

PRINCIPAL COMPONENTS BASED SUPPORT VECTOR REGRESSION MODEL FOR ON-LINE INSTRUMENT CALIBRATION MONITORING IN NPPS

IN-YONG SEO*, BOK-NAM HA, SUNG-WOO LEE, CHANG-HOON SHIN and SEONG-JUN KIM¹

Transmission and Distribution Laboratory, KEPCO Research Institute

65 Munji-Ro, Yuseong-Gu, Daejeon 305-760, Republic of Korea

¹ Department of Industrial Engineering, Kangnung National University

120 Daehangno Gangneung-shi, Gangwon-do 210-702, Republic of Korea

*Corresponding author. E-mail : iyseo@kepri.re.kr

Received September 21, 2009

Accepted for Publication March 2, 2010

In nuclear power plants (NPPs), periodic sensor calibrations are required to assure that sensors are operating correctly. By checking the sensor's operating status at every fuel outage, faulty sensors may remain undetected for periods of up to 24 months. Moreover, typically, only a few faulty sensors are found to be calibrated. For the safe operation of NPP and the reduction of unnecessary calibration, on-line instrument calibration monitoring is needed. In this study, principal component-based auto-associative support vector regression (PCSVR) using response surface methodology (RSM) is proposed for the sensor signal validation of NPPs. This paper describes the design of a PCSVR-based sensor validation system for a power generation system. RSM is employed to determine the optimal values of SVR hyperparameters and is compared to the genetic algorithm (GA). The proposed PCSVR model is confirmed with the actual plant data of Kori Nuclear Power Plant Unit 3 and is compared with the Auto-Associative support vector regression (AASVR) and the auto-associative neural network (AANN) model. The auto-sensitivity of AASVR is improved by around six times by using a PCA, resulting in good detection of sensor drift. Compared to AANN, accuracy and cross-sensitivity are better while the auto-sensitivity is almost the same. Meanwhile, the proposed RSM for the optimization of the PCSVR algorithm performs even better in terms of accuracy, auto-sensitivity, and averaged maximum error, except in averaged RMS error, and this method is much more time efficient compared to the conventional GA method.

KEYWORDS : Support Vector Regression, On-line Calibration Monitoring, Principal Component, Response Surface Methodology

1. INTRODUCTION

For the past two decades, the nuclear industry has attempted to move toward a condition-based maintenance philosophy using new technologies developed to monitor the condition of plant equipment during operation. Specifically, techniques have been developed to monitor the condition of sensors and their associated instrument loops while a plant is operating. Traditionally, instruments must be recalibrated at each refueling outage in accordance with nuclear regulations. One concern with periodic calibrations is that only the sensor's operating status is checked at every fuel outage, meaning that faulty sensors may remain undetected for periods of up to 24 months. Also, the traditional periodic maintenance method can lead to equipment damage, incorrect calibrations due to adjustments made under nonservice conditions, increased radiation

exposure of maintenance personnel, and possibly, increased downtime. In fact, recent studies have shown that less than 5% of the process instruments are in a degraded condition that requires maintenance [13]. Therefore, plant operators are interested in finding ways to monitor sensor performance during operation and to manually calibrate only sensors that require correction. Hence, in this study we developed an OLM model for tracking instrument performance.

Considerable research efforts have been devoted to the development of OLM algorithms. The application of artificial intelligence techniques to NPPs was investigated for instrument condition monitoring [1]. The Multivariate State Estimation Technique (MSET) was developed in the late 1980s [2], and Plant Evaluation and Analysis by Neural Operators (PEANO) was developed by researchers at the Halden Reactor Project in Norway [4]. The underlying

algorithm for PEANO is AANN, and it has been tested by several nuclear facilities throughout Europe.

The SVR algorithm developed by Vapnik [5,6] is based on the statistical learning theory. The support vector machine (SVM) method was applied for data-based state estimation in nuclear power reactors [7,8,19]. In these papers, the Vector State Estimation Technique (VSET) kernel and Hermitian kernel were implemented, yielding promising results. Meanwhile the fuzzy SVR was developed for an inferential modeling of the feedwater flow rate in pressurized water reactors [16,17]. Recently, PCA-SVR was applied to many prediction areas and showed good performance [9-11]. In this paper, we applied the PCA-SVR algorithm for an auto-associative model for instrument calibration monitoring. The objective of the present study is to propose a very efficient and accurate optimization method for the hyperparameters of SVR.

In this paper, we propose a PCSVR for OLM and signal validation, expanding the concept of PCA-SVR for an auto-associative model. This approach exploits the attractive merits of a principal component analysis (PCA) for extracting predominant feature vectors and AASVR. Moreover, RSM is proposed to optimize the SVR hyperparameters and is compared to the conventional GA. With the use of real plant startup data from the Kori NPP Unit 3, the model's performance was compared with those of AANN, which is a fundamental algorithm of PEANO, and AASVR in terms of accuracy and sensitivity.

2. PCSVR MODEL

The outputs of an auto-associative model are trained to emulate its inputs over an appropriate dynamic range. An auto-associative model will estimate the correct input values using the correlations embedded in the model during its training. The estimated correct value from the auto-associative model can then be compared to the actual process parameter to determine if a sensor has drifted or has been degraded by another fault type.

Figure 1 shows a schematic diagram of the proposed PCSVR algorithm for modeling measurements in NPPs. The PCSVR is composed of PCA and AASVR. A PCA is first applied to extract principal components from the

input measurements for dimensionality reduction. The predominant feature vectors in conjunction with the input measurements are then fed into the AASVR algorithm to formulate regression models. Each SVR model in PCSVR has the same three hyperparameters determined by GA and RSM.

Two parameters, accuracy and sensitivity, were used for the performance evaluation of the algorithm [3]. The accuracy metric is simply defined as the mean squared error (MSE) between the model's predictions and the target values. The equation for a single variable is simply

$$A = \frac{1}{N} \sum_{i=1}^N (\hat{x}_i - x_i)^2, \quad (1)$$

where N is the number of test observations, \hat{x}_i is the model prediction of the i th test observation, and x_i is the i th observation of the test data. Although this metric is termed "accuracy," it is actually a measure of error, and a low value is desired. A robust model would produce few to no changes in any of its outputs for errors in each of its inputs.

Model sensitivity is generally defined as a measure of the change in the prediction of the i th variable (\hat{x}_i) produced by a change in its respective input (x_i):

$$S_i = \frac{\Delta \hat{x}_i}{\Delta x_i}.$$

Auto-sensitivity (S_A) is a measure of an empirical model's ability to make correct sensor predictions when its respective input sensor value is incorrect due to some sort of fault. Therefore, this metric involves the following values: sensor i 's prediction with no fault in the input \hat{x}_i , sensor i 's prediction with a faulted input \hat{x}_i^{drift} , sensor i 's unfaulted input value x_i , and sensor i 's drifted input value x_i^{drift} . Using these definitions, the auto-sensitivity for sensor i is found in Eq. (2).

$$S_{A,i} = \frac{1}{N} \sum_{k=1}^N \left| \frac{\hat{x}_{ki}^{drift} - \hat{x}_{ki}}{x_{ki}^{drift} - x_{ki}} \right|, \quad (2)$$

$$S_{C,ij} = \frac{1}{N} \sum_{k=1}^N \left| \frac{\hat{x}_{kj}^{drift} - \hat{x}_{kj}}{x_{kj}^{drift} - x_{kj}} \right| \text{ for } i \neq j \quad (3)$$

The next performance metric is cross-sensitivity (S_C). This value measures the effect a faulty sensor input (i) has on the predictions of sensor (j). This is illustrated in Eq. (3), in which j is the index of the unfaulted variable whose spillover metric is being calculated.

