Neurocomputing 74 (2011) 2083–2095

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

On-line analysis of out-of-control signals in multivariate manufacturing processes using a hybrid learning-based model

Mojtaba Salehi [a], Ardeshir Bahreininejad [a,b,*], Isa Nakhai [a]

[a] Faculty of Engineering, Tarbiat Modares University, 14115 Tehran, Iran
[b] Department of Engineering Design and Manufacture, University of Malaya, Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 26 June 2010
Received in revised form
6 November 2010
Accepted 22 December 2010
Communicated by J. Zhang
Available online 21 March 2011

Keywords:
Multivariate manufacturing processes
Neural network
χ^2 chart
Statistical process control
Support vector machine

ABSTRACT

Advanced automatic data acquisition is now widely adopted in manufacturing industries and it is common to monitor several correlated quality variables simultaneously. Most of multivariate quality control charts are effective in detecting out-of-control signals based upon an overall statistics in multivariate manufacturing processes. The main problem of such charts is that they can detect an out-of-control event but do not directly determine which variable or group of variables has caused the out-of-control signal and what is the magnitude of out of control. This study presents a hybrid learning-based model for on-line analysis of out-of-control signals in multivariate manufacturing processes. This model consists of two modules. In the first module using a support vector machine-classifier, type of unnatural pattern can be recognized. Then by using three neural networks for shift mean, trend and cycle it can be recognized magnitude of mean shift, slope of trend and cycle amplitude for each variable simultaneously in the second module. The performance of the proposed approach has been evaluated using two examples. The output generated by trained hybrid model is strongly correlated with the corresponding actual target value for each quality characteristic. The main contributions of this work are recognizing the type of unnatural pattern and classification major parameters for shift, trend and cycle and for each variable simultaneously by proposed hybrid model.

1. Introduction

Control charts are the most widely applied statistical process control (SPC) tools used to reveal abnormal variations of monitored measurements. Such charts are useful in identifying the presence of assignable causes in manufacturing processes. A process is out-of-control when points fall outside the control chart limits or the control charts display unnatural (non-random) patterns [1]. These unnatural or non-random patterns include shift mean, trend, cycle and mixture of them. Those non-random control chart patterns (CCPs) can be associated with a specific set of assignable causes provided that appropriate process knowledge is available. Hence effective identification of non-random patterns can greatly narrow down a set of possible assignable causes to be investigated, and significantly speed up the diagnostic process.

Control charts do not provide any pattern-related information when the process is out-of-control. Many supplementary rules, like zone tests or run rules and expert systems have also been implemented in control chart pattern recognition (CCPR). But according to the reported works, the overall percentages of correctly recognized for these approaches is low. Recently, many studies used artificial neural networks (ANNs) in order to detect patterns more effectively than the conventional approach and their aim is the automatic diagnosis of the patterns. On the other hand, since some advanced automatic data collection and inspection techniques are widely adopted in manufacturing industries. The tasks of SPC traditionally performed by quality engineers have to be automated. Therefore, various artificial intelligence approaches and machine learning techniques have been applied into SPC. Neural networks (NNs) have excellent noise tolerance in real-time, requiring no hypothesis on statistical distribution of monitored measurements. This important feature makes NNs promising and effective tools that can be implemented to improve data analysis in manufacturing quality control applications. In addition, in recognition problems, NNs can recall learned patterns from noisy representations. This feature makes NNs highly appropriate for CCPR because unnatural CCPs are generally contaminated by natural variations in the process. Such applications have been reported to outperform the conventional methods in terms of recognition accuracy and speed. We briefly review development NNs in CCPR as follows.

Pugh [2,3] compared the performance of Shewhart Charts with that of back propagation network (BPN) models in detecting the process mean shift. The trained network generated average run

* Corresponding author.
E-mail address: bahreininejad@gmail.com (A. Bahreininejad).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.12.020

length (ARL) results of around the ARL of an X-bar chart for large shifts, for a sample size of five. Hwarng [4] specified a unique approach for training BPN solely for analyzing cycle patterns on control charts. The recognition output of the network was the amplitude of the cycle pattern, such as 0.5, 1.0 and 1.5 s (where s is the standard deviation of process). Hwarng and Chong [5] proposed an Adaptive Resonance Theory (ART)-based pattern recognizer to detect unnatural CCPs. In their work, the ART-based recognizer could perform fast and cumulative learning due to the unsupervised-learning nature of the ART networks. The ART-based recognizer was superior to the BPN-based recognizer for cycle patterns, inferior for mixture patterns and similar for other patterns. Pham and Chan [6,7] described a type of neural network for control chart pattern recognition. The neural network is self organizing and can learn to recognize new patterns in an on-line incremental manner. Sagiroglu et al. [8] described a type of neural network for speeding up the training process and to compare three training algorithms in terms of speed, performance and parameter complexity for control chart pattern recognition. The networks are multilayered perceptrons trained with a resilient propagation, back propagation (BP) and extended delta-bar-delta algorithms. The recognition results of CCPs show the BP algorithm is accurate and provides better and faster results, than other algorithms. Chiu et al. [9] used a BPN to identify shifts in process parameter values according to time series models with various autocorrelation coefficients. Their results revealed that NNs successfully separated data that were shifted by one, two and three standard deviations from non-shifted data. The conventional control charts could not identify the same process shifts. Hwarng [10] presents a neural-network based methodology for monitoring a process shift in the presence of autocorrelation. These works suggest a competitive alternative to the existing univariate control charts, and some also provide much further information lacked by conventional univariate control charts. Niaki and Abbasi [11] developed a special two levels-based model using T^2 control chart for detecting the out-of-control signals and a multilayer perceptron neural network for identifying the source(s) of the out-of-control signals. Guh [12] presented a hybrid learning-based model, which integrates NN and decision tree (DT) learning techniques, to detect and discriminate typical unnatural CCPs, while identifying the major parameter (such as the shift displacement or trend slope) and starting point of the CCP detected. Chen et al. [13] presented a hybrid approach by integrating wavelet method with ANNs for on-line recognition of CCPs including concurrent patterns. The main advantage of his approach is its capability of recognizing coexisted or concurrent patterns without training by concurrent patterns.

In many quality control settings, the manufacturing process may have two or more correlated. The usual practice has been to maintain a separate (univariate) chart for each characteristic. Unfortunately, this could result in some fault out-of-control alarms when the characteristics are highly correlated. Hotelling's T^2 statistic might be the most common tool in multivariate analysis for identifying whether the whole process is in out-of-control state. While neural network approaches are generally focused on univariate charts, the application of the neural network approach to multivariate processes has rarely been considered that we review them here briefly. Chen and Wang [14] developed an artificial neural network-based model for identifying the characteristic or group of characteristics that cause the signal and for classifying the magnitude of the shifts. Guh and Shiue [15] proposed a straightforward and effective model to detect the mean shifts in multivariate control charts using decision tree learning techniques. Experimental results using simulation showed that the proposed model could not only efficiently detect the mean shifts but also accurately identify the variables that have deviated from their original means. Cheng and Cheng [16] formulated the interpretation of out-of-control signal as a classification problem. The proposed system includes a shift detector and a classifier. When an out-of-control signal is generated, a classifier will determine which variable is responsible for the variance shift by two classifiers based on neural networks (NN) and support vector machines (SVM). Yu and Xi [17] presented a learning-based model for monitoring and diagnosing out-of-control signals in a bivariate process. In their model, a selective neural network ensemble approach was developed for performing these tasks. El-Midany et al. [18] proposed a framework for multivariate process control chart recognition. The proposed methodology uses the artificial neural networks to recognize set of subclasses of multivariate abnormal patterns, identify the responsible variable(s) on the occurrence of abnormal pattern and classify the abnormal pattern parameters. In the most presented approaches, recognition problem is limited to identifying the characteristic or group of characteristics that cause the mean shift, some of researches consider the trend also separately. But in this study we try to propose a new approach that can identify shift, trend and cycle for each quality variables simultaneously and identify the major parameter for each deviated quality variable.

To clarify the main problem, Let $X_{ij}=(X_{ij1},X_{ij2},\ldots,X_{ijp})$ be a p dimension vector that represents the p quality characteristics in the jth observation of the ith subgroup (sample), where $i=1, 2, \ldots$ and $j=1, 2, \ldots, n$. The lth component of X_{ij}, X_{ijl} denotes the lth quality characteristic, $l=1, 2, \ldots, p$. It is assumed that X_{ij} is independent and identically multivariate normal distribution with known mean μ and covariance matrix Σ when the process is in control. Let $\overline{X_i}$ represent the mean vector for the ith subgroup. The statistic plotted on a multivariate χ^2 control chart for the ith subgroup is given by

$$\chi_i^2 = n(\overline{X}_i-\mu)\Sigma^{-1}(\overline{X}_i-\mu) \tag{1}$$

When the process is in control, it follows a χ^2 central distribution with p degrees of freedom. Therefore, a multivariate χ^2 control chart can be constructed by plotting $\chi_i{}^2$ versus time with an upper control limit (UCL) given by $\chi^2_{\alpha,p}$ where α is an appropriate significance level for performing the test.

Most previous works consider only one type of unnatural patterns for multivariate process or they consider multiunnatural pattern for univariate process. In addition few works that consider recognition problem of multiunnatural pattern for multivariate process, do not obtain any information about magnitude of deviations. This information can help quality participators for rapid recognition of root unnatural patterns. Type of unnatural patterns and major parameters in shift, trend and cycle for each variable will be recognized by proposed hybrid learning-based model simultaneously. The proposed model consists of two modules. In the first module using a support vector machine-classifier, type of unnatural pattern can be recognized. Then by using three special-neural networks for shift mean, trend and cycle, it can be recognized magnitude of mean shift, slope of trend or magnitude of mean shift for each quality variable simultaneously.

