

Determining Parameters of Support Vector Machines by Genetic Algorithms—Applications to Reliability Prediction

Ping-Feng Pai^{1,*}, Wei-Chiang Hong², and Yu-Shen Lee¹

¹ Department of Industrial Engineering and Technology Management, Da-Yeh University, 112 Shan-Jiau Road, Da-Tsuen, Changhua, 51505, Taiwan

² School of Management, Da-Yeh University, 112 Shan-Jiau Road, Da-Tsuen, Changhua, 51505, Taiwan

Received August 2004; Revised September 2004; Accepted October 2004

Abstract—Due to the lack of a structure way in determining the free parameters of support vector machines (SVMs), this study uses genetic algorithms (GAs) to select parameters of SVMs. In addition, the developed SVMG (support vector machine with genetic algorithms) model is applied to reliability prediction. Two numerical examples in the literature are employed to illustrate the performances of various prediction models. Empirical results reveal that the proposed model provides more accurate prediction results than the other forecasting models. Therefore, the presented SVMG model offers a promising alternative in reliability prediction.

Keywords—Support vector machines (SVMs), Genetic algorithms (GAs), Prediction, Reliability

1. INTRODUCTION

Modeling and forecasting of reliability had received increasing attentions in the field of production and operations management systems. In general, system reliability changes with time. Therefore, the reliability changes could be viewed as a time series process. Predicting the variability of reliability with time is difficult. The difficulty arises from assumptions concerning failure distributions and a lack of appropriate reliability models. The methods for forecasting reliability include lifetime distribution, Markov models, part count and part stress, and fault tree analysis. Traditional reliability prediction is usually based on some probability distributions of time to failure. The probability distributions are mostly obtained through analysis of testing data, which is sampled from testing population. However, these works are limited in some variations of individual systems under dynamic operating conditions, such as life time of systems, features of production, and operation conditions. Therefore, the determination of the parameters in a forecasting model is important to the forecasting performance.

Recently, Support vector machines (SVMs) have been developed for solving pattern recognition and nonlinear regression estimation problems. The SVM model is able to increase forecasting accuracy by selecting suitable parameter. Therefore, constructing an adapted procedure to select suitable parameters is an essential task. In this study, a SVM model with genetic algorithms (GAs) is proposed to examine the feasibility of predicting system reliability. The rest of this paper is organized as follows. Literatures of reliability prediction are reviewed in section

two. The proposed SVMG model is introduced in section three. Two numerical examples are used in section four to demonstrate the forecasting performance of the proposed model. Finally, conclusions are presented in the fifth section.

2. RELIABILITY PREDICTION

Duane model is one of the most popular models in reliability growth prediction (Duane, 1964). This approach is a non-homogeneous Poisson process (NHPS) and expresses a relationship between the expected cumulative Mean Time Between Failure (MTBF, θ) and the cumulative test time (t). The slope of the fitted line is estimated by the historical reliability data. Ho and Xie (1998) proposed an ARIMA model to analyze the failures of repairable systems. The experimental results indicated that the proposed model substantially outperform the Duane model in terms of MAD (mean absolute deviation). Ho et al. (2002) conducted a comparative analysis of neural networks and ARIMA techniques to forecast the reliability of repairable systems. Their experimental results demonstrated that both recurrent neural networks and the ARIMA approach are superior to multi-layer forward neural networks in terms of the mean square error (MSE) and MAD. Su et al. (1997) combined ARIMA models with neural network techniques to predict engine reliability. They reported that the proposed model results in more accurate forecasting results than ARIMA models and BPNN (back-propagation neural networks) models. Sitte (1999) applied neural network technology to predict software-reliability-growth, two neural networks, namely the fee-forward-network and the

* Corresponding author's email: paipf@ncnu.edu.tw

Elman-recurrent-network were used in this investigation. The input is the normalized failure-occurrence time; the output is the accumulated failure number. The empirical results indicate that these two neural networks outperform the parametric-recalibration models in terms of forecasting accuracy. Cai et al. (2001) applied a multi-layer perceptron (MLP) neural network to forecast software reliability. The authors claimed that proposed models are more suitable for a smooth reliability data set than a fluctuating one. Xu et al. (2003) applied feed-forward MLP neural networks and radial basis function (RBF) neural networks to forecast engine reliability. The sensitivity analysis is employed to determine the appropriate architectures of neural networks. The experimental results illustrate that the proposed model has better forecasting performance than traditional MLP model and the ARIMA model.

