An Approach to Estimating Product Design Time Based on Fuzzy ν-Support Vector Machine

Hong-Sen Yan and Duo Xu

***Abstract*—This paper presents a new version of fuzzy support vector machine (FSVM) developed for product design time estimation. As there exist problems of finite samples and uncertain data in the estimation, the input and output variables are described as fuzzy numbers, with the metric on fuzzy number space defined. Then, the fuzzy ν-support vector machine (Fν-SVM) is proposed on the basis of combining the fuzzy theory with the ν-support vector machine, followed by the presentation of a time estimation method based on Fν-SVM and its relevant parameter-choosing algorithm. The results from the applications in injection mold design and software product design confirm the feasibility and validity of the estimation method. Compared with the fuzzy neural network (FNN) model, our Fν-SVM method requires fewer samples and enjoys higher estimating precision.**

***Index Terms*—Design time estimation, fuzzy neural network (FNN), fuzzy number, optimal parameters, ν-support vector machine (ν–SVM).**

I. Introduction

PRODUCT design is a complex dynamic process, whose duration is affected by various factors, most of which are of random, fuzzy, and uncertain characteristics. Between the factors and design time exists a kind of nonlinear mapping relationship difficult to describe by definite mathematical models. Traditionally, approximate design time is determined by qualitative analysis. With the development of computer and estimation techniques, new quantitative estimation methods are put forward. Through traditional regression analysis, Bashir and Thomson [1] propose two types of parametric models. Griffin [2], [3] explores main time factors by a statistical method, and has developed several multivariable models. Based on a structure and process decomposition approach, a model for estimating efforts at electronic product design is presented by Jacome and Lapinskii [4]. However, all these models take only a small portion of time factors into consideration. Moreover, defects are apparent in the modeling processes, such as the prior assumption about the properties of the model, and the simplification of parameters. Due to the limited practicability and accuracy of these models, the demand is high for more accurate and general methods. On the other hand, intelligent techniques, such as neural network, fuzzy logic, and expert system, have developed fast and found a wide application, especially in modeling and approximating nonlinear systems [5], [6]. Xu and Yan [7] have successfully used fuzzy neural network (FNN) to fuse data and estimate the design time. Unlike the statistical models, this FNN is a data-driven and nonparametric weak model. Thus, the FNN works well in time estimation when the sample data are sufficient. Nevertheless, the available preexisting design cases in companies are often finite, which hinders the FNN's approximation ability and generalization performance. For this problem, a new approach should be explored.

Recently, a novel machine learning technique, called support vector machine (SVM), has drawn much attention in the fields of pattern classification and regression estimation. SVM was first introduced by Vapnik *et al.* in 1995 [8], [9] as an approximate implementation of the structure risk minimization (SRM) principle in statistical learning theory instead of the empirical risk minimization (ERM) method. The SRM principle is based on the fact that the generalization error is bounded by the sum of the empirical error and a confidence interval term depending on the Vapnik–Chervonenkis (VC) dimension [8]. By minimizing this bound, better generalization performance can be achieved. Compared with traditional neural networks, SVM can provide a unique global optimal solution and avoid the curse of dimensionality. These attractive properties undoubtedly make SVM promising. Recently, with the introduction of Vapnik's ε-insensitive loss function, SVM, initially designed to solve pattern recognition problems [10]–[12], has been extended to function approximation and regression estimation problems [13]–[18].

In many real applications, the observed input data cannot be measured precisely and tend to be described at linguistic levels or in ambiguous metrics and the traditional support vector regression (SVR) method fails to cope with qualitative information. As fuzzy logic is known to be a powerful tool for processing fuzzy and uncertain data, some scholars have explored the fuzzy support vector machine (FSVM). For pattern classification problems, Lin and Wang [19] apply a fuzzy membership to each input point and reformulate the SVM so that each input point contributes to the learning of decision surface. However, Liu *et al.* [20] question the ambiguity of the fuzzy membership. Inoue and Abe [21] put forward another FSVM for the unclassifiable regions of multiclass classification problems. For regression estimation, Sun and Sun [22] add fuzzification to the front of a least-square support vector machine (LS-SVM), and do pretreatments for the input data. However, fuzzy rules are trained indeed in this FSVM. After analyzing fuzzy linear and nonlinear

Manuscript received January 13, 2005; revised June 4, 2006; accepted November 12, 2006. This work was supported in part by the National Natural Science Foundation of China under the Grant 50475075 and in part by the Program for the Ministry of Education of China to Subsidize Key Teachers of Colleges and Universities.

H.-S. Yan is with the Research Institute of Automation, Southeast University, Nanjing 210096, China (e-mail: hsyan@seu.edu.cn).

D. Xu was with the Research Institute of Automation, Southeast University, Nanjing 210096, China. He is now with the Nanjing Institute of Zhongxing Telecommunication Ltd. Corporation, Nanjing 210007, China

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2007.894080

regression approaches, Hong and Hwang [23] propose a kind of support vector fuzzy regression machine. However, the model is somewhat defected and short of real-world application analysis. Hao and Chiang [24] also incorporate the concept of fuzzy set theory into their SVM regression model. Although the components of weight vector and bias term are all expressed in fuzzy numbers, only traditional linear operations are used for these fuzzy numbers in their model.

In the SVR approach, the parameter ε controls the sparseness of the solution in an indirect way. However, it is difficult to generate a reasonable value of ε without the prior information about the accuracy of output values. Schölkopf *et al.* [25], [26] modify the original ε-SVM and introduce ν-SVM, where a new parameter ν controls the number of support vectors and the points that lie outside of the ε-insensitive tube. Then, the value of ε in the ν-SVM is traded off between model complexity and slack variables via the constant ν.

In this paper, by combining the fuzzy theory with ν-SVM, we put forward a new FSVM, called Fν-SVM. Unlike the model of Hao and Chiang [24], we define fuzzy operations and rebuild the nonlinear optimization problem. The distance metric is deduced from the Hausdorff distance. The Fν-SVM is not only of the parameter adjusting and structure control mode of ν-SVM, but of fuzzy regression. Attributed to the distance metric and parameter ν, this fuzzy model and that of [23] vary greatly, such as in the constraint conditions of optimization problems and in the computing expression of vector $\boldsymbol{w}$. Based on this model, an estimation method for product design time and its relevant parameter-choosing algorithm are put forward.

The rest of this paper is organized as follows. The Fν-SVM is described in Section II. The time estimation method is presented in Section III. In Section IV, applications in injection mold design and software product design are given, and Fν-SVM is compared with FNN and traditional ν-SVM. An extended application of the Fν-SVM is also presented in this section. Section V draws the conclusions.

II. Fν-SVM

A. Preliminary Definitions

Let $A = (m_A, s_A, t_A)$ and $B = (m_B, s_B, t_B)$ be two triangular fuzzy numbers in which $m \in R$ is the center, $s > 0$ the left spread, and $t > 0$ the right spread. In the space of $T(R)$ all triangular fuzzy numbers, we define linear operations by the extension principle [27], [28]: $A + B = (m_A + m_B, \max(s_A, s_B), \max(t_A, t_B)), kA = (km_A, s_A, t_A)$ if $k \geq 0, kA = (km_A, t_A, s_A)$ if $k < 0$, and $A - B = (m_A - m_B, \max(s_A, t_B), \max(t_A, s_B))$.

A λ-cut of A can be labeled as $A_\lambda = [\underline{A}(\lambda), \bar{A}(\lambda)]$ for $\lambda \in [0,1]$, where $\underline{A}(\lambda)$ and $\bar{A}(\lambda)$ are two boundaries of the λ-cut, as shown in Fig. 1.

The λ-cut of a fuzzy number is always a closed and bounded interval. In terms of the Hausdorff distance of real numbers, we can define a metric in $T(R)$ as [29], [30]

$$D(A,B) = \sup_\lambda \max\left\{|\underline{A}(\lambda) - \underline{B}(\lambda)|, |\bar{A}(\lambda) - \overline{B}(\lambda)|\right\} \quad (1)$$

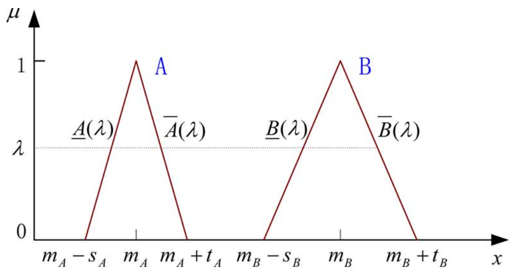


Fig. 1. The λ-cuts of two triangular fuzzy numbers.

where $A_\lambda = [\underline{A}(\lambda), \bar{A}(\lambda)]$ and $B_\lambda = [\underline{B}(\lambda), \overline{B}(\lambda)]$ are λ-cuts of two fuzzy numbers.