2.1 Principal Component Analysis

In multivariate regression, highly correlated data could

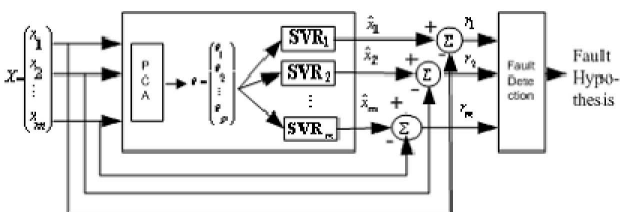


Fig. 1. Schematic Diagram of PCA-AASVR

result in a multicollinearity problem and are prone to producing unstable regression estimates. It is, therefore, desirable to reduce the data to a smaller subset of predominant feature vectors, which give rise to more stable estimates of regression coefficients [14].

The PCA method involves linearly transforming the input space into an orthogonal space that can be chosen from a lower dimension with a minimal loss of information and is used to reduce the dimension of input space into the AASVR system. A lower dimensional input space will reduce the time necessary to train the AASVR system. The PCA method can be chosen as a method of preprocessing data to extract uncorrelated features from the data.

Given a signal vector \mathbf{x} of m dimensions, $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_m]^T$, its true mean and covariance matrix are replaced with the sample mean \mathbf{m} and the sample covariance matrix \mathbf{S} because they are seldom known. The eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_m$ and the corresponding orthonormal eigenvectors $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_p$ of the covariance matrix \mathbf{S} are calculated and then arranged according to their magnitude: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$.

The eigenvectors $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_p$ are called the principal components. The eigenvalues are proportional to the amount of variance represented by the corresponding principal component. The transformation to the principal component space can be written as:

$$\boldsymbol{\theta} = \mathbf{x}^T \mathbf{P}, \quad (4)$$

where $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_p]$.

The feature vector $\boldsymbol{\theta}$ can be transformed back into the original data vector \mathbf{x} without a loss of information as long as the number of features, p , is equal to the dimension of the original space, m . For $p < m$, some information is usually lost. The objective is to choose a small p that does not lose much information. Generally, there is variability in the data with random noise; in most cases, this variability is of no concern and, by transformation to a lower dimensional space, this noise can sometimes be removed [15].

2.2 Auto-Associative SVR

In this paper, the SVM regression method is used for signal validation of the measurements in NPPs. The SVM regression entails nonlinearly mapping the original data into a higher dimensional feature space. Hence, given a set of data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n \in \mathbb{R}^m \times \mathbb{R}^m$, where \mathbf{x}_i is the input vector to SVMs, \mathbf{y}_i is the actual output vector, and n is the total number of data patterns. The multivariate regression function for each output signal is approximated by the following function:

$$y_k = f_k(\mathbf{x}) = \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{x}) + b_k, \quad (5)$$

where $\mathbf{w}_k = [w_{k1}, w_{k2}, \dots, w_{kn}]^T$, $\boldsymbol{\phi} = [\phi_1, \phi_2, \dots, \phi_n]^T$, $k=1, 2, \dots, m$, and m is the number of sensor measurements. Also, the

function $\boldsymbol{\phi}(\mathbf{x})$ is called a feature. Equation (5) is a nonlinear regression model because the resulting hyper-surface is a nonlinear surface hanging over the m -dimensional input space. The parameters \mathbf{w} and b are a support vector weight and a bias that are calculated by minimizing the following regularized risk function:

$$R(\mathbf{w}_k) = \frac{1}{2} \mathbf{w}_k^T \mathbf{w}_k + C_k \sum_{i=1}^n L_k(y_{k,i}), \quad (6)$$

where

$$L_k(y_{k,i}) = \begin{cases} 0, & |y_{k,i} - f_k(\mathbf{x})| < \varepsilon_k \\ |y_{k,i} - f_k(\mathbf{x})| - \varepsilon_k, & \text{otherwise} \end{cases}. \quad (7)$$

The first term of Equation (6) characterizes the complexity of the SVR models. C_k and ε_k are user-specified parameters, and $L_k(y_{k,i})$ is called a ε -insensitive loss function [12]. The loss equals zero if the estimated value is within an error level, and for all other estimated points outside the error level, the loss is equal to the magnitude of the difference between the estimated value and the error level. That is, minimizing the regularized risk function is equivalent to minimizing the following constrained risk function:

$$R(\mathbf{w}, \xi, \xi^*) = 1/2 \mathbf{w}_k^T \mathbf{w}_k + C_k \sum_{i=1}^n (\xi_{k,i} + \xi_{k,i}^*) \quad (8)$$

subject to the constraints

$$\begin{aligned} y_{k,i} - \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{x}) - b_k &\leq \varepsilon_k + \xi_{k,i} \\ \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{x}) + b_k - y_{k,i} &\leq \varepsilon_k + \xi_{k,i}^* \\ \varepsilon_k, \xi_{k,i}, \xi_{k,i}^* &\geq 0 \quad \text{for } i = 1, 2, \dots, n \end{aligned} \quad (9)$$

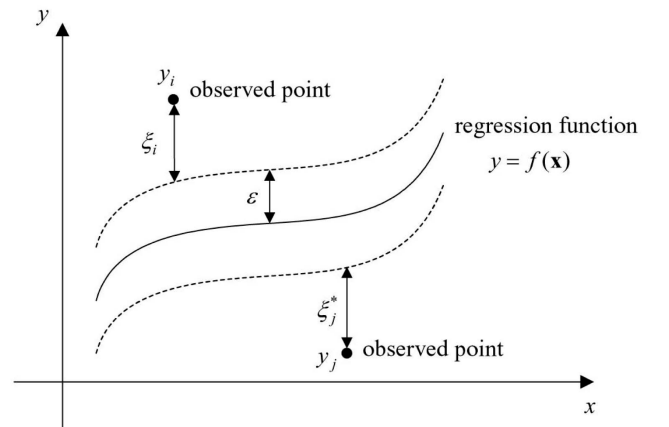


Fig. 2. Parameters for the SVR Models

where the constant C determines the trade-off between the flatness of $f(x)$ and the amount up to which deviations larger than ε are tolerated, and ξ and ξ^* are slack variables representing upper and lower constraints on the outputs of the system and take positive values.

The constrained optimization problem can be solved by applying the Lagrange multiplier technique to (8) and (9), followed by the use of a standard quadratic programming technique. Finally, the regression function of (5) becomes

$$y_k = \sum_{i=1}^n (\lambda_{k,i} - \lambda_{k,i}^*) K(\mathbf{x}_i, \mathbf{x}) + b_k^*, \quad (10)$$

where $K(\mathbf{x}_i, \mathbf{x}) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x})$ is called a kernel function.

By using different kernel functions for inner product evaluations, various types of nonlinear models in the original space can be constructed. The kernel function used in this study is the following radial basis function:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left[-\frac{(\mathbf{x} - \mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i)}{2\sigma^2}\right], \quad (11)$$

where σ is the kernel function parameter.