SVMs, proposed by Vapnik(1995), is a novel learning technique applying statistical learning theory (SLT) to structural risk minimization (SRM). SVMs were originally developed to solve pattern recognition and classification problems. Recently, with the introduction of Vapnik's ε -insensitive loss function, SVMs have been extended to solve forecasting problems in many fields. Lu et al. (2002) applied SVM techniques to forecast air quality parameters. The empirical results indicate that SVMs outperformed the RBF model in terms of MAD. Trafalis and Ince (2000) proposed a SVM model to predict the stock prices. The numerical results indicate that the proposed model provide better forecasting results than back-propagation and RBF neural networks. Tay & Cao (2001) used SVMs in forecasting financial time series. Their numerical results show that SVMs are superior to the multi-layer back-propagation neural network in financial time series forecasting. Mohandes et al. (2004) applied SVMs to predict wind speed. Their experimental results indicate that the SVM model has more accurate forecasting results than MLP neural networks. Pai and Lin (2004a) employed SVMs to forecast the production values of the machinery industry in Taiwan. They reported that SVMs outperform seasonal ARIMA model and general regression neural networks model (GRNN). Pai and Lin (2004b) proposed a hybrid model with the strength of an ARIMA model and the SVM model in forecasting the stock prices. By using ten stocks to examine the performance, numerical results show that the proposed hybrid model outperforms the ARIMA model and random walk model. Hong and Pai (2004) applied a SVM model in forecasting electric load in Taiwan. They found that the proposed model outperform the ARIMA model and the GRNN model in terms of forecasting accuracy.

3. METHODOLOGY

3.1 Support vector machines model

The basic concept of the SVM regression is to map nonlinearly the original data x into a higher dimensional feature space. Hence, given a set of data $G = \{(x_i, d_i)\}_{i=1}^n$ (where x_i is the input vector; d_i is the historical actual value,

and n is the total number of data patterns), the SVM regression function is

$$y = f(x) = w\phi(x) + b \quad (1)$$

where $\phi(x)$ is the feature of inputs, and both w and b are coefficients.

The coefficients (w and b) are estimated by minimizing the following regularized risk function,

$$R_{\text{SVM}}(C) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^n |d_i - y_i|_{\varepsilon}\right) \quad (2)$$

Eq.(2) implies the SVM processes employed the concepts of ε -insensitivity loss function (namely empirical risk, shown in Eq. (3)) to represent the risk minimization in a regression model.

$$R_{\text{emp}}^{\varepsilon}(w, b_i) = \frac{1}{n} \sum_{i=1}^n |d_i - w_i\phi(x_i) - b_i|_{\varepsilon} \quad (3)$$

Therefore, the risk minimization problem should be focused on minimizing Eq.(3) and $\|w\|^2$. The ε -insensitivity loss function is defined as follows, the loss equals zero if the forecasted value is within the ε -tube, else, the loss equals $|d - y| - \varepsilon$ (Eq. (4)).

$$|d - y|_{\varepsilon} = \begin{cases} 0 & , \text{ if } |y - d| \leq \varepsilon \\ |d - y| - \varepsilon & , \text{ otherwise} \end{cases} \quad (4)$$

Meanwhile, in Eq. (2), the first term, $\frac{1}{2}\|w\|^2$, is called regularized term, measures the flatness of the function; the second term, $C\left(\sum_{i=1}^n |d_i - y_i|_{\varepsilon}\right)$, is called empirical error; C is considered to specify the trade-off between the empirical risk and the model flatness. Both C and ε are user-determined parameters. There is lack of standard procedure to determine the values of ε and C to be suited to all kinds of forecasting fields.

Two positive slack variables ζ and ζ^* , which represent the distance from actual values to the corresponding boundary values of ε -tube, are introduced. Then, Eq. (2) is transformed into the following constrained form,

$$\text{Minimize } R_{\text{SVM}}(w, \zeta, \zeta^*) = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^n (\zeta_i + \zeta_i^*) \quad (5)$$

with the constraints, $d_i - w_i\phi(x_i) - b_i \leq \varepsilon + \zeta_i$,
 $w_i\phi(x_i) + b_i - d_i \leq \varepsilon + \zeta_i^*$,
 $\zeta_i, \zeta_i^* \geq 0$.

