Application of Genetic Algorithm-Support Vector Machine for Prediction of Spinning Quality

Zhi-jun Lv, Qian Xiang, Jian-guo Yang

Abstract—**Although many works have been done to construct prediction models on yarn spinning performance, the relation between the spinning variables and the yarn properties has not been established conclusively so far. Support vector machines (SVMs), a new learning-by-example paradigm, are receiving increasing attention in different application domains for which artificial neural networks (ANNs) have had a prominent role. The SVM regression algorithms are briefly introduced in this study, and then SVM models for predicting yarn properties have been presented. Model selection which amounts to search in hyper-parameter space is performed for study of suitable parameters with Genetic Algorithms. The predictive powers of the SVM models are estimated by comparison with ANN models. The experimental results indicate that in the small data sets and real-life production, SVM models are capable of remaining the stability of predictive accuracy, and more suitable for noisy and dynamic spinning process.**

Index Terms—**artificial neural network, genetic algorithms, support vector machines, yarn quality**

I. Introduction

The textile manufacturing is a complex industrial process, ranging from the growth of plants and animals (for production of cotton and wool) to synthetic chemical processing (for fibers such as nylon, polyester, etc.), to rapid processing for converting fibers to yarns and yarns to fabrics. Along the textile chain, yarn quality always gets the privilege of greatest attention as it exerts decisive influence on the efficiency of winding, warping and weaving machines [1]. Spinners have always attempted to estimate yarn quality from existing spinning data in advance in order to select appropriate spinning parameters accordingly. However, the relation between the spinning variables and the yarn properties has not been established conclusively so far. The reasons are the high degree of variability in raw materials, multistage processing, and a lack of precise control on process control [2]. Therefore, for a long time, many works have been done to construct prediction models, which can forecast the various yarn properties from the input fiber and process parameters. One of the first approaches has been the use of mechanistic models by some researchers such as Bogdan, Hearle et al [3]. The mathematical or "white" models developed again by Aggarwal and Frydrych are based on the physical knowledge of spun yarn mechanics [4]. These models are not only very appealing but also give thorough insight into the mechanism of yarn properties. However, to copy with the inherent complexity of yarn structural mechanics, some assumptions have to be introduced into these models. This ultimately leads to relatively low prediction accuracy of mathematical models. In addition, the empirical models such as linear multiple regressions, which use statistical techniques, have relatively higher predictive power. But they do not provide as deep an understanding of relationship between inputs and outputs as mathematical models.

The widespread use of artificial intelligence (AI) has created a revolution in the domain of prediction modeling once more[5,6,7], which includes fuzzy logic model, case based reasoning (CBR) model and more popular artificial neural network (ANN) model etc. These models are termed as "gray box" and "black box" methods as they can connect the inputs and outputs without unearthing explicit physical information about spinning process. Among them, artificial neural network (ANN) has been one of the most universal methods for many prediction-related problems in textile production. Many researchers have reported the use of ANN for the prediction of yarn characteristics from fiber properties and processing information [7]. They found that the neural network provided a worthwhile alternative to regression techniques whenever the fiber/textile structural relationship contained significant nonlinearities.

Although, ANNs are valuable machine learning modeling tools, provided enough rich data is available for modeling the process nonlinear relationships, they rely on empirical risk minimization (ERM). A shortcoming of artificial neural networks is that it is very difficult to intuitively know at what point the system will over-fit the data, which could result in ANN model instability [8]. The theoretical understanding of models which are based on minimization of the generalization error increase the degree of confidence of their use, particularly in noisy and dynamic environments such as those found in industry. The increasing quality demands from the spinner make clear the need to explore novel ways of

Manuscript received March 6, 2011; revised April 20, 2011. This work was supported in part by the Leading Academic Discipline Project of Shanghai Municipal Education Commission (B602) and the Technology innovation Program of Shanghai Municipal Science and Technology Commission (10DZ2250800, 09DZ1122400)

Dr. Zhi-Jun Lv is with College Of Mechanical Engineering, Donghua University，Shanghai 200051, P.R.China (phone: +86-021-67792583; fax: +86-021-67792581; e-mail: lvzj@ dhu.edu.cn).

Qian Xiang is with College Of Mechanical Engineering, Donghua University，Shanghai 200051, P.R.China (e-mail: xqsir@dhu.edu.cn).

