International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: http://www.tandfonline.com/loi/tprs20

On-line classifying process mean shifts in multivariate control charts based on multiclass support vector machines

Shichang Du , Jun Lv & Lifeng Xi

To cite this article: Shichang Du , Jun Lv & Lifeng Xi (2012) On-line classifying process mean shifts in multivariate control charts based on multiclass support vector machines, International Journal of Production Research, 50:22, 6288-6310, DOI: 10.1080/00207543.2011.631596

To link to this article: http://dx.doi.org/10.1080/00207543.2011.631596

Published online: 09 Jan 2012.

International Journal of Production Research
Vol. 50, No. 22, 15 November 2012, 6288–6310

Taylor & Francis
Taylor & Francis Group

On-line classifying process mean shifts in multivariate control charts based on multiclass support vector machines

Shichang Du[a]*, Jun Lv[b] and Lifeng Xi[ac]

[a]*Department of Industrial Engineering and Logistics Engineering, School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai, 200240, People's Republic of China;* [b]*School of Business, East China Normal University, Shanghai 200241, People's Republic of China;* [c]*State Key Lab of Mechanical System and Vibration, Shanghai, 200240, People's Republic of China*

(*Received 27 January 2011; final version received 3 October 2011*)

In multivariate statistical process control (MSPC), most multivariate control charts can effectively monitor anomalies based on overall statistic, however, they cannot provide guidelines to classify the source(s) of out-of-control signals. Classifying the source(s) of process mean shifts is critical for quality control in multivariate manufacturing process since the immediate identification of them can greatly help quality engineer to narrow down the set of possible root causes and take corrective actions. This study presents an improved particle swarm optimisation with simulated annealing-based selective multiclass support vector machines ensemble (PS-SVME) approach, in which some selective multiclass SVMs are jointly used for classifying the source(s) of process mean shifts in multivariate control charts. The performance of the proposed PS-SVME approach is evaluated by computing its classification accuracy. Simulation experiments are conducted and a real application is illustrated to validate the effectiveness of the developed approach. The analysis results indicate that the developed PS-SVME approach can perform effectively for classifying the source(s) of process mean shifts.

Keywords: multivariate statistical process control; support vector machine; ensemble classification; process mean shifts

1. Introduction

The ability to monitor and reduce process variation for quality improvement in manufacturing process plays a critical role in the success of one manufacturing enterprise in today's globally competitive marketplace (Montgomery 2001, Du *et al.* 2008). There are many processes in which the simultaneous monitoring of two or more related quality characteristics is necessary. Processes monitoring in which several related variables are of interest are collectively known as multivariate statistical process control (MSPC) (Woodall and Montgomery 1999, Bersimis *et al.* 2007). Multivariate control charts have been widely used as effective tools in MSPC. The main shortcoming of multivariate control charts is that they can detect an unusual event but do not directly provide the information required by a practitioner to determine which variables or combination of variables causes the out-of-control signals. This raises the issue of identification in relation to multivariate control procedure. Among various abnormal conditions, this study focuses on classifying the source(s) of process mean shifts. Mean shifts are defined as unanticipated sudden shifts in process mean vector. The primary possible causes for the mean shifts result from the introduction of new workers, machines or methods, a change in the measurement method or standard, etc.

Kourti and MacGregor (1996) reviewed several approaches to solve the identification problem. Guh and Shiue (2008) proposed a decision tree-based model for online detect mean shifts and classify the source(s) of the mean shifts. Some authors suggested using decomposition techniques for classifying the source(s) of the mean shifts, which are based mostly on T^2 statistic (Mason *et al.* 1995, 1997, Li *et al.* 2008, Verron *et al.* 2010).

Advanced automatic data inspection and collection techniques have been undergoing tremendous growth during the past few years, significantly transforming many aspects of manufacturing processes and providing unprecedented opportunities for applying various machine learning approaches into statistical process control (SPC). In the recent decade, neural networks (NNs) have been used as effective tools to monitor and classify the source(s) of the process mean and/or variance shifts (Pugh 1989, Smith 1994, Cheng 1995, Chang and Aw 1996,

*Corresponding author. Email: lovbin@sjtu.edu.cn

ISSN 0020–7543 print/ISSN 1366–588X online

http://dx.doi.org/10.1080/00207543.2011.631596
http://www.tandfonline.com

Cook and Chiu 1998, Chang and Ho 1999, Ho and Chang 1999, Cook *et al*. 2001, Hwarng 2004, Pacella 2004, Niaki and Abbasi 2005, 2008, Guh 2007, Yu and Xi 2009, Du and Xi 2010).

Support vector machine (SVM) has recently become a new generation learning system based on recent advances on statistical learning theory for solving a variety of learning, classification and prediction problems (Cortes and Vapnik 1995, Gunn 1998, Vapnik 1998, Cristianini and Shawe-Taylor 2000, Gupta 2010). SVMs calculate a separating hyperplane that maximises the margin between data classes to produce good generalisation abilities. The main difference between NNs and SVMs is in their risk minimisation (Gunn 1998). In case of SVMs, structural risk minimisation principle is used to minimise an upper bound based on an expected risk, whereas in NNs, traditional empirical risk minimisation is used to minimise the error in the training of data. The difference in risk minimisation leads to a better generalisation performance for SVMs than NNs (Gunn 1998).

Some applications of SVM to monitor and classify the process variation have been reported. Chinnam (2002) demonstrated that support vector machines (SVMs) can be extremely effective in minimising both Type-I and Type-II errors for detecting shifts in the auto-correlated processes, and performed as well or better than traditional Shewhart control charts as well as other machine learning methods. Sun and Tsung (2003) developed a kernel-distance-based multivariate control chart using support vector, which can perform better than conventional charts when the underlying distribution of the quality characteristics is not multivariate normal. Cheng and Cheng (2008) developed one SVMs-based classifier to identify the source of variance changes in multivariate process (MP).

In recent years, combining machines instead of using a single one for increasing learning accuracy is an active research area. An ensemble of classifiers is a collection of several classifiers whose individual decisions are combined in some way to classify the testing examples. It is known that an ensemble often exhibits a better performance than the individual classifiers that compose it. Ensemble classification learning generates a set of base classifiers using different distributions of training data and then aggregates their outputs to classify new samples. These ensemble learning methods enable users to achieve more accurate predictions with higher generalisation abilities than the predictions generated by individual models or experts on average. Though different applications of SVM ensembles have been reported, e.g. land cover (Chan *et al*. 2001), bacterial transcription start sites prediction (Gordon *et al*. 2006), text-independent speaker recognition (Lei *et al*. 2006), and cardiovascular disease level prediction (Eom *et al*. 2008), few researches have been done to apply SVMs ensemble technique into classifying the source(s) of the process mean shifts in multivariate control charts.

The rest of this paper is organised as follows. The approach based on multivariate control charts and selective multiclass SVMs ensemble for monitoring and classifying the source(s) of mean shifts in MP is developed in Section 2. In Section 3, the details of SVMs ensemble classifier are explored. In Section 4, the simulation experiments and performance analysis are conducted based on the evaluation index. In Section 5, a case study is presented and the developed approach is validated through the real-world data. Finally, the conclusions are given in Section 6.

2. Overview of the approach

2.1 *The architecture of the approach*

This section provides an overview with respect to monitoring and classifying the source(s) of the mean shifts in MP. The developed approach consists of a multivariate control chart used as a detector of out-of-control signals, and selective SVMs ensemble as a classifier of the mean shifts, respectively.

Among various abnormal conditions, this study is concerned with process mean shifts. Assume there are q quality characteristic variables $X_1, X_2, \ldots, X_q$ need to be monitored and identified, which is expressed as one vector $\mathbf{X} = (X_1, X_2, \ldots, X_q)$. When increases δ occur at time t_p, the observations of quality characteristic vector $\mathbf{X}$ can be expressed as follows:

$$\mathbf{X}(t) = \boldsymbol{\mu}_0 + \mathbf{K}(t) + \boldsymbol{\delta}(t, t_p) \tag{1}$$

where

- $\mathbf{X}(t)$ quality characteristics vector observed at time t;
- $\boldsymbol{\mu}_0$ the process mean vector $\boldsymbol{\mu}_0 = (\mu_1, \mu_2, \ldots, \mu_q)$ when the process is under control;
- $\mathbf{K}(t)$ $N(0, \boldsymbol{\Sigma}_0)$, multivariate normal distribution;
- $\boldsymbol{\Sigma}_0$ the process covariance when the process is under control;
- $\boldsymbol{\delta}(t, t_p)$ $(\delta_1, \delta_2, \ldots, \delta_q)$, where δ_j is the magnitude of shifts.

For example, in a bivariate process, $\mathbf{X}(t)$ follows $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ when the process is under control. The covariance matrix $\boldsymbol{\Sigma}_0$ can be expressed:

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} \sigma_{X_1}^2 & \sigma_{X_1X_2} \\ \sigma_{X_1X_2} & \sigma_{X_2}^2 \end{bmatrix} \tag{2}$$

where σ_{X_1} and σ_{X_2} are the standard deviations of the first variable X_1 and second quality variable X_2, respectively.

There are three abnormal classes of patterns for a bivariate process: (1) the first variable being out of control (i.e., $\mu_1 + \delta,\ \mu_2$); (2) the second variable being out of control (i.e., $\mu_1,\ \mu_2 + \delta$); (3) both the first and second variable being out of control (i.e., $\mu_1 + \delta_1,\ \mu_2 + \delta_2$).

The architecture of the proposed approach is shown in Figure 1, in which three modules are in series: Module I, Module II and Module III utilised for effectively monitoring and classifying the source(s) of mean shifts in MPs. In Module I, T^2 control chart is used in this study for monitoring the process and judge whether the process is under control.

2.2 *Data pre-processing*

2.2.1 *Evaluating multivariate normality*

Module II provides various data pre-processing, such as normality evaluating, data cleaning, noise data deletion etc., which are very important to improve the performance of the process monitoring and classifying.

The assumption of normality for q quality characteristic variables usually needs to be checked. Plots are always useful tools in data processing. For univariate process, special plots called Q-Q plots can be used to assess the assumption of normality. However, for the process with multiple variables, it is more difficult to check.

