DOI: 10.2298/CSIS120223048G

# Study On Prediction Models For Integrated Scheduling In Semiconductor Manufacturing Lines

Lu Guo, Jinghua Hao, and Min Liu*

Department of Automation, Tsinghua University,
Beijing , China,100084
guo-l08@mails.tsinghua.edu.cn, {haojinghua, lium}@tsinghua.edu.cn

**Abstract.** Quality prediction of lot operations is significant for integrated scheduling in semiconductor production line. The model-training algorithm needs to be fast and incremental to satisfy the online applications where data comes one by one or chunk by chunk. This paper presents a novel prediction model referred to as Incremental Extreme Least Square Support Vector Machine (IELSSVM), which transforms the data into ELM feature space and then minimizes the structural risk like LSSVM. The transformation into ELM feature space can be regarded as a good dimensionality reduction. The incremental formula is proposed for on-line industrial application to avoid retraining when data comes one by one or chunk by chunk. Detailed comparisons of the IELSSVM algorithm with other incremental algorithms are achieved by simulation on benchmark problems and real overlay prediction problem of lithography in semiconductor production line. The results show that IELSSVM has better performance than other incremental algorithms like OS-ELM.

**Keywords:** quality prediction model; ELM feature space; semiconductor production line; IELSSVM; overlay prediction; integrated scheduling

## 1. Introduction

With the advancement of technology and more fierce competition in the markets, scheduling has played an increasingly important role in maximizing the manufacturing capability and reducing the product makespan. Thus more and more enterprises and researchers have paid more attention to it [1]. Although many interesting research results have been attained, there are still a lot of difficulties in actual application because the lot state may have changed when a scheduling policy is carried out. For example, the lots may be held, reworked or prohibited from being processed by some machines because of quality problems. If the developed scheduling strategy does not

* To whom correspondence should be addressed.

consider these quality factors, the unexpected lot state change will severely deteriorate the scheduling performance. Integration of scheduling and quality prediction can effectively deal with the difficulties. The scheduling strategy decided on the basis of the accurate quality prediction will keep consistent with the actual environment, thus enhancing the validness of scheduling policy.

From above, we can see that quality prediction is very important in semiconductor manufacturing. Accurate prediction model is the key problem. There are three typical features for the datum attained from actual semiconductor production line. The first is high dimensional. There are hundreds of steps in the semiconductor line and hundreds of check points to measure some parameters of the lot. All these measure data is combined into one high dimensional sample point in the space. The second feature is that the data measured in actual production line is contaminated by stochastic noises. It is inevitable because of the limited precision of measuring equipments and the complexity of production environment. The third feature is originated from running time pressure in actual applications where data may come one by one or chunk by chunk. Retraining based on all the data when new data comes is very time consuming and can not meet the demand of online applications.

In this paper, a novel Incremental Extreme Least Square Support Vector Machine (IELSSVM) is presented to meet the demand of actual application and overcome the deficiencies of other intelligent modeling methods. The datum is at first transformed into ELM feature space and then the structural risk minimization principal is adopted as the objective to be optimized like LSSVM. The transformation of ELM can be regarded as a good dimensionality reduction. The incremental formula is proposed for online industrial applications to avoid retraining when data comes one by one or chunk by chunk, thus improving the training speed and efficiency.

## 2. Related Works

Over the past decades, batch learning algorithms have been discussed and investigated thoroughly. Many intelligent modeling methods have been proposed and implemented, such as Back-Propagation Neural Network (BP-NN)[2], Support Vector Machine (SVM) [3], Least Square Support Vector Machine (LSSVM) [4], Proximal Support Vector Machine (PSVM) [5], Extreme Support Vector Machine (ESVM) [6], Extreme Learning Machine (ELM) [7], and etc. BP-NN and ELM may be over-fitting because of the modeling criteria of Empirical Risk Minimization principal. SVM and LSSVM are constructed on the basis of Structural Risk Minimization principal, thus avoiding over-fitting and showing great prospects and having become very popular in recent years. But training of SVM may take a long time because of a quadratic programming problem to be solved. For ESVM, there is a time consuming matrix inversion calculation process, which restricts its online application. In [8], Huang et al. developed a constrained optimization based Extreme Learning Machine as a unified learning framework for LS-SVM, PSVM and other regularization algorithms. The algorithm can perform much better than other algorithms and showing great prospects.