This constrained optimization problem is solved using the

following primal Lagrangian form:

$$L(w_i, b_i, \zeta, \zeta^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*) = \frac{1}{2} \|w\|^2 + C \left(\sum_{i=1}^n (\zeta_i + \zeta_i^*) \right) - \sum_{i=1}^n \alpha_i [w_i \phi(x_i) + b_i - d_i + \varepsilon + \zeta_i] - \sum_{i=1}^n \alpha_i^* [d_i - w_i \phi(x_i) - b_i + \varepsilon + \zeta_i^*] - \sum_{i=1}^n (\beta_i \zeta_i + \beta_i^* \zeta_i^*) \quad (6)$$

Eq. (6) is minimized with respect to primal variables w , b , ζ and ζ^* , and maximized with respect to nonnegative Lagrangian multipliers α_i , α_i^* , β_i and β_i^* . Finally, Karush-Kuhn-Tucker conditions are applied to the regression, and Eq. (5) thus yields the dual Lagrangian,

Maximize

$$R(\alpha, \alpha^*) = -\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n d_i (\alpha_i - \alpha_i^*) \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \quad (7)$$

with the constraints, $0 \leq \alpha_i \leq C$

$$0 \leq \alpha_i^* \leq C, \quad i = 1, 2, \dots, n$$

The Lagrange multipliers in Eq. (6) satisfy the equality $\alpha_i \alpha_i^* = 0$. The Lagrange multipliers α_i and α_i^* , are calculated and an optimal desired weight vector of the regression hyperplane is shown in Eq. (8), bias term is shown in Eq. (9), SVM regression function is as Eq. (10),

$$w_o = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x_i, x_j) \quad (8)$$

$$b_o = \frac{1}{n} \left(\sum_{i=1}^n (d_i - w_o^T \phi(x)) \right) \quad (9)$$

$$f(x, \alpha, \alpha^*) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x, x_i) + b \quad (10)$$

Here, $K(x_i, x_j)$ is called the Kernel function, which transferred the input space into feature space. The value of the Kernel equals the inner product of two vectors, x_i and x_j in the feature space $\phi(x_i)$ and $\phi(x_j)$, that is $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. There are three types of common examples of kernel function, the polynomial

kernel, $K(x_i, x_j) = (a_1 x_i^T x_j + a_2)^d$ (with degree d , a_1 and a_2 represent the coefficients); the Gaussian RBF kernel function, $K(x_i, x_j) = \exp\left(-\frac{1}{2} * \left(\frac{\|x_i - x_j\|}{\sigma}\right)^2\right)$; and the

multi-layer perceptron kernel function, $K(x_i, x_j) = \tanh(x_i^T x_j - b)$ (where b is the constant). Till now, it is hard to determine the type of kernel functions for specific data patterns (Amari and Wu (1999), Vojislav(2001)). Smola et al. (1998) reported that the Gaussian RBF kernel function is suitable for the data pattern with smoothness and independence. Therefore, the Gaussian RBF kernel function is specified in this study

The selection of three parameters, σ , ε and C , of a SVM model is important to the accuracy of forecasting. However, structural methods for determining parameters efficiently are lacking. The traditional procedure for selecting three parameters is expressed as follows and shown as Figure 1.

Step 1. Set fixed values of the parameters ε and C . Then, adjust the value of σ till a minimum testing error is achieved. The finalized σ value is denoted as σ' .

Step 2. Set a fixed value of the parameter ε and the value of σ is set to σ' . Then, adjust the value of C to achieve a minimum testing error. The finalized C is defined as C' .

Step 3. Values of σ and C are set to σ' and C' . Then, adjust ε till a minimum testing error is obtained. The finalized ε is defined as ε' . Therefore, values of σ , ε and C are obtained as σ' , ε' , and C' .

The traditional way presented in Figure 1 is not a structural way in determining SVMs parameters. Therefore, GAs are applied in the proposed SVMG model to select parameters.