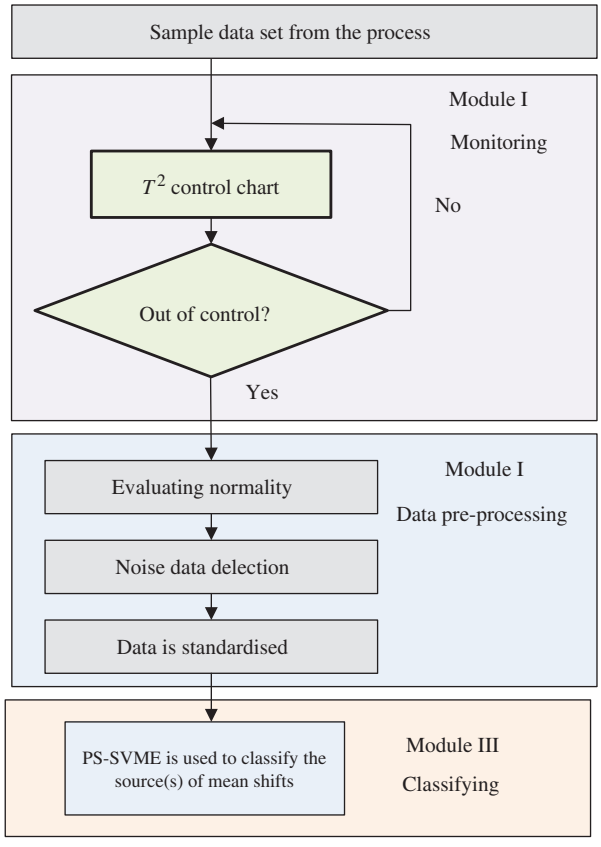

Figure 1. The architecture of the proposed monitoring and classifying approach.

A formal method for judging the joint normality for multiple variables based on the squared generalised distances is popular,

$$d_j^2 = (\mathbf{x}_j - \bar{\mathbf{x}})' \mathbf{S}^{-1} (\mathbf{x}_j - \bar{\mathbf{x}}), \quad j = 1, 2, \ldots, n \tag{3}$$

where $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are the sample observation, $\bar{\mathbf{x}}$ is the sample mean and $\mathbf{S}$ is the sample covariance, respectively.

When the parent population is multivariate normal and both n and n-q are greater than 25 or 30, each of the squared distances $d_1^2, d_2^2, \ldots, d_n^2$ should behave like a chi-square random variable (Anderson 2003). It is very helpful to plot these distances, and the resulting plot is called a chi-square plot.In this study the chi-square plot is used for evaluating normality of a data set, and the evaluating steps are presented as follows:

Step 1: Calculating the squared distances in Equation (3).

Step 2: Order the squared distances in Step 1 from smallest to largest as

$$d_{(1)}^2 \leq d_{(2)}^2 \leq \cdots \leq d_{(n)}^2.$$

Step 3: Graph the pairs $(Q_{c,q}((j-1/2)/n), d_{(j)}^2)$,

where $Q_{c,q}((j-1/2)/n)$ is the $100(j-1/2)/n$ quantile of the chi-square distribution with q degrees of freedom.

Step 4: Judge the normality of the data set. The multivariate normality is indicated if

(1) Rough half of the d_j^2 are less than or equal to $Q_{c,q}(0.5)$.
(2) A plot of the ordered squared distances $d_{(1)}^2 \leq d_{(2)}^2 \leq \cdots \leq d_{(n)}^2$ versus $Q_{c,q}(\frac{1-0.5}{n})$, $Q_{c,q}(\frac{2-0.5}{n}), \ldots, Q_{c,q}(\frac{n-0.5}{n})$, respectively, resembles nearly a straight line having slope 1 and that passed through the origin.

If the plot is a systematic curved pattern, it suggests lack of normality. Quantiles are specified in terms of proportions, whereas percentiles are specified in terms of percentages. The quantiles $Q_{c,q}((j-0.5)/n)$ are related to the upper percentiles of a chi-squared distribution. In particular, $Q_{c,q}((j-0.5)/n) = \chi_q^2((n-j+0.5)/n)$.

The chi-square plot for evaluating multivariate normality has been widely used. Johnson and Wichern (2007) presented an example of a chi-square plot for a four-variable data set. In the example, four different measures of stiffness, x_1, x_2, x_3, and x_4, of each of $n = 30$. The square distances $d_j^2 = (\mathbf{x}_j - \bar{\mathbf{x}})' \mathbf{S}^{-1} (\mathbf{x}_j - \bar{\mathbf{x}})$ are calculated and for more details, the readers are referred to Johnson and Wichern (2007). According to the 4th step, although the last two points with the largest squared distances are clearly removed from the straight line (the last two points make the plot appear curved at the upper end), the distribution of the whole data set is normal.

2.2.2 *Procedure for transforming data set to near normality*

In practice, the multivariate normality of data set may not be a viable assumption. One important way to handle this situation is to transform the non-normal data to near normality. Transformation is a re-expression of the data set in different units. For instance, when a histogram of positive observations exhibits a long right hand tail, transforming the observations by taking their logarithms or square roots will often markedly improve the symmetry about the mean and the approximation to a normal distribution.

Different methods for transformations have been developed, which can be divided into several classes according to transformed scale, such as 'square roots transformation', 'logit transformation', 'Fisher's z-transformation', and 'power transformation'. Among these transformations, the 'power transformation' is a very useful transformation method. Box and Cox (1964) modified the transformation method,

$$x^{(\lambda)} = \begin{cases} \frac{x^{\lambda}-1}{\lambda} & \lambda \neq 0 \\ \ln x & \lambda = 0 \end{cases} \tag{4}$$

which is continuous in λ for $x > 0$.

Given the observations $x_1, x_2, \ldots, x_n$, the solution for the choice of an appropriate power λ is the solution that maximised the expression

$$\ell(\lambda) = -\frac{n}{2} \ln \left[\frac{1}{n} \sum_{j=1}^{n} \left(x_j^{(\lambda)} - \bar{x}^{(\lambda)} \right)^2 \right] + (\lambda - 1) \sum_{j=1}^{n} \ln x_j \tag{5}$$

$x_j^{(\lambda)}$ is defined in Equation (4) and

$$\bar{x}^{(\lambda)} = \frac{1}{n}\sum_{j=1}^{n} x_j^{(\lambda)} = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{x_j^{\lambda}-1}{\lambda}\right) \tag{6}$$

is the arithmetic average of the transformed observations. The first term in Equation (5) is, apart from a constant, the logarithm of a normal likelihood function, after maximising it with respect to the population mean and variance parameters.

The calculation of $\ell(\lambda)$ for many values of λ is an easy task for computer. It is helpful to have a graph of $\ell(\lambda)$ versus λ, as well as a tabular display of the pairs $(\lambda, \ell(\lambda))$, in order to study the behaviour near the maximising value $\hat{\lambda}$. For instance, if either $\lambda = 0$ (logarithm) or $\lambda = 0.5$ (square root) is near $\hat{\lambda}$, one of these may be preferred because of its simplicity.

For multivariate data set, a power transformation should be selected for each of the variables. In this study, let $\lambda_1, \lambda_2, \ldots, \lambda_q$ be the power transformations for the q measured characteristics. Each λ_k can be selected by maximising

$$\ell_k(\lambda) = -\frac{n}{2}\ln\left[\frac{1}{n}\sum_{j=1}^{n}\left(x_{jk}^{(\lambda_k)} - \bar{x}_k^{(\lambda_k)}\right)^2\right] + (\lambda_k - 1)\sum_{j=1}^{n}\ln x_{jk} \tag{7}$$

where $x_{1k}, x_{2k}, \ldots, x_{nk}$ are the n observations on the kth variable, $k = 1, 2, \ldots, q$.

Here

$$\bar{x}_k^{(\lambda_k)} = \frac{1}{n}\sum_{j=1}^{n} x_{jk}^{(\lambda_k)} = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{x_{jk}^{\lambda_k}-1}{\lambda_k}\right) \tag{8}$$

is the arithmetic average of the transformed observations. The jth transformed multivariate observation is

$$\mathbf{x}_j^{(\hat{\boldsymbol{\lambda}})} = \begin{bmatrix} \frac{x_{j1}^{\hat{\lambda}_1}-1}{\hat{\boldsymbol{\lambda}}_1} \\ \frac{x_{j2}^{\hat{\lambda}_2}-1}{\hat{\boldsymbol{\lambda}}_2} \\ \vdots \\ \frac{x_{jq}^{\hat{\lambda}_q}-1}{\hat{\boldsymbol{\lambda}}_q} \end{bmatrix} \tag{9}$$

where $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_q$ are the values that individually maximise Equation (7).

Through using this procedure, the multivariate non-normal data set can be transformed to near normality.

2.2.3 *Standardisation of data*

Before the process data over time series stream are input to Module III, without loss of generality, all variables are scaled so that each process variable has mean zero and variance one. The original data is transformed into a form suitable for training and recalling in Module II. The data is standardised by using Equation (10) to result in each quality variable having a mean of zero and a variance of unity.

$$X'_{i,q} = \frac{X_{i,q} - \mu_q}{\sigma_q} \tag{10}$$

where $X'_{i,q}$ will follow $N(0, \mathbf{R})$ when $X_{i,q}$ follows $N(\mu_0, \Sigma_0)$.

The covariance matrix $\mathbf{R}$ is expressed in correlation form, namely, the main diagonal elements are all one and the off-diagonal elements are the pairwise correlation (ρ) between the process variables (Equation (11)). The proofs are

given in the Appendix.

$$\mathbf{R} = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1q} \\ \rho_{12} & 1 & \cdots & \rho_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1q} & \rho_{2q} & \cdots & 1 \end{bmatrix} \tag{11}$$

For example, in one process with three variables, the mean vector $\boldsymbol{\mu}_0$ and the variance-covariance matrix $\mathbf{R}$ in correlation form are expressed as follow, and if all variables are mutually independent, the matrix $\mathbf{R}$ becomes an identity matrix.

$$\boldsymbol{\mu}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad \mathbf{R} = \begin{bmatrix} 1 & 0.6 & 0.6 \\ 0.6 & 1 & 0.6 \\ 0.6 & 0.6 & 1 \end{bmatrix}$$

2.3 *SVMs*

2.3.1 *Brief review of SVMs*

In Module III, an improved particle swarm optimisation with simulated annealing-based selective multiclass support vector machines ensemble (PS-SVME) is explored for classifying the source(s) of mean shifts in MP due to high recognition power of SVMs ensembles. Utilisation of PS-SVME aims to improve the generalisation performance of identification system in comparison with using single multiclass SVMs classifiers, and to improve the ability to escape from a local optimum.

The basic idea of SVMs is to transform the data to a higher dimensional feature space and find the optimal hyperplane in the space that maximises the margin between the two classes. Consider a training data set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$, $i = 1, 2, \ldots, N$, where N is the total number of training vectors, $\mathbf{x}_i \in R^d \subset R$ is the ith d-dimensional input vector, and $y_i \in \{1, -1\}$ is known target. The training of SVM involves the solution of the following quadratic optimisation problem:

Minimise

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{12}$$

Subject to

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{13}$$

where ξ_i are slack variables, measuring the degree of misclassification of the sample $\mathbf{x}_i$, C is the error penalty factor, penalising the non-zero ξ_i, the bias b is a scalar, representing the bias of the hyperplane, $\mathbf{w}$ is the vector of hyperplane coefficients, defining a direction perpendicular to the hyperplane, the index i labels the N training cases, and the map function ϕ is a non-linear transformation to map the input vectors into a high-dimensional feature space (as shown in Figure 2). The optimisation problem becomes a trade-off between the margin maximisation and training errors minimisation.

