Pattern Anal Applic (2015) 18:75–86
DOI 10.1007/s10044-012-0312-8

THEORETICAL ADVANCES

Recognition of control chart patterns using a neural network-based pattern recognizer with features extracted from correlation analysis

Chuen-Sheng Cheng · Kuo-Ko Huang · Pei-Wen Chen

Received: 3 March 2012 / Accepted: 14 November 2012 / Published online: 25 November 2012

Abstract Control chart pattern analysis is a crucial task in statistical process control. There are various types of nonrandom patterns that may appear on the control chart indicating that the process is out of control. The presence of nonrandom patterns manifests that a process is affected by assignable causes, and corrective actions should be taken. From a process control point of view, identification of nonrandom patterns can provide clues to the set of possible causes that must be searched; hence, the troubleshooting time could be reduced in length. In this paper, we discuss two implementation modes of control chart pattern recognition and introduce a new research issue concerning pattern displacement problem in the process of control chart analysis. This paper presents a neural network-based pattern recognizer with selected features as inputs. We propose a novel application of statistical correlation analysis for feature extraction purposes. Unlike previous studies, the proposed features are developed by taking the pattern displacement into account. The superior performance of the feature-based recognizer over the raw data-based one is demonstrated using synthetic pattern data.

Keywords Nonrandom patterns · Features · Pattern recognition · Correlation analysis · Neural networks

C.-S. Cheng (✉) · K.-K. Huang · P.-W. Chen
Department of Industrial Engineering and Management,
Yuan-Ze University, 135 Yuan-Tung Road,
Chung-Li 320, Taoyuan, Taiwan R.O.C
e-mail: ieccheng@saturn.yzu.edu.tw

1 Introduction

Since today's industrial firms are moving towards agile manufacturing, rapid and economic on-line statistical process control (SPC) solutions are in great demand. Statistical process control concepts and methods have been successfully implemented in manufacturing industries for several decades. As one of the primary SPC tools, control chart plays a very important role in maintaining an acceptable level of process variability. The control charts may signal an out-of-control condition when either one or more points fall outside the control limits or plotted points show some nonrandom patterns of behavior. Any nonrandom patterns shown on control charts imply possible assignable causes that may deteriorate the process performance. In many situations, the pattern of the plotted points will provide useful process diagnostic information which can be used to make process changes or adjustments that reduce variation. Hence, timely detecting and recognizing control chart patterns (hereafter referred to as CCPs) are very important in the implementation of SPC. Under the pattern recognition approach, numerous researchers [1–3] have defined several types of out-of-control nonrandom patterns with a specific set of possible causes. Various supplementary rules known as runs rules or runs tests have been suggested to the practitioners to detect nonrandom patterns. The Western Electric Handbook [3] first proposed a set of zone rules for identifying the out-of-control process by observing systematic patterns on the control charts. Nelson [1, 2] further developed a set of runs rules for nonrandom patterns. For purposes of control chart pattern recognition, the application of runs rules is not without drawbacks. As already pointed out by Cheng [4], there is no one-to-one mapping relation between a runs rule and a nonrandom pattern. There might be several patterns associated with a particular rule. For this reason, interpretation of the process data often relies on the skill and experience of the quality control personnel to identify the existence of a nonrandom pattern in the process.

In the increasingly common situation of automated production, the intelligent pattern recognition of control charts has been an important and widely studied topic. An efficient control chart pattern recognition system can ensure consistent and unbiased interpretation of CCPs leading to fewer false alarms and better implementation of control charts. Several techniques have been proposed to control chart pattern recognition. These include statistical [5, 6], rule-based system [7, 8], artificial neural network (ANN) techniques [4, 9–13], and support vector machines [14]. Among different methods, neural networks have been popularly used for control chart pattern recognition over the past few years. Many researchers have obtained promising results on control chart pattern recognition applying different neural networks methods. For a comprehensive review of the applications of neural networks for process monitoring, the interested reader is referred to Zorriassatine and Tannock [15], Barghash and Santarisi [16], and Masood and Hassan [17]. In the following, we review some important issues related to this work, including input vector and implementation modes of pattern recognition.

Most of the previous works in the literature used raw (unprocessed) process data as the input vector for control chart pattern recognition. Recently, a few researchers [18–22] have attempted to recognize control chart patterns using extracted features from raw process data as the input vector. The reported advantages include improvement of recognition performance, robustness to the amount of noise embedded in monitoring data, and saving of computation time. In the existing literature, features could be obtained in various forms, including statistical features [23], shape features [19, 21], and Wavelet features [18].

For purposes of control chart pattern recognition, it is not enough to look at each observation in time sequentially. Rather, one has to deal with sliding analysis (monitoring) window. The window size, m, is usually determined with the consideration of computational effort and recognition performance. Typical values of m range from 8 to 60 samples, with $m = 16$ and 32 being popular choices [15]. The approaches in the literature to the pattern recognition for control charts can be divided into two main categories. One is the direct continuous recognition [4, 10, 11]. Another is referred to by Hassan and Baksh [23] as "recognition only when necessary" approach. In the continuous recognition mode, the recognition activities are performed in a continuous manner for all data streams as they appear in the sliding analysis window. One significant problem of the continuous monitoring mode is unnecessary recognition of stable process. Besides, premature attempts of recognition may increase the possibility of incorrect diagnosis because of insufficient information. This point is evident from an example shown in Fig. 1 where a cycle is misrecognized as a trend or a shift.

