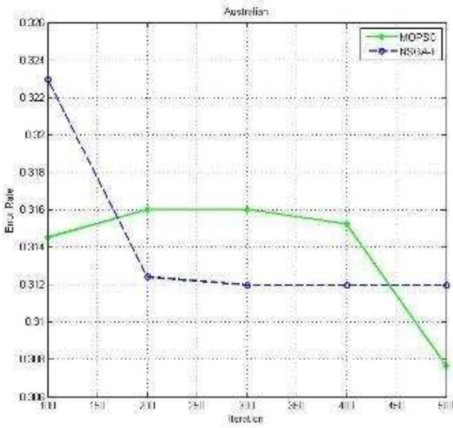

Fig. 2 Classification error rate of SVM classifier

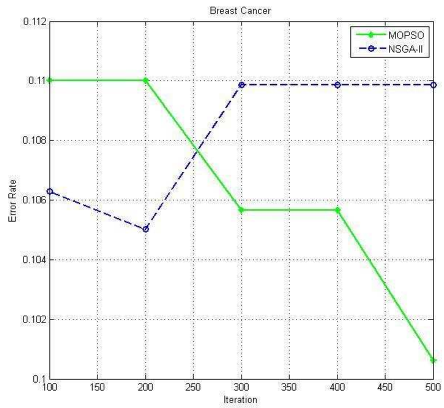

Fig. 3 Classification error rate of SVM classifier

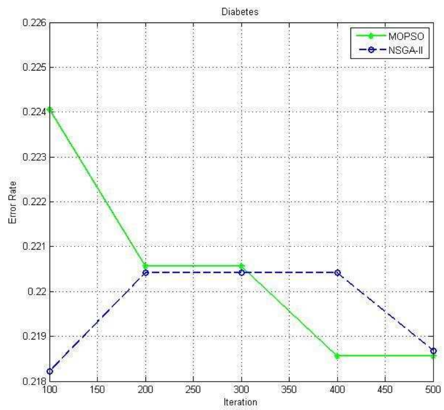

Fig. 4 Classification error rate of SVM classifier

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

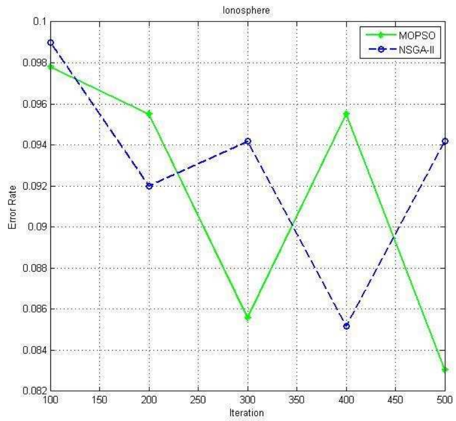

Fig. 5 Classification error rate of SVM classifier

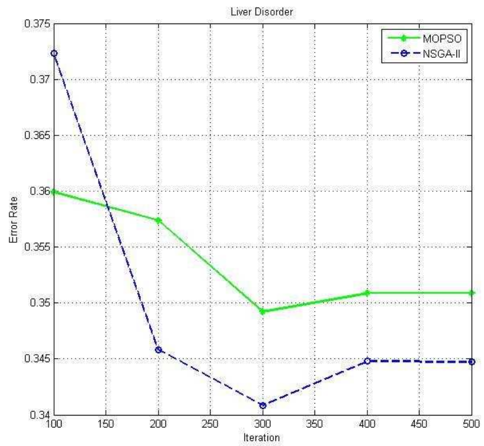

Fig. 6 Classification error rate of SVM classifier

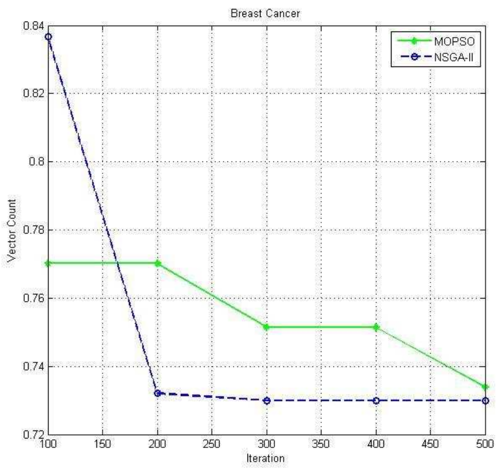

Fig. 7 Classification error rate of SVM classifier

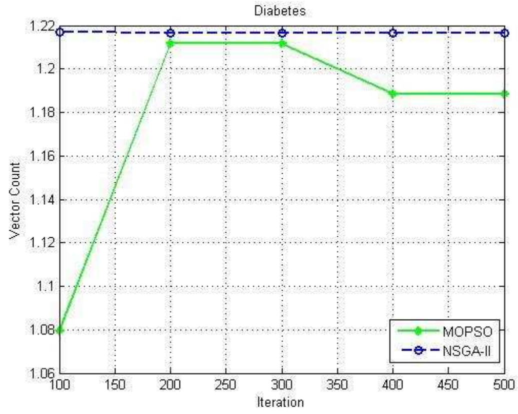

Fig. 8 Classification error rate of SVM classifier

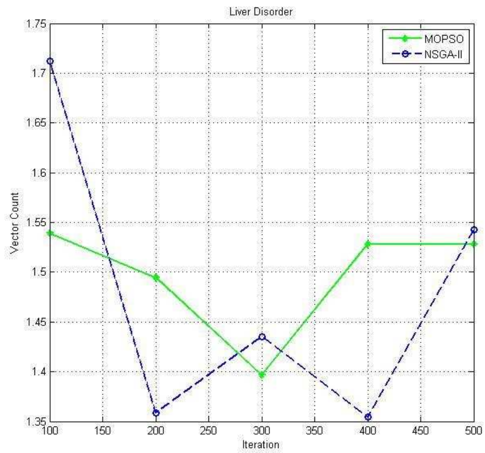

Fig. 5 Classification error rate of SVM classifier

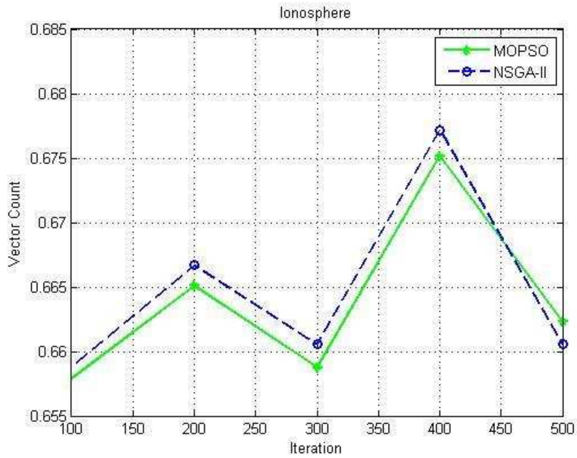

Fig. 9 Classification error rate of SVM classifier

VI. CONCLUSION

In this paper, we presented a MOPSO based approach to SVM model selection. A SVM model selection is multi-objective optimization problem, therefore, MOPSO based approach has been applied to optimize the parameters of SVM according to two objectives such as error rate and vector count.We conducted experiments to evaluate the performance of the proposed approach with five different database and the results obtained were compared with those obtained with NSGAII algorithm. The results obtained show that the proposed approach has less error rates and vector count across some of the datasets as compared to NSGAII algorithm.