Originated from the batch learning methods that have been developed, many online sequential algorithms are presented to meet the actual application demand. In [9], a generalized growing and pruning RBF (GGAP-RBF) is presented. The authors first introduce the significance of the neurons and develop an online sequential algorithm by generalized growing and pruning methods. In [10], an online sequential extreme learning machine (OS-ELM) is introduced which is much faster and produces better generalization performance compared with other sequential learning algorithms, such as GGAP-RBF. Based on batching learning algorithm of SVM, an incremental support vector machine is presented in [11]. The basic idea is that the new SVM is built based on the new arrival data and the trained support vectors. The method is called SV-incremental algorithm in [12]. Fixed-size LSSVM (FS-LSSVM) is developed in [13] for large scale regression problems. Based on quadratic Renyi entropy criteria, an active support vector selection method is developed. Compared with LSSVM, FS-LSSVM needs much less support vectors and produces better performance.

GGAP-RBF and OS-ELM may be over-fitting because of the modeling basis of Empirical Risk Minimization principal. SV-incremental algorithm and FS-LSSVM are constructed on the basis of Structural Risk Minimization principal, thus avoiding over-fitting and having been very popular in recent years. However in the SV-incremental algorithm, the old support vectors may have only a little influence on the new SVM [12] and the performance can't be satisfied. Also, the computation cost of FS-LSSVM is very high because of the eigenvectors and eigenvalues computation.

## 3. Problem Description

ELM feature space is a special space into which data is mapped by a special single-hidden-layer feed-forward networks (SLFNs) with randomly chosen parameters for activation functions. It has been proved that with sufficient nodes, given any infinite differentiable activation functions, such as sigmoid function, RBF function and etc., the ELM network can approximate any continuous target functions [7,8,10]. In this paper, we take the RBF activation function for example, which has the form like $G(a_i,b_i,x_i)=e^{-b_i(\|x_i-a_i\|)}$ .The parameters of $\{a_i,b_i\}_{i=1}^{L}$ are selected randomly and $L$ is the number of hidden nodes.

Given the input training data set $X=\{x_i\}_{i=1}^{N}$ and the corresponding output training data set $Y=\{y_i\}_{i=1}^{N}$, where $N$ is the total number of training data, all the data is mapped into the ELM feature space by the mapping matrix $\Phi(X)$ with the form:

$$\Phi(X)=\begin{bmatrix} G(a_1,b_1,x_1) & G(a_2,b_2,x_1) & \cdots & G(a_L,b_L,x_1) \\ G(a_1,b_1,x_2) & G(a_2,b_2,x_2) & \cdots & G(a_L,b_L,x_2) \\ \cdots & \cdots & \cdots & \cdots \\ G(a_1,b_1,x_N) & G(a_2,b_2,x_N) & \cdots & G(a_L,b_L,x_N) \end{bmatrix} \quad (1)$$

According to the structural risk minimization principal, the risk minimization in ELM feature space is constructed as follows:

$$\min_{w,\xi,b} E=\frac{1}{2}\|w\|^2+\frac{1}{2}\nu\|\xi\|^2 \quad (2)$$

$$s.t.\quad D[\Phi(X)w+Eb]=E-\xi \quad (3)$$

Where $w$ 、 $b$ are the weight vector and the constant offset respectively, and $D$ is the diagonal matrix with plus one or minus one along its diagonal, which indicates the class of each sample point. Clearly, $D^2$ equals $I$ . $E$ is the vector with all elements are one and dimension is $N\times 1$. Parameter $\nu$ can be seen as the cost coefficient for misclassification. The specific formation of $D$ and $E$ is given as

$$D=\begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & y_N \end{bmatrix} \quad (4)$$

$$E=\begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix}_{N\times 1} \quad (5)$$

So, the classifier is to determine the parameters of $w$ 、 $b$ when training data set $\{X,Y\}$ is given.