3.2 Genetic algorithms in selecting SVM parameters

Holland (1975) first proposed genetic algorithms. The algorithms are based on the survival principle of the fittest member in a population, which retains genetic information by passing it from generation to generation. The process of a GA is described as follows.

Step 1. (Initialization): Establish a random initial population of chromosomes.

Step 2. (Evaluating fitness): Evaluate the fitness of each chromosome. In this investigation, two indices are used as the fitness function. The negative root mean square error (-RMSE) is for the first numerical example; the negative mean absolute deviation (-MAD) is for the second example. The formulas of these fitness functions are indicated as Eq. (11) and Eq. (12), respectively.

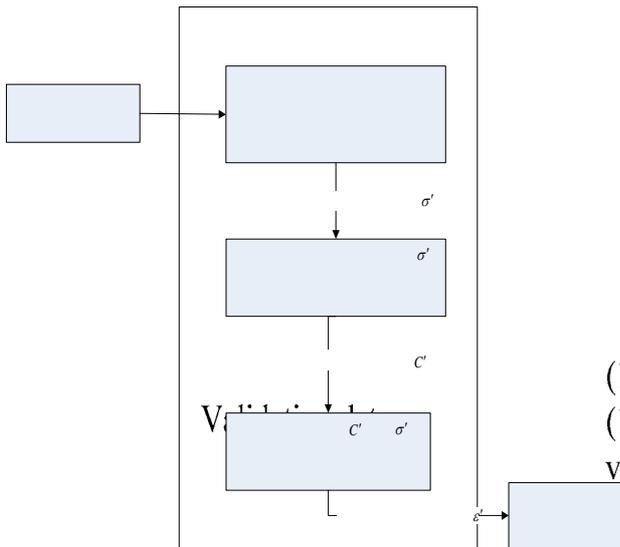


Figure 1. The traditional selecting processes of three parameters in SVM model.

$$\text{Fitness function(example 1)} = -\sqrt{\frac{\sum_{i=1}^n (d_i - f_i)^2}{n}} \quad (11)$$

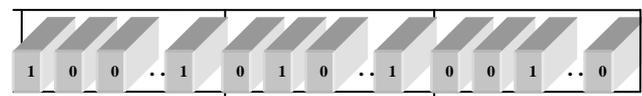
$$\text{Fitness function(example 2)} = -\frac{\sum_{i=1}^n |d_i - f_i|}{n} \quad (12)$$

where d_i and f_i represent the actual and forecast values, and n is the number of forecasting periods.

- Step 3. (Selection): Select the mating pair, #1 parent and #2 parent, for reproduction.
- Step 4. (Crossover and mutation): Create new offspring by performing crossover and mutation operations. The crossover rate is often set between 0.5 and 1. The mutation process is then employed to avoid local optimum problem, the mutation date is empirically set between 0.01 and 0.08 (Greenwell et al. (1995), and Wang(1997)).
- Step 5. (Next generation): Form a population for the next generation.
- Step 6. (Stop conditions): If the number of generations exceeds a given threshold, then the best chromosomes are presented as a solution; otherwise go to Step 2.

Three free parameters, σ , C and ϵ are represented by a chromosome that consists of three genes in the form of binary numbers (Figure 2). The size of the population is set to 200 herein. Each gene contains 40 bits. If each gene contains 40 bits, for example, then a chromosome contains 120 bits. More bits in a gene correspond to a finer partition of the searched space. Parent selection is a procedure in which two chromosomes from the parent population are chosen according to the fitness functions. Fitter chromosomes are more likely to generate offspring to the next generation. The roulette wheel selection principle

(Holland, 1975) is used to select chromosomes for reproduction. In the crossover operation, chromosomes are paired randomly. The single-point-crossover principle is employed in this investigation. Segments of paired chromosomes between two determined break-points are swapped.



- (1.a) Set fixed values of ϵ and C .
- (1.b) Adjust σ to obtain minimum validation error

Figure 2. The binary coding of a chromosome.