The bias, b , is calculated as follows:

$$b_k^* = -\frac{1}{2} \sum_{i=1}^n (\lambda_{k,i} - \lambda_{k,i}^*) [K(\mathbf{x}_r, \mathbf{x}_i) + K(\mathbf{x}_s, \mathbf{x}_i)], \quad (12)$$

where \mathbf{x}_r and \mathbf{x}_s are called support vectors (SVs) and are data points positioned at the boundary of the ε -insensitivity zone.

2.3 PCA-Based AAVR

By replacing θ derived from Section 2.1 with \mathbf{x} , we can combine PC and AASVR as follows:

$$y_k = f_k(\theta) = \sum_{i=1}^n (\lambda_{k,i} - \lambda_{k,i}^*) K(\theta_i, \theta) + b_k^* \quad (13)$$

$$b_k^* = -\frac{1}{2} \sum_{i=1}^n (\lambda_{k,i} - \lambda_{k,i}^*) [K(\theta_r, \theta_i) + K(\theta_s, \theta_i)]. \quad (14)$$

The three most relevant design parameters for the PCSVR model are the insensitivity zone, ε , the regularization parameter, C , and the kernel function parameter, σ . An increase in the insensitivity zone, ε , reduces the accuracy requirements of the approximation and allows a decrease in the number of SVs. In addition, an increase in the regularization parameter, C , reduces larger errors, thereby minimizing the approximation error. The kernel function parameter σ determines the sharpness of the radial basis kernel function [16].

2.4 Model Regularization by Response Surface Methodology

The PCSVR model was optimized by a process of learning from available data. As mentioned, the performance of the PCSVR model depends strongly on the three major design parameters: the insensitivity zone, ε , the regularization parameter, C , and the kernel function parameter, σ . Therefore, these parameters must be optimized in order to maximize the performance of the PCSVR model. Response surface methodology (RSM) is a good alternative to efficiently determine the optimal values of SVR hyperparameters [6]. This method is very efficient in terms of time compared to the genetic algorithm.

2.4.1 Conventional Genetic Optimization

The genetic algorithm is widely used for the optimization of SVR parameters. Although the GA is less susceptible to being stuck at local minima than other search methods since it starts from many points, simultaneously climbing many peaks in parallel, it requires too much time for the process, especially when the amount of observation data increases. We used the GA for comparison with our proposed RSM method.

To optimize three parameters in SVR, we encoded them as a bit string in each chromosome of the genetic algorithm. Our specified multiple objectives were to minimize the root mean squared (RMS) error and the maximum error, and we achieved these objectives by maximizing the following fitness function [16]:

$$F = \exp(-w_1 E_1 - w_2 E_2 - w_3 E_3 - w_4 E_4), \quad (15)$$

where w_1 , w_2 , w_3 , and w_4 are the weighting coefficients and E_1 , E_2 , E_3 , and E_4 are defined as follows:

$$E_1 = \sqrt{\frac{1}{m \cdot N_t} \sum_{i=1}^m \sum_{k=1}^{N_t} (y_i^t(k) - \hat{y}_i^t(k))^2} \quad (16)$$

$$E_2 = \sqrt{\frac{1}{m \cdot N_o} \sum_{i=1}^m \sum_{k=1}^{N_o} (y_i^o(k) - \hat{y}_i^o(k))^2} \quad (17)$$

$$E_3 = \max_i \left\{ \max_k \{y_i^t(k) - \hat{y}_i^t(k)\} \right\} \quad (18)$$

$$E_4 = \max_i \left\{ \max_k \{y_i^o(k) - \hat{y}_i^o(k)\} \right\}. \quad (19)$$

The variable $y_i(k)$ denotes the measured (target) output of sensor i , and the variable $\hat{y}_i(k)$ denotes the output predicted from the PCSVR model. The superscripts, t and o , indicate the training and optimization data, respectively, and N_t

and N_o represent the number of training and optimization data points. Meanwhile m denotes the number of sensors of input and the subscript i indicates the sensor number. The design parameters of the genetic algorithm that optimizes the PCSVR model are the crossover probability, the mutation probability, and the population size. Note that our objectives are different from those in [16] in minimizing the RMS error and the maximum error for all sensor outputs.

2.4.2 Proposed RSM Optimization

The method we proposed is a statistical design of experiments called RSM to optimize the design parameters of the PCSVR model. RSM is an efficient method, which involves a reduced number of experiments, for fitting a second-order multiple regression model and then finding an optimum parameter setting.

Central composite design (CCD) is the most popular among all the experimental designs for RSM. CCD usually consists of three types of experimental points: 2^k factorial points, $2k$ axial points, and n_c center points, where k implies the number of parameters. These points are illustrated in Figure 3. For the sake of constructing a rotatable design, α is typically chosen as $\alpha = \sqrt[4]{2^k}$.

Note that, because $k=3$, the axial value is obtained by $k=2^{3/4}=1.6818$ and there are three repetitions at the center point.

For every experimental point, each PCSVR model was constructed, optimization data were inputted into the model, and then the corresponding MSE was measured by

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (z_{ij} - \hat{z}_{ij})^2, \quad (20)$$

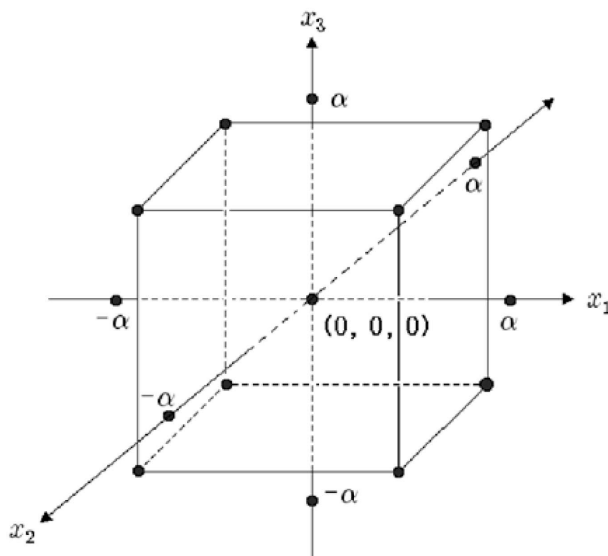


Fig. 3. Experimental Points in CCD when $k=3$

where z_{ij} is the j th input value of the i th sensor in the normalized scale and \hat{z}_{ij} is its PCSVR estimator. The MSE above is a measure of model accuracy; therefore, a smaller MSE implies better accuracy.

3. APPLICATION OF THE PCSVR

The proposed algorithm was confirmed with the real plant startup data of the Kori Nuclear Power Plant Unit 3. These data are values measured from the primary and secondary systems of the NPP. The data is derived from the following 11 types of measured signals: the reactor power (Sensor 1), the pressurizer water level (Sensor 2), the steam generator (SG) steam flow rate (Sensor 3), the SG narrow range level (Sensor 4), the SG pressure (Sensor 5), the SG wide-range level (Sensor 6), the SG main feedwater flow rate (Sensor 7), the turbine power (Sensor 8), the charging flow rate (Sensor 9), the residual heat removal flow rate (Sensor 10), and the reactor head coolant temperature (Sensor 11). All 11 signals were the target output of the PCSVR model, and these were input signals for the PCSVR model.