In particular, if the data are perfect linearly separable, then $\xi_i = 0$, and the separating hyperplane that creates the maximum distance between the plane and the nearest data (i.e. the maximum-margin equals $||\mathbf{w}||^{-2}$) is the optimal separating hyperplane. To solve non-linear classification tasks, a mapping function is usually employed to map the training samples from the input space into a higher-dimensional feature space. This allows the SVM to fit the maximum-margin hyperplane in the transformed feature space. Any function that satisfies Mercer's theorem (Cristianini and Shawe-Taylor 2000) can be used as a kernel function. Some popular SVM kernel functions include:

Linear function:

$$K(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i \times \mathbf{y}_j$$

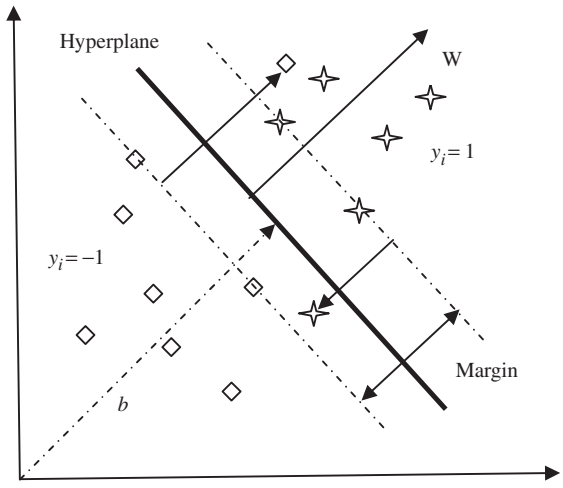

Figure 2. A geometric interpretation of the classification of SVM for a data set with two classes.

Gaussian radial basis function (RBF):

$$K(\mathbf{x}_i,\ \mathbf{y}_j) = \exp\left(-\gamma\left\|\mathbf{x}_i - \mathbf{y}_j\right\|^2\right) \quad \gamma \succ 0$$

Polynomial function with degree d:

$$K(\mathbf{x}_i, \mathbf{y}_j) = ((\mathbf{x}_i \times \mathbf{y}_j) + \gamma)^d \quad \gamma \succ 0$$

Sigmoid:

$$K(\mathbf{x}_i, \mathbf{y}_j) = \tan h(\gamma \mathbf{x}_i \times \mathbf{y}_j) + r) \quad \gamma \succ 0$$

2.3.2 *Multiclass SVMs*

SVMs are originally designed for binary classification (Cortes and Vapnik 1995, Vapnik 1995, 1998), which are not suitable for classifying the source(s) of the mean shifts because there are usually several quality variables needed to be monitored and identified in a MP. Thus, it is necessary to develop an approach to deal with a multi-classification problem. Multi-classification can be obtained by the combination of binary classification (Crammer and Singer 2000). In order to extend a basic SVM to solve multi-classification problem, 'one-against-one', 'one-against-all' and direct acyclic graph (DAG) are three popular approaches. Hsu and Lin (2002) conducted a comprehensive comparison of these three multiclass SVM classification approaches, and they suggested that the one-against-one method is most suited for practical use than other methods. Therefore, one-against-one method is used to classify the source(s) of the mean shifts in this study.

3. PS-SVME

3.1 *PS-SVME framework*

The main steps of PS-SVME are given as follows.

Step 1: Creation of single SVM for binary classification. Several accurate and diverse binary classification SVMs (i.e., $SVM_1, SVM_2, \ldots, SVM_{q(q-1)/2}$) are initially created.

Step 2: Creation of one component multiclass SVM classifier through combining single SVM (SVM_1, $SVM_2, \ldots, SVM_{q(q-1)/2}$). SVM is originally designed for binary classification, which is not suitable for classifying

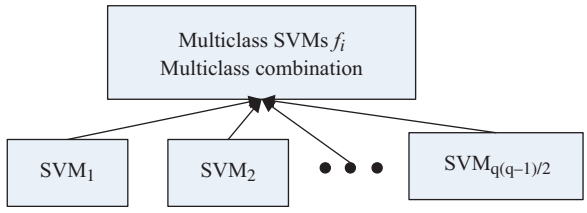

Figure 3. Creation of one component multiclass SVM classifier through combining single SVMs.

multiple variables with means shift simultaneously. One component multiclass SVM classifier is obtained by combining the single binary SVM classifiers ($SVM_1, SVM_2, \ldots, SVM_{q(q-1)/2}$) (Figure 3) created in Step 1 using 'one-against-one' method (discussed in Section 3.2).

Step 3: Creation of several component multiclass SVM classifiers. By repeating Step 2, several multiclass SVM classifiers are created and used as component multiclass SVMs classifiers for optimal selection in Step 4, and they are denoted by $f_1, f_2, \ldots, f_T$.

Step 4: Selection of optimal subset. Instead of attempting to ensemble all independent component multiclass SVM classifiers created in Step 3, the optimal subset is selected from the component multiclass SVM classifiers through an improved particle swarm optimisation with simulated annealing (PSA) algorithm explored in this study (discussed in Section 3.3).

Step 5: Combining 'good' outputs. Finally, the classified outputs of the optimal subset are combined to produce the output of the ensemble using majority voting method (Zhou *et al*. 2002). Step 4 and Step 5 are shown in Figure 4.

3.2 *Creation of one component multiclass SVM classifier*

This section discusses how to create one component multiclass SVM classifier through combining single SVM (Step 2 in Section 3.1). For q-class event, the 'one-against-one' method constructs $M = C_q^2 = q(q-1)/2$ binary classifiers, and each of them is trained by binary-class data. For training data from the ith and the jth class, the following binary classification problem needs to be solved:

$$\begin{aligned} &\min \frac{1}{2}(\mathbf{w}^{ij})^T\mathbf{w}^{ij} + C\sum_t \xi_n^{ij}(\mathbf{w}^{ij})^T \\ &\text{s.t.} \begin{cases} (\mathbf{w}^{ij})^T K(x_n) + b^{ij} \geq 1 - \xi_n^{ij} & \text{if} \quad y_n = i \\ (\mathbf{w}^{ij})^T K(x_n) + b^{ij} \leq \xi_n^{ij} - 1 & \text{if} \quad y_n = j \\ \xi_n^{ij} \geq 0 \end{cases} \end{aligned} \tag{14}$$

where, similarly to binary classification SVMs, $K(x_n)$ is kernel function, (x_n, y_n) is the ith or jth training sample, $\mathbf{w} \in R^N$ and $b \in R$ are the weighting factors, ξ_n^{ij} is the slack variable and C is the penalty parameter.

3.3 *Selective SVMs ensemble*

3.3.1 *Ensemble algorithm*

The most prevailing ensemble constructing techniques are 'bagging' and 'boosting'. Bagging is proposed based on bootstrap sampling of the training set (Breiman 1996). The general idea of boosting is to develop the classifier ensemble incrementally, adding one classifier at a time (Schapire 1990). There are many boosting algorithms. In this study, the AdaBoost algorithm, short for adaptive boosting and proposed by Freund and Schapire (1997), is used to ensemble learning. The proposed algorithm of SVMs ensemble is shown in Figure 5. The first ensemble member is formed by applying the SVMs to the entire training set. Subsequent ensemble members are formed by applying the SVMs to the training set but with cases re-weighted to place higher weight on cases that are misclassified by existing ensemble members. The optimal subset is selected from component multiclass SVMs using PSA algorithm, which is

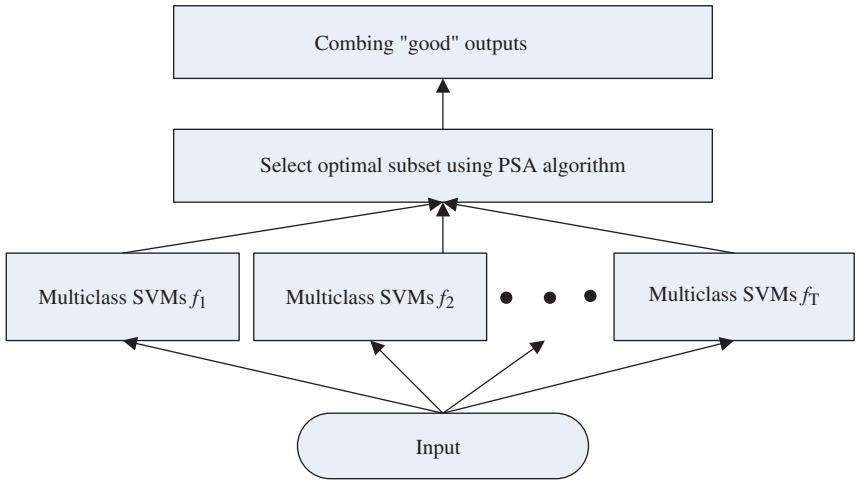

Figure 4. Selection of optimal subset and combining 'good' outputs.

Input: Training data set $TR=\{x_i, y_i\}_{i=1}^{N}$, where $\mathbf{x}_i \in R^d \subset R$, $y_i \in Y \subset \{l_1, l_2, \cdots l_k\}$ represents class label, validation data set V; one-against-one SVM; integer T specifying the number of SVMs ensemble.

Initialise: initial weights $\mathbf{w}=\{w_i = 1/N, i=1,2,\cdots,N\}$. $t=0, \varepsilon_0 = 0$

While ($t \le T$ and $\varepsilon_t < 0.5$)

Step 1: Normalise $\mathbf{w}^t$, so that $\sum_{i=1}^{N} w_i^t = 1$.

Step 2: Train the tth SVMs f_t, providing that the weight $\mathbf{w}^t$.

Step 3: Compute the prediction error of f_t on the original training set TR as

$$\varepsilon_t = \sum_{i=1}^{N} w_i^t \left[f_t(x_i) \neq y_i \right].$$

Step 4: Set $\beta_t = 0.5 \ln(\frac{1-\varepsilon_t}{\varepsilon_t})$.

Step 5: Update weight vector: $w_i^{t+1} = w_i^t \times \begin{cases} \exp(-\beta_t) & if\ f_t(x_i) = y_i \\ \exp(\beta_t) & if\ f_t(x_i) \neq y_i \end{cases}$

Step 6: $t = t+1$.