## 4. IELSSVM Algorithm

First, we define the Lagrange relaxation function as follows

$$L(w,\xi,b,s)=\frac{1}{2}\|w\|^2+\frac{1}{2}\nu\|\xi\|^2-s^T\{D[\Phi(X)w+Eb]-E-\xi\} \quad (6)$$

where $s$ is the Lagrange multipliers with $N\times 1$ dimensions, and $s^T$ is the transpose of $s$ . The optimized solution of (4) is the saddle point of

$L(w,\xi,b,s)$ according to KKT optimality conditions. So, we get the following equations

$$\frac{\partial L(w,\xi,b,s)}{\partial w} = w - \Phi(X)^T Ds = 0 \tag{7}$$

$$\frac{\partial L(w,\xi,b,s)}{\partial b} = s^T DE = 0 \tag{8}$$

$$\frac{\partial L(w,\xi,b,s)}{\partial \xi} = v\xi - s = 0 \tag{9}$$

$$D[\Phi(X)w + Eb] - E + \xi = 0 \tag{10}$$

Rewrite the above equations as

$$\begin{cases} w - \Phi(X)^T Ds = 0 \\ s^T DE = 0 \\ v\xi - s = 0 \\ D[\Phi(X)w + Eb] - E + \xi = 0 \end{cases} \tag{11}$$

The main parameters to be solved are $w$ 、 $b$ , so we substitute $w$ 、 $b$ for the parameters $\xi$ 、 $s$ . The result of substitution is given as

$$\begin{cases} v\Phi(X)^T \Phi(X)w + w + v\Phi(X)^T Eb = v\Phi(X)^T DE \\ E^T \Phi(X)w + E^T Eb = E^T DE \end{cases} \tag{12}$$

So we get the matrix expression of parameters $w$ 、 $b$ as

$$\begin{bmatrix} \frac{1}{v}I + \Phi(X)^T \Phi(X) & \Phi(X)^T E \\ E^T \Phi(X) & E^T E \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \Phi(X)^T DE \\ E^T DE \end{bmatrix} \tag{13}$$

Clearly, parameters of $\begin{bmatrix} w \\ b \end{bmatrix}$ can be attained by a direct matrix inverse which is time consuming.

$$\begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{v}I + \Phi(X)^T \Phi(X) & \Phi(X)^T E \\ E^T \Phi(X) & E^T E \end{bmatrix}^{-1} \begin{bmatrix} \Phi(X)^T DE \\ E^T DE \end{bmatrix} \tag{14}$$

In actual application, data may come chunk by chunk or one by one. When one new data comes, retraining all the data by a direct matrix inverse is time consuming and can not meet the requirements of online use. So we need to construct an incremental algorithm for online use in actual applications.

Suppose at time $t$, the datum that has been learned is denoted as $\{X_t, Y_t\}$, where $X_t = \{x_i\}_{i=1}^{N}$ is the input sample data set, and $Y_t = \{y_i\}_{i=1}^{N}$ is the output sample data set. At time $t+1$, new data set $\{X_{IC}, Y_{IC}\}$ is received, where $X_{IC} = \{x_i\}_{i=N+1}^{N+k}$ and $Y_{IC} = \{y_i\}_{i=N+1}^{N+k}$. The ELM mapping matrix of the last received data set is given by

$$H(X_{IC}) = \begin{bmatrix} G(a_1, b_1, x_{N+1}) & G(a_2, b_2, x_{N+1}) & \cdots & G(a_L, b_L, x_{N+1}) \\ G(a_1, b_1, x_{N+2}) & G(a_2, b_2, x_{N+2}) & \cdots & G(a_L, b_L, x_{N+2}) \\ \cdots & \cdots & \cdots & \cdots \\ G(a_1, b_1, x_{N+k}) & G(a_2, b_2, x_{N+k}) & \cdots & G(a_L, b_L, x_{N+k}) \end{bmatrix} \quad (15)$$

So, the ELM mapping matrix of all the data that has been received up to time $t+1$ can be written as

$$\Phi(X_{t+1}) = \begin{bmatrix} \Phi(X_t) \\ H(X_{IC}) \end{bmatrix} \quad (16)$$

For time $t+1$, we have the following equations

$$\begin{bmatrix} \frac{1}{v} I + \Phi(X_{t+1})^T \Phi(X_{t+1}) & \Phi(X_{t+1})^T E_{t+1} \\ E_{t+1}{}^T \Phi(X_{t+1}) & E_{t+1}{}^T E_{t+1} \end{bmatrix} \begin{bmatrix} w_{t+1} \\ b_{t+1} \end{bmatrix} = \begin{bmatrix} \Phi(X_{t+1})^T D_{t+1} E_{t+1} \\ E_{t+1}{}^T D_{t+1} E_{t+1} \end{bmatrix} \quad (17)$$