The rest of this research is organized as follows. Section 2 describes proposed model for solving CCPR problem. Section 3 presents a case study and a simulated example CCPR problems that are solved with proposed algorithm and conclusions will be drawn in Section 4.

2. Material and methods

Since many CCPs often appear separately or together in a manufacturing process, a CCPR system can be developed and

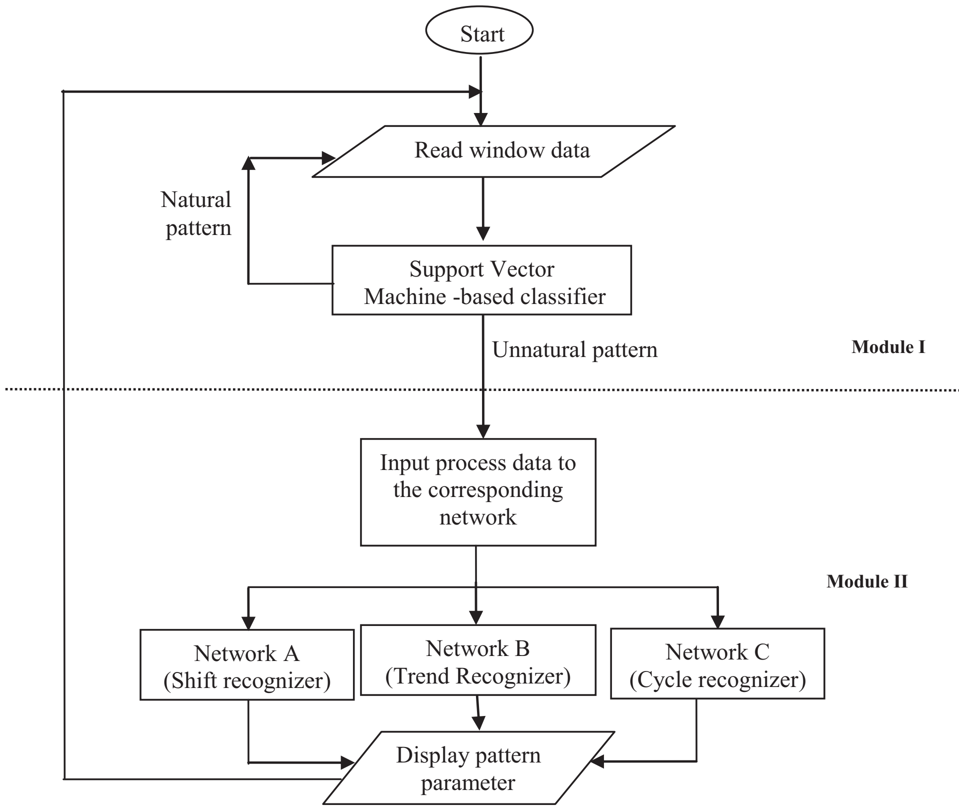

Fig. 1. Proposed approach for CCPR.

trained either as a general-purpose system that can recognize several types of CCP, or as a special-purpose system that can recognize only a particular type of CCP [12]. A suitably developed special-purpose system should be more effective than a general-purpose system for analyzing a particular type of CCP [4]. In this research a modular framework was presented for using advantages of general-purpose and special-purpose simultaneously. Approximately it is impossible that one network would be employed to perform all the required recognition and analysis functions. Because in this situation, the network would have to be complex and convergence in network training would be very difficult. By modular structure we can split the main recognition problem into more manageable sub-problems.

In this research module I is a support vector machine-based classifier as a general propose system for recognizing unnatural pattern chart and module II is a neural network-based classifier as a special propose system for estimating the major parameters of the unnatural CCPs for all quality variable simultaneously. As Fig. 1 shows, at first the proposed system reads data process, then by support vector machine it classifies the situation of processes in natural, shift, trend and cycle. If module I showed a special unnatural pattern, the corresponding network in module II was implemented for estimation the major parameter of the unnatural CCPs, otherwise collecting of data would be continued that this property can save computation cost.

2.1. Module I

In this module using a support vector machine various types of unnatural CCPs were detected. SVM is a related supervised learning method used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. We describe SVM as follows.

2.1.1. SVM-classier

Firstly, SVM transforms the input space to a higher dimensional space. Second SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. A good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

To clarify SVM Consider set $\{(X_i,y_i)\}_{i=1}^{N}$ as training vectors, where $X_i \in R^m$ is the ith m-dimensional input vector, $y_i \in \{-1,+1\}$ is known target. We know that the decision function can be written as

$$y_i(W^T\phi(X_i)+1) \geq 1 \tag{2}$$

where W is the vector of hyperplane. The index i labels the N training cases. A separating hyperplane may not exist in many practical applications. The possibility of examples violating Eq. (2) may be allowed for which the slack variables are restricted to $\xi_i \geq 0$. It means that the variables ξ_i are introduced to allow for some classification errors. That is, ξ_i are parameters (errors) for handling no separable data. The mapping function ϕ represents a non-linear transformation to map the input vectors into a high-dimensional feature space. By rewriting the formula for relaxing the constraints we have: $y_i(W^T\phi(X_i)+1) \geq 1-\xi_i$. Thus, the formula of quadratic optimization can be written as follows:

$$\min \quad \frac{1}{2}W^TW + C\sum_{i=1}^{N}\xi_i$$
$$\text{subject to } y(W^T\phi(X_i)+b) \geq 1-\xi_i$$
$$\xi_i \geq 0 \quad (3)$$

where C is the penalty factor and b is a bias term. The parameter C determines the degree of penalty assigned to an error. This parameter can be used to control the trade-off between maximizing the margin and the classification error. The larger the C, the more the error is penalized. Thus, C should be chosen with care to avoid over-fitting.

In 1992, Boser et al. [19] suggested a way to create non-linear classifiers by applying the kernel trick to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. Some popular kernel functions are defined in Table 1.

Since in the SVM, the data was separated by a hyperplane defined by a number of support vectors. The SVM was designed for binary classification problem. However, many real-world problems have more than m two classes. Constructing multi-class SVM is still an on-going research issue. Basically, two types of approaches usually applied to multi-class classification problem. The first case, the SVM was modified in order to incorporate multi-class learning in the quadratic solving algorithm. That is, the SVM treat all classes at once considering only one optimization problem. Unfortunately, this approach can lead to a high computational cost. The second case, several binary SVM classifiers were combined. The one-against-all and one-against-one methods were used in the second case [16].

For m-class problem, the one against-all method constructs m binary SVMs. The ith SVM is trained with all the samples in the ith class with positive labels $y=+1$, and all the other samples with negative labels $y=-1$. The decision function chooses the class of a sample that corresponds to the maximum value of m binary decision functions specified by the furthest positive hyperplane. Thus, it uses a winner takes-all scheme. The one-against-all method is the earliest approach for multi-class SVM. Another method is the one-against-one which consists on a pairwise classification. The basic idea is to use $m(m-1)/2$ binary classifiers covering all pairs of classes instead of using only m classifiers as in the one-against-all. Each of the $m(m-1)/2$ binary classifiers casts one vote for its favored class, finally the class with most votes wins [16].

2.1.2. Model selection

The most important in design a SVM is the selection of an appropriate kernel function that defines the feature space and the mapping function. The choice of kernel functions is strongly problem-dependent. In control chart pattern recognition problem, different kernel functions from Table 1 are used and compared in terms of their classification performance for the selection of kernel function. Consequently, radial basis function showed the best performance.

Table 1
Kernel functions.

Comments	Kernel function
Polynomial (homogeneous)	$k(X_i,X_j)=(X_i.X_j)^d$
Polynomial (inhomogeneous)	$k(X_i,X_j)=(X_i.X_j+1)^d$
Radial basis function	$k(X_i,X_j)=\exp(-\gamma\Vert X_i-X_j\Vert^2)$ for $\gamma>0$
Gaussian radial basis function	$k(X_i,X_j)=\exp(-\frac{\Vert X_i-X_j\Vert^2}{2\sigma^2})$
Hyperbolic tangent	$k(X_i,X_j)=\tanh(\kappa X_iX_j+c)$ for $k>0$ and $c>0$

2.1.3. Training and testing

For setting parameter in a particular kernel function, training is necessary. The RBF kernel function, the parameters must be determined are the kernel parameter γ and the penalty factor C. In this research, the Monte-Carlo simulation method was used to generate the required data sets of normal and abnormal examples for training and testing. However collecting various data from real-world manufacturing systems is better than multivariate simulation. Nevertheless, simulation does provide a platform from which investigation into potential problems associated with occurrence and detection of abnormal patterns in a multivariate system can begin [20].

In order to train and test the SVM-classifier in this study, a process with tree variables is considered which is just an example of a limited case of the general multivariate when p, the number of variables, equals 3. The reference mean vector and covariance matrix for simulated example are:

$$(\mu_1,\mu_2,\mu_3)=(0,0,0),\ \Sigma=\begin{pmatrix}1 & 1.5 & 0.5\\ 1.5 & 3 & 1\\ 0.5 & 1 & 2\end{pmatrix}$$

In this work, SVM was implemented in STATISTICA software. For determining the best parameters, a cross-validation experiment was designed. In this research, γ and C were checked in [0.03, 1] and [1,30] ranges, respectively. Note that in this work, the number of class k is set to 19 as shown in Table 2. After training the SVM by simulated examples $\gamma=0.08$ and $C=12$ are determined. After training the SVM, we implement the test data on the SVM. Table 2 indicated that the proposed approach can recognize the type of unnatural pattern with a good classification rate.