Theorem 1: In $T(R)$, the Hausdorff metric can be obtained as follows:

$$D(A,B) = \max\{|(m_A - s_A) - (m_B - s_B)|, |m_A - m_B|, |(m_A + t_A) - (m_B + t_B)|\}. \quad (2)$$

Corollary 1: If A and B are two symmetric triangular fuzzy numbers in $T(R)$, where $A = (m_A, s_A)$ and $B = (m_B, s_B)$, then the Hausdorff metric of A and B can be written as

$$D(A,B) = \max\{|(m_A - s_A) - (m_B - s_B)|, |(m_A + s_B) - (m_B + s_B)|\}. \quad (3)$$

B. Linear Regression Estimation

The classical fuzzy linear regression (FLR) theory, first proposed by Tanaka *et al.* [31], involves crisp input–output data and fuzzy regression coefficients. However, many actual problems consist of qualitative or uncertain observation values, and that of the estimation of product design time is just of this kind. It is thus necessary to study the regression models composed of fuzzy observation data and crisp coefficients, which requires that an SVM model for linear regression estimation be built first.

Suppose a set of fuzzy training samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^l$, where $\boldsymbol{x}_i \in T(R)^n$ and $\boldsymbol{y}_i \in T(R)$. $T(R)^n$ is the set of n-dimensional vectors of triangular fuzzy numbers. For computational simplicity, only symmetric triangular fuzzy numbers are taken into account, i.e., $\boldsymbol{s}_{x_i} = \boldsymbol{t}_{x_i}$, $\boldsymbol{s}_{y_i} = \boldsymbol{t}_{y_i}$, $\boldsymbol{x}_i = (\boldsymbol{m}_{x_i}, \boldsymbol{s}_{x_i})$ and $\boldsymbol{y}_i = (\boldsymbol{m}_{y_i}, \boldsymbol{s}_{y_i})$.

Let the approximation function $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$, where $\boldsymbol{w} = (w^1, w^2, \ldots, w^n)$ and $\boldsymbol{w} \cdot \boldsymbol{x}$ denotes an inner product of $\boldsymbol{w}$ and $\boldsymbol{x}$. In $T(R)$, $f(\boldsymbol{x})$ can be written as

$$f(\boldsymbol{x}) = (\boldsymbol{w} \cdot \boldsymbol{m}_x + b, \rho(\boldsymbol{s}_x)) \quad (4)$$

where $\rho(\boldsymbol{s}_x) = \max(s_x^1, s_x^2, \ldots, s_x^n)$.

From Corollary 1 and the idea of ν-SVM, the regression coefficients in $T(R)$ can be estimated by the following constrained optimization problem (Fig. 2):

$$\min_{\boldsymbol{w}, b, \varepsilon, \xi^{(*)}} \frac{1}{2}\|\mathbf{w}\|^2 + C\left[\nu\varepsilon + \frac{1}{l}\sum_{k=1}^{2}\sum_{i=1}^{l}(\xi_{ki} + \xi_{ki}^*)\right]$$

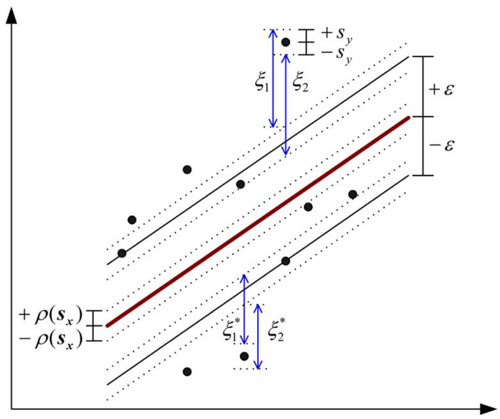


Fig. 2. The ε-insensitive tube of Fν-SVM.

$$\text{s.t.}\begin{cases}(m_{y_i}+s_{y_i})-(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b+\rho(\boldsymbol{s}_{x_i}))\le\varepsilon+\xi_{1i}\\(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b+\rho(\boldsymbol{s}_{x_i}))-(m_{y_i}+s_{y_i})\le\varepsilon+\xi_{1i}^*\\(m_{y_i}-s_{y_i})-(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b-\rho(s_{x_i}))\le\varepsilon+\xi_{2i}\\(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b-\rho(\boldsymbol{s}_{x_i}))-(m_{y_i}-s_{y_i})\le\varepsilon+\xi_{2i}^*\\\xi_{ki},\xi_{ki}^*\ge 0,\qquad k=1,2\\\varepsilon\ge 0\end{cases}\tag{5}$$

where $C>0$ is a penalty factor, $\xi_{ki}^{(*)}(k=1,2;i=1,\ldots,l)$ are slack variables, and $\nu\in(0,1]$ is an adjustable regularization parameter.

Problem (5) is of quadratic programming (QP). In terms of Lagrangian multipliers, the Lagrangian function is

$$\begin{aligned}L&\left(\boldsymbol{w},b,\varepsilon,\xi^{(*)},\alpha^{(*)},\beta,\eta^{(*)}\right)\\&=\frac{1}{2}\|\boldsymbol{w}\|^2+C\left[\nu\varepsilon+\frac{1}{l}\sum_{k=1}^{2}\sum_{i=1}^{l}(\xi_{ki}+\xi_{ki}^*)\right]\\&\quad-\sum_{k=1}^{2}\sum_{i=1}^{l}(\eta_{ki}\xi_{ki}+\eta_{ki}^*\xi_{ki}^*)-\beta\varepsilon\\&\quad+\sum_{i=1}^{l}\alpha_{1i}\left[(m_{y_i}+s_{y_i})-(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b+\rho(\boldsymbol{s}_{x_i}))-\varepsilon-\xi_{1i}\right]\\&\quad+\sum_{i=1}^{l}\alpha_{1i}^*\left[(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b+\rho(\boldsymbol{s}_{x_i}))-(m_{y_i}+s_{y_i})-\varepsilon-\xi_{1i}^*\right]\\&\quad+\sum_{i=1}^{l}\alpha_{2i}\left[(m_{y_i}-s_{y_i})-(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b-\rho(\boldsymbol{s}_{x_i}))-\varepsilon-\xi_{2i}\right]\\&\quad+\sum_{i=1}^{l}\alpha_{2i}^*\left[(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b-\rho(\boldsymbol{s}_{x_i}))-(m_{y_i}-s_{y_i})-\varepsilon-\xi_{2i}^*\right]\end{aligned}\tag{6}$$

where $\alpha_{ki}^{(*)},\beta,\eta_{ki}^{(*)}\ge 0(k=1,2;i=1,\ldots,l)$ are Lagrangian multipliers. Differentiating the Lagrangian function (6) with regard to $\boldsymbol{w},b,\varepsilon,\xi_{ki}^{(*)}$, we have

$$\frac{\partial L}{\partial\boldsymbol{w}}=0\Rightarrow\boldsymbol{w}=\sum_{k=1}^{2}\sum_{i=1}^{l}(\alpha_{ki}-\alpha_{ki}^*)\boldsymbol{m}_{x_i}\tag{7}$$

$$\frac{\partial L}{\partial b}=0\Rightarrow\sum_{k=1}^{2}\sum_{i=1}^{l}(\alpha_{ki}-\alpha_{ki}^*)=0\tag{8}$$

$$\frac{\partial L}{\partial\varepsilon}=0\Rightarrow\beta=C\nu-\sum_{k=1}^{2}\sum_{i=1}^{l}(\alpha_{ki}+\alpha_{ki}^*)\tag{9}$$

$$\frac{\partial L}{\partial\xi_{ki}^{(*)}}=0\Rightarrow\eta_{ki}^{(*)}=C/l-\alpha_{ki}^{(*)}.\tag{10}$$

Substituting (7)–(10) into (6) generates the corresponding dual form of function (5)

$$\begin{aligned}\max_{\alpha,\alpha^*}&-\frac{1}{2}\|\boldsymbol{w}\|^2+\sum_{i=1}^{l}(m_{y_i}+s_{y_i}-\rho(\boldsymbol{s}_{x_i}))(\alpha_{1i}-\alpha_{1i}^*)\\&+\sum_{i=1}^{l}(m_{y_i}-s_{y_i}+\rho(\boldsymbol{s}_{x_i}))(\alpha_{2i}-\alpha_{2i}^*)\\\text{s.t.}\quad&\sum_{k=1}^{2}\sum_{i=1}^{l}(\alpha_{ki}-\alpha_{ki}^*)=0,\qquad\alpha_{ki}^{(*)}\in[0,C/l],\\&\sum_{k=1}^{2}\sum_{i=1}^{l}(\alpha_{ki}+\alpha_{ki}^*)\le C\nu\end{aligned}\tag{11}$$

where $\|\boldsymbol{w}\|^2=\sum_{i,j=1}^{l}(\alpha_{1i}-\alpha_{1i}^*+\alpha_{2i}-\alpha_{2i}^*)(\alpha_{1j}-\alpha_{1j}^*+\alpha_{2j}-\alpha_{2j}^*)(\boldsymbol{m}_{x_i}\cdot\boldsymbol{m}_{x_j})$.