For simplicity, suppose a gene has four bits. A chromosome contains 12 bits (Figure 3). Before crossover is performed, the values of the three parameters in #1 finalized σ value is denoted as 0.09375. For #2 parent, the three values are 0.375, 8.75 and 0.1875. After crossover, for #1 offspring, the three values are 1.375, 3.75 and 0.1875. For #2 offspring, the three values are 0.125, 8.125 and 0.09375. Mutations are performed randomly by converting a 0 bit into a 1 bit or a 1 bit into a “0” bit. The rates of crossover and mutation are probabilistically determined.

Due to calculating forecasting errors is time consuming, it is necessary to set the boundaries of the three parameters. Then, the total historical actual data is divided into two finalized C value is denoted as

- (2.a) Set fixed values of ϵ and C .
- (2.b) Adjust σ to obtain minimum validation error
- (3.a) Fix values of σ and C .
- (3.b) Adjust ϵ to obtain minimum validation error

4. NUMERICAL EXAMPLES

Two numerical examples are employed to demonstrate the forecasting performance of the SVMG model. Reliability data for periodic vehicle repaired, from Su et al. (1997), are used in the first example. The data include the number of instances of vehicle damage repaired, and the period reliability ratio. Totally, there are 36 data in this example. These data are divided into training data set and testing data set. The number of data in the training data set and the testing data set are 24 and 12, respectively. To compare the forecasting performance of the proposed model with the models of Su et al. (1997), the data division principle is the same as that used by Su et al. (1997). Meanwhile, this study uses the RMSE (Eq. (13)) to measure the forecasting accuracy.

In the second example, the machine failure data used by Ho et al. (1998) are specified to investigate the performance of the proposed model in forecasting machine reliability. The data include the number of machine failure. Totally, 15 data are available. the data are

Values of σ , ϵ and C are obtained

divided into two sets: the training data set and the testing data set. The number of data in the training data set and the testing data set are 10 and 5, respectively. Similarly, the data division principle is the same as that used by Ho et al. (1998), the index of forecasting accuracy is MAD (Eq.(14)).

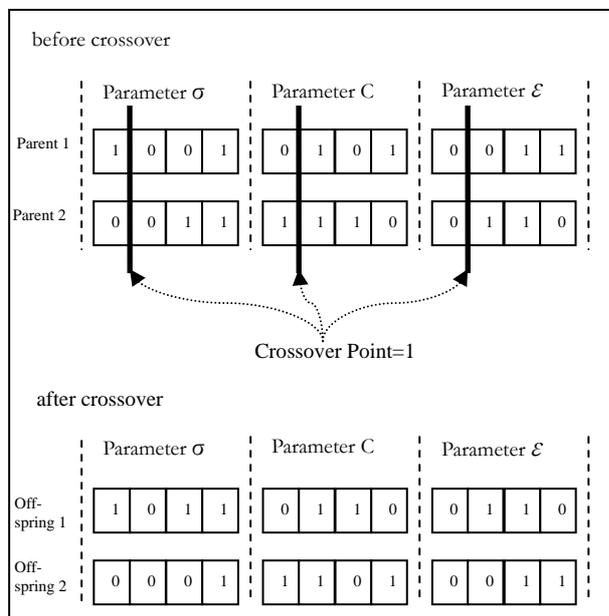


Figure 3. A simplified example of parameter representation.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (d_i - f_i)^2}{n}} \quad (13)$$

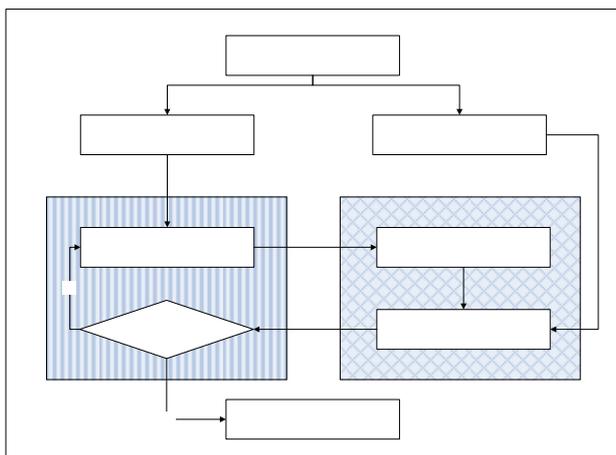


Figure 4. The architecture of a SVMG model.

$$MAD = \frac{\sum_{i=1}^n |d_i - f_i|}{n} \quad (14)$$

where d_i and f_i represent the actual and forecast values, and n is the number of forecasting periods.