For simplicity, we denote $A_t$, $O_t$, $\beta_t$ as follows:

$$A_t = \begin{bmatrix} \frac{1}{v} I + \Phi(X_t)^T \Phi(X_t) & \Phi(X_t)^T E_t \\ E_t{}^T \Phi(X_t) & E_t{}^T E_t \end{bmatrix} \quad (18)$$

$$O_t = \begin{bmatrix} \Phi(X_t)^T D_t E_t \\ E_t{}^T D_t E_t \end{bmatrix} \quad (19)$$

$$\beta_t = \begin{bmatrix} w_t \\ b_t \end{bmatrix} \quad (20)$$

According to the relationship between time $t$ and $t+1$, $E_{t+1}$, $D_{t+1}$, $A_{t+1}$, $O_{t+1}$ can be written as

$$E_{t+1} = \begin{bmatrix} E_t \\ E_{k\times 1} \end{bmatrix} \quad (21)$$

$$D_{t+1} = \begin{bmatrix} D_t & 0 \\ 0 & D_{k\times k} \end{bmatrix} \quad (22)$$

$$A_{t+1} = A_t + \begin{bmatrix} H_{IC} \\ E_{k\times 1} \end{bmatrix} \begin{bmatrix} H_{IC} & E_{k\times 1} \end{bmatrix} \tag{23}$$

$$O_{t+1} = O_t + \begin{bmatrix} H_{IC} \\ E_{k\times 1} \end{bmatrix} \begin{bmatrix} y_{N+1} \\ y_{N+2} \\ \cdots \\ y_{N+k} \end{bmatrix} \tag{24}$$

Also, we denote $B_t$ and $Y_{t+1}$ as

$$B_t = \begin{bmatrix} H_{IC} & E_{k\times 1} \end{bmatrix} \tag{25}$$

$$Y_{t+1} = \begin{bmatrix} y_{N+1} \\ y_{N+2} \\ \cdots \\ y_{N+k} \end{bmatrix} \tag{26}$$

We have a simple expression of $A_{t+1}$ and $O_{t+1}$ as follows

$$A_{t+1} = A_t + B_t^T B_t \tag{27}$$

$$O_{t+1} = O_t + B_t^T y_{t+1} \tag{28}$$

The solution of $\beta_{t+1}$ is written as

$$\beta_{t+1} = A_{t+1}^{-1} O_{t+1} = (A_t + B_t^T B_t)^{-1} (O_t + B_t^T Y_{t+1}) \tag{29}$$

The Sherman-Morrison-Woodbury (SMW) formula [14] can be used to deal with the matrix inverse problem

$$(A_t + B_t^T B_t)^{-1} = A_t^{-1} - A_t^{-1} B_t^T (B_t A_t^{-1} B_t^T + I_{k\times k})^{-1} B_t A_t^{-1} \tag{30}$$

So, we rewrite the formula of $\beta_{t+1}$ as

$$\beta_{t+1} = (A_t^{-1} - A_t^{-1} B_t^T (B_t A_t^{-1} B_t^T + I_{k\times k})^{-1} B_t A_t^{-1})(O_t + B_t^T Y_{t+1}) \tag{31}$$

We denote $K_t$ as

$$K_t = I_{N\times 1} - A_t^{-1} B_t^T (B_t A_t^{-1} B_t^T + I_{k\times k})^{-1} B_t \tag{32}$$

So, the estimation of $\beta_{t+1}$ can be given as

$$\beta_{t+1} = K_t \beta_t + K_t A_t^{-1} B_t^T Y_{t+1} \tag{33}$$

$$A_{t+1}^{-1} = K_t A_t^{-1} \quad (34)$$

According to above procedure, a summarization of the presented IELSSVM algorithm can be given as follows.

**IELSSVM Algorithm:**

Given the hidden node number $L$ of ELM mapping feature space and a small number of training data to boost the algorithm, we present the algorithm as follows.