2.2. Module II

This module is responsible for identification of the key parameter of each unnatural pattern. It consists of three separated specialist-NNs for identification the key pattern parameters. It consists of three types of ANN used for identifying the key pattern parameters. It must be noted that If only one network was employed to perform all the required recognition problems, the network would have to be large and complex. This situation would make training and convergence the network very difficult. Modular design splits the original problem into more manageable sub-problems. The selection of each ANN is based on the output of module A. A brief description of ANNs used in this research is given as follows.

2.2.1. Selecting input feature vector

The selection of the feature vector in training set significantly affects the performance of an NN. The input feature vector must be able to intensify the pattern feature of the data set. Most of researches input raw data into a recognizer as an input feature vector but in this study, the statistical features to be extracted from the raw data are added to the input vectors. The use of individual data usually results in a higher type I error (i.e., shorter in-control ARL), it can reveal an out-of-control situation quickly (i.e., lower type II error). In-control ARL means the average number of observations that must be taken before an observation indicates an out-of-control condition when the process is actually in control. In a high-speed automatic production scenario,

Table 2
The result on the test data by the SVM.

No.	Number of errant variables	Type of error	Aggregate CR with different magnitudes of error for variables
1	One quality characteristic	(Shift)	0.947
2	One quality characteristic	(Trend)	0.872
3	One quality characteristic	(Cycle)	0.853
4	Two quality characteristic	(Shift, shift)	0.932
5	Two quality characteristic	(Trend, trend)	0.864
6	Two quality characteristic	(Cycle, cycle)	0.851
7	Two quality characteristic	(Shift, trend)	0.873
8	Two quality characteristic	(Shift, cycle)	0.861
9	Two quality characteristic	(Trend, cycle)	0.831
10	Three quality characteristic	(Shift, shift, shift)	0.891
11	Three quality characteristic	(Trend, trend, trend)	0.854
12	Three quality characteristic	(Cycle, cycle, cycle)	0.832
13	Three quality characteristic	(Shift, shift, trend)	0.882
14	Three quality characteristic	(Shift, shift, cycle)	0.873
15	Three quality characteristic	(Trend, trend, shift)	0.876
16	Three quality characteristic	(Trend, trend, cycle)	0.832
17	Three quality characteristic	(Cycle, cycle, shift)	0.843
18	Three quality characteristic	(Cycle, cycle, trend)	0.12
19	Three quality characteristic	(Shift, trend, cycle)	0.765

detecting and correcting the inceptive problem as early as possible is important in preventing the possible high-speed manufacturing of defects. Battiti [21] and Smith [22] showed that the BPN integrating raw data and statistical features as input feature vectors has improved performance. Montgomery [1] implemented a two-level resolution IV fractional factorial experimental design for screening and selecting a minimal set of representative statistical features from a list of 10 candidate features. Hassan et al. [23] conducted an experimental study and indicated that the BPN using statistical features as input vectors has better performance than those of the other BPN using raw data as input vectors. In this experiment, we implement mean, standard deviation, skewness, mean-square value, autocorrelation and cumulative sum control chart (CUSUM) with raw data as an input feature vector.

Among various out of control conditions, this study is concerned with mean shifts, trends and cycles. When special disturbance at time d_t (zero when no unnatural pattern present) occur at time t_η the observations X of a quality characteristic is expressed as follows:

$$X(t)=\mu+Y(t)+d(t,t_\eta) \quad t\geq t_\eta \tag{4}$$

where, $X(t)$ is the quality characteristics measured at time t, μ is the process mean vector when the process is in control

$$Y(t) \quad N\left(0,\sum\right)$$

$$d(t,t_\eta)=u\times b$$

u is the parameter to determine the position of shifting ($u=0$ before shifting, $u=1$ after shifting).

$$b=(k_1\sigma_1,k_2\sigma_2,\ldots,k_p\sigma_p)$$

where k_l is the magnitude of shifts in terms of σ_l, which is the lth quality characteristic

$$d(t,t_\eta)=s\times t$$

$$s=(r_1\sigma_1,r_2\sigma_2,\ldots,r_p\sigma_p)$$

Table 3
Relationship between target value and type of shift for the lth quality characteristic.

Target value	Magnitude of shift for the lth quality characteristic	Slope of trend for the lth quality characteristic	Magnitude of cycle for the lth quality characteristic
$+1$	$+3\sigma$	$+0.3\sigma$	$+3\sigma$
$+0.7$	$+2\sigma$	$+0.2\sigma$	$+2\sigma$
$+0.4$	$+1\sigma$	$+0.1\sigma$	$+1\sigma$
0	0	0	0
-0.4	-1σ	-0.1σ	–
-0.7	-2σ	-0.2σ	–
-1	-3σ	-0.3σ	–

where r_l is the trend slope in terms of σ_l, which is the lth quality characteristic

$$d(t,t_\eta)=l\sin\left(\frac{2\pi t}{\Omega t}\right)$$

$$l=(h_1\sigma_1,h_2\sigma_2,\ldots,h_p\sigma_p)$$

where l is the cycle amplitude in terms of σ_l. Ω is cycle period.

The feature vector comprises raw observations X_{ij} and the corresponding features.

This study considers seven distinct types of shift, trend and cycle associated with the lth quality characteristic; hence k_l and l_l has seven possible values, from -3 to $+3$ in increments of one and r_l has seven possible values, from -0.3 to $+0.3$ in increments of 0.1 (shown in Table 3).

2.2.2. Designing the network architecture

The size of the input feature vector that is referred as the identification window size can significantly influence the performance of the proposed model. A small input feature vector will typically detect unnatural patterns more quickly, and may also yield a short in-control ARL (equivalent to a high Type I error). A large window can reduce the recognition efficiency by increasing the time required to detect patterns (higher Type II error, or longer out-of-control ARL). The suitable feature vector size here should balance the Type I and Type II errors. Preliminary experiments were implemented to choose an appropriate size of moving window. The proposed model with number of quality variables $p=2$, 3, 5 and 10 and input feature vector size $w=5$, 8, 12, 20 and 30 were studied. To simplify the required simulation and get an approximate result, the correlation between the quality variables were set at 0.5 for all the p values considered. The results are shown in Fig. 2. Each in-control ARL value in Fig. 2 was obtained based on the Average of 800 simulation runs. In these experiment, A threshold value ($\lambda\in[0, 1]$) was implemented to the output the neuron in the output layer. Any value above λ was considered to sign the presence of an unnatural pattern. According to the previous study [24], $\lambda=0.9$ gives appropriate window size with satisfied Type I error. With an in-control state, a Type I error occurred when any of the output values from the output neurons were equal to or greater than λ. Fig. 2 indicates that for a specific number of quality variables (p), the in-Control ARL decreases when the feature vector size decreases. Additionally, the proposed model needs a larger feature vector size to maintain a fixed in-control ARL as then number of quality variables increases. However, it is seen that when p changes between 2 and 10, window size can is considered 12.

In this study a three-layer fully connected feed-forward network with aback-propagation training algorithm was implemented. Fig. 3 shows the neural network architecture, which includes an input layer with 18p nodes that are used as input data for 12 consecutive points in a control chart, a hidden layer with 6p nodes and an output layer with p nodes. In the input layer $X_{ij,l}$ is the value of jth observation ($j=1, 2, \ldots, 12$) for lth quality characteristic in ith batch ($i=1,2, \ldots$). It must be noted that in addition of raw data $X_{if,l}$ shows selected feature (mean, standard deviation, skewness, mean-square value, CUSUM) for lth quality characteristic in ith batch ($i=1\ 2, \ldots$). Each output node represents a unique target value associated with a type of shift, trend or cycle for a particular quality characteristic. Table 3 shows the relationship between target value and type of shift, cycle and trend associated with a particular quality characteristic. Cheng [25] claimed that the hyper tangent transfer function effectively detects process changes in different directions. The hyper tangent (tansig) function was used as the activation function of the hidden and output layers of NNs for shift and trend and the logsig function was used for cycle.

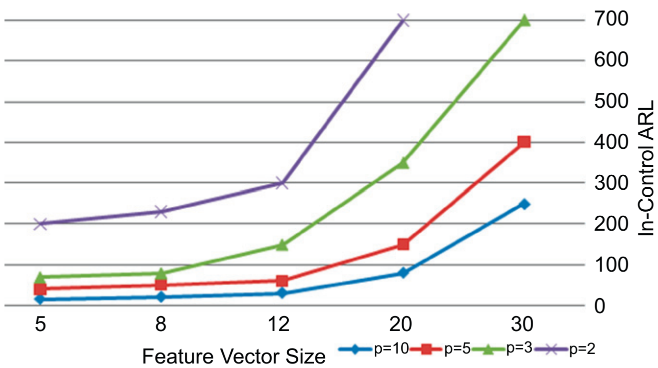

Fig. 2. In-control ARL of the learning-based model with different feature vector size (w) and various number of quality variables (p).

A large recognition window corresponds to higher recognition accuracy. The minimal window size that yielded satisfactory identification accuracy was determined empirically to be 36 (12 nodes for three variables or 12p). Many theoretical and simulative investigations of engineering applications have demonstrated that the number of hidden layers need not exceed two [26]. Since one hidden layer can approximate any continuous mapping from the input patterns to the output patterns in back propagation network, one hidden layer was considered. The optimal number of node depends on the problem. If a network has too few hidden neurons, it cannot learn the training set and cannot generalize well. On the other hand, a network with too many neurons can tend to memorize the training set and will also damage the ability of the network to generalize. In this study, the number of hidden neurons was selected based on trial and error experiments. The proposed neural network is shown in Fig. 3.