REFERENCES

[1] Zhiyu Zhu, Bing Zhang, and Weiting Liu, "A Speech Recognition Method Based on Fuzzy SVM", Computer Engineering, Vol. 32, No. 2, pp. 180-182, 2006.

[2] S. Gao-li and D. Fang-ping. "Introduction to Model selection of SVM Regression[J]", Bulletin of Science and Technology, Vol. 22, No. 6, pp. 154-157,2013.

[3] O. Chapelle and V. Vapnik and O. Bousquet and S. Mukherjee. "Choosing multiple parameters for support vector", machines.Machine Learning, Vol. 46. No.1, pp.131–159,2002.

[4] C. Coello Coello G. Lamont D. Van Veldhuizen. "Handling Multiple Objectives With Particle Swarm Optimization", IEEE Transaction on Evolutionary Computation, Vol. 8, NO. 3, 2004.

[5] D. Wang and W. Xiangbin. "The Optimization of SVM Parameters Based on PSO", Computer Applications, 2008, Vol. 28, No. 1, pp. 134-139.

[6] LIBSVM - A Library for Support Vector Machines. (Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/).

[7] K. A. Saroj. "Multi-Objective Genetic Algorithm Approach to SVM Kernel Functions Parameters Selection.Computer Applications". The International Journal Of Engineering And Science (IJES), Vol. 2, No. 6, pp. 92-101,2013.