The Lagrangian multipliers $\alpha_{ki}^{(*)}$ is determined by solving the previous QP problem. Based on the Karush–Kuhn–Tucker (KKT) conditions, we have

$$\begin{cases}\alpha_{1i}(m_{y_i}+s_{y_i}-\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}-b-\rho(\boldsymbol{s}_{x_i})-\varepsilon-\xi_{1i})=0\\\alpha_{1i}^*(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b+\rho(\boldsymbol{s}_{x_i})-m_{y_i}-s_{y_i}-\varepsilon-\xi_{1i}^*)=0\\\alpha_{2i}(m_{y_i}-s_{y_i}-\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}-b+\rho(\boldsymbol{s}_{x_i})-\varepsilon-\xi_{2i})=0\\\alpha_{2i}^*(\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b-\rho(\boldsymbol{s}_{x_i})-m_{y_i}+s_{y_i}-\varepsilon-\xi_{2i}^*)=0\\\left(C/l-\alpha_{ki}^{(*)}\right)\xi_{ki}^{(*)}=0\\\left[C\nu-\sum_{k=1}^{2}\sum_{i=1}^{l}(\alpha_{ki}+\alpha_{ki}^*)\right]\varepsilon=0\end{cases}.\tag{12}$$

Theorem 2: In $T(R)$, the Lagrangian multipliers $\alpha_{ki}^{(*)}$ satisfy $\alpha_{ki}\alpha_{ki}^*=0,k=1,2$.

Proof: Suppose that there exist Lagrangian multipliers α_{1i} and α_{1i}^*, which satisfy $\alpha_{ki}\alpha_{ki}^*\ne 0$. Then, there must exist $\alpha_{1i}\ne 0$ and $\alpha_{1i}^*\ne 0$. According to the first and the second equations of (12), we get

$$\begin{cases}m_{y_i}+s_{y_i}-\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}-b-\rho(\boldsymbol{s}_{x_i})-\varepsilon-\xi_{1i}=0\\\boldsymbol{w}\cdot\boldsymbol{m}_{x_i}+b+\rho(\boldsymbol{s}_{x_i})-m_{y_i}-s_{y_i}-\varepsilon-\xi_{1i}^*=0\end{cases}.$$

Incorporating the previous two equations, we obtain $2\varepsilon+\xi_{1i}+\xi_{1i}^*$ resulting in $\varepsilon=\xi_{1i}=\xi_{1i}^*=0$. This equation conflicts with the conditions $\varepsilon,\xi_{1i},\xi_{1i}^*\ge 0$. Therefore, $\alpha_{1i}\alpha_{1i}^*=0$ holds.

In the same way, $\alpha_{2i}\alpha_{2i}^*=0$ also holds. This completes the proof of Theorem 2.

According to Theorem 2, if $\alpha_{ki}\ne 0$ holds, $\alpha_{ki}^*=0$ must hold, and vice versa. For $k=1$ and $k=2$, the data points inside the tube satisfy $\alpha_{ki}=\alpha_{ki}^*=0$. Those data points outside the tube are boundary support vectors (BSVs), where $\alpha_{ki}=C/l$ and $\alpha_{ki}^*=0$ (or $\alpha_{ki}=0$ and $\alpha_{ki}^*=C/l$). In addition, the ones lying on the edge of the tube are normal support vectors (NSVs), where $\alpha_{ki}\in(0,C/l)$ and $\alpha_{ki}^*=0$ (or $\alpha_{ki}=0$ and $\alpha_{ki}^*\in(0,C/l)$). BSVs and NSVs are referred to as support vectors (SVs) [13], [25].

For $k=1$, NSVs satisfy $\alpha_{1i} \in (0, C/l)$ or $\alpha^*_{1i} \in (0, C/l)$. By the first and the second equations of (12), we obtain (13), as shown at the bottom of the page. For $k=2$, we also get (14), as shown at the bottom of the page. Thus, parameter b can be determined from (13) and (14) by taking into account input data $\boldsymbol{x}_i$ for which the Lagrangian multipliers are in open interval $(0, C/l)$. Usually, the parameter is obtained by taking the mean value obtained for all data that satisfy the conditions of (13) and (14). In practice, (13) and (14) are equivalent, i.e., $b = b_1 = b_2$, where b_1 is the mean value of (13) and b_2 that of (14). Let N_{NSV1} and N_{NSV1^*} denote the NSV numbers corresponding to the two conditions of (13); then, we obtain (15), as shown at the bottom of the page.

If $N_{\mathrm{NSV1}^{(*)}} \neq 0$, by combining the previous two equations, we obtain

$$b = b_1 = \frac{\sum_{x_i \in \mathrm{NSV1}} (m_{y_i} + s_{y_i} - \boldsymbol{w} \cdot \boldsymbol{m}_{x_i} - \rho(\boldsymbol{s}_{x_i}))}{2N_{\mathrm{NSV1}}} + \frac{\sum_{x_i \in \mathrm{NSV1}^*} (m_{y_i} + s_{y_i} - \boldsymbol{w} \cdot \boldsymbol{m}_{x_i} - \rho(\boldsymbol{s}_{x_i}))}{2N_{\mathrm{NSV1}^*}}. \quad (16)$$

The regression function (4) is determined by

$$f(\boldsymbol{x}) = \left(\sum_{k=1}^{2} \sum_{i=1}^{l} (\alpha_{ki} - \alpha^*_{ki})(\boldsymbol{m}_{x_i} \cdot \boldsymbol{m}_x) + b, \rho(\boldsymbol{s}_x) \right). \quad (17)$$

According to the last equation of (12), if $\varepsilon \neq 0$, $\sum_{k=1}^{2} \sum_{i=1}^{l} (\alpha_{ki} + \alpha^*_{ki}) = C\nu$. Let N_{BSV} and N_{SV}, respectively, denote the numbers of BSVs and SVs, where $N_{\mathrm{SV}} = N_{\mathrm{NSV}} + N_{\mathrm{BSV}}$. For BSVs, we obtain

$$\frac{N_{\mathrm{BSV}} C}{l} \leq \left(\sum_{k=1}^{2} \sum_{j=1}^{N_{\mathrm{BSV}}} (\alpha_{kj} + \alpha^*_{kj}) + \sum_{k=1}^{2} \sum_{j=1}^{N_{\mathrm{NSV}}} (\alpha_{kj} + \alpha^*_{kj}) \right) = \sum_{k=1}^{2} \sum_{i=1}^{l} (\alpha_{ki} + \alpha^*_{ki}) = C\nu. \quad (18)$$

Thus, $\nu \geq (N_{\mathrm{BSV}}/l)$ holds. For SVs, we get

$$\frac{N_{\mathrm{SV}} C}{l} \geq \sum_{k=1}^{2} \sum_{j=1}^{N_{\mathrm{SV}}} (\alpha_{kj} + \alpha^*_{kj}) = \sum_{k=1}^{2} \sum_{i=1}^{l} (\alpha_{ki} + \alpha^*_{ki}) = C\nu. \quad (19)$$

Thus, $\nu \leq (N_{\mathrm{SV}}/l)$ holds. Then, we have $N_{\mathrm{BSV}} \leq l\nu \leq N_{\mathrm{SV}}$.