2.2.3. *Training and parameter setting*

To train the three special neural networks, 50 examples were generated using CCP example simulator for each type of unnatural pattern. This quantity that can change was determined by an experimental study that revealed that increasing the number of examples did not significantly improve the learning performance.

The initial weights were randomly set between [−0.01, +0.01]. The epochs of the iteration were 350. The learning rate and momentum factor were set to 0.15 and 0.3, respectively. These NN training parameters were set mainly based on trial-and-error experiments performed to determine the best NN training parameters, however for more investigation for parameter setting interested readers are referred to Guh [27] that used the genetic algorithm (GA) to evolve the NN structure, while simultaneously determining a training parameter set (including learning rate, momentum factor, initial range of weights and others) to yield

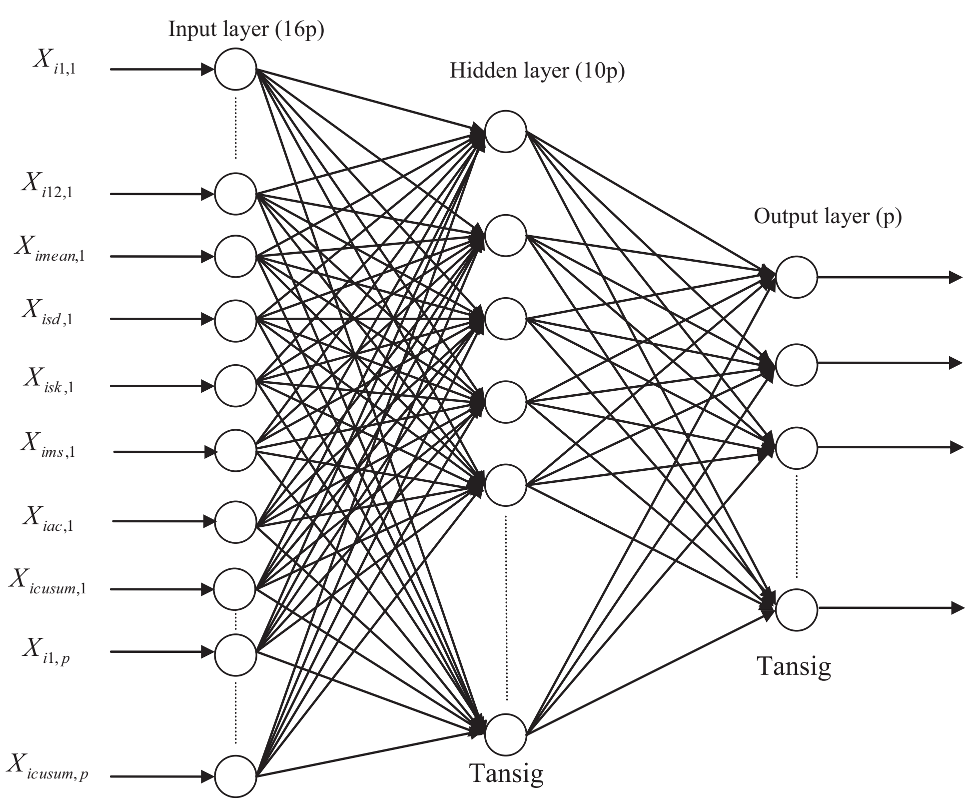

Fig. 3. The neural network architecture.

efficiently a near-optimal NN model for the specific application encountered. Barghash and Santarisi [28] also attempted to explore the effect of the training parameters on the performance of the NNs. Their results showed that many parameters usually assigned by experience have significant effect on the performance of the NN. The BPN were trained by the implementation of back-propagation algorithm in MATLAB toolbox. During training, the convergence condition was reached within 100 training epochs.

This module implements the Levenberg–Merquardt Quasi-Network approach as training algorithm. The input vectors are

Table 4
Matrix of shift test result on the training data for module II.

Shift	Magnitude of shift	Target value	Average output of major parameter special network		
			Mean	Aggregate standard deviation	Errors
1nd quality characteristic	(+2, 0, 0)	(+0.7, 0, 0)	(+0.73, +0.08, +0.11)	0.180	(0.03, 0.08, 0.11)
	(−2, 0, 0)	(−0.7, 0, 0)	(−0.64, −0.03, +0.09)	0.203	(0.04, 0.03, 0.09)
2nd quality characteristic	(0, +2, 0)	(0, +0.7, 0)	(+0.03, +0.74, +0.06)	0.196	(0.03, 0.04, 0.06)
	(0, −2, 0)	(0, −0.7 ,0)	(−0.07, −0.75, +0.01)	0.222	(0.07, 0.05, 0.01)
3nd quality characteristic	(0, 0, +2)	(0, 0, +0.7)	(−0.09,−0.07,+0.67)	0.214	(0.09, 0.07, 0.03)
	(0, 0, −2)	(0, 0, −0.7)	(+0.13, +0.09, −0.78)	0.219	(0.13, 0.09, 0.08)
1nd and 2nd quality characteristic	(+2, −2, 0)	(+0.7, −0.7, 0)	(+0.64, −0.69,+0.03)	0.195	(0.06, 0.01, 0.03)
	(−2, +2, 0)	(−0.7, +0.7, 0)	(−0.75, +0.71, −0.08)	0.224	(0.05, 0.01, 0.08)
1nd and 3nd quality characteristic	(+2, 0, −2)	(+0.7, 0, −0.7)	(+0.69, −0.10, −0.74)	0.234	(0.01, 0.1, 0.04)
	(−2, 0, +2)	(−0.7, 0, +0.7)	(−0.74, +0.06, +0.72)	0.246	(0.04, 0.06, 0.02)
2nd and 3nd quality characteristic	(0, +2, −2)	(0, +0.7, −0.7)	(+0.07, +0.67, −0.64)	0.216	(0.07, 0.03, 0.06)
	(0, −2, +2)	(0, −0.7, +0.7)	(+0.08, −0.73, +0.67)	0.240	(0.08, 0.03, 0.03)
1nd and 2nd and 3nd quality characteristic	(+2, −2, +2)	(+0.7, −0.7, +0.7)	(+0.74, −0.75, +0.64)	0.289	(0.04, 0.05, 0.06)
	(−2, +2, −2)	(−0.7, +0.7, −0.7)	(−0.69, +0.76, −0.77)	0.273	(0.01, 0.06, 0.07)

Table 5
Matrix of trend test result on the training data for module II.

Trend	Slope of trend	Target value	Average output of major parameter special network		
			Mean Standard deviation	Aggregate	Errors
1nd quality characteristic	(+0.2, 0, 0)	(+0.7, 0, 0)	(+0.67, −0.07, +0.09)	0.169	(0.03, 0.07, 0.09)
	(−0.2, 0, 0)	(−0.7, 0, 0)	(−0.74, −0.13, +0.12)	0.198	(0.04, 0.13, 0.12)
2nd quality characteristic	(0, +0.2, 0)	(0, +0.7, 0)	(+0.10, +0.74, −0.06)	0.231	(0.10, 0.04, 0.06)
	(0, −0.2, 0)	(0, −0.7, 0)	(+0.08, −0.74, +0.08)	0.206	(0.08, 0.04, 0.08)
3nd quality characteristic	(0, 0, +0.2)	(0, 0, +0.7)	(−0.07, +0.09, +0.77)	0.257	(0.07, 0.09, 0.07)
	(0, 0, −0.2)	(0, 0, −0.7)	(−0.07, −0.11, −0.67)	0.243	(0.07, 0.11, 0.03)
1nd and 2nd quality characteristic	(+0.2, −0.2, 0)	(+0.7, −0.7, 0)	(+0.66, −0.78, +0.03)	0.285	(0.04,0.08,0.03)
	(−0.2, +0.2, 0)	(−0.7, +0.7, 0)	(−0.75, +0.71, +0.11)	0.176	(0.05,0.01,0.11)
1nd and 3nd quality characteristic	(+0.2, 0, −0.2)	(+0.7, 0, −0.7)	(+0.76, −0.09, −0.67)	0.213	(0.06, 0.09, 0.03)
	(−0.2, 0, +0.2)	(−0.7, 0, +0.7)	(−0.73, −0.15, +0.79)	0.246	(0.03, 0.15, 0.09)
2nd and 3nd quality characteristic	(0, +0.2, −0.2)	(0, +0.7, −0.7)	(−0.12, +0.73, −0.73)	0.261	(0.12, 0.03, 0.03)
	(0, −0.2, +0.2)	(0, −0.7, +0.7)	(+0.05, −0.79, +0.71)	0.232	(0.05, 0.09, 0.01)
1nd and 2nd and 3nd quality characteristic	(+0.2, −0.2, +0.2)	(+0.7, −0.7, +0.7)	(+0.64, −0.79, +0.68)	0.275	(0.06, 0.09, 0.02)
	(−0.2, +0.2, −0.2)	(−0.7, +0.7, −0.7)	(−0.76, +0.65, −0.76)	0.291	(0.06, 0.05, 0.06)

Table 6
Matrix of cycle test result on the training data for module II.