The data were sampled every 1 minute for about 38 hours. The total observation number of measurement data is 2,290, which were divided into five subsets of equal size, i.e., one training subset, one test subset, and three optimization subsets. The total dataset was indexed using Arabic numerals, i.e., $i = 1, 2, \dots, 2,290$. 458 patterns with indices, $i=5j+3$, $j=0, 1, \dots, 457$, named KR3 were used to train SVRs to capture the quantitative relation between 11 inputs and outputs. The first subset KR1, which has indices of $5j+1$, was used to test the model, while the remaining three subsets (KR2, KR4, KR5), which have indices of $5j+2$, $5j+4$, $5j+5$, were used for the optimization. Note that data in all subsets were sampled every 5 minutes because the original dataset was divided into five subsets. All the data subsets were normalized in each dimension for modeling and were named z1 through z5. They were de-normalized after the prediction process for the original signals.

Let $(\theta_1, \theta_2, \dots, \theta_{11})$ denote PCs obtained by applying PCA to the above plant data. As noted earlier, variance is used in selecting dominant PCs. We found that θ_1 is the most dominant PC and explains about 84.12% of the total variation in the original data. However, in order to minimize loss of information, seven PCs are considered in this study. The selected PCs explain more than 99.98% of the total variation. The loss of information is less than 0.1%.

3.1 Optimization Using the Genetic Algorithm

The first method we adopted to optimize the design parameters of the PCSVR model was the conventional genetic algorithm. In order to optimize the proposed PCSVR model with the genetic algorithm (GA-PCSVR), we set the parametric values of the genetic algorithm as 1

for the crossover probability, 0.05 for the mutation probability, and 20 for the population size. The optimized parameters are as follows:

$$\varepsilon = 4.8283 \times 10^{-4}, C = 122.8180, \sigma = 0.8568.$$

3.2 Optimization Using RSM

Experimental ranges of σ , ε , and C for the RSM are 0.2-2.0, 0.0005-0.05, and 0.1-10.0, respectively. The experimental design for PCSVR parameter regularization is shown in Table 1.

In order to check the statistical significance of the fitted model, an analysis of variances (ANOVA) is conducted, and the results are shown in Table 2. Note that for the sake of convenience for the statistical data analysis, a natural log of MSE is used rather than the original MSE. Since P -value approaches zero, we can conclude that, although three interaction terms are pooled into the error

term, the second order regression model between MSE and model parameters is statistically very significant. This suggests that, prior to evaluating the model performance, parameter regularization is of high importance. Though not shown in the table, the coefficients of determination can be obtained as $R^2=94.9\%$ and $R^2_{adj}=91.8\%$, respectively.

From the subsequent analysis results, it is shown that σ and ε have a significant effect on MSE, while C is of little significance. A response surface plot of $\log(MSE)$ versus σ and ε is depicted in Figure 4. From the plot, we can find where the MSE is minimized. The optimized parameters are as follows:

$$\sigma=1.4909, \varepsilon=0.0005, C=7.60.$$

3.3 PCSVR Modeling Using RSM

Empirical model building was carried out for the proposed PCSVR using RSM (RSM-PCSVR). This was accomplished with the parameter settings found in Section 3.2. The numbers of support vectors needed for each SVR are 383 (83.6%), 222 (48.5%), 378 (82.5%), 101 (22.1%), 332 (72.5%), 207 (45.2%), 418 (91.3%), 383 (83.6%), 162 (35.4%), 102 (22.3%), and 433 (94.5%). The average number of support vectors is 283 (61.8%). Data subset z3 was used for the training of the PCSVR

Table 1. Experimental Design for SVR Parameter Regularization

No	Sigma	Epsilon	C
1	0.564856	0.010534	2.1067
2	1.635144	0.010534	2.1067
3	0.564856	0.039967	2.1067
4	1.635144	0.039967	2.1067
5	0.564856	0.010534	7.9933
6	1.635144	0.010534	7.9933
7	0.564856	0.039967	7.9933
8	1.635144	0.039967	7.9933
9	0.2	0.02525	5.05
10	2.0	0.02525	5.05
11	1.1	0.0005	5.05
12	1.1	0.05	5.05
13	1.1	0.02525	0.1
14	1.1	0.02525	10.0
15	1.1	0.02525	5.05
16	1.1	0.02525	5.05
17	1.1	0.02525	5.05

Table 2. Analysis of Variances

Source	Degrees of freedom	Sum of squares	Mean of squares	F	P
Regression	6	4.6883	0.7814	30.80	0.000
Residual	10	0.2537	0.0254		
Total	18	4.9420			

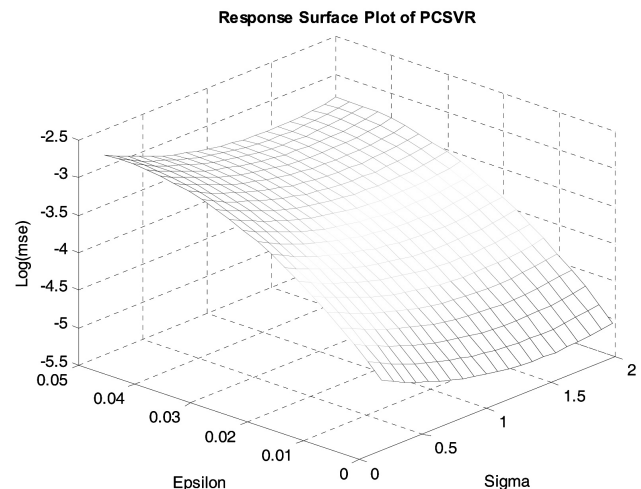
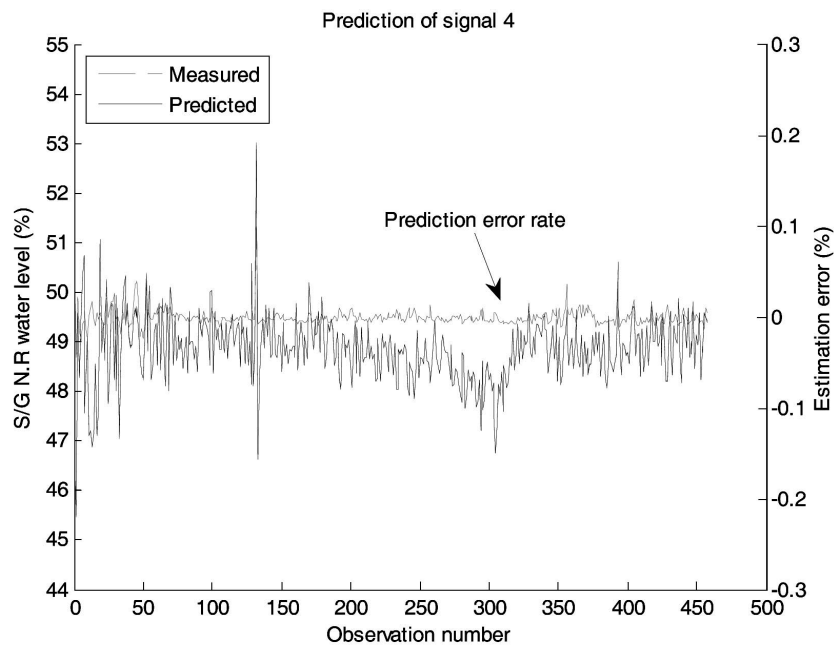


Fig. 4. Response Surface Plot of $\log(MSE)$ Versus σ and ε when C is Fixed at 7.6 for PCA-AAVSR