Step 7: Implement *PSA* algorithm to obtain the optimal particle vector on the validation data set V.

Output: weighted majority voting, for a testing set $\mathbf{z}$ with class label $y_i \subseteq Y = \{l_1, l_2, \cdots l_k\}$,

ensemble $f^{opt}(x)$, $f^{opt}(x) = \arg\max \sum_{j=1}^{t} \beta_j \left| f_j(\mathbf{z}) = y_i \right|$

Figure 5. The proposed selective multiclass SVMs ensemble algorithm.

illustrated in Section 3.3.2. The votes of ensemble members are weighted by a function that lowers the vote of classifier that has lower accuracy on the weighted training set from which it is learned.

Once a number of component multiclass SVM classifiers (i.e., $f_1, f_2, \ldots, f_T$) are obtained, PS-SVME will select several SVMs from the component multiclass SVM classifiers to construct the SVMs ensemble. Both theoretical and empirical studies demonstrate that ensemble several 'good' SVMs with accuracy and diversity could be better than ensemble all component multiclass SVM classifiers for both regression and classification problems (Zhou *et al.* 2002). Here, 'ensemble all' means ensemble the output results of all component multiclass SVM classifiers. Furthermore, the size of the ensemble is reduced without worsening the performance. However, excluding those 'bad' component multiclass SVM classifiers and selecting the optimal subset is not an easy task as one may have imagined since the space of possible subsets is very large ($2^T - 1$) for a population of size T (T is the number of all component multiclass SVM classifiers). Thus, it is impractical to use exhaustive search to find an optimal subset. There is still no agreement about which method is the most appropriate for selection of SVMs. Some intelligent evolution algorithms show distinguished performance for solving this problem.

Particle swarm optimisation (PSO) is a population based search algorithm, which was inspired by the social behaviour of bird flocks as originally developed by Eberhart and Kennedy (1995). It is widely reported that the PSO algorithm is very easy to implement and has fewer parameters to adjust when compared with other evolutionary algorithms. In this study, for PS-SVME, PSA algorithm is explored to obtain an optimal subset from the component multiclass SVM classifiers. In PSA algorithm, each particle is 'flown' through the multidimensional search space, adjusting its position in search space according to its own experience and that of neighbouring particles. The particle therefore makes use of the best position encountered by itself and that of its neighbours to position itself toward an optimal solution. The performance of each particle is evaluated using a predefined fitness function, which encapsulates the characteristics of the optimisation problem.

It is an iterative process in which the change in weights of a particle at beginning of an iteration are calculated using Equation (15) and new position of every particle is found using Equation (16).

$$v_i(t+1) = kv_i(t) + c_1 r_1(p_{id} - x_i(t)) + c_2 r_2(p_{gd} - x_i(t)) \tag{15}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{16}$$

where t is the current step number, k is the inertia weight, c_1 and c_2 are the acceleration constants, r_1 and r_2 are two random numbers in the range [0, 1], $x_i(t)$ is the current position of the particle, p_{id} is the best one of the solutions this particle has reached, p_{gd} is the best one of the solutions all the particles have reached.

For PSA algorithm, simulated annealing (SA) is used to improve the ability to escape from a local optimum. SA is normally introduced as a heuristic approach to solve numerous combinatorial optimisation problems to replace those schemes where the solution could get stuck on local optimum. In the search process, the SA accepts not only better but also worse neighbouring solutions with a certain probability. Such mechanism can be regarded as a trial to explore new space for new solutions, either better or worse. The probability of accepting a worse solution is larger at higher initial temperature. As the temperature decreases, the probability of accepting worse solutions gradually approaches zero. This feature means that the SA technique makes it possible to jump out of a local optimum to search for the global optimum. The relative success of SA algorithm has been tapped by many researchers, who discussed the useful, theoretical and practical details of SA (Lv *et al.* 2009, Pandey *et al.* 2006). In this study, the SA technique is used to deal with every particle according to the following rules:

(1) If $p_{id} > p_{gd}$, accept $p_{gd} = p_{id}$ with the probability 1.
(2) If $p_{id} < p_{gd}$, accept $p_{gd} = p_{id}$ with the probability *prob* defined as:

$$prob = 1 - \exp\left(-\frac{p_{gd} - p_{id}}{temp}\right) \tag{17}$$

$$temp = initemp \times h \tag{18}$$

where *prob* is the probability, temp is the current temperature, *initemp* is a constant selected as initial temperature, p_{id} is the best one of the solutions this particle has reached, p_{gd} is the best one of the solutions all the particles have reached, and h is the current step number.

The above treatment can increase the diversity in the particles and enable PSO to accept a good solution with high probability. Through continuous evolution of these particles, the most optimal global solution (i.e. the most optimal particle vector) can be obtained.

3.3.2 *Combination of outputs of component multiclass SVMs*

As for combining the predictions of component multiclass SVMs, the most prevailing approaches are plurality voting or majority voting for classification tasks, and simple averaging or weighted averaging for classification tasks (Zhou *et al*. 2002). In this study, the majority voting method is used to classify the source(s) of mean shifts.

4. Simulations and results

4.1 *Training data*

In order to evaluate the proposed approach, some simulation experiments have been conducted, in which the multivariate normal training and testing data set is generated by Monte Carlo simulation. Mean shifts are investigated for their shift magnitudes δ from 1.0 to 3.0 in steps of 0.5 (total five δ). The training dataset is summarised in Table 1. For each δ, 800 testing examples are generated in the same way as the generation of training examples. Correct classification percentage (CCP), representing the ability of one classification system to correctly classify the status of one process, is used as evaluation index to evaluate the performance of the developed approach.

4.2 *Parameters of PS-SVME*

The performance of learning in PS-SVME is influenced by its relative parameters. There are no absolute rules for tuning these key factors, which are dependent on the characteristics of the real-world problems. A number of factors discussed possibly affect the performance of PS-SVME. These factors include different training data set, sample size, number of component SVMs T, and number of particles etc. The relative parameters of PS-SVME, which achieved a relatively better performance on the training dataset in this study, are summarised as follows.

(1) The number of all component multiclass SVM classifiers T: in the step of constructing PS-SVME, the number of component SVMs, 2, 4, 8, 10, 16, 32 are considered.
(2) Kernel function: Gaussian radial basis function (RBF) $K(\mathbf{x}_i, \mathbf{y}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{y}_j||^2)$ is used in this study, and $\gamma = 2$.
(3) Penalty parameter: $C = 10$.
(4) The number of particles in PSA: when using PSA algorithm to select the subset from component SVMs, the number of particles 5, 10, 15, 20, 25, 30, 35 and 40 are considered.
(5) The inertia weights: $w_{\min}$ equals 0.2 and $w_{\max}$ equals 0.8.
(6) The speed of particles: $v_{\max}$ equals 4 and $v_{\min}$ equals -4.

Table 1. Types of data set.

Mean shift type		Number of training examples	Comments
$q=2$	$(\delta, 0)$	4000	800 for each δ
	$(0, \delta)$	4000	800 for each δ
	(δ, δ)	4000	800 for each δ
$q=3$	$(\delta, 0, 0)$	4000	800 for each δ
	$(0, \delta, 0)$	4000	800 for each δ
	$(0, 0, \delta)$	4000	800 for each δ
	$(\delta, \delta, 0)$	4000	800 for each δ
	$(\delta, 0, \delta)$	4000	800 for each δ
	$(0, \delta, \delta)$	4000	800 for each δ
	(δ, δ, δ)	4000	800 for each δ

(7) The initial velocities: the initial velocities of the initial particles are generated at random in the range [−4, 4].
(8) The study factors: c_1 equals 2 and c_2 equals 2.
(9) The initial temperature *initemp* is 0.55.

4.3 *Performance analysis*

4.3.1 *The overall percentage of correct classification of PS-SVME*

Table 2 presents classification results of PS-SVME trained and tested for different shift magnitudes in one bi-variate process. The possible mean shifts are summarised for the following cases: shift in first variable, shift in second variable, and shift in both variables. Columns 3 to 5 in Table 2 show the number of times PS-SVME

Table 2. Correct classification percentage (CCP) (%) for PS-SVME.

		Actual classification results			
Abnormal pattern	Shift magnitude	$(\delta, 0)$	$(0, \delta)$	(δ, δ)	CCP (%)
$(\delta, 0)$	(1.0, 0)	703	5	92	87.9
	(1.5, 0)	727	0	73	90.9
	(2.0, 0)	712	0	88	89.0
	(2.5, 0)	698	3	99	87.3
	(3, 0)	695	0	105	86.9
	Average	707	1.6	91.4	**88.4**
$(0, \delta)$	(0, 1.0)	0	712	88	89.0
	(0, 1.5)	4	687	109	85.9
	(0, 2.0)	1	714	85	89.3
	(0, 2.5)	0	703	97	87.9
	(0, 3.0)	0	711	89	88.9
	Average	1	705.4	93.6	**88.2**
(δ, δ)	(1.0, 1.0)	0	52	748	93.5
	(1.0, 1.5)	19	22	759	94.9
	(1.0, 2.0)	2	33	765	95.6
	(1.0, 2.5)	0	47	753	94.1
	(1.0, 3.0)	0	29	771	96.4
	(1.5, 1.0)	5	26	769	96.1
	(1.5, 1.5)	89	5	706	88.3
	(1.5, 2.0)	1	45	754	94.3
	(1.5, 2.5)	2	51	747	93.4
	(1.5, 3.0)	1	47	752	94.0
	(2.0, 1.0)	1	62	737	92.1
	(2.0, 1.5)	62	0	738	92.3
	(2.0, 2.0)	49	2	749	93.6
	(2.0, 2.5)	33	9	758	94.8
	(2.0, 3.0)	1	51	748	93.5
	(2.5, 1.0)	44	0	756	94.5
	(2.5, 1.5)	47	0	753	94.1
	(2.5, 2.0)	51	5	744	93.0
	(2.5, 2.5)	56	2	742	92.8
	(2.5, 3.0)	45	0	755	94.4
	(3.0, 1.0)	1	27	772	96.5
	(3.0, 1.5)	0	49	751	93.9
	(3.0, 2.0)	5	41	754	94.3
	(3.0, 2.5)	2	35	763	95.4
	(3.0, 3.0)	3	36	761	95.1
	Average	20.76	27.04	752.20	**94.04**
	Total	242.92	244.68	312.40	**90.21**

approach reaches the corresponding classification patterns, and column 6 shows the correct classification percentage (CCP).

The number of component SVMs T and particles in PSA is eight and 25 and other parameters are referred to Section 4.2. The overall percentage of correct recognition of PS-SVME for first, second and third class of abnormal pattern is 88.4, 88.2 and 94.04, respectively. The overall average CCP of PS-SVME is 90.21, which indicates that the developed approach exhibits a strong ability to online identify which quality characteristic variable's mean shifts.