Prof. Jian-guo Yang is with College Of Mechanical Engineering, Donghua University ， Shanghai 200051, P.R.China (e-mail: jgyangm@dhu.edu.cn).

quality prediction furthermore.

In recent years, support vector machines (SVMs), a new learning-by-example paradigm, are receiving increasing attention in different application domains for which artificial neural networks (ANNs) have had a prominent role, due to its many attractive features and promising empirical performance [8]. This systematic approach motivated by statistical learning theory led to a class of algorithms characterized by the use of kernels, the absence of local minima, the sparseness of the solution and the capacity control obtained by acting on the margin. Unlike ANN models, SVM models are based on the principle of structure risk minimization (SRM), which equips the later with greater potential to generalize. Since the foundation of the SVMs paradigm was laid down by Vapnik and co-workers [9], applications in many engineering fields have emerged, such as wind speed prediction, quality monitoring, fault detection, and process optimization [8,10]. In this work, an attempt have been made to construct SVM models from spinning data sets in order to predict the yarn properties, while SVM model parameters optimized with Genetic Algorithms. Results have been compared with those of ANN models. The investigation shows that SVM predictive models have a reasonably good accuracy in most of cases, and more suitable for noisy and dynamic spinning process. The relative algorithm, model selection and experiments are presented in detail.

II. SVM REGRESSION ALGORITHMS

Consider a training data set $G=\{(x_1,y_1),(x_2,y_2),\cdots,(x_M,y_M)\}$, such that $x_i \in R^N$ is a vector of input variables and $y_i \in R$ the corresponding scalar output (target) value. Here, the modeling objective is to find a regression function, $y = f(x)$, such that it accurately predicts the outputs {y} corresponding to a new set of input–output examples, {(x, y)}, which are drawn from the same underlying joint probability distribution, $P(x, y)$, as the training set. To fulfill the stated goal, SVM considers the following linear estimation function:

$$f(x)=\sum_{i=1}^{D} w_i\phi_i(x)+b \qquad (1)$$

where $\{w_i\}_{i=1}^{D}$ denotes the weight vector, *b* a constant known as “bias”, $\{\phi_i(x)\}_{i=1}^{D}$ a function termed feature. In SVM, the input data vector, *x*, is mapped into a high-dimensional feature space, *F*, via a nonlinear mapping function, $\{\phi_i(x)\}_{i=1}^{D}$, and a linear regression is performed in this space for predicting y. Thus, the problem of nonlinear regression in lower-dimensional input space R^N is transformed into a linear regression in the high-dimensional feature space, *F*. Accordingly, the original optimization problem involving nonlinear regression is transformed into finding the flattest function in the feature space *F* and not in the input space, *x*. The unknown parameters $\{w_i\}_{i=1}^{D}$ and *b* in Equation (1) are estimated using the training set, *G*. To avoid over fitting and thereby improving the generalization capability, following regularized functional involving summation of the empirical risk and a complexity term $\|w\|^2$, is minimized:

$$R_{reg}=R_{emp}+\lambda\|w\|^2=\frac{1}{M}\sum_{i=1}^{M}\left|f(x_i)-y_i\right|_{\varepsilon}+\lambda\|w\|^2 \quad (2)$$

where R_{reg} and R_{emp} denote the regression and empirical risks, respectively, $\|w\|^2$ the Euclidean norm, $\left|f(x)-y\right|_{\varepsilon}$ a cost function measuring the empirical risk and $\lambda > 0$ a regularization constant. For a given function, *f*, the regression risk (test set error), R_{reg} , is the possible error committed by the function *f* in predicting the output corresponding to a new (test) example input vector drawn randomly from the same sample probability distribution, $P(x, y)$, as the training set. The empirical risk R_{emp} , represents the error (termed “training set error”) committed in predicting the outputs of the training set inputs. Minimization task described in Equation (2) involves: (i) minimization of the empirical loss function R_{emp} and, (ii) obtaining as small *w* as possible, using the training set *G*. Vapnik (1999) showed that the following function possessing finite number of parameters can minimize the regularized function in Equation (2):

$$f(x,\alpha,\alpha^*)=\sum_{i=1}^{M}(\alpha_i-\alpha_i^*)k(x_i,x)+b \quad (2)$$

with $\alpha_i\alpha_i^*=0$, $\alpha_i,\alpha_i^*\geq 0$ and the kernel function $k(x_i,x)$ describes the dot product in the D-dimensional feature space.