Step1: Initialization:

(1) Generate $\{a_i, b_i\}_{i=1}^{L}$ randomly to formulate the RBF activation function $G(a_i, b_i, x_i) = e^{-b_i(\|x_i - a_i\|)}$

(2) Generate the mapping matrix as

$$\Phi(X) = \begin{bmatrix} G(a_1,b_1,x_1) & G(a_2,b_2,x_1) & \cdots & G(a_L,b_L,x_1) \\ G(a_1,b_1,x_2) & G(a_2,b_2,x_2) & \cdots & G(a_L,b_L,x_2) \\ \cdots & \cdots & \cdots & \cdots \\ G(a_1,b_1,x_N) & G(a_2,b_2,x_N) & \cdots & G(a_L,b_L,x_N) \end{bmatrix} \quad (35)$$

for the given small number of training data.

(3) Initialize the parameters of $\begin{bmatrix} w \\ b \end{bmatrix}$ for time $t$ as

$$\begin{bmatrix} w_t \\ b_t \end{bmatrix} = \begin{bmatrix} \frac{1}{\nu} I + \Phi(X_t)^T \Phi(X_t) & \Phi(X_t)^T E_t \\ E_t^T \Phi(X_t) & E_t^T E_t \end{bmatrix}^{-1} \begin{bmatrix} \Phi(X_t)^T D_t E_t \\ E_t^T D_t E_t \end{bmatrix} \quad (36)$$

$$A_t^{-1} = \begin{bmatrix} \frac{1}{\nu} I + \Phi(X_t)^T \Phi(X_t) & \Phi(X_t)^T E_t \\ E_t^T \Phi(X_t) & E_t^T E_t \end{bmatrix}^{-1} \quad (37)$$

$$O_t = \begin{bmatrix} \Phi(X_t)^T D_t E_t \\ E_t^T D_t E_t \end{bmatrix} \quad (38)$$

Step 2: Online incremental learning procedure.

Repeat the following steps:

(1) Generate the matrix $H(X_{IC})$ for the last arrival data

$$H(X_{IC}) = \begin{bmatrix} G(a_1,b_1,x_{N+1}) & G(a_2,b_2,x_{N+1}) & \cdots & G(a_L,b_L,x_{N+1}) \\ G(a_1,b_1,x_{N+2}) & G(a_2,b_2,x_{N+2}) & \cdots & G(a_L,b_L,x_{N+2}) \\ \cdots & \cdots & \cdots & \cdots \\ G(a_1,b_1,x_{N+k}) & G(a_2,b_2,x_{N+k}) & \cdots & G(a_L,b_L,x_{N+k}) \end{bmatrix} \quad (39)$$

(2) Generate the other matrix such as

$$B_t = \begin{bmatrix} H_{IC} & E_{k\times 1} \end{bmatrix} \quad (40)$$

$$Y_{t+1} = \begin{bmatrix} y_{N+1} \\ y_{N+2} \\ \cdots \\ y_{N+k} \end{bmatrix} \quad (41)$$

(3) Estimate the parameters of $\beta = \begin{bmatrix} w \\ b \end{bmatrix}$ using the newest data

$$K_t = I_{N\times 1} - A_t^{-1} B_t^T (B_t A_t^{-1} B_t^T + I_{k\times k})^{-1} B_t \quad (42)$$

$$A_{t+1}^{-1} = K_t A_t^{-1} \quad (43)$$

$$\beta_{t+1} = K_t \beta_t + K_t A_t^{-1} B_t^T Y_{t+1} \quad (44)$$

Step 3: Online Use:

Use the learned parameters of $\beta = \begin{bmatrix} w \\ b \end{bmatrix}$ to evaluate the output $\hat{Y}_{test}$ for the given $n_{test}$ input vector $X_{test}$ in actual application by

$$\hat{Y}_{test} = \Phi(X_{test})w + E_{n_{test}\times 1} b \quad (45)$$

## 5. Simulation Results

### 5.1. Simulation Results and Analysis of Benchmark Problems

In this section, the performance of the presented IELSSVM algorithm is shown by comparison with OS-ELM algorithm presented by [10] on a lot of Benchmark problems which are very famous and can be downloaded from UCI Repository of machine learning databases[15] conveniently. The specification of datasets of the benchmark problems is listed in table 1.

In table I, the Secom dataset is a typical semiconductor manufacturing process dataset, in which all the data is collected from monitoring sensors or measurement points. Each data represents a lot with a number of measured feature values and a label for passing or failing in house line testing.