4.3.2 *Sensitivity analysis to the number of the examples*

In Sections 4.3.2 to 4.3.4, the influences of the key factors of PS-SVME on their generalisation performance are analysed by implementing the following experiments. The analysis can help us to obtain the suitable parameter setting to improve the generalisation performance of PS-SVME. In each of the following tests, only one factor is varied while the remaining ones are kept constant.

Training sets of different sizes are generated to train PS-SVME. Table 3 and Figure 6 present the test results of PS-SVME trained by different examples. From these results, one can find that increasing the training examples can improve the performance of PS-SVME up to a certain level of accuracy. This could be explained by the fact that with enough large training sets there is a better chance of true representation of a problem space. However, any further increase of the training size after reaching such limits will not greatly improve the performance of PS-SVME. Moreover, the larger training set results in higher time cost of training.

4.3.3 *Sensitivity analysis to the number of multiclass SVM classifier T*

The number of component multiclass SVM classifier T affects the performance of SVMs ensemble. Table 4 and Figure 7 present CCPs for a different number of component multiclass SVM classifier T. The results show that increasing T results in improved performance of PS-SVME. This may be explained by the fact that with more component multiclass SVM classifiers there is a better chance of selection subsets for SVM ensemble. In this study, the overall average CCP of PS-SVME is 90.21 when the number T is eight. Though PS-SVME has better performance when T is bigger than eight (the overall average CCPs are 91.19, 92.08, 92.12 for the number T 10, 16, 32, respectively), a large T results in high time cost of training and testing, and it also decreases the effectiveness of detecting out-of-control signals as quickly as possible. Therefore, the number of component multiclass SVM classifier T for training PS-SVMN must be varied based on the balance between time cost and CCPs.

4.3.4 *Sensitivity analysis to the number of particles in PS-SVME*

One of the key factors affecting the performance of PS-SVME is the number of particles. Thus, it might affect the generalisation performance of PS-SVME. As shown in Table 5 and Figure 8, increasing the number of particles can improve the performance of PS-SVME. However, 25 particles are enough to find the optimal subset for PS-SVME.

Table 3. CCPs of PS-SVME for different number of training examples ($q = 2$, $\rho = 0.6$).

	CCP			
Training examples	$(\delta, 0)$	$(0, \delta)$	(δ, δ)	Average
500	79.41	79.95	83.23	**80.86**
1000	80.29	80.81	84.26	**81.79**
2000	83.32	83.95	87.49	**84.92**
3000	85.08	85.81	90.01	**86.97**
4000	88.40	88.20	94.04	**90.21**
5000	89.45	90.11	94.78	**91.45**
5500	89.65	89.99	94.88	**91.51**

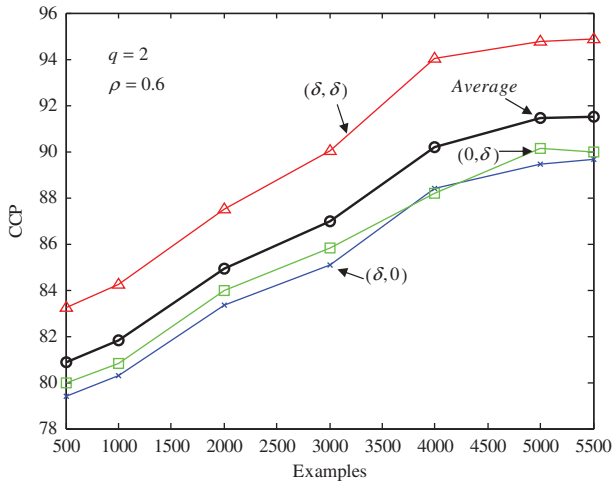

Figure 6. The average CCPs of PS-SVME for different number of training examples.

Table 4. CCPs of PS-SVME for the number of multiclass SVM classifier T ($q = 2$, $\rho = 0.6$).

	CCP			
T	$(\delta, 0)$	$(0, \delta)$	(δ, δ)	Average
2	80.91	81.35	86.32	**82.86**
4	83.29	83.27	85.71	**84.09**
8	88.40	88.20	94.04	**90.21**
10	88.32	89.45	95.80	**91.19**
16	90.21	89.56	96.47	**92.08**
32	89.45	90.11	96.80	**92.12**

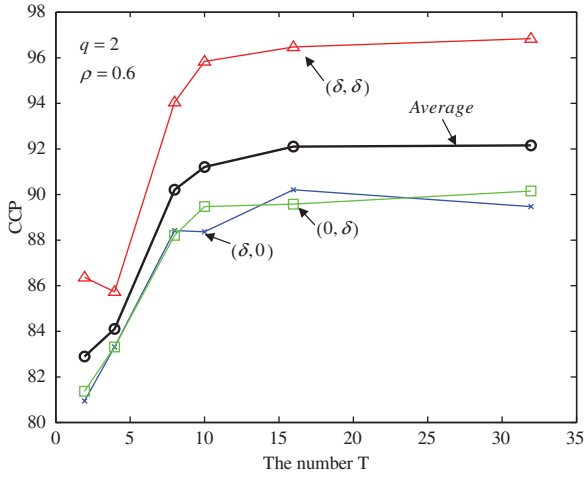

Figure 7. The average CCPs of PS-SVME for different multiclass SVM classifier T.

Table 5. CCPs of PS-SVME for the number of particles ($q = 2$, $\rho = 0.6$).

	CCP			
T	$(\delta, 0)$	$(0, \delta)$	(δ, δ)	Average
5	84.27	84.65	90.34	**86.42**
10	86.01	86.17	90.65	**87.61**
15	87.28	87.41	92.34	**89.01**
20	88.31	88.07	92.96	**89.78**
25	88.40	88.20	94.04	**90.21**
30	88.95	88.52	94.12	**90.53**
35	88.98	89.63	94.00	**90.87**
40	89.05	90.02	93.69	**90.92**

4.3.5 *Comparison analysis for PS-SVME, single SVMs and Ensemble ALL*

In order to further evaluate the performance of the PS-SVME, it is compared with other SVMs approach including single SVMs and Ensemble ALL in the bi-variate process with $\rho = 0.6$. Table 6 presents the CCPs for PS-SVME, the best single multiclass SVM classifier (i.e. the component multiclass SVM classifier f_i showing the best performance) and Ensemble ALL (i.e. ensemble all component multiclass SVM classifier $f_1, f_2, \ldots, f_T$).

It should be noted that one component multiclass SVM classifier is actually one SVM classifier for the bi-variate process ($q = 2$) since it is one binary classification process, and also be noted that the results are the average of three abnormal classes of patterns for each shift magnitude. The overall average percentage of correct recognition of PS-SVME, the best single multiclass SVM classifier, and Ensemble ALL is 90.21, 86.57, and 88.81 in bi-variate process ($q = 2$, $\rho = 0.6$), respectively. These results show that PS-SVME outperform the best single multiclass SVM classifier by 4.04% and Ensemble ALL by 1.55%. This proves that PS-SVME has better performance compared with those of the commonly used single multiclass SVM classifier approach and Ensemble ALL approach.

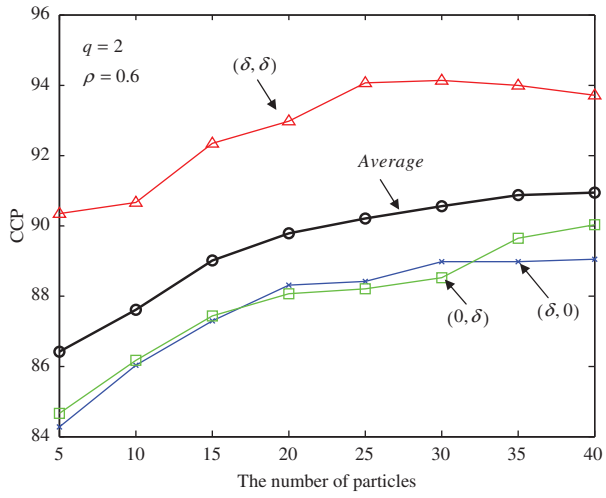

Figure 8. The CCPs of PS-SVME for the different number of particles.

Table 6. Comparison of CCPs for different approaches ($q = 2$, $\rho = 0.6$).

	CCP			
Approaches	$(\delta, 0)$	$(0, \delta)$	(δ, δ)	Average
PS-SVME	88.40	88.20	94.04	**90.21**
Best single multiclass SVM classifier	85.28	85.30	89.14	**86.57**
Ensemble ALL	87.41	87.67	91.35	**88.81**

Table 7. Comparison of CCPs of PS-SVME for bi-variate process and three-variables process.

	CCP		
Approaches	Type A	Type B	Type C
The bi-variate process	88.30	94.04	N/A
The process with three variables	85.79	86.21	89.07

4.3.6 *Performance analysis for processes with three variables*

In this section, the proposed approach is used to analyse the correct classification percentage in a process with three quality variables. For this simulated process, the reference mean vector and variance-covariance matrix are

$$\boldsymbol{\mu}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 1 & 0.6 & 0.6 \\ 0.6 & 1 & 0.6 \\ 0.6 & 0.6 & 1 \end{bmatrix}$$

In order to study the influences of the number of quality variables upon the performance of the proposed approach, a simulated bi-variate process is also considered in this simulation experiment. For this simulated bi-variate process, the reference mean vector and variance-covariance matrix are

$$\boldsymbol{\mu}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

To monitor the out-of-control signals of mean shifts, the T^2 control chart with the type I error $\alpha = 0.5\%$ is used. The training and testing dataset are generated and shown in Section 4.1. The possible mean shifts are summarised for the following cases: type A, one variable mean shifts; type B, two variables means shift; type C, three variables means shift.

Table 7 presents the results for the two-variable case and the three-variable case. It can be observed that the percentage of correct classification for the proposed approach is decreased when it is used for the three-variable case, which proves that the number of quality variables in a process affects the performance of the proposed hybrid learning-based approach. However, the proposed approach also does well in the three-variable case. Therefore, the proposed approach can be generalised to multivariate processes.

4.4 *Performance comparison with other work*

As stated in the introduction, the decomposition approach based mostly on T^2 statistic (Mason *et al.* 1995, 1997, Li *et al.* 2008, Verron *et al.* 2010) and neural networks approach (Pugh 1989, Smith 1994, Cheng 1995, Chang and Aw 1996, Cook and Chiu 1998, Chang and Ho 1999, Ho and Chang 1999, Cook *et al.* 2001, Hwarng 2004, Pacella 2004, Niaki and Abbasi 2005, 2008, Guh 2007, Yu and Xi 2009, Du and Xi 2010) are two of most classical and

popular approaches in classifying the source(s) of the mean shifts in the published works. Therefore, in this section, the proposed approach is compared with these two kinds of approaches.