$$k(x_i,x_j)=\left\langle \phi(x_i),\phi(x_j)\right\rangle \qquad (3)$$

It is important to note that the features $\{\phi_i(x)\}_{i=1}^{D}$ need not be computed; rather what is needed is the kernel function that is very simple and has a known analytical form. The only condition required is that the kernel function has to satisfy Mercer’s condition. Some of the mostly used kernels include polynomial, radial basis function, and sigmoid. Note also that for Vapnik’s ε -insensitive loss function [9], the Lagrange multipliers α_i,α_i^* are sparse, i.e. they result in nonzero values after the optimization (2) only if they are on the boundary, which means that they satisfy the so-called Karush–Kuhn–Tucker conditions. The coefficients α_i,α_i^* are obtained by maximizing the following form:

$$Max.R(\alpha^*,\alpha)=-\frac{1}{2}\sum_{i,j=1}^{M}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(x_i,x_j)-\varepsilon\sum_{i=1}^{M}(\alpha_i^*+\alpha_i)+\sum_{i=1}^{M}y_i(\alpha_i^*-\alpha_i) \quad (4)$$

$$.S\ .T\ .\ \sum_{i=1}^{M}(\alpha_i^*-\alpha_i) \qquad (5)$$
$$0\leq\alpha_i^*,\alpha_i\leq C$$

Only a number of coefficients α_i,α_i^* will be different from zero, and the data points associated to them are called support vectors. Parameters *C* and ε are free and have to be decided by the user. Computing b requires a more direct use of the Karush–Kuhn–Tucker conditions that lead to the quadratic programming problems stated above. The key idea is to pick those values for a point x_k on the margin, i.e. α_k or α_k^* in the open interval (0, C). One x_k would be sufficient but for stability purposes it is recommended that one take the average over all points on the margin. Some advances and more detailed description of SVM for regression can be found in references [8, 10].

III. SVM Models Development

A. Model selection

In any predictive learning task, an appropriate representation of examples as well as the model and parameter estimation method should be selected to obtain a high level of performance of the learning machine. In fact, the task of learning amounts to selecting the model of optimal complexity and estimating parameters from training data. In our study, ν-support vector regression machines were used. Under the approach, the usually parameters to be chosen are the following:

- The penalty term C which determines the tradeoff between the complexity of the decision function and the number of training examples misclassified;
- The sparsity parameter ν in accordance with the noise that is in the output values in order to get the highest generalization accuracy.
- The kernel function such that $k(x, y)$.

In this work, radial basis function (RBF) kernel, given by Equitation 6 is used:

$$K(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2) \quad (6)$$

where σ is the width of the RBF kernel parameter.

The RBF kernel nonlinearly maps samples into a higher dimensional space, so it, unlike the linear kernel, can handle the case when the relation between inputs and outputs is nonlinear. Furthermore, the linear kernel is a special case of RBF as Keerthi and Lin [11] shows that the linear kernel with a penalty parameter *C* has the same performance as the RBF kernel with some parameters (*C*, σ). In addition, the sigmoid kernel behaves like RBF for certain parameters. The second reason using RBF kernels is the number of hyper-parameters which influences the complexity of model selection. The polynomial kernel has more hyper-parameters than the RBF kernel. Finally, for the RBF kernel, it has less numerical difficulties; and a key point is $0 < k(x, y) < 1$ in contrast to polynomial kernels of which kernel values may go to infinity or zero while the degree is large. Moreover, it is noted that the sigmoid kernel is not valid (i.e. not the inner product of two vectors) under some parameters.