C. Nonlinear Regression Estimation

In order to move from linear functions to nonlinear functions, the following generalization can be made [8], [9]: mapping the input vector $\boldsymbol{x}$ onto a high-dimensional feature space $\Re$ through a nonlinear mapping $\varphi(\cdot)$. Then, a nonlinear regression function in the feature space can be constructed as

$$f(\boldsymbol{x}) = (\boldsymbol{w} \cdot \varphi(\boldsymbol{m}_x) + b, \rho(\boldsymbol{s}_x)) \quad (20)$$

where the dimension of $\boldsymbol{w}$ is equal to that of the feature space $\Re$, with $\rho(\boldsymbol{s}_x)$ still defined as $\max(s^1_x, s^2_x, \ldots, s^n_x)$.

The constrained optimization problem (5) can be reformulated as

$$\min_{\boldsymbol{w}, b, \varepsilon, \xi^{(*)}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\left[\nu\varepsilon + \frac{1}{l}\sum_{k=1}^{2}\sum_{i=1}^{l}(\xi_{ki} + \xi^*_{ki})\right]$$
$$\text{s.t.} \begin{cases} (m_{y_i} + s_{y_i}) - (\boldsymbol{w} \cdot \varphi(\boldsymbol{m}_{x_i}) + b + \rho(\boldsymbol{s}_{x_i})) \leq \varepsilon + \xi_{1i} \\ (\boldsymbol{w} \cdot \varphi(\boldsymbol{m}_{x_i}) + b + \rho(\boldsymbol{s}_{x_i})) - (m_{y_i} + s_{y_i}) \leq \varepsilon + \xi^*_{1i} \\ (m_{y_i} - s_{y_i}) - (\boldsymbol{w} \cdot \varphi(\boldsymbol{m}_{x_i}) + b - \rho(\boldsymbol{s}_{x_i})) \leq \varepsilon + \xi_{2i} \\ (\boldsymbol{w} \cdot \varphi(\boldsymbol{m}_{x_i}) + b - \rho(\boldsymbol{s}_{x_i})) - (m_{y_i} - s_{y_i}) \leq \varepsilon + \xi^*_{2i} \\ \xi_{ki}, \xi^*_{ki} \geq 0, \quad k = 1, 2 \\ \varepsilon \geq 0 \end{cases}. \quad (21)$$

Then, we obtain the corresponding dual form of function (21) in the same way

$$\max_{\alpha, \alpha^*} -\frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{l} (m_{y_i} + s_{y_i} - \rho(\boldsymbol{s}_{x_i}))(\alpha_{1i} - \alpha^*_{1i}) + \sum_{i=1}^{l} (m_{y_i} - s_{y_i} + \rho(\boldsymbol{s}_{x_i}))(\alpha_{2i} - \alpha^*_{2i})$$
$$\text{s.t.} \quad \sum_{k=1}^{2}\sum_{i=1}^{l} (\alpha_{ki} - \alpha^*_{ki}) = 0, \alpha^{(*)}_{ki} \in [0, C/l], \quad \sum_{k=1}^{2}\sum_{i=1}^{l} (\alpha_{ki} + \alpha^*_{ki}) \leq C\nu \quad (22)$$

where $\|\boldsymbol{w}\|^2 = \sum_{i,j=1}^{l} (\alpha_{1i} - \alpha^*_{1i} + \alpha_{2i} - \alpha^*_{2i})(\alpha_{1j} - \alpha^*_{1j} + \alpha_{2j} - \alpha^*_{2j})(\varphi(\boldsymbol{m}_{x_i}) \cdot \varphi(\boldsymbol{m}_{x_j}))$.

In that case, the inner product $\boldsymbol{m}_{x_i} \cdot \boldsymbol{m}_{x_j}$ of a linear problem is transformed into an inner product $\varphi(\boldsymbol{m}_{x_i}) \cdot \varphi(\boldsymbol{m}_{x_j})$ of the corresponding icons in the feature space. This new inner product can be replaced by a suitable kernel function $K(\boldsymbol{m}_{x_i}, \boldsymbol{m}_{x_j}) = \varphi(\boldsymbol{m}_{x_i}) \cdot \varphi(\boldsymbol{m}_{x_j})$.

Therefore, we can rewrite (12) as (23) and (24), as shown at the bottom of the next page. In fact, SVMs can be described as neural networks. Fig. 3 shows the structure of Fν-SVM.

III. Time-Estimation Method Based On Fν-SVM

Estimating product design time is essentially a nonlinear regression estimation problem, with two specialties worth pointing out. First, product design is a discrete problem, and

$$b = \begin{cases} m_{y_i} + s_{y_i} - \boldsymbol{w} \cdot \boldsymbol{m}_{x_i} - \rho(\boldsymbol{s}_{x_i}) - \varepsilon, & \text{if } \alpha_{1i} \in (0, C/l) \\ m_{y_i} + s_{y_i} - \boldsymbol{w} \cdot \boldsymbol{m}_{x_i} - \rho(\boldsymbol{s}_{x_i}) + \varepsilon, & \text{if } \alpha^*_{1i} \in (0, C/l) \end{cases}. \quad (13)$$

$$b = \begin{cases} m_{y_i} - s_{y_i} - \boldsymbol{w} \cdot \boldsymbol{m}_{x_i} + \rho(\boldsymbol{s}_{x_i}) - \varepsilon, & \text{if } \alpha_{2i} \in (0, C/l) \\ m_{y_i} - s_{y_i} - \boldsymbol{w} \cdot \boldsymbol{m}_{x_i} + \rho(\boldsymbol{s}_{x_i}) + \varepsilon, & \text{if } \alpha^*_{2i} \in (0, C/l) \end{cases}. \quad (14)$$

$$\begin{cases} N_{\mathrm{NSV1}} b_1 = \sum_{x_i \in \mathrm{NSV1}} (m_{y_i} + s_{y_i} - \boldsymbol{w} \cdot \boldsymbol{m}_{x_i} - \rho(\boldsymbol{s}_{x_i})) - N_{\mathrm{NSV1}}\varepsilon, & \text{if } \alpha_{1i} \in (0, C/l) \\ N_{\mathrm{NSV1}^*} b_1 = \sum_{x_i \in \mathrm{NSV1}^*} (m_{y_i} + s_{y_i} - \boldsymbol{w} \cdot \boldsymbol{m}_{x_i} - \rho(\boldsymbol{s}_{x_i})) + N_{\mathrm{NSV1}^*}\varepsilon, & \text{if } \alpha^*_{1i} \in (0, C/l) \end{cases}. \quad (15)$$

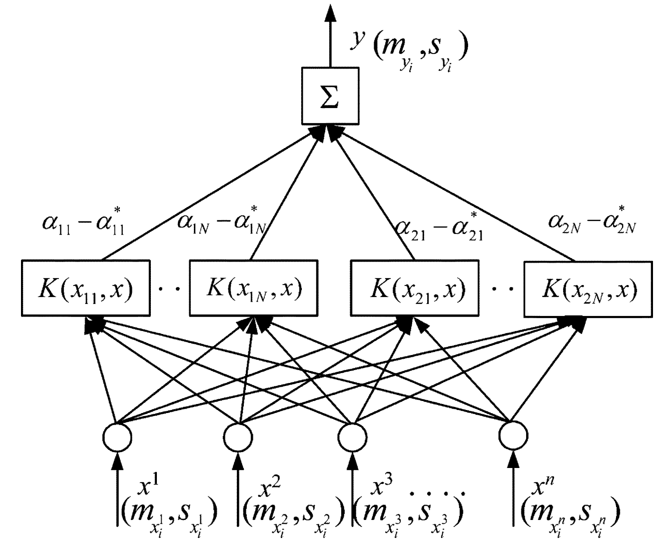


Fig. 3. Architecture of Fν-SVM.

design cycle time cannot form continuous time-series data. Thereby, the estimation of design time is not a general forecasting problem. Second, there exist problems of finite samples and uncertain data in design time estimation. Therefore, the fuzzy support vector regression machine is just suitable for the problem of design time estimation. Here, the input and output variables are described as fuzzy numbers. There are various factors that influence product design time. These time factors, corresponding to the input variables of Fν-SVM, can be sorted into the four main categories: product characteristics, design process, design condition, and design team [7]. For the particular design team of a certain company, changes of design conditions and design process are often infrequent. Therefore, only product characteristics are taken into account in this paper. Information on product characteristics can be extracted from customer demands at an early stage of product development [32].