4.1 Example 1

To demonstrate the improvement of GAs in selecting parameters of the SVM model, the traditional selecting procedure is applied to deal with the example. First, set the values of the parameters ϵ and C fixed at 0 and 10 respectively. Then, adjust the value of σ till a minimum testing error (RMSE) is reached. In Figure 5, the finalized σ value equaled to 87, the minimum value of RMSE is achieved. Therefore, set finalized σ value as 87. Secondly, fix the values of the parameters σ and ϵ at 87 and 0 respectively. Then, adjust the value of C (Figure 6). The finalized value of C is 2. The minimum value of RMSE is obtained. Therefore, the finalized C value is 2. Finally, set the values of the parameters σ and C fixed at 87 and 2 respectively. Then, adjust the value of ϵ (Figure 7). The ϵ value equals to 0.008 when the minimum value of RMSE is achieved. Therefore, the values of three parameters (σ , C , and ϵ) are 87, 2, 0.008 correspondingly. The minimum value of RMSE is 0.002345.

Then, GAs are used to determine the values of the three parameters (σ , C , and ϵ) in the SVM model (Figure 4). The intervals of three parameters are $[0, 250]$, $[0, 1000]$, and $[0, 0.1]$ respectively. The number of bits is set to 120, crossover rate is 0.5, mutation rate is 0.05. A rolling-based forecasting procedure is conducted and a one-step-ahead forecasting policy is adopted. In the training stage, several types of data-rolling are applied. Different number of reliability data is fed into the proposed model to forecast reliability in the next period. These data-rolling types include “5-input,” “10-input,” “15-input,” “19-input,” “21-input” and “22-input.” The model with the minimum testing RMSE value is selected as the most appropriate model for this example. Table 1 presents the forecast results and the parameters of the proposed models. It is shown that the SVMG model has the best performance when 21 input data are used. The RMSE value is 0.002297. Therefore, SVMG model is superior to the traditional method (RMSE=0.002345) in forecasting accuracy. Additionally, Table 2 lists the forecast results of the best SVMG model and three other models (including ARIMA, BPNN · ICBPNN3) proposed by Su et al. (1997). The proposed SVMG model yields the best forecast results in terms of the RMSE.

4.2 Example 2

In example two, GAs are employed to determine the values of the three parameters (σ , C , and ϵ) in the SVM model. The boundaries of three parameters, the number of bits, crossover rate, and mutation rate, are exactly the same as example one. In the training stage, several types of data-rolling are used. These data-rolling types include “2-input,” “4-input,” “6-input,” and “8-input.” The model with the minimum testing MAD value is selected as the most appropriate model for use with this example. Table 3 presents the forecast results and the parameters of the proposed models. It is indicated that the SVMG model has the best performance when 8-input data are used. Table 4 presents the forecast results of the best SVMG model and

two other models (ARIMA and Duane models) proposed by Ho et al. (1998). The proposed SVMG model yields the best forecast results in terms of the MAD.

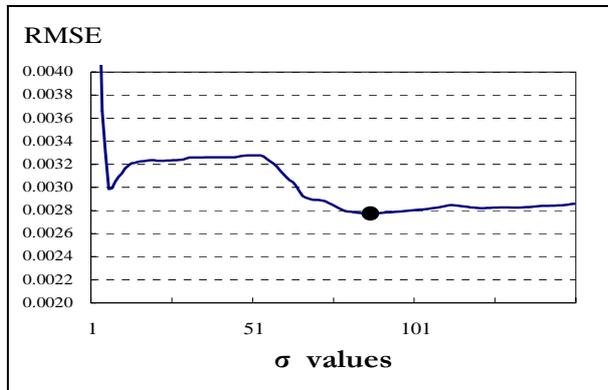


Figure 5. MSE values with respect to σ values.