B. Cross-validation

There are three key parameters need choosing while using RBF kernels to model: (ν, C and σ). Unfortunately, it is not known beforehand which (ν, C and σ) are the best for one problem. The goal is just about to identify good (ν, C, and σ) so that the model can accurately predict unknown data (i.e., testing data). A common way is to separate training data to two parts of which one is considered unknown in training the model. Then the prediction accuracy on this set can more precisely reflect the performance on predicting unknown data. The procedure for improved model is called as cross-validation. The cross-validation procedure can also prevent the over-fitting problem furthermore. In this study, the regression function was built with a given set of parameters (ν, C, and σ) tuned by genetic algorithms (GAs). The performance of the parameter set is measured by the computational risk, here mean squared error on the last subset. The above procedure is repeated p times, so that each subset is used once for testing. Averaging the mean squared error (MSE, see Equation 7) over the p trials gives an estimate of the expected generalization error for training on sets of size $\frac{p-1}{p} \cdot l$, *l* is the number of training data.

$$MSE = \frac{1}{pq} \sum_{j=1}^{p} \sum_{i=1}^{q} (y_{ti}^{(j)} - y_{pi}^{(j)})^2 \qquad (7)$$

where *q* is the sample number of tested subset in the training set; $y_{ti}^{(j)}$ and $y_{pi}^{(j)}$ are the i^{th} observed value and predicted value under j^{th} tested subset, respectively.

C. Optimal Parameters with Genetic Algorithm

Obviously, the selection of three parameters (ν, *C*, and σ) of a SVM model is very important to the accuracy of prediction. Because structural methods for determining three parameters efficiently are lacking, genetic algorithms (GAs) based on cross-validation are applied in the proposed SVM model to optimize parameters and improve generalization. Such algorithms are based on the principle of the survival of the fittest, which attempts to retain genetic information from generation to generation [12]. The major advantages of GAs are the capabilities for finding or near optimal solutions with relatively modest computational requirements. Fig.1 presents the iteration loop of the genetic algorithm, which is briefly described below.

Step1 (Initialization) Construct randomly the initial population of chromosomes.

Step2 (Evaluating fitness) Evaluate the fitness of each chromosome. In this study, the computational risk of SVM model, i.e. averaging the mean squared error (MSE, Equation 8) is used as the fitness function.

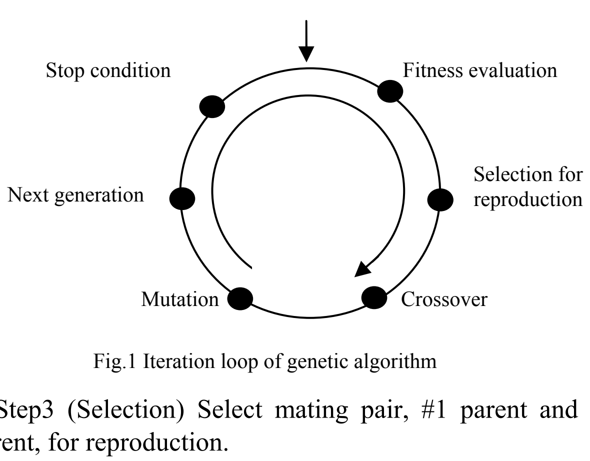


Fig.1 Iteration loop of genetic algorithm

Step3 (Selection) Select mating pair, #1 parent and 2# parent, for reproduction.

Step4 (Crossover and mutation) Create new offspring by crossover and mutation operations.

Step5 (Next generation) Form a population for the next generation.

Step6 (Stop condition) If the number of generation is equal to a given scale, or the value of fitness function is not changed for a long time, then the best chromosomes are presented as a solution, otherwise go back to step2.

In the proposed GA-SVM model, the SVM parameters are dynamically optimized by implementing the real-valued GA evolutionary process and the SVM model then performs the prediction task using these optimal values. Namely, the real-valued GA tries to search the optimal values to enable SVM to fit various datasets. The optimal values of SVM's parameters are searching by GAs with a randomly generated initial populations consisting of chromosomes. The values of the three parameters (ν, C, and σ), are directly coded in the chromosomes with real-valued data. The proposed model can implement either the roulette-wheel method or the tournament method for selecting chromosomes. Adewuya's crossover method and boundary mutation method were used to modify the chromosome. The single best chromosome in each generation is survives to the succeeding generation. The proposed model was developed and implemented in the MATLAB v7.1 environment. The major tool for training and validating the SVM were those developed by Lin et al. [11]. The proposed model is able to handle huge data sets and easily be combined with the real-valued genetic algorithm in the MATLAB environment.