4.4.1 *Comparison with MTY decomposition*

Classifying the source(s) of mean shifts can also be done using decomposition approaches. One of most important decomposition approaches is the Mason-Tracy-Yong (MTY) decomposition approach (Mason *et al.* 1995, 1997). The main idea behind this approach is to decompose the T^2 statistic into independent and orthogonal parts, each of which reflects the contribution of an individual variable. The main drawback of the MTY approach is that the decomposition of the T^2 statistic into p independent T^2 components is not unique as $p!$ different non-independent partitions are possible (Bersimis *et al.* 2007), which results in extensive computation.

The performance of the MTY approach was tested by Aparisi *et al.* (2006) using different shift type, magnitude of the shift, the sample sizes, etc. Two phases are used to implement the simulations. A bi-variate process with standard normal distribution is simulated in phase I by generating $N = k \times n = 100$ (i.e. using k samples with size n) values for the in-control condition, and different correlation coefficients ρ are considered (i.e. $\rho = 0.2,\ 0.4,\ 0.6,\ 0.8$) in the simulation experiments. The in-control mean vector is shifted to simulate an out-of-control signal. The components of the out-of-control mean vector μ_1 are computed to obtain the required Mahalanobis distance d between this vector and the in-control mean vector, where

$$d = \sqrt{(\mu_1 - \mu_0)' \sum_{0}^{-1} (\mu_1 - \mu_0)}.$$

Aparisi *et al.* (2006) simulated the shifts given the value of the Mahalanobis distance d, as shown in Figure 9. Five cases (i.e. five out-of-control T^2 statistics) for each combination of the ρ and d are generated for testing the performance of the MTY approach. For more details, ones are referred to the work (Aparisi *et al.* 2006).

In order to further evaluate the performance of the PS-SVME approach, it is compared with the MTY approach using the type I error $\alpha = 3$ % and the sample size is 3. Once T^2 control chart gives an abnormal alarm for a signal, it is fed to PS-SVME immediately to identify the variables that have shifted. The training dataset is generated in the same way as the generation of training dataset in Section 4.1. The parameters of PS-SVME are same as those in Section 4.2.

Table 8 presents the comparison results of the PS-SVME approach with those of the MTY approach provided by Apasisi *et al.* (2006) when all the various Mahalanobis distances (shown in Figure 9) and correlation coefficient values (i.e. $\rho = 0.2,\ 0.4,\ 0.6,\ 0.8$) are considered in the simulation experiments. The overall percentages of correct classification of the PS-SVME approach and the MTY approach are shown in the last row of the table, respectively. The MTY approach outperforms the PS-SVME approach in cases of the abnormal model ($\mu_1 \pm \delta, \mu_2 \pm \delta$), i.e. shift type S_1, S_2, S_5, and S_7. These points correspond to the cases where both variables shift. The PS-SVME approach outperforms the MTY approach in cases of abnormal model ($\mu_1 \pm \delta, \mu_2$) and ($\mu_1, \mu_2 \pm \delta$), i.e. shift type S_3, S_4, S_6 and S_8. These points correspond to the cases where only one variable shifts. These results demonstrate that the PS-SVME approach has the capability to identify source(s) of process mean vector in the bi-variate process.

4.4.2 *Comparison with neural networks*

Another one competing approach is neural network approach. In order to further evaluate the performance of the proposed approach, it is compared with some classical NNs classifiers including probabilistic neural networks (PNN) (Specht 1990), radial basis function neural networks (RBFNNs) (Chen 1991) and multilayered perceptron (MLP) neural networks (Pal and Mitra 1992) with different training algorithm such as back propagation (BP) learning algorithm and with the resilient propagation (RP) learning algorithm. They comprise parameters that should be readjusted in any new classification. Furthermore, those parameters regulate the classifiers to be best fitted in the classification task. In most cases, there is no classical method for obtaining their values, and therefore they are experimentally specified through try and error. The comparison result for one bi-variate process is indicated in Table 9. It can be seen from Table 9 that the proposed approach in this study outperforms the PNN recogniser by 4.17%, RBFNN by 4.42%, MLP (BP) by 5.35%, and MLP (RP) by 5.21%. These results prove that the proposed approach has better performance compared with those of the commonly used NNs approach.

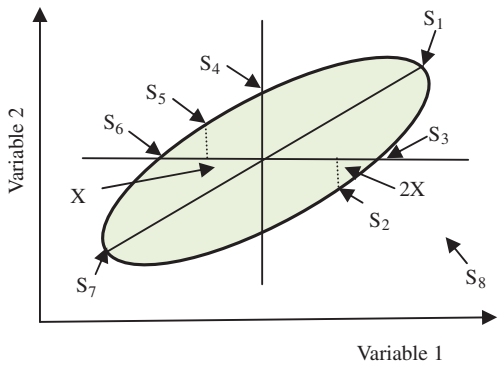

Figure 9. The simulated shift types in the bi-variate process.

Table 8. Comparison of CCPs between PS-SVME and MTY approach for bivariate process.

	Correct classification percentage (CCP, %)	
Shift type	PS-SVME approach	MTY approach*
S_1	84.54	86.02
S_2	67.71	71.35
S_3	78.20	39.06
S_4	88.07	31.25
S_5	71.19	74.02
S_6	64.38	42.79
S_7	78.42	82.29
S_8	73.25	32.08
Average	75.72	**57.36**

*Results are from Aparisi *et al*. (2006).

Table 9. Recognition performance comparison of PS-SVME with other approaches.

Approach	Parameters	CCR (%)
PNN	Spread = 50	86.45
RBFNN	Spread = 64, goal = 0.01	86.22
MLP(BP)	Hidden neurons = 32	85.38
MLP(RP)	Hidden neurons = 24	85.51
PS-SVME	The number of particles is 20; $T = 8$; other parameters are referred to Section 4.2	90.21

5. A case study

To further validate the effectiveness of the developed approach, its real application in an aircraft horizontal stabiliser assembly process for monitoring and identification of process mean shifts is illustrated. Most components of an aircraft have a very high demand of high quality, reliability, and low variation. Aircraft horizontal stabiliser is one of these most important components and has very strict dimensional specification. The assembly processes can directly affect the reliability and performance of the final aircraft. However, the complexity of aircraft horizontal stabiliser assembly processes make it very difficult to efficiently identify the mean shifts depend solely on control charts and the operator' experience. The immediate location of abnormal variation sources in aircraft horizontal stabiliser assembly processes can greatly narrow down the set of possible root causes, facilitating rapid analysis and corrective action by quality engineers.

5.1 *Product and assembly process*

Figure 10 shows a computer-aided design (CAD) model of an aircraft horizontal stabiliser, which is mainly composed of two beams (front beam and back beam), two rear spars (front rear spar and back rear spar) and several webs. There are a total of 16 measurement points (eight measurement points on front beam and back beam, respectively) numbered in the CAD model. Among these 16 measurement points, six measurement points are most important (③, ④, and ⑤ on front beam and back beam, respectively) called key product characteristics (KPCs), which directly affect the final dimensional quality and product reliability of aircraft horizontal stabiliser. These six measurement points are observed and monitored based on the developed online monitoring and classification approach. If the system gives the alarm of mean shifts, quality variables causing the alarm can be identified online.

Figure 11 is the fixture layout in horizontal stabiliser assembly process. Firstly, the rear spars are loaded and located into fixture system, and is completed by assembling two rear spars into one rear spar component. Then the

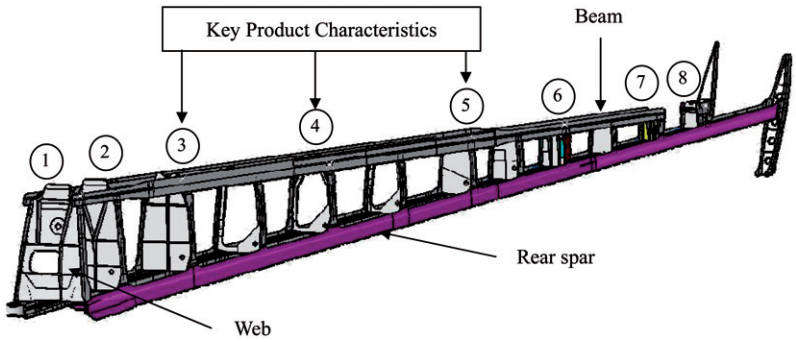

Figure 10. CAD model of aircraft horizontal stabiliser and key product characteristics.

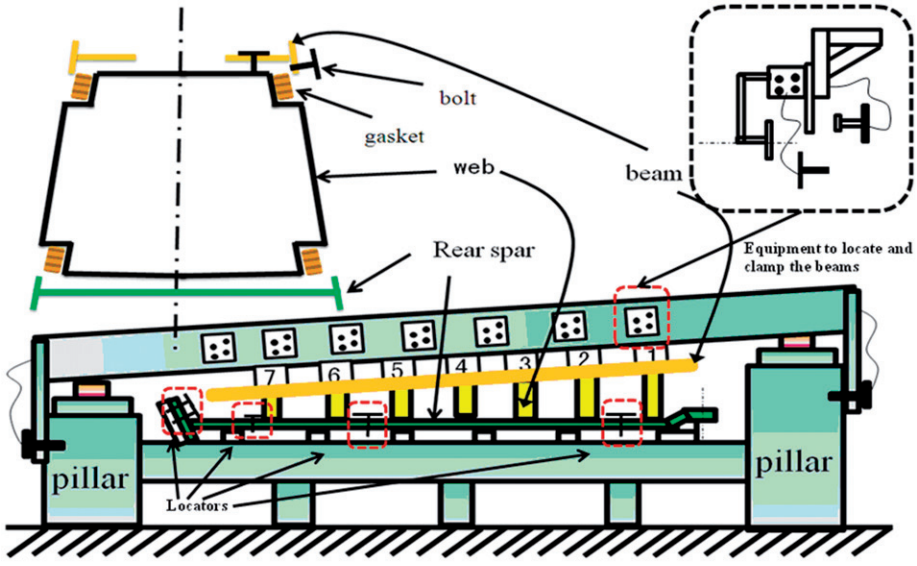

Figure 11. Fixture scheme in aircraft horizontal stabiliser assembly process.

webs are loaded, located, clamped and assembled into the rear spar component. Finally, two beams are located and assembled. The six KPCs are formed on the aircraft horizontal stabiliser.

5.2 *Normality test of data set*

The 1500 samples for front beam and back beam from real-world assembly processes are collected respectively. Since the front beam and back beam are assembled independently, the three measurement points (three variables, $q = 3$) of KPCs on front beam and back beam are monitored and identified independently. The generated data set in simulation experiments of Section 4 follows the multivariate normal distribution through the computer programs; however, the normality of the collected data set from real-world processes needs to be checked.