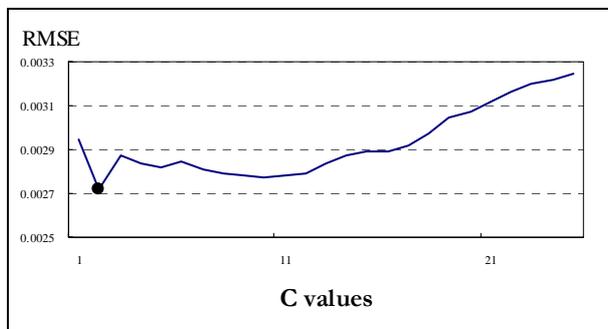


Figure 6. MSE values with respect to C value.

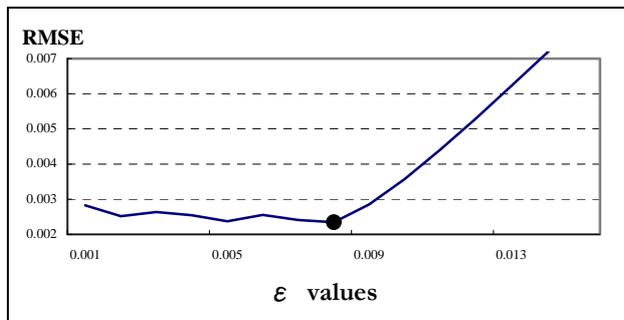


Figure 7. MSE values with respect to ϵ value.

Table 1. Forecasting results of example 1

Number of input data	Parameters			Testing RMSE
	σ	C	ϵ	
5	106.48	638.44	0.00361	0.002510
10	46.933	152.78	0.00345	0.002932
15	58.902	294.69	0.00631	0.002526
19	45.658	289.73	0.00811	0.002387
20	236.58	682.33	0.00386	0.002531
21	208.55	183.48	0.00444	0.002297
22	167.91	466.07	0.00517	0.002358

Table 2. Forecasting results of SVMG model and other models

(example 1)

Models	Actual	ARIMA	BPNN	ICBPNN3	SVMs*	SVMG
		(absolute error)				
Month						
25	0.010023	0.003266	0.005366	0.003925	0.001962	0.000947
26	0.008911	0.001290	0.001392	0.000386	0.000855	0.000398
27	0.011470	0.001950	0.000514	0.000690	0.003421	0.001928
28	0.007078	0.001292	0.003462	0.001958	0.000971	0.002698
29	0.011408	0.003550	0.001460	0.002500	0.003359	0.001693
30	0.006335	0.002022	0.004334	0.002090	0.001714	0.003560
31	0.010884	0.003760	0.002070	0.003160	0.002835	0.001085
32	0.010689	0.001911	0.004014	0.001625	0.002640	0.001562
33	0.005024	0.004420	0.003230	0.004310	0.003025	0.004121
34	0.006509	0.001390	0.000351	0.001950	0.001539	0.002941
35	0.007277	0.000130	0.000778	0.000720	0.000772	0.002305
36	0.010774	0.003410	0.003788	0.002822	0.002726	0.001073
RMSE		0.002668	0.003024	0.002484	0.002345	0.002297

*: Three parameters of SVM model are obtained by traditional selecting method.

Table 3. Forecasting results of example 2

Number of input data	Parameters			Testing MAD
	σ	C	ϵ	
2	4.3656	60.1005	0.8681	3.6
4	1.1297	9.9897	0.2461	3.4
6	2.7660	88.6485	2.9855	3.3
8	0.9811	48.1709	3.7940	3.0

Table 4. Forecasting results of SVMG model and other models (example 2)

Models	Actual	ARIMA	Duane	SVMG
		(absolute error)	(absolute error)	(absolute error)
Time space				
11	15	0.9	18.9	1.6
12	18	4.9	29.6	3.2
13	15	2.8	45.2	0.6
14	7	4.3	70.8	4.2
15	3	7.4	102.3	5.4
MAD		4.1	53.4	3.0

5. CONCLUSIONS

Reliability forecasting accuracy plays an important role in manufacturing systems. The proposed SVMG model yields lower forecasting errors than other forecasting models in both numerical examples. The superior performance of the SVMG model over other approaches is raised from the following reasons. First, with the nonlinear mapping capabilities, the SVMG model more easily captures data patterns of system reliability than the other approaches. Secondly, the GAs properly select free parameters in the proposed SVMG models to predict system reliability. Moreover, the SVMG model minimizes structural risk, instead of minimizing training errors. This process seeking to minimize an upper bound of the generalization error improves generalization performance. Therefore, SVMG model can outperform traditional SVM model. For future

research, some other search techniques can be considered to select free parameters in the SVM model to improve the prediction performance.