IV. THE EXPERIMENTAL STUDY

To demonstrate the generalization performance of SVM model used in yarn processing, the relevant experiments were completed, and comparisons with ANN models were given in our work. Artificial neural network (ANN) is also a powerful data modeling tool simulating the behavior of biological human neurons. The most representative back propagation algorithm in ANN modeling was employed herein. One hidden layer was used in the ANN models (for most cases, it was sufficient). The relationship between inputs and outputs of ANN model was completely same as the SVM model. The ANN algorithms and detail procedures of ANN modeling can refer to the report [13].

A. Data acquisition and preprocessing

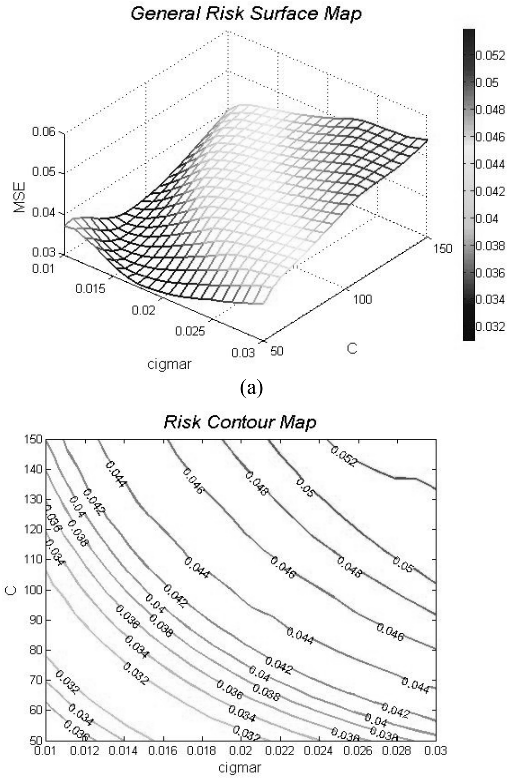
In this work, a small data set (a total of twenty-six different data samples) from real industrial process (worsted spinning) was acquired. To make problem more simply, like most ANN models[7], some fiber properties and processing information were selected as the SVM model's inputs, which were mean fiber diameter (MFD, μm), diameter distribute (CVD, %), hauteur (HT, mm), fiber length distribution (CVH, %), short fiber content (SFC, %), yarn count (CT, tex), twist (TW, t.p.m), draft ratio (DR), spinning speed (SS, r.p.m), traveler number (TN). The two most important quality characteristics of the resulting yarn, namely elongation at break (EB, %) and break force (BF, cN), served as the SVM model's outputs. Before training and modeling, the data in the process database had to be normalized so that they were bounded within the prescribed range of 1 and 0. Scaling of the original value (v_i) from the process database was carried out according to Equation8.

$$x_i = \frac{v_i - \min(v_{1,\cdots,n})}{\max(v_{1,\cdots,n}) - \min(v_{1,\cdots,n})} \qquad (8)$$

where x_i is the scaled value, and $\min(v_{1,\cdots,n})$ and $\max(v_{1,\cdots,n})$ are the respective maximum and minimum values within each original data array.

B. Model training

After selecting a suitable kernel function in SVM model, one of the primary aspects of training a SVM regression model is the selection of the sparsity parameter ν, the penalty term *C* and the width of the RBF kernel parameter σ. To search the three parameters adapting to our problems, the Genetic Algorithm Tool from the Matlab 7.1 software has been used to realize the optimization of three parameters (ν, *C*, and σ) of the SVM models. Besides, in the our study an SVM-implementation known as "ν-SVM" in the LIBSVM 2.8 software library has also been used to train the two SVM-based models. The LIBSVM package utilizes a fast and efficient method known as sequential minimal optimization (SMO) for solving large quadratic programming problems and thereby estimating function parameters α , α^* , and *b* (see Eq.(2)). Table1 lists the final searching area and optimal values of the four SVM models, respectively. Before optimization parameters with genetic algorithm, "grid-search" method is used to explore the possible area of minimal risk in order to prevent the problem of local minimization in genetic algorithm. The "grid-search" experiments are illustrated as 3D surface plots and the