Suppose the number of variables is n and $n = n_1 + n_2$, where n_1 and n_2 denote the number of fuzzy linguistic variables and crisp numerical variables, respectively. The linguistic variables are evaluated at several description levels, and a real number between 0 and 1 can be assigned to each of them. Distinct numerical variables, being of different dimensions, should be normalized first as in the following:

$$\bar{x}_i^d = \frac{x_i^d - \min\left(x_i^d\big|_{i=1}^l\right)}{\max\left(x_i^d\big|_{i=1}^l\right) - \min\left(x_i^d\big|_{i=1}^l\right)}, \qquad d = 1, 2, \ldots, n_2 \tag{25}$$

where l is the number of samples, x_i^d denotes the original value, and $\overline{x}_i^d$ is the normalized value. In fact, all the numerical variables in (4)–(24) are the normalized values although unmarked by bars.

Fuzzification is used to process the linguistic and the normalized numerical variables. The centers of those corresponding triangular fuzzy numbers are assigned the normalized values. The spreads of those fuzzy numbers are determined by evaluation or by the function of the observed values, such as $\boldsymbol{s}_{x_i} = \theta^* \boldsymbol{m}_{x_i}$ where θ is a coefficient for fuzzification.

On the basis of the Fν-SVM model, a time-estimation algorithm can be summarized as follows.

Algorithm 1:

Step 1) Initialize the original data by normalization and fuzzification, and then, form training patterns.

Step 2) Select the kernel function K, control constant ν, and penalty factor and C. Construct the QP problem (22) of the Fν-SVM.

Step 3) Solve the optimization problem and obtain the parameters $\alpha_{ki}^{(*)}$. Compute the regression coefficient b by (23).

Step 4) For a new design task, extract product characteristics and form a set of input variables $\boldsymbol{x}$.

Then, compute the estimation result $\hat{y}$ by (24).

Many actual applications suggest that radial basis functions (RBFs) tend to perform well under general smoothness assumptions, so that they should be considered specifically if no additional knowledge of the data is available [33]. In this paper, Gaussian RBF is used as the kernel function of Fν-SVM

$$K\left(\boldsymbol{m}_{x_i}, \boldsymbol{m}_{x_j}\right) = \exp\left(-\frac{\left\|\boldsymbol{m}_{x_i} - \boldsymbol{m}_{x_j}\right\|^2}{2\sigma^2}\right). \tag{26}$$

There is no structured way to choose the optimal parameters of SVMs, and cross validation is generally adopted. Fν-SVM involves three main parameters: C, ν, and σ. A number of our experiments show that the learning error is insensitive to the change of C unless the value of C is too small. Therefore, we can choose C based on the dynamic optimization of ν and σ. The parameter-choosing algorithm goes as follows.

Algorithm 2:

Step 1) Initialize the parameter C with a big number, such as $C = 300$. Set initial intervals of ν and σ: $\nu \in (0, 1], \sigma \in (0.1, 10]$.

Step 2) Select ν several values and σ values to a certain scale within the ranges of intervals. Assume the number of ν values is m and that of σ values n. Thus, we have $m \cdot n$ couples of parameter values.

$$b = \frac{\sum_{x_i \in \mathrm{NSV1}} \left(m_{y_i} + s_{y_i} - \sum_{k=1}^{2} \sum_{j=1}^{l} (\alpha_{kj} - \alpha_{kj}^*) K\left(\boldsymbol{m}_{x_j}, \boldsymbol{m}_{x_i}\right) - \rho(\boldsymbol{s}_{x_i})\right)}{2N_{\mathrm{NSV1}}} + \frac{\sum_{x_i \in \mathrm{NSV1}^*} \left(m_{y_i} + s_{y_i} - \sum_{k=1}^{2} \sum_{j=1}^{l} (\alpha_{kj} - \alpha_{kj}^*) K(\boldsymbol{m}_{x_j}, \boldsymbol{m}_{x_i}) - \rho(\boldsymbol{s}_{x_i})\right)}{2N_{\mathrm{NSV1}^*}} \tag{23}$$

$$f(\boldsymbol{x}) = \left(\sum_{k=1}^{2} \sum_{i=1}^{l} (\alpha_{ki} - \alpha_{ki}^*) K\left(\boldsymbol{m}_{x_i}, \boldsymbol{m}_{x}\right) + b, \rho(\boldsymbol{s}_{x_i})\right) \tag{24}$$

TABLE I
TIME FACTORS OF PRODUCT CHARACTERISTICS FOR INJECTION MODEL

Product characteristics	Unit	Expression	Weight
Structure complexity (SC)	Dimensionless	Linguistic information	0.9
Model difficulty (MD)	Dimensionless	Linguistic information	0.7
Wainscot gauge variation (WGV)	Dimensionless	Linguistic information	0.7
Cavity number (CN)	Dimensionless	Numerical information	0.8
Mold size (height/diameter) (MS)	Dimensionless	Numerical information	0.55
Form feature number (FFN)	Dimensionless	Numerical information	0.55

TABLE II
LEARNING AND TESTING DATA OF INJECTION MODEL DESIGN

Molds		Input data						Desired outputs (h)
No.	Name	SC	MD	WGV	CN	MS	FFN	
1	Global handle	L	L	L	4	3.1	3	23
2	Water bottle lid	H	L	H	4	0.56	7	45.5
3	Medicine lid	H	M	VL	4	1.5	6	37
4	Footbath basin	VL	VL	VL	1	0.5	3	10
5	Litter basket	L	M	H	1	2.1	12	42.5
6	Plastic silk flower	L	M	M	1	7.1	4	29.5
7	Dining chair	M	H	L	1	0.5	15	48
8	Spindling bushing	H	VL	L	2	8.07	2	30
9	Three-way pipe	H	L	L	1	0.45	5	24.5
10	Hydrant shell	VH	H	M	1	0.3	7	49
…	…	…	…	…	…	…	…	…
71	Paper-lead pulley	L	M	H	8	6.1	6	55
72	Winding tray	M	M	VH	1	2.18	7	41.5

Step 3) For these $m \cdot n$ couples of parameter values, $m \cdot n$ results are obtained by learning of the Fν-SVM. The parameter couple with minimum error is taken as the optimal couple of parameters.

Step 4) If the minimum error satisfies the given precision demand, go to Step 5). Otherwise, take the current optimal couple of parameter values obtained in Step 3) as the centers and shrink the intervals of ν and σ. Then, select $m \cdot n$ new couples of parameter values adjacent to the centers, and go to Step 3).

Step 5) The optimal parameters ν and σ having been determined, change C values within $[0.1, 10\,000]$ and compute the learning errors of the Fν-SVM. The parameter with minimum error is taken as the optimal C.

For l training samples, the QP problem (22) consists of $4l$ variables, one linear equality constraint, and $8l+1$ bound constraints. Therefore, the size of QP problem (22) is directly proportional to the number of training samples, independent of the input dimensionality of the Fν-SVM. The sample number in a time-estimation problem tends to be finite, allowing the computation of the QP problem (22) by traditional optimization algorithms, such as the active set method, projection method, and interior point method [14]. For cases of a large scale, chunking procedures [11], sequential minimal optimization (SMO) [14], [34], [35] and decomposition method [36]–[39] all work well.

IV. EXPERIMENTS

To illustrate the time-estimation method, the designs of plastic injection mold and software product are studied in this section, where the effectiveness of the Fν-SVM and parameter-choosing algorithm is verified not only by the estimation of product design time but also by an extended application.

A. Plastic Injection Mold Design

The injection mold is a kind of single-piece-designed product, the design process of which is usually driven by customer orders. Some time factors with large influencing weights are gathered to develop a factor list, as shown in Table I. The first three factors are expressed as linguistic information and the last three as numerical data.