Table 3. RMS Error of the PCSVR Model Using RSM

Sensor No.	1	2	3	4	5	6	7	8	9	10	11	AVG
No. of SVs	383	222	378	101	332	207	418	383	162	102	433	283
Training data	No. of data	458	458	458	458	458	458	458	458	458	458	458
	RMS error (%)	0.239	0.024	0.372	0.013	0.014	0.005	0.399	0.319	0.008	0.007	0.132
	Max error (%)	1.671	0.187	4.391	0.249	0.064	0.051	3.238	3.276	0.066	0.053	1.225
Optimization data	No. of data	1374	1374	1374	1374	1374	1374	1374	1374	1374	1374	1374
	RMS error (%)	0.252	0.027	0.359	0.018	0.019	0.010	0.495	0.318	0.012	0.008	0.143
	Max error (%)	1.741	0.237	3.993	0.437	0.328	0.252	10.09	3.181	0.129	0.066	1.889
Test data	No. of data	458	458	458	458	458	458	458	458	458	458	458
	RMS error (%)	0.259	0.025	0.407	0.010	0.018	0.014	0.549	0.334	0.013	0.009	0.154
	Max error (%)	1.511	0.168	4.938	0.099	0.207	0.289	5.213	3.047	0.204	0.051	1.471


Fig. 5. Prediction of Sensor 4 Using RSM-PCSVR

model, and subset z1 was used for testing.

Table 3 summarizes the prediction accuracy of 11 signals for the RSM-PCSVR model. The relative RMS errors are compared with the rated values (100%, 100%, 2.0Mkg/hr, 100%, 100Kg/cm², 100%, 2.0Mkg/hr, 1000Mw, 30m³/hr, 100m³/hr, 330°C). The averaged relative RMS errors for the 11 sensors are 0.132% for the training data, 0.143% for the optimization data, and 0.154% for the test data. From Table 3 we can observe that the prediction accuracy of sensor 4 for the test data is the best among the 11 sensors, except for sensor 10, for which two-thirds of the data are zeros, while sensor 7 performs the worst.

Figure 5 shows the measured, predicted, and estimated errors of the SG narrow range water level by the RSM-PCSVR model. The predicted signal is accurate to a degree that we can hardly distinguish it from the measured signal.

Figure 6 shows the histograms of the estimation errors for the training data, the optimization data, and the test data, respectively. In Fig. 6, it is shown that the relative RMS errors of sensor 4 compared with the rated value (100 %) are 0.0133%, 0.0183%, and 0.0103% for the training data, the optimization data, and the test data, respectively. The maximum relative RMS errors are 0.2494%, 0.4369%, and 0.0987%, respectively.

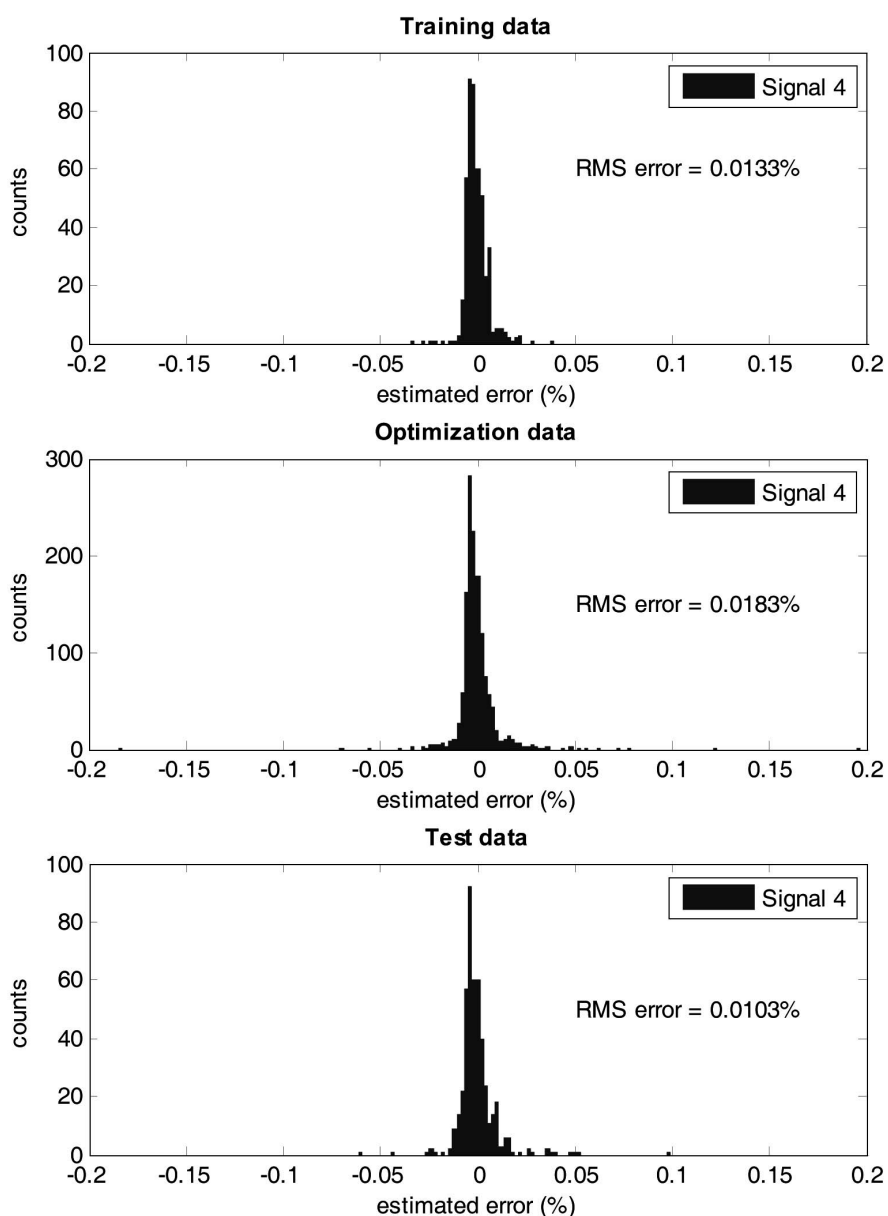


Fig. 6. SG Narrow Range Level Estimation Error Histogram for Training, Optimization and Test Data for Training, Optimization and Test Data

4. AASVR AND AANN

For the performance comparison, we constructed AASVR and AANN and tested them with the same datasets used for PCSVR.

4.1 AASVR Modeling Using GA

A schematic of the AASVR is shown in Fig. 7. Three parameters of AASVR were optimized by the genetic algorithm (GA-AASVR), and the parametric values of the genetic algorithm were set to the same values as those for the PCSVR described in Section 3.1.

The optimized parameters for GA-AASVR are as

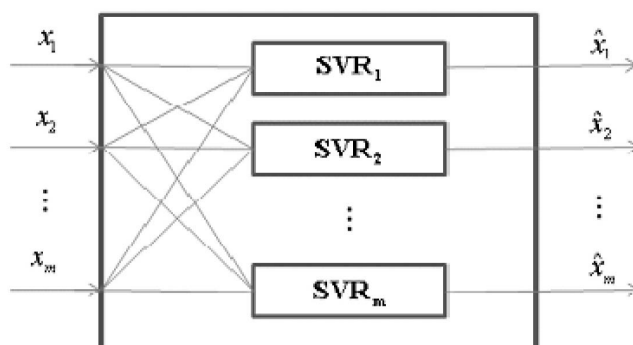


Fig. 7. Schematic Diagram of AASVR

follows:

$$\sigma=7.0145, \varepsilon=1.0207 \times 10^{-4}, C=2.7973 \times 10^3.$$

Note that the σ and C values are much bigger than those of GA-PCSVR, while ε is almost the same. This set of the three parameters was used for all 11 SVRs in GA-AASVR, as they were for the GA-PCSVR.