Cycle	Magnitude of cycle	Target value	Average output of major parameter special network		
			Mean	Aggregate standard deviation	Errors
1nd quality characteristic	(+2, 0, 0)	(+0.7, 0, 0)	(−0.68, −0.07, −0.09)	0.130	(0.02, 0.07, 0.09)
2nd quality characteristic	(0, +2, 0)	(0, +0.7, 0)	(+0.05, −0.73, −0.05)	0.241	(0.05, 0.03, 0.05)
3nd quality characteristic	(0, 0, +2)	(0, 0, +0.7)	(−0.11, −0.12, +0.73)	0.187	(0.11, 0.12, 0.03)
1nd and 2nd quality characteristic	(+2, −2, 0)	(+0.7, −0.7, 0)	(−0.66, +0.72, −0.12)	0.261	(0.04, 0.02, 0.12)
1nd and 3nd quality characteristic	(+2, 0, −2)	(+0.7, 0, −0.7)	(+0.73, +0.11, −0.75)	0.194	(0.03, 0.11, 0.05)
2nd and 3nd quality characteristic	(0, +2, −2)	(0, +0.7, −0.7)	(−0.09, +0.71, +0.68)	0.214	(0.09, 0.01, 0.02)
1nd and 2nd and 3nd quality characteristic	(+2, −2, +2)	(+0.7, −0.7, +0.7)	(−0.75, +0.73, −0.72)	0.183	(0.05, 0.03, 0.02)

Table 7
Evaluation results of neural network A, B and C.

	Average error	Aggregate standard deviation of average error
Shift	(0.0731, 0.0728, 0.0723)	0.0216
Trend	(0.0741, 0.0758, 0.0731)	0.0231
Cycle	(0.0792, 0.0737, 0.0783)	0.0301

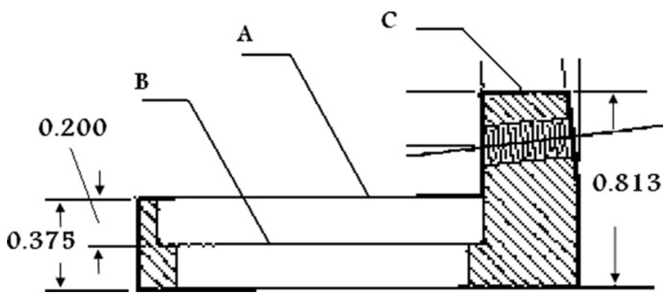

Fig. 4. Plans A and B in typical part.

implemented to the network and propagated forward to yield the output. Connection weights were adjusted after each pattern was presented. The mean square error (MSE) associated with the output layer is propagated backward through the network, by modifying the weights. Since numbers of example is too many, Table 4–6 detail only some of the test examples.

2.2.4. Performance evaluation of module II

For evaluating the performance of module II the simulated examples were run. For having more practical state in this evaluation by a moving window analysis approach, it is assumed that the process starts under an in-control condition. The initial recognition window contains no unnatural CCP points. Unnatural CCPs begin to appear as the recognition window moves along the time series. The pattern features slowly strengthen as the recognition window moves forwards through the process data stream [12]. Table 7 summarizes the average errors and aggregate standard deviation of average error of the CCP parameter identification. Table 7 shows that the overall performance of CCP parameter identification is reasonably good. Notably, the results here were calculated only for the CCPs whose type was correctly recognized by module I.

3. Result and discussion

In this section, we implement the proposed model for a case study (industrial example) and a simulated example. By these examples, we try to discuss advantages and contributions of the model.

3.1. Case study

For implementing the proposed approach, consider the part that is shown in Fig. 4. In this part the plans A, B and C shown are finished on the same machine having the same tool holder. This part is assembled with other parts and the height of plans is correlated to each other. These three heights are considered as x_1, x_2 and x_3, and the variance covariance matrix, S is as follows:

$$S=\begin{bmatrix} 1 & 0.65 & 0.86 \\ 0.65 & 1 & 0.73 \\ 0.86 & 0.73 & 1 \end{bmatrix}$$

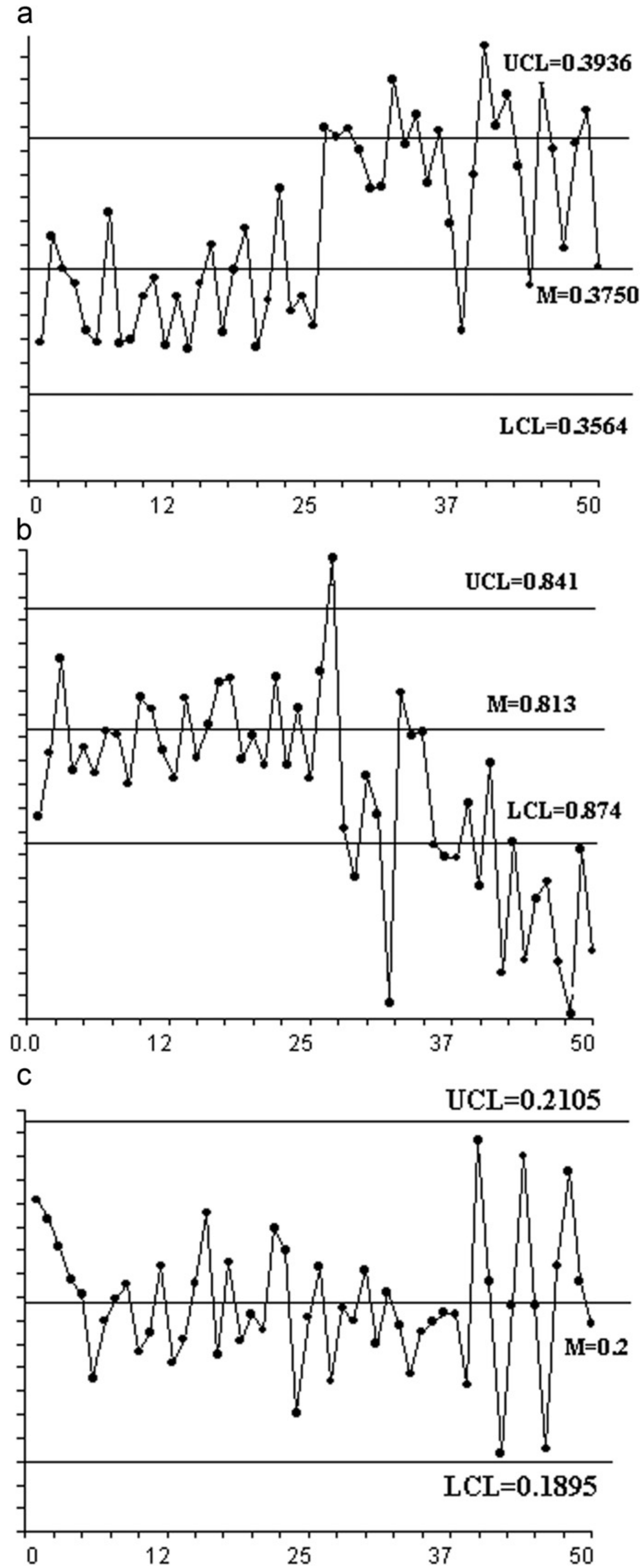

Fig. 5. Individual control chart for x_1, x_2 and x_3 for presented part. (a) Displacement of the mean is $+2\sigma$ for x_1. (b) Slope of trend is -0.2σ for x_2. (c) Magnitude of cycle is $+2\sigma$ for x_3.

In this situation, we have a control chart pattern recognition problem with three variables. Fig. 5 shows the measurements of 50 consecutive batches of the part for each quality characteristic respectively. The in-control mean and standard deviation of the process for x_1 are 0.3750, 0.0062, for x_2 are 0.8130, 0.0096 and for x_3 are 0.200, 0.0035, respectively. The UCL and LCL are calculated to be 0.3936 and 0.3564 for x_1, 0.8418 and 0.7842 for x_2 and 0.2105 and 0.1895 for x_3, respectively. An assignable cause makes a shift for mean x_1 to 0.3874 (displacement of the mean $= +2\sigma$) at sample no. 25, a trend for x_2 (slope of trend $= -0.2\sigma$) at sample no. 25 and a cycle for x_3 (magnitude of the cycle $= +2\sigma$) at sample no. 40. Fig. 5 shows the individual control charts for each variable. Fig. 6 shows χ^2 control chart for these observations.

If we presented collected data to the proposed model in real-time with window size 12, output of the SVM detected an unnatural pattern in 33th and 34th observation. We can see a shift in x_1 and a trend in x_2. With these simultaneous deviations (trend and shift) in quality characteristics, ones may be able to recognize cause of deviations but for more accurate the outputs of networks A and B must be analyzed. Magnitude of shift $+2\sigma$ was recognized by network A in x_1 and magnitude of trend -0.2σ was recognized by network B in x_2. Output of the SVM detected another unnatural pattern for x_3 in 45th observation too. Magnitude of cycle $+2\sigma$ was recognized by network C in x_3. These results are highly correlated with real situations. Outputs of SVM and networks A, B and C for case study were shown in Table 8.

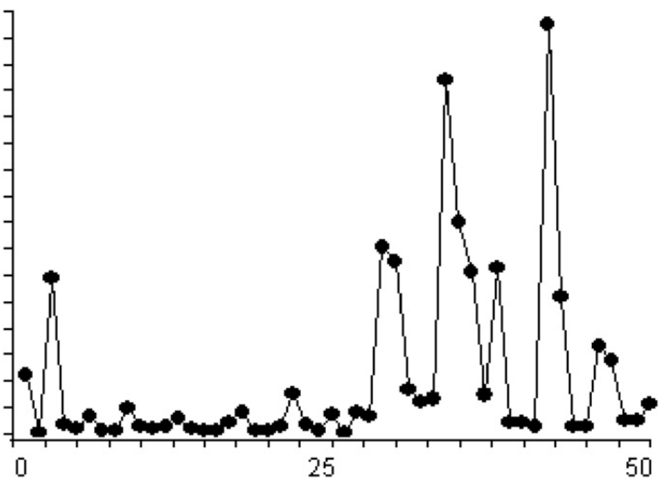

Fig. 6. χ^2 Control chart for x_1 and x_2 for presented part.