Simulation is carried out with the parameters selected as [10]. The parameters for initialization is that for regression problem $N_0 = L + 50$ and for classification problem $N_0 = L + 100$, where $N_0$ is the number of data randomly selected from the data set to initialize the algorithm and $L$ is the number of hidden nodes. For IELSSVM, the parameters of $\nu$ for regression problem is selected as $e^{N_0-5}$ while for classification problem the parameter is selected as $e^{N_0-20}$. The detailed comparison is listed in table 2. We did not deliberately select the hidden nodes for Secom datasets because of the universal approximation capability of ELM based algorithm [7,8], for which good generalization performance can be attained as long as the hidden nodes is enough. Also, to predetermine the optimal hidden nodes is not very practical as the data comes chunk by chunk or one by one in actual application. Since [10] didn't give the standard deviation for the OS-ELM algorithm, we fill the corresponding items with "---" in the table 2 and table 3.

**Table 1.** Spcification of the dataSets

| Dataset | Training Data | Testing Data | Attributes | Classes |
|---|---|---|---|---|
| Auto-MPG | 320 | 72 | 7 | --- |
| Abalone | 3000 | 1177 | 8 | --- |
| California Housing | 8000 | 12640 | 8 | --- |
| Image Segment | 1500 | 810 | 19 | 7 |
| Satellite Image | 4435 | 2000 | 36 | 6 |
| Secom | 1200 | 367 | 591 | 2 |

**Table 2.** Comparison for classification problems

| Datasets | Algorithm | Nodes | Training accuracy(%) | | Testing Accuracy (%) | |
|---|---|---|---|---|---|---|
| | | | Mean | Std | Mean | std |
| Image Segment | OS-ELM (RBF)[10] | 180 | 96.65 | --- | 94.53 | --- |
| | IELSSVM | 180 | 97.19 | 0.28 | 95.14 | 0.45 |
| Satellite Image | OS-ELM (RBF)[10] | 400 | 93.18 | --- | 89.01 | --- |
| | IELSSVM | 400 | 94.73 | 0.28 | 91.42 | 0.36 |
| Secom | OS-ELM (RBF) | 200 | 94.57 | 0.12 | 90.28 | 0.33 |
| | IELSSVM | 200 | 94.59 | 0.12 | 90.31 | 0.40 |

From table 2 and table 3, we can see that IELSSVM attains better performance than OS-ELM for both classification problems and regression problems in most cases on benchmark problems.

**Table 3.** ComparIson for regression problems

| Datasets | Algorithms | Nodes | Training RMSE($10^{-3}$) | | Testing RMSE($10^{-3}$) | |
|---|---|---|---|---|---|---|
| | | | Mean | Std | Mean | std |
| Auto-MPG | OS-ELM (RBF)[10] | 25 | 69.6 | --- | 75.9 | --- |
| | IELSSVM | 25 | 68.4 | 1 | 64.4 | 1.6 |
| Abalone | OS-ELM (RBF)[10] | 25 | 75.9 | --- | 78.3 | --- |
| | IELSSVM | 25 | 73.8 | 0.3 | 73.2 | 0.3 |
| California Housing | OS-ELM (RBF)[10] | 50 | 132.1 | --- | 134.1 | --- |
| | IELSSVM | 50 | 129.9 | 0.7 | 134.4 | 1.9 |

### 5.2. Simulation results of actual overlay prediction in lithography of semiconductor production line for integrated scheduling

Lithography is the most important part in the semiconductor production line which transfers the patterns of a reticle onto the surface of a silicon wafer. Overlay is the key parameter for the lithography process to ensure that the relative position of current layer and its pre-layer is within the regulatory value. If the overlay value exceeds the regulatory value, the electrical properties will deteriorate and even become ineffective, thus resulting in lot scraps and reworks. In real semiconductor line, the offset value range of X-axis and the offset value range of Y-axis are strictly set. There are many factors to influence the overlay performance, such as the machine performance, the variety of the lot, the current layer of the lot, the machine used to process the pre-layer and etc. For different lithography machine, the machine performance may be different sharply. So the offset value may remarkably be different when processing the same lot. Prohibiting the lots from being processed by the machines with exceeding offset values will help to avoid unexpected quality events.