4.2 AASVR Modeling Using RSM

AASVR parameters were optimized by the RSM (RSM-AASVR) over the same ranges with RSM-PCSVR described in Section 3.2. The response surface plot of $\log(MSE)$ versus σ and ε for the RSM-AASVR is depicted in Figure 8. The optimum point of the response surface for RSM-AASVR is obtained as follows:

$$\sigma=1.40, \varepsilon=0.0005, C=6.30.$$

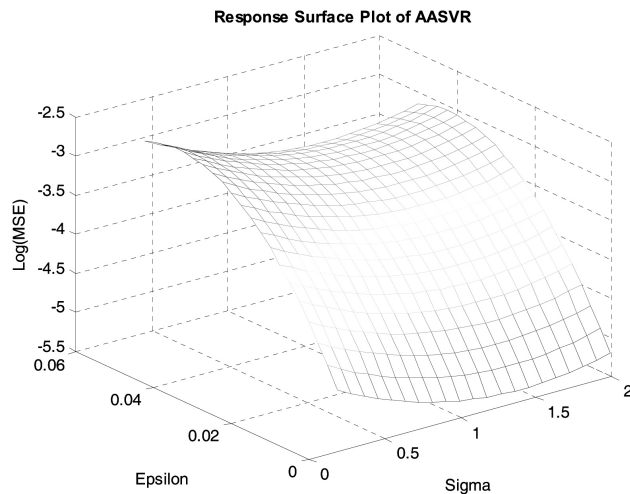


Fig. 8. Response Surface Plot of $\log(MSE)$ Versus σ and ε when C is Fixed at 6.3 for RSM-AASVR

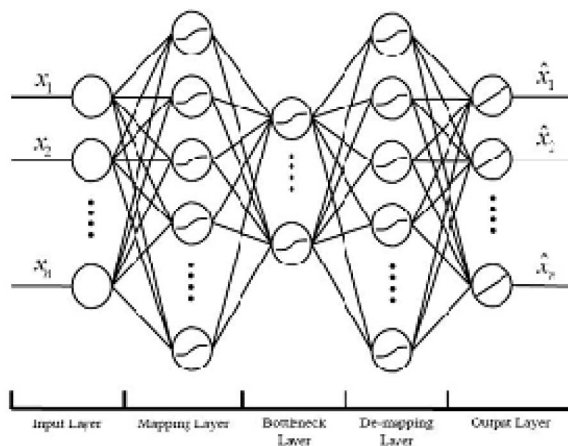


Fig. 9. AANN Architecture

4.3 AANN

The architecture of the AANN underlying algorithm of PEANO is presented in Fig. 9 along with neurons. AANN is composed of four major neuron layers, specific mapping, bottleneck, de-mapping, and output layers. The activation functions used in the processing elements are a hyperbolic tangent sigmoid function in the input layer and three hidden layers, whereas a linear activation function is used in the output layer. A rule of thumb that prevents the model from overfitting the data is that the total number of weights should be less than the number of patterns. The maximum number of neurons in each layer of AANN that meets this rule is [6,11,13], and, hence, we performed a combinatorial grid search of the model architectures for the optimum number of neurons in each layer using the optimization data. We found that the best architecture of the AANN model is [6,11,12], which gives the smallest error for the optimization data.

5. PERFORMANCE COMPARISON

Accuracy and sensitivity were used as measures to compare the performance of the three algorithms. Figure 10 shows the accuracies of the 11 sensors for the AANN, GA-AASVR, RSM-AASVR, GA-PCSVR, and RSM-PCSVR models using the normalized test data.

Figure 11 shows the averaged accuracies of the 11 sensors for the five algorithms. They are 1.363×10^{-5} , 5.195×10^{-9} , 0.126×10^{-5} , 1.148×10^{-5} , and 1.039×10^{-5} for AANN, GA-AASVR, RSM-AASVR, GA-PCSVR, and RSM-PCSVR,

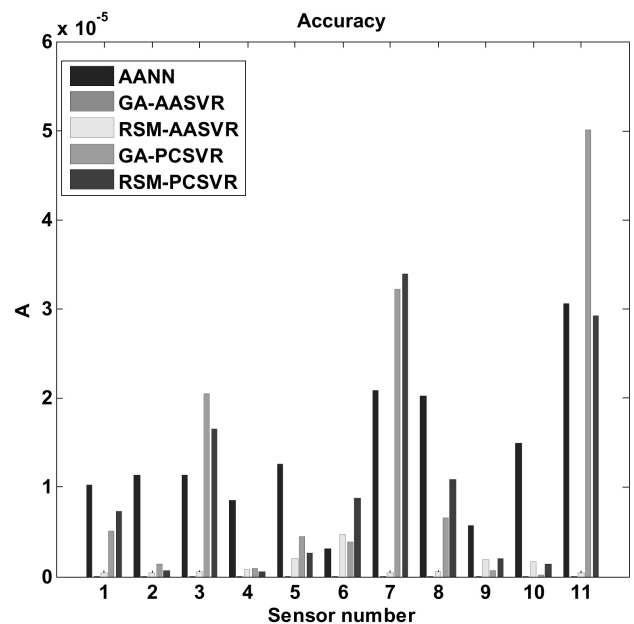


Fig. 10. Accuracy Comparison

and RSM-PCSVR, respectively. As shown in Figure 11, the GA-AASVR model gives the best accuracy among the five algorithms. This good accuracy of GA-AASVR is supposedly caused by the large bandwidth of the Gaussian radial basis kernel (σ).

One major purpose of OLM is to detect sensor drift as early as possible. This can be achieved by constructing

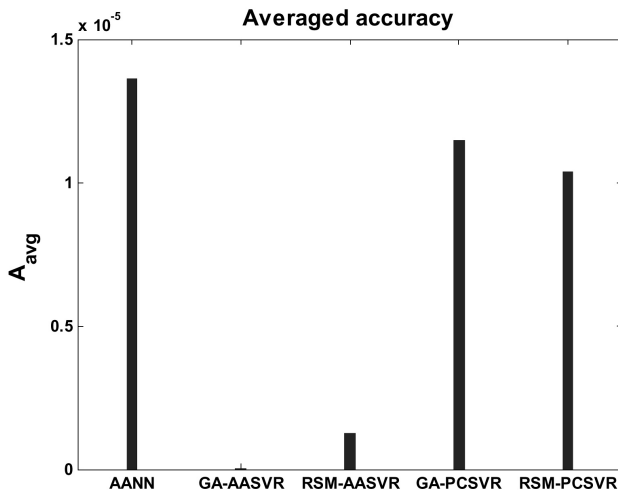


Fig. 11. Averaged Accuracies of Three Models

an auto-association model that is insensitive to drifted sensor input. The model sensitivity, which is a critical performance index for instrument calibration monitoring, therefore, should be sufficiently low. In order to investigate model sensitivity, we artificially degraded the SG feed water flow rate channel (unit: Mkg/hr). The degraded signal, shown in Fig. 12, linearly increases at a rate of 3.14% per day from the first observation, i.e. 5% positive drift at the end of the observation. Residuals are usually used to detect sensor drift. For example, when a residual exceeds a predetermined interval, it can be determined that a drift has occurred. The residuals ($\hat{x}_7 - x_7$) produced by five algorithms are depicted in Fig. 12.