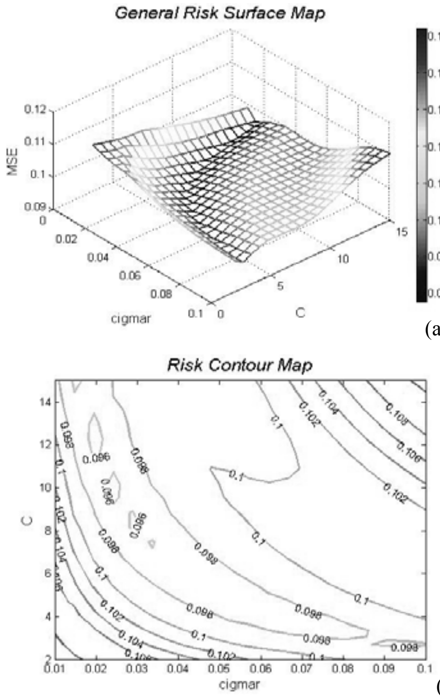


(a)

(b)

Fig.2 3D surface plot (a) and contour map
(b) for the predictive risk of BF (ν=0.54)

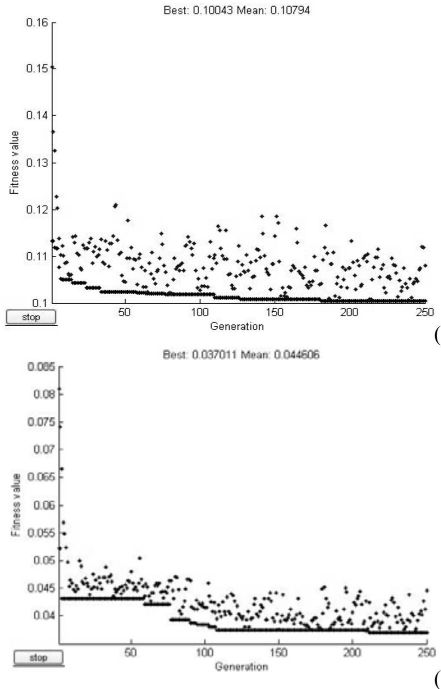
corresponding contour map for the computational risk (MSE) (see Fig.2 ~Fig.3). Usually, the "grid-search" need an iterative process to estimate the rough area of minimal risk, and then GAs are used to optimize the model parameters more accurately. The results of the model training based on GAs are shown in Fig.4.



(a)

(b)

Fig.3 3D surface plot (a) and contour map (b) for the predictive risk of EB (ν=0.54)



(a)

(b)

Fig.4 Plots of the best and mean values of the fitness function at each generation in SVM Models: (a) EB and (b)

C. *Experimental results*

After the completion of model development or training, the models based on SVM (and ANN) were both subjected to the unseen testing data set. Statistical parameters such as the correlation coefficient (R) between the actual and predicted values, mean squared error, and mean error (%), were used to compare the predictive power of the SVM and ANN models. Experimental results are shown in Table2. It has been observed that for ANN models, the mean error (%) of two predictive indices (EB and BF) is 22.80% and 13.67%, and especially, the correlation coefficient (R) of the EB is very low, shown as only 0.58. However, for SVM models, the mean error (%) of two predictive indices (EB and BF) is 12.71% and 5.52%, and the correlation coefficient (R) of EB and BF is improved to 0.87 and 0.99. On the other hand, the cases with over 10% error also decrease from 4 and 3 in ANN models to 2 and 1 in SVM models respectively

V. CONCLUSIONS

Support vector machines are a learning-by-example paradigm with many potential applications in science and engineering. The salient features of SVM include the absence of local minima, the sparseness of the solution and the improved generalization. SVMs being a relatively new technique, their application on textile production have hitherto been quite limited. However, the elegance of the formalism involved and their successful use in diverse science and engineering applications confirm the expectations raised in this appealing learning from examples approach. In our study, the SVM models for predicting the yarn properties are presented and compared with the BP neural network models. The important parameters of SVM models to improve the model generalization are optimized using genetic algorithms. From the experiments, it has been found that the SVM model is able to predict to a reasonably good accuracy in most of cases. And a more interested phenomenon is that in small data set and real-life production, the predictive power of ANN models appears to decrease, while SVM models are still capable of remaining the stability of predictive accuracy to some extent. The experimental results indicate that SVM models are more suitable for noisy and dynamic spinning process. Of course, like other emerging industrial techniques, applied issues on SVM reaffirm the due commitment to their further development and investigation, such as the problems how to design the kernel function and how to set the SVM hyper-parameters (to make the industrial model development more easily). This study thus far has demonstrated that SVMs are able to provide an alternative solution for the spinners to predict yarn properties more correctly and reliably. With advancement of artificial intelligent techniques, one looks forward to the era in which much of the subjective decision-making in textile production will finally be replaced by more intelligent systems.