In order to accurately predict the overlay value of the lot, history data and theoretical offset compensation data of current lot are used together. The input attributes are selected as a 11-dimension vector, which includes the theoretical offset compensation data of current lot and the related data of past five lots that have been processed with the same product variety and the same layer. The theoretical offset compensation and the overlay measurement attributes of the last five lots are selected. So there are total 11 attributes selected as the input attributes. The overlay measurement value of current lot is selected as the output attribute. Data of each input attribute is normalized into [-1,1]. All the data are collected from real semiconductor production line and the time is between 2010-8-2 and 2012-7-27. There are about 1192 records. 500 records selected randomly are used as the testing data while the other 692 records are used as the training data.

The parameters for initialization is that $N_0 = L + 200$ ,where $N_0$ is the number of data randomly selected from the data set to initialize the algorithm and $L$ is the number of hidden nodes. For IELSSVM, the parameters of $\nu$ is selected by a wide range of trials from the 50 parameters $\{2^{-24}, 2^{-23}, \cdots, 2^{24}, 2^{25}\}$. The best value $\nu = 2^{-6}$ is selected. The simulation results are given in table 4. Clearly, IELSSVM achieves better performance than OS-ELM on real overlay prediction problem.

**Table 4.** ComparIson for real overlay prediction problem

| Datasets | Attribute name | Algorithms | Nodes | Training RMSE($10^{-3}$) | | Testing RMSE($10^{-3}$) | |
|---|---|---|---|---|---|---|---|
| | | | | Mean | Std | Mean | std |
| Real overlay prediction dataset | X-offset | OS-ELM (RBF) | 100 | 19.3 | 0.11 | 27.2 | 0.70 |
| | | IELSSVM | 100 | 20.2 | 0.19 | 25.5 | 0.29 |
| | Y-offset | OS-ELM (RBF) | 100 | 17.4 | 0.98 | 27.9 | 1.5 |
| | | IELSSVM | 100 | 18.3 | 0.10 | 22.0 | 0.27 |

In order to further evaluate the performance of the overlay prediction, the error curve of IELSSVM of all the data points is given in figure 1. Clearly, the prediction error of all the data is within the interval of [-0.1,0.1], which indicates that the prediction results can meet the accuracy requirements and can be used as the prediction model for integrated scheduling.

In actual integrated scheduling application, the overlay performance of each machine to process each lot is predicted by the IELSSVM algorithm. If the predicted value exceeds the permissible range predetermined, the lot should not be processed by the machine. The prediction results will be regarded as additional constraints when optimizing the scheduling strategy. The scheduling strategy developed on the basis of quality prediction will keep consistent with the actual environments, thus effectively reducing unexpected quality events and improving the scheduling performance.

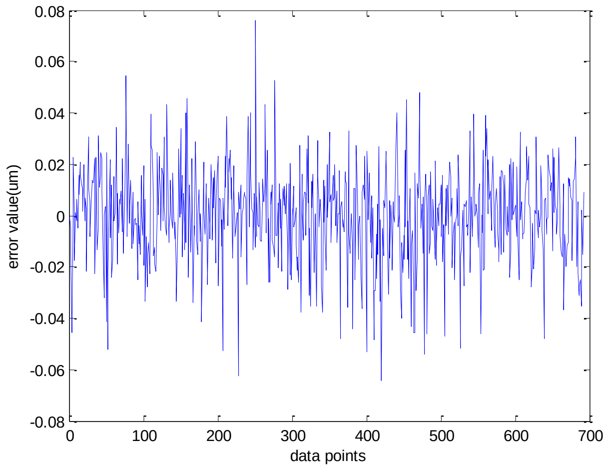


**Fig. 1.** The overlay error curve for all the real data points of lithography in semiconductor production line

## 6. Conclusion

In this paper, a novel Incremental Extreme Least Square Support Vector Machine for quality prediction and integrated scheduling of semiconductor production line is presented, in which data is first mapped into ELM feature space and then minimized under the structural risk principal like LSSVM. An incremental formula is given to accommodate the actual application where data may come one by one or chunk by chunk. Comparisons with the OS-ELM on classification and regression benchmark problems and real overlay prediction problems show that IELSSVM outperforms OS-ELM in most cases. That reason lies in that the IELSSVM is constructed on the basis of Structural Risk Minimization principal, thus avoiding over-fitting which OS-ELM constructed on the basis of Empirical Risk Minimization principal cannot avoid.

**Acknowledgment.** This work was supported by the National Key Basic Research and Development Program of China (2009CB320602), the National Natural Science Foundation of China (No. 60834004, 61025018, 61021063, 61104172) and the National Science and Technology Major Project of China (2011ZX02504-008).