The case study is provided to explain further the method developed above. Excluding the natural pattern, $4 \times 4 \times 4 - 1 = 63$ unnatural patterns (four states for each variable are natural, shift, trend and cycle) must be considered for the output of SVM-Classifier. For each type of unnatural pattern, a Monte Carlo simulation generates 800 input vectors when the χ^2 statistic exceeds the UCL. These 800 input vectors for each unnatural pattern include different magnitudes of deviation from natural pattern for each variable according the Table 3. Two-thirds of them were used for training, and the rest of the data were for testing. The training was terminated since the MSE of the trained network was 0.075. Then, the test data set that composed of $16800 (= 1/3 \times 63 \times 800)$ input vectors with a sample size of 12 were classified in the on-line test stage to test the classification capability. The trained model starts to check an abnormal pattern based on the most recent 12 observations. The result was presented in Table 8.

As a result of checking data by the proposed hybrid learning-based model, as it is in above the following ranges of the pattern parameters are classified:

1. The samples 1–21 indicate to the normal pattern.
2. The sample 22 indicates a shift in x_1, therefore the collected data were presented in network A and the real magnitude of shift for x_1 was recognized in sample 26. As a result, magnitude of a shift that has been recognized by SVM-Classifier can be determined after four sampling.
3. The samples 23 indicates a trend in x_2, the real magnitude of slope of trend for x_2 was recognized in sample 28. It means the real magnitude of slope of trend was determined after five sampling.
4. The samples 33 indicates a cycle in x_3, the magnitude of cycle for x_3 was recognized in sample 37 that is after four sampling.

According the results for case study, if an unnatural pattern is caused in collected data, after about six sampling the proposed model can recognize type of unnatural pattern and after about twelve sampling the corresponding network can determine the major parameter.

Table 8
Outputs of SVM and network A, B, C for case study.

No.	Output SVM	Output A	Output B	Output C	No.	Output SVM	Output A	Output B	Output C
1–12	(N, N, N)	–	–	–	**21–32**	(N, N, N)	(0.43, −0.04, 0.03)	(0.08, 0.33, −0.04)	(0.02, −0.01, 0.02)
2–13	(N, N, N)	–	–	–	**22–33**	(S, N, N)	(0.52, 0.06, 0.08)	(0.18, 0.39, −0.03)	(−0.04, 0.06, 0.05)
3–14	(N, N, N)	–	–	–	**23–34**	(S, T, N)	(0.51, 0.04, −0.04)	(−0.03, 0.51, 0.01)	(−0.1, 0.1, −0.01)
4–15	(N, N, N)	–	–	–	**24–35**	(S, T, N)	(0.56, −0.03, 0.01)	(0.07, 0.55, 0.05)	(−0.08, 0.11, 0.01)
5–16	(N, N, N)	–	–	–	**25–36**	(S, T, N)	(0.62, −0.07, −0.01)	(−0.05, 0.63, −0.01)	(0.00, 0.09, 0.1)
6–17	(N, N, N)	–	–	–	**26–37**	(S, T, N)	(0.73, −0.06, 0.05)	(0.03, 0.62, 0.00)	(0.02, 0.02, 0.04)
7–18	(N, N, N)	–	–	–	**27–38**	(S, T, N)	(0.70, −0.09, −0.06)	(0.02, 0.68, 0.02)	(−0.01, −0.04, 0.09)
8–19	(N, N, N)	–	–	–	**28–39**	(S, T, N)	(0.69, −0.11, −0.1)	(−0.08, 0.71, 0.04)	(0.03, 0.05, 0.14)
9–20	(N, N, N)	–	–	–	**29–40**	(S, T, N)	(0.71, −0.12, 0.1)	(−0.10, 0.72, 0.1)	(−0.05, −0.02, 0.21)
10–21	(N, N, N)	–	–	–	**30–41**	(S, T, N)	(0.77, −0.12, 0.03)	(−0.11, 0.70, 0.05)	(−0.06, 0.07, 0.20)
11–22	(N, N, N)	–	–	–	**31–42**	(S, T, N)	(0.63, −0.13, 0.00)	(−0.05, 0.71, 0.06)	(0.01, 0.01, 0.28)
12–23	(N, N, N)	–	–	–	**32–43**	(S, T, N)	(0.67, −0.16, 0.05)	(−0.06, 0.72, 0.06)	(0.05, 0.08, 0.35)
13–24	(N, N, N)	–	–	–	**33–44**	(S, T, C)	(0.66, −0.13, 0.07)	(0.01, 0.75, 0.06)	(−0.01, 0.00, 0.36)
14–25	(N, N, N)	–	–	–	**34–45**	(S, T, C)	(0.64, −0.14, 0.04)	(0.05, 0.71, 0.09)	(−0.06, 0.04, 0.46)
15–26	(N, N, N)	–	–	–	**35–46**	(S, T, C)	(0.71, −0.11, 0.07)	(−0.09, 0.68, 0.1)	(−0.04, −0.02, 0.58)
16–27	(N, N, N)	–	–	–	**36–47**	(S, T, C)	(0.72, −0.18, 0.09)	(0.05, 0.73, 0.12)	(0.01, 0.05, 0.63)
17–28	(N, N, N)	–	–	–	**37–48**	(S, T, C)	(0.70, −0.19, 0.11)	(−0.03, 0.68, 0.11)	(−0.06, 0.11, 0.70)
18–29	(N, N, N)	–	–	–	**38–49**	(S, T, C)	(0.71, −0.21, 0.13)	(0.06, 0.69, 0.15)	(0.03, 0.05, 0.69)
19–30	(N, N, N)	–	–	–	**39–50**	(S, T, C)	(0.68, −0.24, 0.12)	(−0.02, 0.72, 0.13)	(−0.04, 0.00, 72)
20–31	(N, N, N)	–	–	–	**40–51**	(S, T, C)	(0.73, −0.23, 0.14)	(−0.09, 0.75, 0.15)	(−0.01, −0.04, 0.68)[a]

[a] N, S, T and C means normal, shift, trend and cycle, respectively.

3.2. Simulated example

In order to evaluate the proposed approach where the numbers of quality characteristics are high such as in chemical processes, consider a numerical example with ten variables. We assume these variables are correlated according to presented covariance matrix in Table 9. These ten variables are considered as $x_1, x_2, \ldots, x_{10}$.

In this situation, we can reduce dimensions of measured data using reduction techniques such as principle component analysis and extract the main features of data [29,30]. Then, using an appropriate control chart, we can recognize any unnatural pattern

Table 9
Covariance matrix for simulated example.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
x_1	1	0.44	−0.28	0.75	−0.57	0.63	0.37	−0.24	0.51	0.27
x_2	0.44	1	0.54	0.80	−0.54	0.58	0.82	0.81	0.85	−0.75
x_3	−0.28	0.54	1	0.48	0.64	−0.73	0.35	0.9	−0.15	0.68
x_4	0.75	0.80	0.48	1	0.47	−0.24	0.32	0.77	−0.34	−0.55
x_5	−0.57	−0.54	0.64	0.47	1	−0.27	0.58	0.94	0.32	−0.57
x_6	0.63	0.58	−0.73	−0.24	−0.27	1	−0.24	0.57	0.74	0.76
x_7	0.37	0.82	0.35	0.32	0.58	−0.24	1	0.67	−0.85	−0.27
x_8	-0.24	0.81	0.9	0.77	0.94	0.57	0.67	1	0.54	0.42
x_9	0.51	0.85	−0.15	−0.34	0.32	0.74	−0.85	0.54	1	0.83
x_{10}	0.27	−0.75	0.68	−0.55	−0.57	0.76	−0.27	0.42	0.83	1

Table 10
Output of Network A, B, C for simulated example.