According to the steps described in Section 2.2.1, the square distances $d_j^2 = (\mathbf{x}_j - \bar{\mathbf{x}})'\mathbf{S}^{-1}(\mathbf{x}_j - \bar{\mathbf{x}})$ are calculated and the chi-square plot is presented in Figure 12 (front beam) and Figure 13 (back beam). Figure 12 (a) and Figure 13 (a) present the complete chi-square plot within the range [0, 1500]; meanwhile, Figure 12 (b) and Figure 13 (b) present the tailored chi-square plot within the range [60, 80]. From these chi-square plots, it is shown that the data set quite follows multivariate normal.

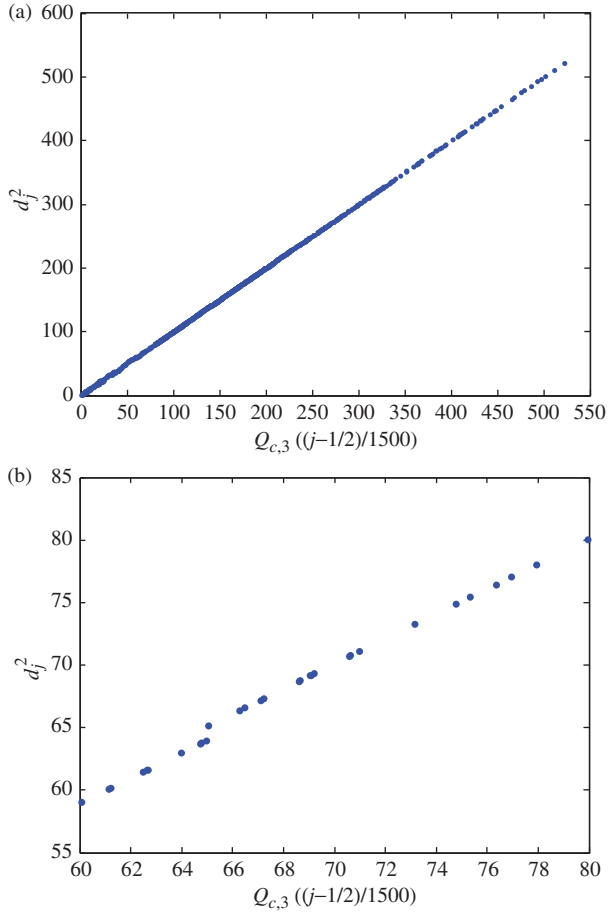

Figure 12. The normality test of the data set for front beam. (a) The complete chi-square plot within the range [0, 1500] for front beam. (b) The tailored chi-square plot within the range [60, 80] for front beam.

5.3 *Classifying results*

This section reports the initial results of applying developed approach into a real aircraft horizontal stabiliser assembly process. In order to further validate the proposed PS-SVME approach, not only the abnormal state (i.e. process mean shifts) but also normal state is considered in this case. Therefore, the output modes can be divided into four categories: normal mode (A_o), one quality variable mean shifts (A_1, A_2, A_3), two quality variables means shift (A_4, A_5, A_6), and three variables means shift (A_7).

Tables 10 and 11 present the output results of the CCP for front beam and back beam KPCs using PS-SVME. Columns 1 to 4 in Tables 10 and 11 show output mode, actual number of times the aircraft horizontal stabiliser assemble process reaches the output mode, the numbers of right classification by PS-SVME and CCP, respectively. The overall percentage of correct recognition for front beam and back beam KPCs is 98.57% and 98.33% respectively, indicating that the developed system exhibits a strong ability to online identify which quality characteristic variable's mean shifts in the aircraft horizontal stabiliser assembly process.

6. Conclusion and future work

In a MP, classifying the source(s) of mean shifts has been a challenging task for traditional MSPC techniques. New approaches need to be developed to accommodate rapid data input rate, and classify the source(s) as soon as possible. In this study, one novel hybrid learning-based approach is explored for classifying the source(s) of mean shifts in MP. By implementing two steps (i.e. creation of component multiclass SVMs and selection of

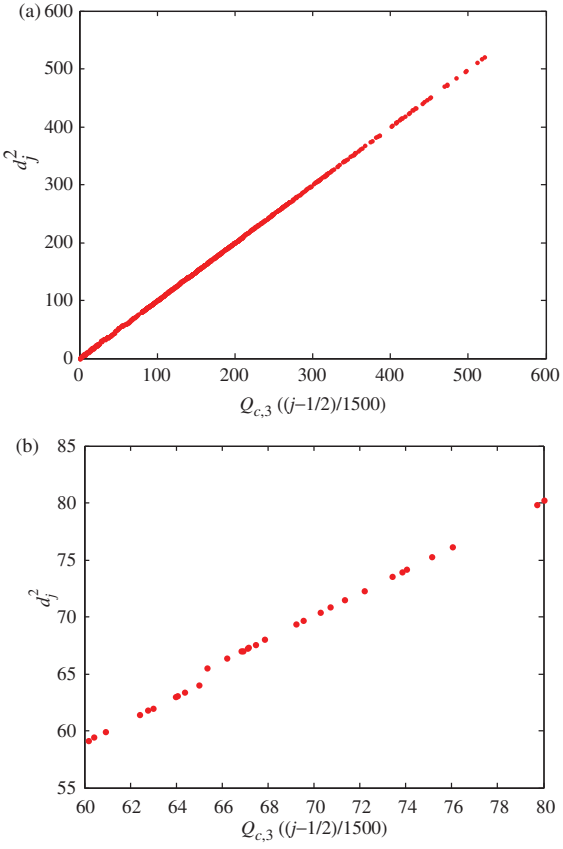

Figure 13. The normality test of the data set for back beam. (a) The complete chi-square plot within the range [0, 1500] for back beam. (b) The tailored chi-square plot within the range [60, 80] for back beam.

Table 10. CCPs for front beam KPCs using PS-SVME.

Output mode	Actual number	The numbers of right classification	CCP%
A_0	1387	1387	100.00
A_1	13	13	100.00
A_2	17	16	94.12
A_3	25	25	100.00
A_4	26	26	100.00
A_5	11	11	100.00
A_6	18	17	94.44
A_7	3	3	100.00
Average			**98.57**

Table. 11. CCPs for back beam KPCs using PS-SVME.

Output mode	Actual number	The numbers of right classification	CCP%
A_0	1380	1380	100.00
A_1	17	17	100.00
A_2	20	19	95.00
A_3	12	12	100.00
A_4	24	22	91.67
A_5	25	25	100.00
A_6	21	21	100.00
A_7	1	1	100.00
Average			**98.33**

optimal subsets), PS-SVME that ensembles accurate and diverse component multiclass SVMs is developed. The simulation results indicate that PS-SVME shows improved generalisation performance, which outperforms those of single best SVMs and ensemble all approach. Moreover, the influences of some key parameters of PS-SVME upon its generalisation are analysed, which aims to find the suitable parameters for constructing PS-SVME. The effectiveness of the developed approach is further validated through the data from the real-world aircraft horizontal stabiliser assembly processes. The analysis from this study provides guidelines in developing selective multiclass SVMs ensemble-based MSPC recognition systems in multivariate processes.

The existing problem for the proposed approach is that although the effectiveness of the proposed approach has been validated in the processes with two and three variables, its effectiveness needs to be further validated in the processes with four or more variables. Further work also can focus on finding out the better optimisation algorithms to find an optimal subset of parameters of SVMs classifier. Another further investigation can be done to apply the proposed approach into monitoring and classifying the covariance changes, which is also important in MSPC, since the proposed approach only monitors and classifies the shifts of process mean in this study. Moreover, in the future more industry cases should be collected in order to further validate and improve the proposed approach.

Acknowledgements

The authors greatly acknowledge the editor and the referees for their valuable comments and suggestions that have led to a substantial improvement of the paper. This work was supported in part by the National Natural Science Foundation of China

(Grant No. 50905114), 863 High-Tech Project of China (Grant No. 2009AA043000, 2009AA043001), the Research Fund of State Key Lab of MSV of China (Grant No. MSV-2010-13), and the Program of Introducing Talents of Discipline to Universities (B06012).

References

Anderson, T.W., 2003. *An introduction to multivariate statistical analysis*. 3rd ed. New York: John Wiley.

Aparisi, F., Avendano, G., and Sanz, J., 2006. Techniques to interpret T^2 control chart signals. *IIE Transactions*, 38 (3), 647–657.

Bersimis, S., Psarakis, S., and Panaretos, J., 2007. Multivariate statistical process control charts: an overview. *Quality and Reliability Engineering International*, 23 (5), 517–543.

Box, G.E.P. and Cox, D.R., 1964. An analysis of transformations. *Journal of the Royal Statistical Society (B)*, 26 (2), 211–252.

Breiman, L., 1996. Bagging predictors. *Machine Learning*, 24 (2), 123–140.

Chan, J., Huang, C., and DeFries, R., 2001. Enhanced algorithm performance for land cover classification from remotely sensed data using bagging and boosting. *IEEE Transactions on Geoscience and Remote Sensing*, 39 (3), 693–695.

Chang, S.I. and Aw, C.A., 1996. A neural fuzzy control chart for detecting and classifying process mean shifts. *International Journal of Production Research*, 34 (8), 2265–2278.

Chang, S.I. and Ho, E.S., 1999. A two-stage neural network approach for process variance change detection and classification. *International Journal of Production Research*, 37 (7), 1581–1599.

Chen, S., Cowan, C.F.N., and Grant, P.M., 1991. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transaction on Neural Networks*, 2 (2), 302–309.

Cheng, C.S., 1995. A multi-layer neural network model for detecting changes in the process mean. *Computer and Industrial Engineering*, 28 (1), 51–61.

Cheng, C.S. and Cheng, H.P., 2008. Identifying the source of variance shifts in the multivariate process using neural networks and support vector machines. *Expert Systems with Applications*, 35 (1–3), 198–206.

Chinnam, R.B., 2002. Support vector machines for recognising shifts in correlated and other manufacturing processes. *International Journal of Production Research*, 40 (17), 4449–4466.

Cook, D.F. and Chiu, C.C., 1998. Using radial basis function neural networks to recognise shifts in correlated manufacturing process parameters. *IIE Transactions*, 30 (3), 227–234.

Cook, D.F., Zobel, C.W., and Nottingham, Q.J., 2001. Utilisation of neural networks for the recognition of variance shifts in correlated manufacturing process parameters. *International Journal of Production Research*, 39 (17), 3881–3887.

Cortes, C. and Vapnik, V., 1995. Support vector network. *Machine learning*, 20 (3), 273–297.