ACKNOWLEDGEMENTS

This research was conducted with the support of National Science Council (NSC 92-2213-E-212-001 & NSC 93-2213-E-212-001)

REFERENCES

1. Amari, S. and Wu, S. (1999). Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12:783-789.
2. Cai, K.Y., Cai, L., Wang, W.D., Yu, Z.Y. and Zhang, D. (2001). On the neural network approach in software reliability modeling. *The Journal of System and Software*, 58: 47-62.
3. Duane, J.T. (1964). Learning curve approach to reliability monitoring. *IEEE Transaction on Aerospace*, AS-2: 563-566.
4. Greenwell, R.N., Angus, J.E. and Finck, M. (1995). Optimal mutation probability for genetic algorithms. *Mathematical and Computer Modelling*, 21: 1-11.
5. Ho, S.L. and Xie, M. (1998). The Use of ARIMA Models for Reliability Forecasting and Analysis. *Computers & Industrial Engineering*, 35: 213-216.
6. Ho, S.L., Xie, M. and Goh, T.N. (2002). A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction. *Computers & Industrial Engineering*, 42: 371-375.
7. Holland, J. (1975). *Adaptation in Natural and Artificial System*. University of Michigan Press, Ann Arbor.
8. Hong, W.C. and Pai, P.F. (Sep., 2004). Support vector machines with genetic algorithms in forecasting electricity load. *Proceeding of Joint 2nd International Conference on Soft Computing and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems*, Yokohama, Japan.
9. Lu, W., Wang, W., Leung, A.Y.T., Lo, S.M., Yuen, R.K.K., Xu, Z. and Fan, H. (2002). Air Pollutant Parameter Forecasting Using Support Vector Machines. *Proceedings of the 2002 International Joint Conference on Neural Networks*, IEEE Service Center, 1: pp. 630-635.
10. Mohandes, M.A., Halawani, T.O., Rehman, S. and Hussain, A.A. (2004). Support vector machines for wind speed prediction. *Renewable Energy*, 29: 939-947.
11. Pai, P.F. and Lin, C.S. (2004a). Using support vector machines in forecasting production values of machinery industry in Taiwan. *International Journal of Advanced Manufacturing Technology*, accepted for publication.
12. Pai, P.F. and Lin, C.S. (2004b). A hybrid ARIMA and support vector machines model in stock price forecasting. *OMEGA-International Journal of Management Science*, accepted for publication.
13. Sitte, R. (1999). Comparison of Software-Reliability- Growth Predictions: Neural Networks vs Parametric recalibration. *IEEE Transactions on Reliability*, 48(3): 285-291.
14. Smola, A.J., Schölkopf, B. and Müller K.R. (1998). The connection between regularization operators and support vector kernels. *Neural Networks*, 11: 637-649.
15. Su, C.T., Tong, L.I. and Leou, C.M. (1997). Combining time series and neural network approaches. *Journal of the Chinese Institute of Industrial Engineers*, 4: 419-430.
16. Tay, F.E.H. and Cao, L.J. (2001). Application of support vector machines in financial time series forecasting. *Omega: The International Journal of Management Science*, 29: 309-317.
17. Tay, F.E.H. and Cao, L.J. (2002). Modified support vector machines in financial time series forecasting. *Neurocomputing*, 48: 847-861.
18. Trafalis, T.B. and Ince, H.Y. (2000). Support Vector Machine for Regression and Application to Financial Forecasting. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, IEEE Computer Society, 6: pp. 348-353.
19. Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.
20. Xu, K., Xie, M., Tang, L. C. and Ho, S.L. (2003). Application of neural networks in forecasting engine system reliability. *Applied Soft computing Journal*, 2: 255-268.
21. Vojislav, K. (2001). *Learning and Soft Computing-Support Vector Machines, Neural Networks and Fuzzy Logic Models*. The MIT Press, Massachusetts.
22. Wang, Q.J. (1997). Using genetic algorithms to optimize model parameters. *Environmental Modeling and Software with Environment Data News*, 12: 27-34.