No.	**Network**	**Output**
23–34	Output A	(0.13, −0.19, 0.11, 0.05, −0.08, −0.13, 0.12, 0.05, 0.08, 0.07)
	Output B	(0.13, −0.08, 0.1, 0.03, −0.04, −0.09, 0.08, −0.04, 0.03, 0.06)
	Output C	(0.09, −0.04, 0.09, 0.04, −0.04, −0.06, 0.13, 0.05, 0.04, 0.07)
24–35	Output A	(0.17, −0.22, 0.10, 0.04, −0.06, −0.15, 0.12, 0.1, 0.1, 0.06)
	Output B	(0.12, −0.06, 0.11, 0.04, −0.07, −0.1, 0.14, 0.08, 0.06, 0.05)
	Output C	(0.13, −0.09, 0.13, 0.03, −0.11, −0.03, 0.15, 0.1, 0.13, 0.12)
25–36	Output A	(0.23, −0.25, −0.05, 0.07, −0.1, −0.13, 0.11, 0.13, 0.00, 0.03)
	Output B	(0.15, −0.19, 0.1, 0.05, −0.12, −0.13, 0.23, 0.04, 0.09, 0.13)
	Output C	(0.11, −0.19, 0.11, 0.06, −0.16, −0.08, 0.22, 0.12, 0.08, 0.1)
26–37	Output A	(0.27, −0.33, −0.06, 0.1, −0.09, −0.12, 0.17, 0.12, 0.04, −0.03)
	Output B	(0.13, −0.15, 0.14, 0.07, −0.13, −0.11, 0.25, −0.09, 0.1, 0.11)
	Output C	(0.12, −0.14, 0.16, 0.09, −0.12, −0.09, 0.23, 0.14, 0.09, 0.1)
27–38	Output A	(0.32, −0.43, 0.07, 0.01, −0.03, −0.1, 0.24, 0.12, 0.05, −0.04)
	Output B	(0.17, −0.12, 0.2, 0.08, −0.13, −0.13, 0.31, −0.04, 0.07, 0.17)
	Output C	(0.13, −0.17, 0.17, 0.07, −0.17, −0.1, 0.27, 0.18, 0.1, 0.12)
28–39	Output A	(0.33, −0.55, 0.09, 0.08, −0.06, −0.09, 0.26, 0.09, 0.06, 0.09)
	Output B	(0.14, −0.18, 0.08, 0.03, −0.15, −0.15, 0.33, 0.07, 0.09, 0.13)
	Output C	(0.11, −0.14, 0.07, 0.06, −0.14, −0.05, 0.33, 0.2, 0.11, 0.14)
29–40	Output A	(0.35, −0.54, −0.1, 0.05, −0.09, −0.11, 0.3, 0.12, 0.07, 0.03)
	Output B	(0.09, −0.15, 0.09, 0.13, −0.14, −0.22, 0.39, 0.06, 0.12, 0.11)
	Output C	(0.15, −0.19, 0.14, 0.14, −0.13, −0.07, 0.41, 0.22, 0.1, 0.13)
30–42	Output A	(0.29, −0.62, −0.15, 0.07, −0.1, −0.08, 0.28, 0.13, 0.02, 0.05)
	Output B	(0.17, −0.22, 0.17, 0.12, −0.11, −0.23, 0.54, −0.13, 0.16, 0.14)
	Output C	(0.12, −0.18, 0.14, 0.17, −0.09, −0.06, 0.43, 0.24, 0.12, 0.11)
31–43	Output A	(0.34, −0.65, 0.12, 0.05, −0.05, −0.17, 0.42, 0.07, 0.08, 0.09)
	Output B	(0.15, −0.11, 0.18, 0.14, −0.09, −0.28, 0.58, −0.1, 0.11, 0.12)
	Output C	(0.13, −0.14, 0.19, 0.18, −0.07, −0.09, 0.53, 0.28, 0.14, 0.09)
32–44	Output A	(0.37, −0.61, 0.11, 0.07, −0.09, −0.13, 0.38, 0.11, 0.03, 0.07)
	Output B	(0.15, −0.11, 0.18, 0.15, -0.06, −0.34, 0.66, 0.06, 0.08, 0.14)
	Output C	(0.12, −0.12, 0.15, 0.13, −0.03, -0.12, 0.53, 0.34, 0.08, 0.1)
33–45	Output A	(0.43, −0.68, -0.19, 0.05, −0.08, −0.2, 0.46, 0.15, −0.03, −0.01)
	Output B	(0.11, −0.14, 0.17, 0.19, −0.11, −0.38, 0.73, 0.07, 0.09, 0.15)
	Output C	(0.14, -0.11, 0.16, 0.18, −0.09, −0.17, 0.67, 0.37, 0.06, 0.11)
34–46	Output A	(0.48, −0.77, −0.2, 0.02, -0.05, −0.14, 0.49, 0.19, 0.06, 0.03)
	Output B	(0.19, −0.19, 0.14, 0.21, −0.1, −0.43, 0.77, −0.07, 0.04, 0.18)
	Output C	(0.18, −0.15, 0.16, 0.21, −0.1, −0.15, 0.73, 0.38, 0.08, 0.14)
35–47	Output A	(0.39, −0.72, 0.14, 0.07, −0.04, −0.16, 0.56, 0.13, 0.05, 0.04)
	Output B	(0.19, −0.2, 0.15, 0.17, −0.04, −0.41, 0.87, −0.05, 0.03, 0.16)
	Output C	(0.17, −0.19, 0.13, 0.16, −0.06, −0.13, 0.81, 0.44, 0.07, 0.17)

for process. But by implementing the proposed approach in these states, in addition to recognition of any unnatural pattern, the main parameters of each unnatural pattern for each variable are obtained even if these unnatural patterns take place simultaneously, that it is an important key for recognition of root of assignable causes.

We simulate process as follows: assignable causes make a shift for mean x_1, x_2, x_7, (displacement of the mean $+\sigma$, -2σ, $+\sigma$, for x_1, x_2, x_7, respectively) and a trend for x_6, x_7, (slope of trend -0.1σ, $+0.2\sigma$, for x_6, x_7, respectively) and a cycle for x_6, x_8 (magnitude of the cycle$+2\sigma$, $+\sigma$ for x_7, x_8 respectively) at sample no. 3 .

The simulated data were presented to the proposed model in real-time with window size 12, output of the SVM detected an unnatural pattern in 27th and 28th observations. In addition by presenting the simulated data to the tree networks A, B, C, magnitude of shift, slope of trend and magnitude of cycle was recognized for each variable that the results were shown in Table 10. These results are correlated with real situations. These experiments were run twenty times; the best recognition was illustrated in Table 10.

In this example we consider, $4^{10}-1$ unnatural patterns and for each type of unnatural pattern, similar to the case study a Monte Carlo simulation generates 800 input vectors when the χ^2 statistic exceeds the UCL with different magnitudes of deviation from unnatural pattern for each variable. Two-thirds of them were used for training, and the rest of the data were used for testing.

The following notes can be understood from the output of model:

1. In the samples 27 and 28 existence of unnatural patterns were recognized.
2. In the samples 34 and 35 magnitudes of unnatural patterns were determined.
3. As you see in Table 10, there are three increasing trends for outputs of networks A, B, C. in the output of network A that indicates shift mean, this trend is observed for x_1, x_2, x_7, and in the output of network B, this trend is observed for x_6, x_7, and in the output of network C, this trend is observed for x_7, x_8. These trends can conduct us to the valid recognition.

According the results for the simulated example with ten quality characteristics, if an unnatural pattern is caused in the collected data, after about thirty samplings the corresponding network can determine the major parameter for the unnatural pattern.

In order to illustrate rigorous of the proposed model, we consider a problem with simultaneous unnatural patterns. In the simulated example, variable x_7 had three unnatural patterns, shift, trend and cycle, simultaneously; since the model uses different neural networks for each type of unnatural pattern, the major parameter for each of them is determined appropriately, but in this situation the measured standard deviation is high. Therefore, on other advantage of the proposed model is that it can diagnose more than one assignable cause simultaneously

3.3. Discussion

For evaluation the performance of the proposed model correlation between the actual output and the output generated by the trained model can be consider as measure. The correlation between the actual output and the output generated by SVM that presented in Table 2 was revealed that the output of the trained model is very strongly correlated with the corresponding actual output for every quality characteristic. Another measure for evaluation the proposed model is classification rate. The classifying rule in Table 3 enables the output of the corresponding trained network can be classified into one of the seven types of unnatural pattern. The proposed model can classify with 78.1% accuracy and can classify adjacent correctly with up to 99% accuracy.

Some of experiments were conducted to check the optimality of model: (1) it was tried without SVM and only by NNs, type of unnatural patterns and the major parameter for unnatural patterns for each variable would been recognized simultaneously but it was impossible because when number of quality characteristics increase NNs cannot consider generality of the problem. (2) Experimental results indicate that large window size can increase the performance of model. In particular, the power of model at lower magnitudes of shift, tend and cycle pattern increases as the window size increases. However, large window size increases the complexity of NNs. Thus, the window size must be varied according to the requirements of real-world applications. (3) Increasing the size of the training set can improve the performance of model up to a certain level of accuracy at any given value of unnatural pattern magnitude. More training examples of abnormal patterns with small unnatural pattern magnitudes are needed. In contrast, fewer examples of unnatural patterns with large magnitudes are needed in generating training examples of model.

The ARL performance of the proposed hybrid learning-based model with $p=2$ and correlation between quality characteristics $q=0.5$ was compared in Table 11 with several statistics-based approaches Hotelling's χ^2 chart, MCUSUM chart [31], MCUSUM chart (MC1) [32], and two MEWMA charts proposed by Lowry et al. [33]. The ARL performances of the statistics-based approaches depend on the value of the non-centrality parameter ($\chi^2=(X_i-\mu)\Sigma^{-1}(X_i-\mu)$). It means that in a bivariate process for shift to μ_1 is to be detected as quickly as a shift to μ_2 if $\mu_1'\Sigma^{-1}\mu_1=\mu_2'\Sigma^{-1}\mu_2$; or the ARL is a function of the statistical distance or Mahalanobis distance (d). On the other hand, the ARL performance of the proposed model is more strongly related to the Euclidean distance than to the statistical distance [24]. Therefore, the ARL performance of the proposed approach cannot

Table 11
Comparison of ARLs between the proposed model and existing MSPC approaches.

d	Learning-based approach	Statistics-based approach				
		χ^2 (UCL=10.6)	MCUSUM (h=5.5, k=0.50)	MC1 (h=4.75, k =0.50)	MEWMA-1 (h=8.79, r =0.10)	MEWMA-2 (h=8.66, r=0.10)
0	200	200	200	203	200	200
1	5.42	42.00	9.35	9.28	7.76	10.20
1.5	3.43	15.80	5.94	5.23	4.07	6.12
2	2.45	6.90	4.20	3.69	2.59	4.41
2.5	2.06	3.50	3.26	2.91	1.89	3.51
3	1.82	2.20	2.78	2.40	1.50	2.92

be compared directly to those of the statistics-based approaches toward specified the statistical distance. For tackling this problem, the average ARL performance of the learning-based approach was estimated by seven shift combination. Theses seven shift combination settings were chosen within the ranges of $[-3.00, -1.00]$ and $[1.00, 3.00]$ to have the same d but different Euclidean distances. The type I errors (expressed here as in-control ARL) of all the control methods in Table 11 are kept approximately the same. Each ARL value of a shift combination setting for the proposed model was obtained using 2000 simulation runs. As a result, the proposed approach performs better than the statistics-based approaches, especially when the shift magnitude is small. As you saw in the previous examples, the ARL of shift mean is approximately same with trend and cycle, hence by this examination, we confirm the appropriate ARL for cycle and trend.