Crammer, K. and Singer, Y., 2000. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47 (2–3), 201–233.

Cristianini, N. and Shawe-Taylor, J., 2000. *A introduction to support vector machines and other kernel-based learning methods*. Cambridge, UK: Cambridge University Press.

Du, S. and Xi, L., 2010. An integrated system for on-line intelligent monitoring and identifying process variability and its application. *International Journal of Computer Integrated Manufacturing*, 23 (6), 529–542.

Du, S., *et al*., 2008. Product lifecycle-oriented quality and productivity improvement based on stream of variation methodology. *Computers in Industry*, 59 (2–3), 180–192.

Eberhart, C.R. and Kennedy, J., 1995. Particle swarm optimisation. *Proceedings of the IEEE international conference on neural networks*, date, Piscataway, NJ. Place: Publisher, 1942–1948.

Eberhart, C.R. and Kennedy, J., 1995. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 27 November–1 December, Perth, Australia. Piscataway, NJ: IEEE, 1942–1948.

Eom, J.H., Kim, S.C., and Zhang, B.T., 2008. AptaCDSS-E: A classifier ensemble based clinical decision support system for cardiovascular disease level prediction. *Expert System with Application*, 34 (4), 2465–2479.

Freund, Y. and Schapire, R., 1997. A decision-theoretic generalisation of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55 (1), 119–139.

Ho, E. S. and Chang, S.I., 1999. An integrated neural network approach for simultaneous monitoring of process mean and variance shifts – a comparative study. *International Journal of Production Research*, 37 (8), 1881–1901.

Hsu, C.W. and Lin, C.J., 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13 (2), 415–425.

Hwarng, H.B., 2004. Detecting process mean shift in the presence of autocorrelation: A neural-network based monitoring scheme. *International Journal of Production Research*, 42 (3), 573–595.

Gordon, J.J., *et al*., 2006. Improved prediction of bacterial transcription start sites. *Bioinformatics*, 22 (2), 142–148.

Guh, R.S., 2007. On-line identification and quantification of mean shifts in bivariate processes using a neural network-based approach. *Quality and Reliability Engineering International*, 23 (3), 367–385.

Guh, R.S. and Shiue, Y.R., 2008. An effective application of decision tree learning for on-line detection of mean shifts in multivariate control charts. *Computers & Industrial Engineering*, 55 (2), 475–493.

Gunn, S.R., 1998. Support vector machines for classification and regression. Technical Report. University of Southampton.

Gupta, A.K., 2010. Predictive modelling of turning operations using response surface methodology, artificial neural networks and support vector regression. *International Journal of Product Research*, 48 (3), 763–778.

Johnson, R. A, and Wichern, D. W., 2007. *Applied multivariate statistical analysis*, 6th ed., Upper Saddle River, NJ: Prentice Hall.

Kourti, T. and MacGregor, J.F., 1996. Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, 28 (4), 409–428.

Lei, Z., Yang, Y., and Wu, Z., 2006. Ensemble of support vector machine for text independent speaker recognition. *International Journal of Computer Science and Network Security*, 6 (5A), 163–167.

Li, J., Jin, J., and Shi, J., 2008. Causation-based T^2 decomposition for multivariate process monitoring and diagnosis. *Journal of Quality Technology*, 40 (1), 46–68.

Lv, P., Yuan, L., and Zhang, J.F., 2009. Cloud theory-based simulated annealing algorithm and application. *Engineering Applications of Artificial Intelligence*, 22 (4–5), 742–749.

Mason, R.L., Tracy, N.D., and Yong, J.C., 1995. Decomposition of it T^2 for multivariate control chart interpretation. *Journal of Quality Technology*, 27 (2), 140–143.

Mason, R.L., Tracy, N.D., and Yong, J.C., 1997. A practical approach of interpreting multivariate T^2 control chart signals. *Journal of Quality Technology*, 29 (4), 396–406.

Montgomery, D.C., 2001. *Statistical quality control*. 4th ed. New York: Wiley.

Niaki, S.T.A. and Abbasi, B., 2005. Fault diagnosis in multivariate control charts using artificial neural networks. *Quality and Reliability Engineering International*, 21 (8), 825–840.

Niaki, S.T.A. and Abbasi, B., 2008. Detection and classification mean-shifts in multi-attribute processes by artificial neural networks. *International Journal of Production Research*, 46 (11), 2945–2963.

Pacella, M., Semeraro, Q., and Anglani, A., 2004. Manufacturing quality control by means of a Fuzzy ART network trained on natural process data. *Engineering Applications of Artificial Intelligence*, 17 (1), 83–96.

Pal, S.K. and Mitra, S., 1992. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transaction on Neural Networks*, 3 (5), 683–697.

Pandey, V., Tiwari, M.K., and Kumar, S., 2006. An interactive approach to solve the operation sequencing problem using simulated annealing. *International Journal of Advanced Manufacturing Technology*, 29 (11–12), 1212–1231.

Pugh, G.A., 1989. Synthetic neural networks for process control. *Computer and Industrial Engineering*, 17 (1–4), 24–26.

Schapire, R.E., 1990. The strength of weak learnability. *Machine Learning*, 5 (2), 197–227.

Smith, A.E., 1994. X-Bar and R control chart interpretation using neural computing. *International journal of Production Research*, 32 (2), 309–320.

Specht, D.F., 1990. Probabilistic neural networks. *Neural Networks*, 3 (1), 109–118.

Sun, R. and Tsung, F., 2003. A kernel-distance-based multivariate control chart using support vector methods. *International Journal of Production Research*, 41 (13), 2975–2989.

Vapnik, V.N., 1995. *The nature of statistical learning theory*. Berlin: Springer.

Vapnik, V.N., 1998. *Statistical learning theory*. New York: Wiley.

Verron, S., Li., J., and Tiplica, T., 2010. Fault detection and isolation of faults in a multivariate process with Bayesian network. *Journal of Process Control*, 28 (8), 902–911.

Woodall, W.H. and Montgomery, D.C., 1999. Research issues and ideas in statistical process control. *Journal of Quality Technology*, 31 (4), 376–386.

Yu, J.B. and Xi, L.F., 2009. A hybrid learning-based model for on-line monitoring and diagnosis of out-of-control signals in multivariate manufacturing processes. *International Journal of Production Research*, 47 (15), 4077–4108.

Zhou, Z.H., Wu, J.X., and Tang, W., 2002. Ensembling neural networks: many could be better than all. *Artificial Intelligence*, 137 (1–2), 239–263.

Appendix

Assume the covariance matrix is denoted by:

$$\boldsymbol{\Sigma} = E(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})' = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1q} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1q} & \sigma_{2q} & \cdots & \sigma_{qq} \end{bmatrix} \tag{A1}$$

The correlation coefficient ρ_{ij} is defined in terms of the covariance σ_{ij} and variances σ_{ii} and σ_{jj} as

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}} \times \sqrt{\sigma_{jj}}} \tag{A2}$$

Let the population correlation matrix be the $q \times q$ symmetric matrix:

$$\rho = \begin{bmatrix} \frac{\sigma_{11}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{11}}} & \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}} & \cdots & \frac{\sigma_{1p}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{pp}}} \\ \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}} & \frac{\sigma_{22}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{22}}} & \cdots & \frac{\sigma_{11}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{qq}}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\sigma_{1q}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{qq}}} & \frac{\sigma_{2q}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{qq}}} & \cdots & \frac{\sigma_{qq}}{\sqrt{\sigma_{qq}}\sqrt{\sigma_{qq}}} \end{bmatrix} = \begin{bmatrix} 1 & \rho_{12} & \cdots \rho_{1q} & \\ \rho_{12} & 1 & \cdots \rho_{2q} & \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1q} & \rho_{2q} & \cdots 1 & \end{bmatrix} \tag{A3}$$

And let the $q \times q$ standard deviation matrix be

$$\mathbf{V}^{1/2} = \begin{bmatrix} \sqrt{\sigma_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{\sigma_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\sigma_{qq}} \end{bmatrix} \tag{A4}$$

Thus,

$$\begin{aligned} \mathbf{V}^{1/2}\boldsymbol{\rho}\mathbf{V}^{1/2} &= \begin{bmatrix} \sqrt{\sigma_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{\sigma_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\sigma_{qq}} \end{bmatrix} \begin{bmatrix} \frac{\sigma_{11}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{11}}} & \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}} & \cdots & \frac{\sigma_{1p}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{pp}}} \\ \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}} & \frac{\sigma_{22}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{22}}} & \cdots & \frac{\sigma_{11}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{qq}}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\sigma_{1q}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{qq}}} & \frac{\sigma_{2q}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{qq}}} & \cdots & \frac{\sigma_{qq}}{\sqrt{\sigma_{qq}}\sqrt{\sigma_{qq}}} \end{bmatrix} \begin{bmatrix} \sqrt{\sigma_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{\sigma_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\sigma_{qq}} \end{bmatrix} \\ &= \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1q} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1q} & \sigma_{2q} & \cdots & \sigma_{qq} \end{bmatrix} \end{aligned} \tag{A5}$$

Thus, $\mathbf{V}^{1/2}\boldsymbol{\rho}\mathbf{V}^{1/2} = \boldsymbol{\Sigma}$.

Therefore, without loss of generality, if all variables have mean zero and variance one, the covariance matrix is in correlation form, namely, the main diagonal elements are all one and the off-diagonal elements are the pair-wise correlation (ρ) between the process variables.

$$\boldsymbol{\Sigma} = \mathbf{R} = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1q} \\ \rho_{12} & 1 & \cdots & \rho_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1q} & \rho_{2q} & \cdots & 1 \end{bmatrix} \tag{A6}$$

$\mathbf{R}$ is a general correlation matrix. For example, if there exist two variables with mean $\mathbf{R}$ zero and variance one, then,

$$\begin{aligned} \mathbf{V}^{1/2}\boldsymbol{\rho}\mathbf{V}^{1/2} &= \begin{bmatrix} \sqrt{\sigma_{11}} & 0 \\ 0 & \sqrt{\sigma_{22}} \end{bmatrix} \begin{bmatrix} \frac{\sigma_{11}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{11}}} & \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}} \\ \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}} & \frac{\sigma_{22}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{22}}} \end{bmatrix} \begin{bmatrix} \sqrt{\sigma_{11}} & 0 \\ 0 & \sqrt{\sigma_{22}} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \sigma_{12} \\ \sigma_{12} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{bmatrix} = \mathbf{R} = \begin{bmatrix} 1 & \sigma_{12} \\ \sigma_{12} & 1 \end{bmatrix} = \boldsymbol{\Sigma} \end{aligned}$$

Namely, $\mathbf{R} = \begin{bmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{bmatrix}$