TABLE1
THE SEARCH AREA AND OPTIMAL VALUES OF ν, σ AND C

Output parameter	Searching area	Optimal value
Breaking force (cN)	$\nu\in[0.3, 0.6]$ $\sigma\in[0.01, 0.03]$ $C\in[50, 150]$	ν =0.31 σ=0.013 C=101.7
Elongation at break (%)	$\nu\in[0.3, 0.6]$ $\sigma\in[0.\ 01, 0.1]$ $C\in[1, 15]$	ν =0.54 σ=0.06 C=1.86

TABLE2
THE PREDICTIVE POWER OF THE SVM AND ANN MODELS

Sample No	Predicted values			
	ANN Model		SVM Model	
	BF	EB	BF	EB
W21	113.89	13.57	116.24	13.40
W22	61.91	14.56	76.87	12.94
W23	153.46	5.32	156.57	15.55
W24	61.91	14.56	76.87	12.94
W25	47.00	9.99	76.86	12.88
W26	66.76	13.00	66.62	13.03
Correlation coefficient (R)	0.96	0.58	0.99	0.87
Mean squared error	165.91	18.91	42.12	2.70
Mean error%	13.67	22.80	5.52	12.71
Cases with over 10% error	3	4	1	2

ACKNOWLEDGMENT

This work was financially supported by the Technology innovation Program of Science and Technology Department of China (2009GJC00026).

REFERENCES

[1] Peter R. Lord, *Handbook of Yarn Production (Technology, Science and Economics)*, Woodhead publishing Limited, Abinhton England,2003,ch.1

[2] Les M.Sztanera, Christopher Pastore, *Soft computing in textile sciences,* Physica-Verlag Heidelberg, New York, 2003,ch.1~4

[3] Bogdan. J. F., "The Characterization of Spinning Quality", *Textile Res. J*, 1956, vol.26(9),pp.720-730

[4] Aggarwal, S.K., "A model to estimate the breaking elongation of high twist ring spun cotton yarns. Part I: Derivation of the model for yarns from single cotton varieties", *Textile Res.J,* 1989, vol.59(12), pp.691-698

[5] Savita Garg, Jai Singh, D.V. Singh, "Mathematical modeling and performance analysis of combed yarn production system: Based on few data", *Applied Mathematical Modeling,* Vol34(11), 2010, pp.3300-3308

[6] Gui Liu, Weidong Yu, "Smart case-based indexing in worsted roving process: Combination of rough set and case-based reasoning". *Applied Mathematics and Computation,* Vol214(1), 2009, pp.280-286

[7] Xianggang Yin, Weidong Yu, "The virtual manufacturing model of the worsted yarn based on artificial neural networks and grey theory". *Applied Mathematics and Computation*, vol 185(1), 2007, pp. 322–332

[8] V. David Sánchez A, "Advanced Support Vector Machines and Kernel Methods. Neurocomputing". 2003, vol55(3),pp.5-20

[9] V. N. Vapnik, Statistical Learning Theory. 2nd ed., 1999, Wiley, New York, ch1~5

[10] Pao-Hua Chou, Menq-Jiun Wu, Kuang-Ku Chen, "Integrating support vector machine and genetic algorithm to implement dynamic wafer quality prediction system", *Expert Systems with Applications*, Vol37(6), 2010, pp. 4413-4424

[11] Keerthi, S.S. and C.J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel", *Neural Computation*, vol15(7), 2003, pp.1667-1689

[12] Michalewicz, Z, *Genetic Algorithms+Data Structures=Evolution Programs*, third ed. Springer, New York,1999, ch.1~4

[13] Chattonpadhyay R. and Guha A., "Artificial Neural Networks: Applications to Textiles", *Textile Progress*, 2004, vol35(1),pp.1~42

.