We can see that AANN, GA-PCSVR, and RSM-PCSVR successfully detected the drift. On the contrary, the residuals of GA-AASVR and RSM-AASVR stay around zero for all of the drifted test data, which suggests that AASVR is relatively incapable of detecting a sensor drift at an early point in time.

Figure 13 presents a sensitivity comparison of the five algorithms. Fig. 13(a) shows the sensitivities of all the sensors caused by the artificially degraded SG feed water flow rate. Figs. 13(b) and (c) show the auto-sensitivities and the cross-sensitivities, respectively. The auto-sensitivities are 0.1408, 0.9972, 0.9205, 0.1516, and 0.1461 for AANN, GA-AASVR, RSM-AASVR, GA-PCSVR, and RSM-PCSVR, respectively. The auto-

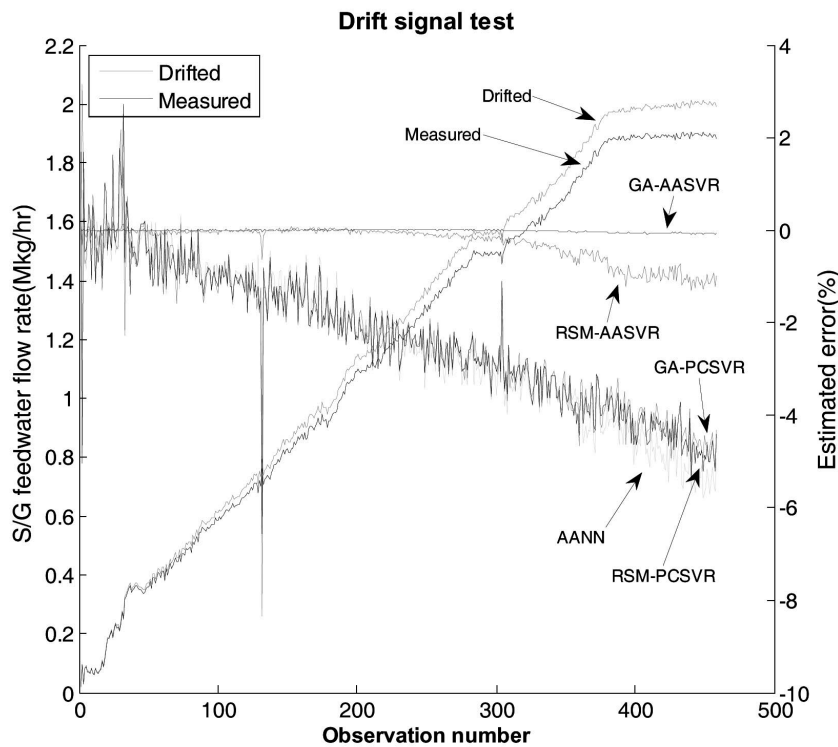


Fig. 12. Residuals of Five Algorithms

sensitivities of GA-AASVR and RSM-AASVR are almost 1, and this means that the drifted input and predicted output are almost the same as shown in Fig. 12. The averaged

cross-sensitivities are 0.0999, 0.0016, 0.0129, 0.0845, and 0.0826 for AANN, GA-AASVR, RSM-AASVR, GA-PCSVR, and RSM-PCSVR, respectively. The cross-sensitivity of GA-AASVR is the lowest.

The AASVR suffers from high auto-sensitivity, and PCA is helpful in improving the auto-sensitivity of AASVR; however, it would be difficult to attain cross-sensitivity improvement by using a PCA. Both GA-AASVR and RSM-AASVR provided low cross sensitivity and high auto-sensitivity. This means the prediction outputs of healthy sensors are not affected by the drifted sensor signal but the model output of the drifted sensor is very sensitive to the drifted input, i.e. the predicted output of the drifted sensor follows the drifted input, resulting in a residual of zero, and the fault cannot be detected. Such a model would be of little use in instrument calibration validation because its predictions follow the faulted inputs rather than correcting them.

Table 4 summarizes the performances of the five algorithms. Comparing the optimization methods of GA and RSM for PCSVR, RSM performs better in terms of accuracy, auto-sensitivity, and averaged maximum error, except in averaged RMS error. Most of all, RSM is more