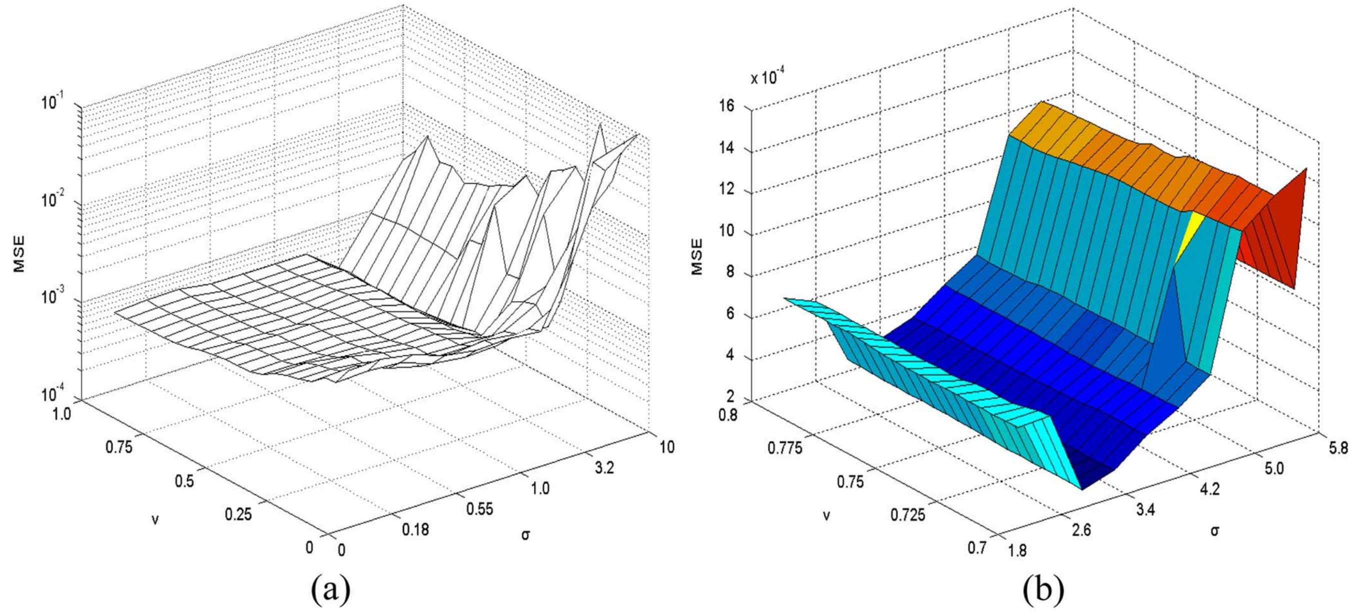


Fig. 4. Risk (MSE) versus parameters ν and σ in $C = 300$. (a) Obtained by the first parameter-choosing process. (b) Obtained by the second parameter-choosing process.

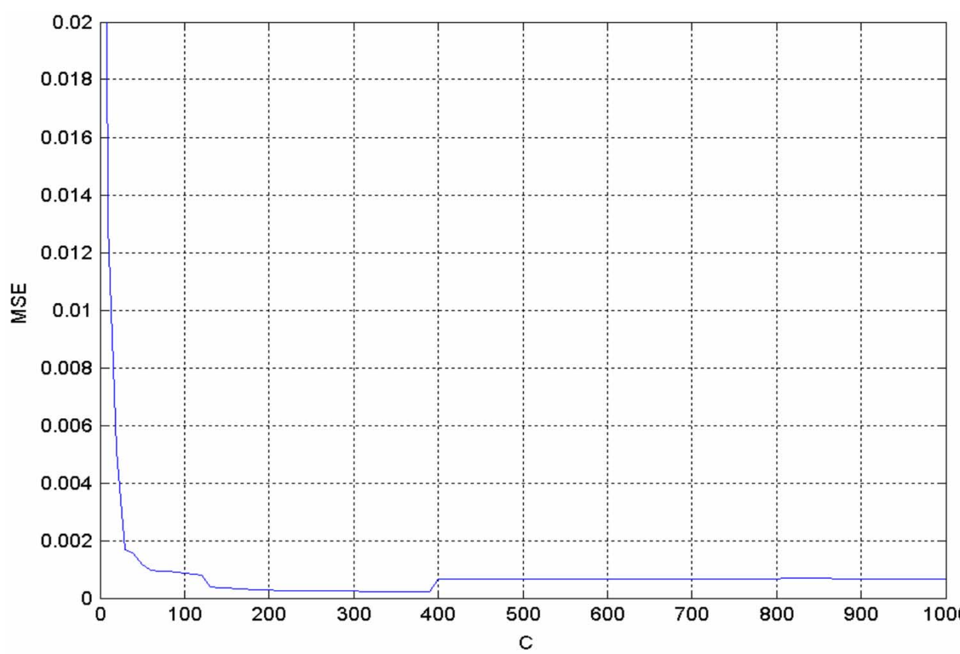


Fig. 5. Risk (MSE) versus parameter C in $\nu = 0.705$ and $\sigma = 2.65$.

In our experiments, 72 sets of molds with corresponding design time are selected from the previous projects of a typical company. The detailed characteristic data and design time of these molds compose the corresponding patterns, as shown in Table II. We train the Fν-SVM with 60 patterns, and the others are used for testing. MATLAB 6.1 is used to implement the time-estimation method. The experiments are conducted on a 2.4-GHz Pentium IV personal computer (PC) with 512-MB memory under Microsoft Windows 2000. Some criteria, such as the sum of square error (SSE), mean square error (MSE), and maximal absolute error (MAXE), are adopted to evaluate the performance of the Fν-SVM method.

Initially, we choose $C = 300$, and then, optimize the parameters ν and σ, with $m = 20$ and $n = 10$. Our first optimal choosing gains a minimal MSE (3.2471×10^{-4}), corresponding to the parameter values $\nu = 0.75$ and $\sigma = 3.2$. On this basis, we shrink the ranges of intervals and carry out the second optimal choosing. Thus, another minimal MSE (2.3755×10^{-4}) is obtained. Here, $\nu = 0.705$ and $\sigma = 2.65$ hold. The results of the previous experiments are shown in Fig. 4 as surface plots.

Then, we vary parameter C while the other parameters ν and σ are fixed. The results of risk versus C values are shown in Fig. 5: The curve is flat while $C > 120$; at $C = 390$ the minimal

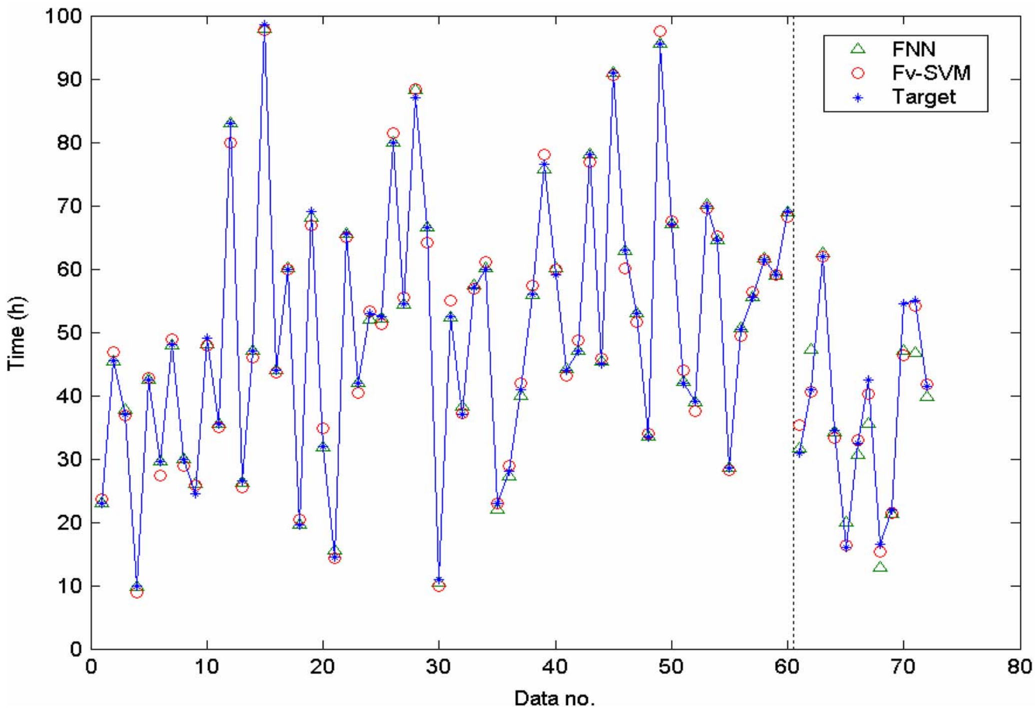


Fig. 6. Outputs of injection mold design.