The paper proposed a hybrid learning-based model which is capable of recognizing concurrent unnatural patterns with satisfied accuracy, in addition determining the major parameter for each unnatural pattern. The proposed model can lead to the finding of multiple assignable causes simultaneously and thus significantly reduce the diagnostic time for out-of-control process. The main contributions of this work are recognizing the type of unnatural pattern and major parameters for shift, trend and cycle and for each variable simultaneously by proposed hybrid learning-based model. Because most previous works consider only one type of unnatural patterns for multivariate process or they consider multiunnatural pattern for univariate process. In addition few works El-Midany et al. [18] that consider recognition problem of multiunnatural pattern for multivariate process, do not obtain any information about magnitude of deviations. This information can help quality participators for rapid recognition of root unnatural patterns and subsequently can reduce defect products.

4. Conclusion

With the widespread use of automated and intelligent production and inspection, various process control tasks, traditionally performed by quality practitioners, have been automated [34,35]. In these cases, process data collected for several quality variables at high frequencies may be correlated. The main problem of traditional approach that use of multivariate χ^2 charts for this situation is that they cannot directly determine which variable or group of variables has caused the out-of-control signal and what is the magnitude of out of control that these are clues for finding the cause of an unnatural process.

This study presents a hybrid learning-based model for on-line analysis of out-of-control signals in multivariate manufacturing processes. It consists of two modules. By using a support vector machine-classifier in the first module, type of unnatural pattern can be recognized. Then, in the second module by using three neural networks, magnitude of mean shift, slope of trend and cycle amplitude for each variable is recognized simultaneously. The performance of the proposed approach was evaluated using a case study and a simulated example. The output generated by trained hybrid model is strongly correlated with the corresponding actual target value for each quality characteristic. Unnatural CCPs provide clues to potential quality problems at an early stage, to eliminate defects before they are produced. By considering the main contributions of this work that are recognizing the type of unnatural pattern and major parameters for shift, trend and cycle and for each variable simultaneously by proposed hybrid model, quality practitioners can recognize potential quality problems as soon as possible and eliminate cause of them.

Future research should address for other multivariate charts and other unnatural patterns such as systematic pattern. In addition an expert system can be integrated with the proposed model for rapid finding and consulting about an unnatural pattern and pattern parameters.

References

[1] D.C. Montgomery, Introduction to statistical Quality Control, 4th ed., Wiley, New York, 2001.
[2] G.A. Pugh, Synthetic neural networks for process control, Computers and Industrial Engineering 17 (1–4) (1989) 24–26.
[3] G.A. Pugh, A comparison of neural networks to SPC charts, Computers and Industrial Engineering 21 (1–4) (1991) 253–255.
[4] H.B. Hwarng, Proper and effective training of a pattern recognizer for cyclic data, IIE Transactions 27 (6) (1995) 746–756.
[5] H.B. Hwarng, C.W. Chong, Detecting process non-randomness through a fast and cumulative learning ART-based pattern recognizer, International Journal of Production Research 33 (7) (1995) 1817–1833.
[6] D.T. Pham, A.B. Chan, Control chart pattern recognition using a new type of self-organizing neural network, Proceedings of the Institution of Mechanical Engineers, Part I 12 (1998) I03597.
[7] D.T. Pham, A.B. Chan, Unsupervised adaptive resonance theory neural networks for control chart pattern recognition, Proceedings of the Institution of Mechanical Engineers, Part B 215 (2001) B04999.
[8] S. Sagiroglu, E. Besdok, M. Erler, Control chart pattern recognition using artificial neural networks, Turkish Journal of Electrical Engineering and Computer Sciences 8 (2) (2000).
[9] C.C. Chiu, M.K. Chen, K.M. Lee, Shifts recognition in correlated process data using a neural network, International Journal of Systems Science 32 (2) (2001) 137–143.
[10] H.B. Hwarng, Detecting process mean shift in the presence of autocorrelation: a neural-network based monitoring scheme, International Journal of Production Research 42 (2004) 573–595.
[11] S.T.A. Niaki, B. Abbasi, Fault diagnosis in multivariate control charts using artificial neural networks, International Quality and Reliability Engineering 21 (2005) 825–840.
[12] R.S. Guh, A hybrid learning-based model for on-line detection and analysis of control chart patterns, Computers & Industrial Engineering 49 (2005) 35–62.
[13] Z. Chen, S. Lu, S. Lam, A hybrid system for SPC concurrent pattern recognition, Advanced Engineering Informatics (21) (2007) 303–310.
[14] L.H. Chen, T.Y. Wang, Artificial neural networks to classify mean shifts from multivariate χ^2 chart signals, Computers and Industrial Engineering 47 (2004) 195–205.
[15] R.S. Guh, Y.R. Shiue, An effective application of decision tree learning for on-line detection of mean shifts in multivariate control charts, Computers & Industrial Engineering 55 (2008) 475–493.
[16] C.-S. Cheng, H.-P. Cheng, Identifying the source of variance shifts in the multivariate process using neural networks and support vector machines, Expert Systems with Applications 35 (2008) 198–206.
[17] J.-b. Yu, L.-f. Xi, A neural network ensemble-based model for on-line monitoring and diagnosis of out-of-control signals in multivariate manufacturing processes, Expert Systems with Applications 36 (2009) 909–921.
[18] T.T. El-Midany, M.A. El-Baz, M.S. Abd-Elwahed, A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks, Expert Systems with Applications 37 (2010) 1035–1042.
[19] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual ACM Workshop on COLT, Pittsburgh, PA, 1992, pp. 144–152.
[20] F. Zorriassatine, J.D.T. Tannock, C.O. Brien, Using novelty detection to identify abnormalities caused by mean shifts in bivariate processes, Computers & Industrial Engineering 44 (2003) 385–408.
[21] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Transaction on Neural Networks 5 (1994) 537–550.
[22] A.E. Smith, X-bar and R control chart interpretation using neural computing, International Journal of Production Research 32 (1994) 309–320.
[23] A. Hassan, M.Shariff Nabi Baksh, A.M. Shaharoun, H. Jamaluddin, Improved SPC chart pattern recognition using statistical features, International Journal of Production Research 41 (7) (2003) 1587–1603.
[24] R.S. Guh, On-line identification and Quantification of mean shifts in bivariate processes using a neural network-based approach, Quality and Reliability Engineering International (2006).
[25] C.S. Cheng, A multi-layer neural network model for detecting changes in the process mean, Computers and Industrial Engineering 28 (1995) 51–61.
[26] V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, MA: MIT press, Cambridge, 2001.
[27] R.S. Guh, Optimizing feedforward neural networks for control chart pattern recognition through genetic algorithms, International Journal of Pattern Recognition and Artificial Intelligence 18 (2) (2004) 75–99.
[28] M.A. Barghash, N.S. Santarisi, Pattern recognition of control charts using artificial neural networks-analyzing the effect of the training parameters, Journal of Intelligent Manufacturing 15 (2004) 635–644.

[29] Y. Zhang, H. Zhou, S. Joe Qin, T. Chai, Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares, IEEE Transactions on Industrial Informatics 1 (2010) 3–12.
[30] Y. Zhang, S. Joe Qin, Improved nonlinear fault detection technique and statistical analysis, AIChE Journal 54 (12) (2008) 3207–3220.
[31] J.J. Pignatiello, G.C. Runger, Comparisons of multivariate CUSUM charts, Journal of Quality Technology 22 (3) (1990) 173–186.
[32] R.B. Crosier, Multivariate generalizations of cumulative sum quality control schemes, Technometrics 30 (3) (1988) 29–1303.
[33] C.A. Lowry, W.H. Woodall, C.W. Champ, S.E. Rigdon, Multivariate exponentially weighted moving average control chart, Technometrics 34 (1) (1992) 46–53.
[34] M. Salehi, R. Tavakkoli-Moghaddam, Application of genetic algorithm to computer-aided process planning in preliminary and detailed planning, Engineering Applications of Artificial Intelligence 22 (2009) 1179–1187.
[35] M. Salehi, A. Bahreininejad, Optimization process planning using hybrid genetic algorithm and intelligent search for job shop machining, Journal of Intelligent Manufacturing, on line publication, doi:10.1007/s10845-010-0382-7.

Mojtaba Salehi is currently a Ph. D. candidate, Department of Industrial Engineering, Tarbiat Modarres University. His research areas of interest include stochastic modeling, applied multivariate analysis, soft computing, data mining and machine learning.

Ardeshir Bahreininejad obtained his Ph.D. in Computational Mechanics from Hatteriot-W University, UK in 1996. From 2000 to 2002 he pursued a postdoctoral research fellowship at Georgia Institute of technology, USA working in the area of Manufacturing Information Systems. He is currently working as an Associate Professor at University of Malaya. His research interests include: computational mechanics, artificial intelligence, neural computing, optimization, parallel, cluster and grid computing.

Isa Nakhai obtained his Ph.D. in industrial. He is currently working as an Associate Professor Tarbiat Modarres University. His research interests include: optimization, operation research and information technology.