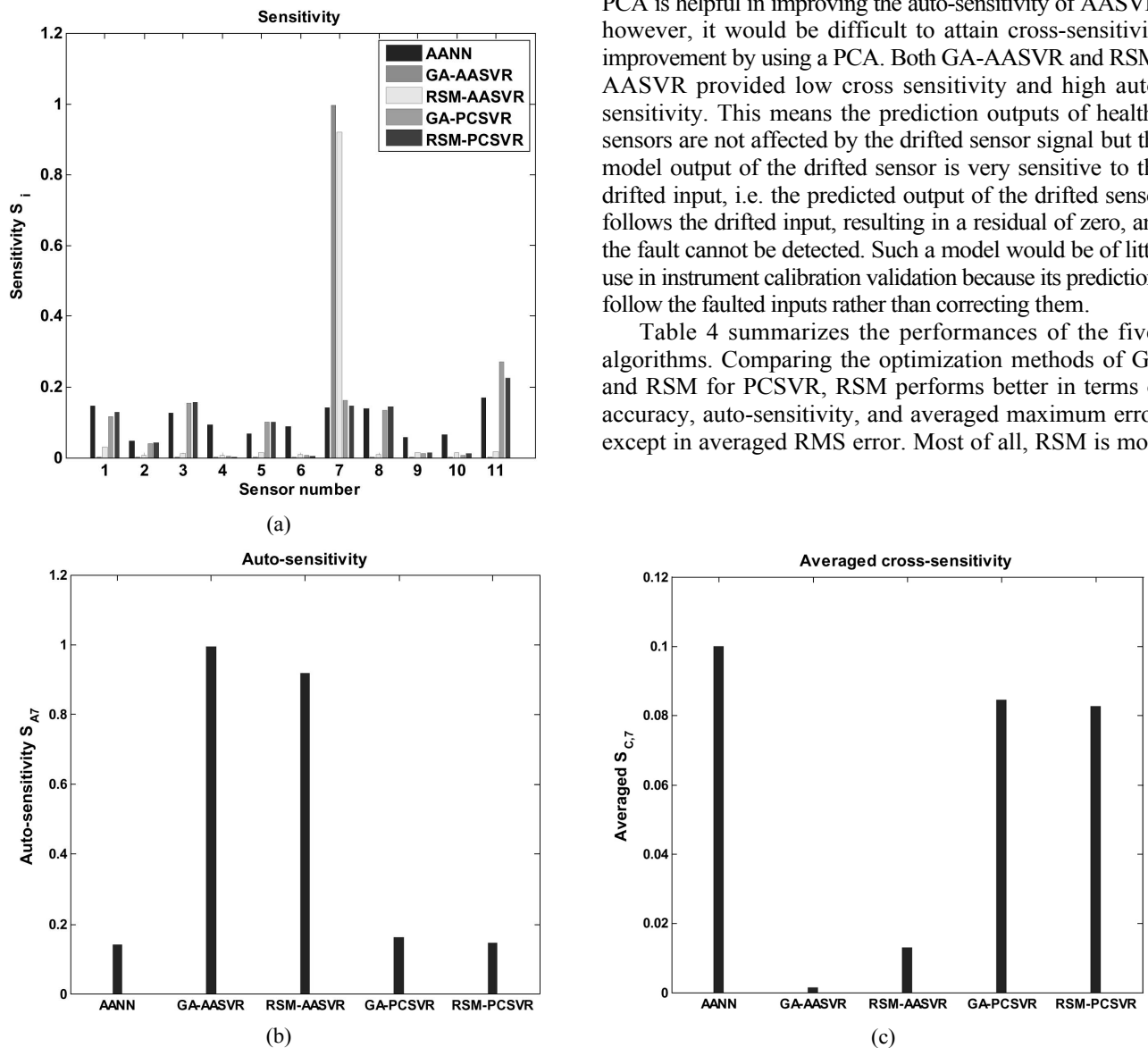


Fig. 13. Sensitivity Comparison. (a) Sensitivity of All Sensors. (b) Auto-sensitivities. (c) Cross-sensitivities

Table 4. Performance Comparison

	Optimi-zation	Exec. Time	Acc.*10 ⁻⁵	Auto Sens.	RMS err. (%)	Avg. Max. Err.(%)
AANN			1.363	0.1408	0.167	1.263
AASVR	GA	83hr	0.00052	0.9972	0.0029	0.0182
	RSM	8hr	0.126	0.9205	0.0330	0.3010
PCSVR	GA	84hr	1.148	0.1516	0.149	1.618
	RSM	8hr	1.039	0.1461	0.154	1.471

efficient in execution time, as shown in Table 4.

In conclusion, AASVRs using GA and RSM performs best in terms of accuracy and cross-sensitivity among the five algorithms, but the auto-sensitivity is almost 1. This implies that the AASVR is not suitable for sensor drift detection. Meanwhile, the proposed PCSVR shows a similar auto-sensitivity to the AANN, and its accuracy is better than that of the AANN. Moreover, the proposed RSM-PCSVR is more efficient than the conventional GA-PCSVR for the OLM.

6. CONCLUSIONS

In this paper, a PCSVR algorithm using RSM for optimization is proposed for the signal validation and instrument calibration monitoring of NPPs. The proposed algorithm utilizes PCA for extracting predominant feature vectors, auto-associative SVR for database statistical learning, and RSM for the optimization of SVR hyperparameters. The RSM-PCSVR model was applied to the data of Kori Nuclear Power Plant Unit 3, and the performance was compared with that of the AASVR using GA and RSM for optimization, and AANN models in terms of accuracy and sensitivity. The auto-sensitivity of AASVR is improved by around six times by using a PCA, resulting in good detection of sensor drift. Compared to AANN, accuracy and cross-sensitivity are better while the auto-sensitivity is almost the same. The proposed RSM for the optimization of PCSVR algorithm shows better performances in accuracy, auto-sensitivity, and averaged maximum error except in averaged RMS error, and this method is very time efficient compared to the conventional GA method. To increase the performance of the RSM-PCSVR, we can consider using other kernel functions or clustering methods, and this is a potential direction for future research.

ACKNOWLEDGEMENTS

The authors would like to express special gratitude to Prof. Man Gyun Na of Chosun University for his suggestions and the use of his genetic algorithm program.

REFERENCES

- [1] Upadhyaya, B. R., and E. Eryurek (1992), "Application of Neural Networks for Sensor Validation and Plant Monitoring", *Nuclear Technology*, 97, 170–176 (February 1992).
- [2] Mott, Y., and R. W. King (1987), *Pattern Recognition Software for Plant Surveillance*, U.S. DOE Report.
- [3] Wegerich, S. (2002), "Performance Comparison of Variable Selection and Grouping Algorithms", Technical Report, SmartSignal Corp.
- [4] Fantoni, P., S. Fignedy, and A. Racz (1998), "A Neuro-Fuzzy Model Applied to Full Range Signal Validation of PWR Nuclear Power Plant Data", FLINS-98, Antwerpen, Belgium.
- [5] C. Cortes and V. Vapnik. *Support vector networks*, Machine Learning 20: 273-297, 1995.
- [6] V. Vapnik. *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.
- [7] N. Zavaljevski and K. C. Gross. "Support Vector Machines for Nuclear Reactor State Estimation", ANS International Topical Meeting May 7-11, 2000, Pittsburgh, USA
- [8] A. V. Gribok, J. W. Hines, R. E. Uhrig. "Use of Kernel Based Techniques For Sensor Validation In Nuclear Power Plants", NPIC&HMIT 2000, Washington, DC, November, 2000.
- [9] Liu X, Chen HC, Liu TA, Li YL, Lu ZR, Lu WC. "Application of PCA-SVR to NIR prediction model for tobacco chemical composition", 2007 Dec; 27 (12):2460-3
- [10] Xuexiang Jin, Yi Zhang, and Danya Yao. "Simultaneously Prediction of Network Traffic Flow Based on PCA-SVR", LNCS 4492, pp. 1022–1031, 2007
- [11] X. G. Hua, Y. Q. Ni, J. M. Ko, F. ASCE and K. Y. Wong. "Modeling of Temperature–Frequency Correlation Using Combined Principal Component Analysis and Support Vector Regression Technique", *journal of computing in civil engineering* march/April 2007, 122-135
- [12] V. N. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer, 1995.
- [13] EPRI, "On-Line Monitoring of Instrument Channel Performance Volume 3: Applications to Nuclear Power Plant Technical Specification Instrumentation," Final Report # 1007930, EPRI, Palo Alto, CA. 2004
- [14] Rencher, A. C. *Methods of multivariate analysis*, 2nd Ed., Wiley, New York (2002).
- [15] M. G . Na, "A Neuro-Fuzzy Inference System for Sensor Failure Detection Using Wavelet Denosing, PCA and SPRT," *J. Korean Nucl. Soc.*, vol 33, no. 5, pp483-497, Oct. 2001.
- [16] M. G. Na, H. Y. Yang, D. H. Lim, "A Soft-sensing Model for Feedwater Flow Rate Using Fuzzy Support Vector Regression," *Nucl. Eng. Tech.*, vol 40, no. 1, pp69-76, Feb. 2008.
- [17] M. G. Na, I. J. Hwang, and Y. J. Lee, "Inferential Sensing and Monitoring for Feedwater Flowrate in Pressurized Water Reactors," *IEEE Trans. Nucl. Sci.*, vol.53, no. 4, pp. 2335–2342, 2006.
- [18] I. Y. Seo, and S. J. Kim, "An On-line Monitoring Technique Using Support Vector Regression and Principal Component Analysis" CIMCA 2008, Vienna, Austria, pp. 663–669. 2008.
- [19] O. Omitaomu, M. K. Jeong, A. Badiru, and J. W. Hines, "On-Line Support Vector Regression Approach for the Monitoring of Motor Shaft Misalignment and Feedwater Flow Rate," *IEEE Transactions on Systems, Man, Cybernetics, Part C*, 37(5), 962-970. 2007.