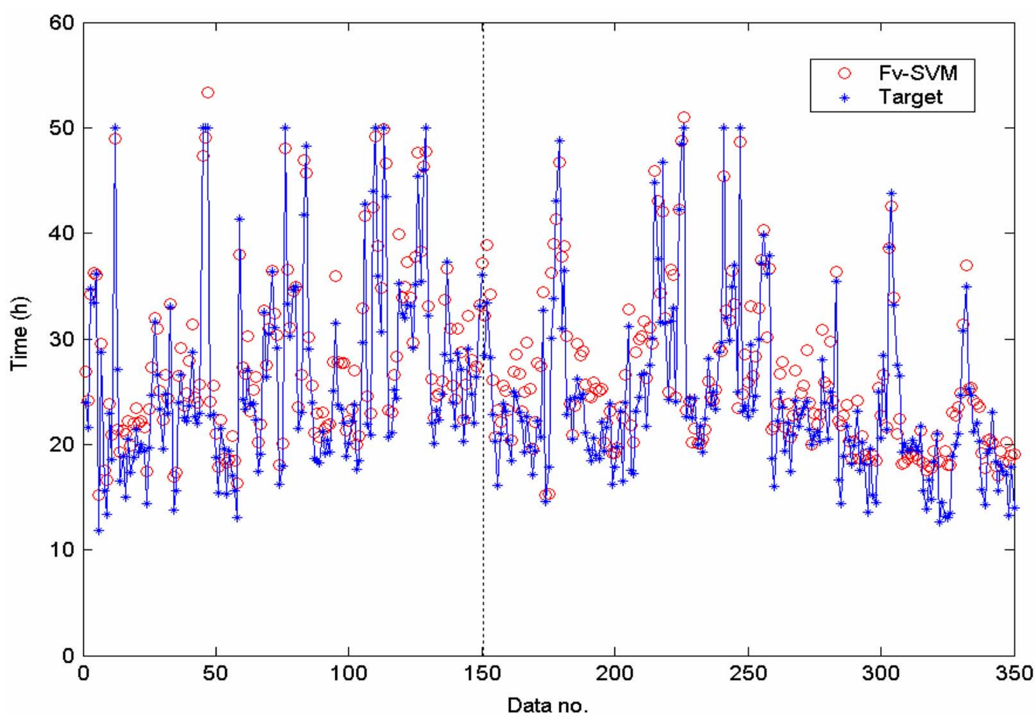


Fig. 7. Outputs of Boston housing data.

risk $(\mathrm{MSE} = 2.1152 \times 10^{-4})$ is achieved; after the optimal parameters ($\nu = 0.705$, $\sigma = 2.65$, and $C = 390$) are obtained, the learning process of the Fν-SVM needs 15.922 s. Here, we use $\theta = 0.1$.

Finally, the Fν-SVM model is verified via testing data. The MSE is 9.9603×10^{-4} and MAXE 0.0916. The Fν-SVM is compared with the FNN presented in [7], and Fig. 6 shows the estimation results of these two models. It can be seen that the Fν-SVM has a better approximation performance than the FNN.

Furthermore, we test the Fν-SVM and FNN by varying the number of training samples. The results are given in Table III. With the decrease of training samples, the estimating accuracy of the FNN declines gradually, which results from the fact that FNNs are based on the empirical risk minimization principle

TABLE III
PERFORMANCES OF THE Fν-SVM AND FNN UNDER DIFFERENT SAMPLE NUMBERS

Training samples	FNN			Fν-SVM		
	SSE	MSE	MAXE	SSE	MSE	MAXE
60	0.0286	0.0024	0.1006	0.0119	0.00099	0.0916
50	0.0711	0.0059	0.1583	0.0132	0.00110	0.0714
40	0.0747	0.0062	0.1760	0.0135	0.00113	0.0721
30	0.1105	0.0092	0.2687	0.0147	0.00123	0.0733
20	0.4617	0.0385	0.4423	0.0135	0.00113	0.0719

TABLE IV
LEARNING AND TESTING DATA OF SOFTWARE PRODUCT DESIGN

No.	Input data				Desired outputs (d)
	PC	MN	CMN	HR	
1	VH	15	1	3.6	40
2	H	18	12	2.4	91
3	L	7	6	4.1	20.5
4	VH	12	2	2.2	43
5	M	19	16	4.6	67
6	L	18	15	6.8	40.5
7	VL	22	7	3.8	26
8	VH	15	6	5.6	43
9	M	10	3	3.7	23.5
10	L	6	1	3.9	9
…	…	…	…	…	…
37	L	20	18	3.9	63
38	M	18	7	7.2	34

and training samples should match fuzzy rules in number in the FNN. However, the Fν-SVM still performs well with a small number of training samples. It is thus clear that the Fν-SVM is of good generalization performance, appropriate to cases with finite samples.

B. Software Product Design

The software development process generally consists of the five phases of *plan, design, implement, mend*, and *issue*. Here, the design and implement phases are studied.

Chosen for this study is a software development company that mainly develops Office Automation System (OAS) for governments, corporations, and financial institutions. Here, four time factors are considered: product complexity (PC), module number (MN), custom-built module number (CMN), and human resource (HR). Product complexity is expressed as linguistic information and the other three as numerical data. Human resource is the weight sum of personal number of all levels. Product complexity is gained through synthetically evaluating the module correlation degree, function complexity, and product novelty. In our experiments, 38 projects developed on Lotus Domino/Notes are selected, with 30 patterns used for training and the rest for testing. These data are shown in Table IV.

We choose $C = 300$, and then, optimize the parameters ν and σ. It takes 263 s to gain a minimal MSE (2.9474×10^{-4}), which corresponds to the parameter values $\nu = 0.9$ and $\sigma = 0.55$. Then, 106 s are taken to optimize C and reach the minimal risk $(\mathrm{MSE} = 2.8686 \times 10^{-4})$ with $C = 130$. For the testing data, the MSE is 1.5×10^{-3} and MAXE 0.0938. Here $\theta = 0.1$.

If we set $\theta = 0$, the triangular fuzzy numbers will become precise numbers, and then, the model is a ν-SVM in fact. On this condition, we train the model and get a result. For the testing data, the MSE is 2.5×10^{-3} and MAXE 0.0996.

C. Extended Application of the Fν-SVM

Besides the problem of design time estimation, Fν-SVM can also be extended to other applications. We tested the Fν-SVM by the Boston housing data set, which is part of the University of

California at Irvine (UCI) machine learning repository [40]. The data set consists of 14 attributes, with binary-valued attribute *CHAS* and deficient attribute *ZN* ignored, where *CHAS* represents the Charles River dummy variable ($= 1$ if tract bounds river; 0, otherwise) and *ZN* denotes the proportion of residential land zoned for lots over 25 000 ft^2. The median housing values (MEDV) are taken as the desired outputs. We choose the frontal 350 cases, of which 100 are used for training and the rest for testing data.

By the parameter-choosing algorithm, we obtain $\nu = 0.65$, $\sigma = 0.7$, and $C = 200$. On this condition, we train the model and get the evaluated output values of testing data: MSE is 6.2×10^{-3} and MAXE 0.18577. Here, θ is 0.15. For normal ν-SVM, MSE is 7.3×10^{-3} and MAXE 0.27362.

V. Conclusion

In this paper, a new version of FSVM, named Fν-SVM, is proposed for product design time estimation. It is based on the combination of the fuzzy theory with ν-SVM. Its performance is evaluated with the illustration of three typical examples. The simulation results show that the Fν-SVM is effective in dealing with uncertain data and finite samples, and the parameter-choosing algorithm proposed allows the Fν-SVM to seek optimized parameters.

Compared with FNNs, the Fν-SVM overcomes the "curse of dimensionality," and has such superior properties as the strong learning capability for small samples, the good generalization performance, the insensitivity to noise or outliers, and the steerable approximation parameters. Moreover, the Fν-SVM is more controllable and computable than traditional SVM.

ν-SVM is an important modification of the original ε-SVM. In many real applications, the observed data cannot be measured precisely and usually described only at linguistic levels or in ambiguous metrics. Therefore, the authors integrate ν-SVM with the fuzzy theory to solve such problems. The Fν-SVM takes into account the fuzzy and uncertain attributes of the vectors. With the fuzzy operations, it is also capable of reflecting the fuzzy mapping relationship between or among the vectors, providing a better generalization.

In our experiments, a fixed fuzzification coefficient is adopted. However, how to choose an appropriate coefficient is not described in this paper. That is sure to be a key problem worthy of future study.

Acknowledgment

The authors would like to thank the Associate Editor, the three anonymous referees, and Prof. L. Lu for their valuable comments and suggestions.