## References


1. Min Liu, Cheng Wu: Genetic algorithm using sequence rule chain for multi-objective optimization in re-entrant micro-electronic production line. Robotics and Computer-Integrated Manufacturing, 20: pp.225-236. (2004)
2. S. Haykin: Neural Nectworks: A comprehensive Foundation. Prentice Hall, New Jersey. (1999)
3. Corinna Cortes and Vladimir Vapnik: Support vector networks. Machine Learning, vol. 20, 3: pp. 273–297. (1995)
4. J. A. K. Suykens and J. Vandewalle: Least squares support vector machine classifiers. Neural Processing Letters, vol. 9, 3: pp. 293–300. (1999)
5. Glenn Fung and Olvi L Mangasarian: Proximal support vector machine classifiers. The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining , San Francisco, USA, pp.77-86. (2001)
6. Qiuge Liu, Qing He and Zhongzhi Shi: Extreme Support Vector Machine Classifier. Advances in Knowledge Discovery and Data Mining ,Lecture Notes in Computer Science, pp. 222-233. (2008)
7. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew: Extreme learning machine:Theory and applications. Neurocomputing, vol. 70, 1–3: pp. 489–501 (2006)
8. Huang, G.-B. , Zhou, H. , Ding, X. , Zhang, R. , "Extreme Learning Machine for Regression and Multiclass Classification ," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2011, Issue 99: 1-17
9. G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," IEEE Trans. Neural Netw., 2005, vol. 16, 1: pp. 57-67
10. Nan-Ying Liang, Guang-Bin Huang, P. Saratchandran, N.Sundararajan: A Fast and Accurate On-line Sequential Learning Algorithm for Feedforward Networks. IEEE Transactions on Neural Networks, vol. 17, 6: pp. 1411-1423. (2006)
11. N. Syed, H. Liu, and K. Sung. "Incremental learning with support vector machines," In Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Articial Intelligence(IJCAI-99), Stockholm, Sweden, 1999
12. S. Ruping, "Incremental learning with support vector machines," Proceedings of the IEEE International Conference on Data Mining, 2001, pp:641-642
13. M. Espinoza, J. A. K. Suykens, and B. De Moor, "Fixed-size least squares support vector machines: A large scale application in electrical load forecasting," Computational Management Science-Special Issue on Support Vector Machines, 2006, vol. 3, 2: pp. 113-129
14. Gene H. Golub ,Charles F. van Van Loan: Matrix Computations, 3rd ed. The Johns Hopkins University Press, Baltimore. (1996)
15. http://archive.ics.uci.edu/ml/


**Lu Guo** received his B.Eng. degree from Harbin Institute of Technology in 2001, Harbin, China and his M.Eng degree from the Second Academy of China Aerospace in 2004, Beijing，China. Now he is working toward his Ph.D. degree in the Department of Automation, Tsinghua University, Beijing, China. His research interests include scheduling optimization, extreme learning machines, machine learning and signal processing.

**Jinghua Hao** received Ph.D. degree of Control Science and Engineering in Tsinghua University. His research interests are modelling, simulation and optimization methods for complex manufacturing process.He has published more than 10 papers. He participated in several projects of the National program of China.

**Min Liu** received the Ph.D. degree from Tsinghua University, Beijing, China, in 1999. He is currently a Professor with the Department of Automation, Tsinghua University, Associate Director of Automation Science and Technology Research Department of Tsinghua National Laboratory for Information Science and Technology, Director of Control and Optimization of Complex Industrial Process, Tsinghua University, Director of China National Committee for Terms in Automation Science and Technology, Director of Intelligent Optimization Committee of China Artificial Intelligence Association. His main research interests are in optimization scheduling of complex manufacturing process and intelligent operational optimization of complex manufacturing process or equipment. He led more than 20 important research projects including the project of the National 973 Program of China, the project of the National Science and Technology Major Project of China, the project of the National Science Fund for Distinguished Young Scholars of China, the project of the National 863 High-Tech Program of China, and so on. He has published more than 100 papers and a monograph supported by the National Defense Science and Technology Book Publishing Fund. Dr. Liu won the National Science and Technology Progress Award.

*Received: February 23, 2012; Accepted: November 20, 2012.*