References

[1] H. A. Bashir and V. Thomson, "Models for estimating design effort and time," *Design Studies*, vol. 22, no. 2, pp. 141–155, 2001.
[2] A. Griffin, "Modeling and measuring product development cycle time across industries," *J. Eng. Technol. Manage.*, vol. 14, no. 1, pp. 1–24, 1997.
[3] A. Griffin, "The effect of project and process characteristics on product development cycle time," *J. Marketing Res.*, vol. 34, no. 1, pp. 24–35, 1997.
[4] M. F. Jacome and V. Lapinskii, "NREC: Risk assessment and planning for complex designs," *IEEE Des. Test Comput.*, vol. 14, no. 1, pp. 42–49, Jan.–Mar. 1997.
[5] M. M. Polycarpou and J. Y. Conway, "Indirect adaptive nonlinear control of drug delivery systems," *IEEE Trans. Autom. Control*, vol. 43, no. 6, pp. 849–856, Jun. 1998.
[6] M. M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 41, no. 3, pp. 447–451, Mar. 1996.
[7] D. Xu and H. S. Yan, "Research on intelligent estimation method for product design time," (in Chinese) *Control Decision*, vol. 19, no. 2, pp. 314–318, 2004.
[8] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[9] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.
[10] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Disc.*, vol. 2, no. 2, pp. 110–115, 1998.
[11] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, 1995.
[12] L. Zhang, W. D. Zhou, and L. C. Jiao, "Hidden space support vector machines," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1424–1434, Sep. 2004.
[13] V. N. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation and signal processing," in *Advances in Neural Information Processing Systems*, M. Mozer, M. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 281–287.
[14] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Royal Holloway College, London, U.K., Neuro COLT Tech. Rep. NC2-TR-1998-030, 1998.
[15] C. C. Chuang, S. F. Su, J. T. Jeng, and C. C. Hsiao, "Robust support vector regression networks for function approximation with outliers," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1322–1330, Nov. 2002.
[16] F. E. H. Tay and L. J. Cao, "Modified support vector machines in financial time series forecasting," *Neurocomput.*, vol. 48, pp. 847–861, 2002.
[17] G. B. Huang, K. Z. Mao, C. K. Siew, and D. S. Huang, "Fast modular network implementation for support vector machines," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1651–1663, Nov. 2005.
[18] V. Cherkassky and Y. Q. Ma, "Multiple model regression estimation," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 785–798, Jul. 2005.
[19] C. F. Lin and S. D. Wang, "Fuzzy support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 464–471, Mar. 2002.
[20] G. L. Liu, A. L. Zhang, and N. Y. Deng, "Uncertainty support vector classification for early warning," (in Chinese) *Chinese J. Manage. Sci.*, vol. 11, no. 4, pp. 58–61, 2003.
[21] T. Inoue and S. Abe, "Fuzzy support vector machines for pattern classification," in *Proc. Int. Joint Conf. Neural Netw.*, 2001, vol. 2, pp. 1449–1454.
[22] Z. H. Sun and Y. X. Sun, "Fuzzy support vector machine for regression estimation," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2003, vol. 4, pp. 3336–3341.
[23] D. H. Hong and C. Hwang, "Support vector fuzzy regression machines," *Fuzzy Sets Syst.*, vol. 138, no. 2, pp. 271–281, 2003.
[24] P. Y. Hao and J. H. Chiang, "A fuzzy model of support vector machine regression," in *Proc. 12th IEEE Int. Conf. Fuzzy Syst.*, 2003, vol. 1, pp. 738–742.
[25] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Comput.*, vol. 12, no. 5, pp. 1207–1245, 2000.
[26] A. Chalimourda, B. Schölkopf, and A. J. Smola, "Experimentally optimal ν in support vector regression for different noise models and parameter settings," *Neural Netw.*, vol. 17, no. 1, pp. 127–141, 2004.
[27] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic Press, 1980.
[28] M. Ming, M. Friedman, and A. Kandel, "General fuzzy least squares," *Fuzzy Sets Syst.*, vol. 88, no. 1, pp. 107–118, 1997.
[29] J. J. Buckley and T. Feuring, "Linear and non-linear fuzzy regression: Evolutionary algorithm solutions," *Fuzzy Sets Syst.*, vol. 112, no. 3, pp. 381–394, 2000.
[30] P. Diamond and P. Kloeden, *Metric Spaces of Fuzzy Sets*. Singapore: World Scientific, 1994.
[31] H. Tanaka, S. Uejima, and K. Asai, "Linear regression analysis with fuzzy model," *IEEE Trans. Syst., Man, Cybern.*, vol. 12, no. 6, pp. 903–907, 1982.
[32] D. Xu and H. S. Yan, "Mapping and analysis of product characteristics based on fuzzy measurable house of quality," *Comput. Integr. Manuf. Syst.*, vol. 10, no. 2, pp. 693–698, 2004, (in Chinese).

[33] A. J. Smola, "Learning with kernels," Ph.D. dissertation, Fakultät für Electrotechnik und Informatik, Technische Universität, Berlin, Germany, 1998.

[34] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1998, pp. 185–208.

[35] X. Y. Zeng and X. W. Chen, "SMO-based pruning methods for sparse least squares support vector machines," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1541–1546, Nov. 2005.

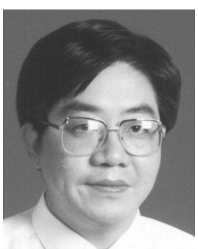
[36] C. C. Chang and C. J. Lin, "Training ν-support vector regression: Theory and algorithms," *Neural Comput.*, vol. 14, no. 8, pp. 1959–1977, 2002.

[37] C. C. Chang, C. W. Hsu, and C. J. Lin, "The analysis of decomposition methods for support vector machines," *IEEE Trans. Neural Netw.*, vol. 11, no. 4, pp. 1003–1008, Jul. 2000.

[38] P. H. Chen, R. E. Fan, and C. J. Lin, "A study on SMO-type decomposition methods for support vector machines," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 893–908, Jul. 2006.

[39] N. Takahashi and T. Nishi, "Global convergence of decomposition learning methods for support vector machines," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1362–1369, Nov. 2006.

[40] P. M. Murphy and D. W. Aha, "UCI repository of machine learning databases," Univ. California at Irvine, Irvine, CA [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html



Hong-Sen Yan received the B.S. degree in automatic control from the Harbin Ship Building Engineering Institute, Harbin, China, in 1982, the M.S. degree in industrial automation and the Ph.D. degree in automatic control theory and application from the Harbin Institute of Technology, Harbin, China, in 1989 and 1992, respectively.

From February 1982 to August 1986, he was an Assistant Engineer at the Yangzhou Marine Electronic Instruments Institute, Yangzhou, China. From July 1992 to June 1994, he was a Postdoctoral Fellow in the CIMS Laboratory, Department of Mechanical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He has been an Associate Professor and a Professor at the Research Institute of Automation, Southeast University, Nanjing, China, since July 1994 and April 1998, respectively. He is the founder of knowledgeable manufacturing and knowledge meshes theory. He has been engaged in or finished the design and development of three computer integrated manufacturing systems in enterprises with his colleagues. He has authored or coauthored more than 100 research papers in refereed journals such as the *Computers and Industrial Engineering*, the *Computers in Industry*, the *Concurrent Engineering: Research & Applications*, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, the IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the *IIE Transactions*, the *International Journal of Computer Integrated Manufacturing*, the *International Journal of Production Research*, the *Journal of Optimization Theory and Applications*, and the *Proceedings of IMechE Part B: Journal of Engineering Manufacture and Robotics and Computer Integrated Manufacturing*. His research interests include flexible manufacturing systems (FMS), computer integrated manufacturing systems (CIMS), agile manufacturing, concurrent engineering, and knowledgeable manufacturing systems.

Dr. Yan is a Senior Member of the Institute of Industrial Engineers (IIE) and a Director of the Jiangsu Association of Automation in China. He received the Best Journal Paper Prize that is one of the best 200 papers selected from all papers published in the 283 journals sponsored by the 105 institutes, societies, institutions, and associations of the China Association for Science and Technology in 2001–2005, and seven prizes from the Nation, the Ministry, the Province, and from different universities.

Duo Xu received the B.S. degree in mechanical engineering from the Hunan University, Hunan, China, in 1997, the M.S. degree in mechanical engineering, and the Ph.D. degree in control theory and engineering from Southeast University, Nanjing, China, in 2000 and 2005, respectively.

Currently, he is an Engineer at the Nanjing Institute of Zhongxing Telecommunication Ltd. Corporation, Nanjing, China. His research interests include network optimization, congestion control, and all-IP network technology.