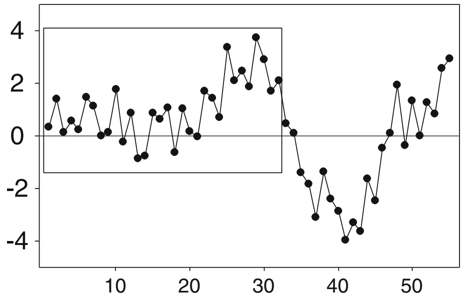


Fig. 1 An example of premature attempt of recognition

A stability test procedure called abnormality detector is required in "recognition only when necessary" approach to determine the necessity to recognize prior to triggering the recognition process. This implementation mode is intended to avoid unnecessary recognition of stable processes and allow timely recognition of unstable processes. Under this approach, all the process data streams will be tested for its stability prior to the recognition process. If a data stream is considered as coming from a stable process, then the process will continue without any recognition. With this configuration, attempts to classify streams of process data are allowed only when they are identified as possibly coming from an unstable process. Hassan and Baksh [23] suggested using runs rules and CUSUM (short for cumulative sum) as a procedure to determine process stability. Wang et al. [8] suggested using correlation analysis to determine if abnormality exists prior to a recognition procedure.

One major assumption made by most previous studies is that the pattern has a fixed starting point in the development stage (training). However, this assumption is not always realistic. Using an analysis window approach, some basic pattern features might change over time due to the time-varying characteristics of nonrandom patterns. The shape of the pattern collected in an analysis window may or may not be the same as a specific pattern considered in the development stage. The variations may be due to the starting point of a pattern in the analysis window (e.g., trend or shift) or the phase difference of a pattern (e.g., cyclic or systematic pattern). This phenomenon is referred to by the authors as "pattern displacement" in this paper. Figure 2 provides several examples to support the above argument. The examples (from left to right) include: (1) difference in starting point of the trend; (2) difference in shift position in terms of sampling time; (3) the phase difference of cycles; and (4) difference in starting location

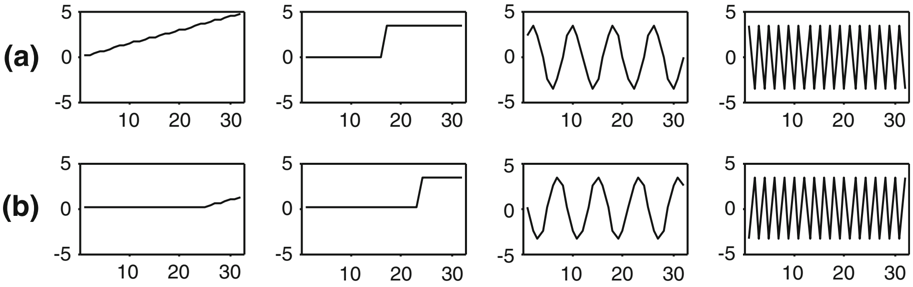


Fig. 2 Illustrations of pattern displacement using noise-free data

of a systematic pattern in terms of above or below average. These examples show that the shape of a particular pattern appeared in the analysis window (the subplots in the lower row) might be different from that of a prototype learned in the development stage (the subplots in the upper row). The level of pattern displacement may be influenced by the disturbance level or the sensitivity of a stability test. For instance, a large shift may result in a pattern with shifting position near the end of analysis window. A sensitive stability test can detect the process shift earlier, thus produces a shift pattern with fewer number of shifted observations.

Al-Assaf [18] conducted several experiments and found that the performance of the pattern recognizer is highly dependent on the location and duration of a pattern appearing in the analysis window. The variations of nonrandom patterns considered in previous research are too simplified. Hassan et al. [20] considered the case that a shift only appears in the middle of an analysis window. In Wang et al. [8] work, the change points of shifts are randomly selected around half of the window. In view of this, it is desirable to construct a set of pattern features that are robust against the pattern displacement. Considering all possible variants of a pattern during the pattern recognizer's development stage is one way to approach the displacement problem. However, this method may complicate the training of the pattern recognizer considerably and consequently degrade the recognition performance. An alternative way is to pre-compute a set of features and use these features as the input vector of a pattern recognizer. However, special care should be given to the pattern displacement problem when constructing the relevant features. The plots in the leftmost column of Fig. 2 are used to illustrate the issues introduced above. The plots in this figure are two-data series which differ in the starting point of a trend. The feature "a_{pm}" proposed by Pham and Wani [21] is used for differentiate a trend from other patterns. Based on 1,000 simulation replicates, the average value of feature "a_{pm}" is (a) 40.21 for data series and (b) 20.89 for data series. The distributions of the values of feature "a_{pm}" are shown in Fig. 3. This example shows that the time-varying nature of pattern displacement may dilute the usefulness of a distinct feature. Similar observations can be found in the feature "RVPEPE" introduced by Gauri and Chakraborty [19]. Figure 4 shows the distributions of the values of this feature for two different data series. We notice that the average values are quite different, (a) 3.49 for data series and (b) 1.26 for series.

The objective of this paper is to provide an appropriate solution to address the aforementioned problems. The proposed feature-based approach consists of two main steps. First, a novel application of correlation analysis is used to extract a set of relevant CCP features that are less affected by the displacement of nonrandom patterns. Then, we develop a neural network-based CCP recognizer with extracted features as the input vector. Its performance is compared with that of a raw data-based recognizer. Although the proposed methodology can be applied to either implementation mode of control chart pattern recognition, the discussions and performance reported are in the context of "recognition only when necessary" mode. Since the abnormality detector is not a major concern of the current study, some pattern parameters are manipulated to simulate the diversity of patterns that may be identified by a detector. The various pattern parameters will be discussed later in this paper.

The remaining parts of the paper are arranged as follows. After describing a set of equations useful for generating nonrandom patterns in Sect. 2, we introduce the

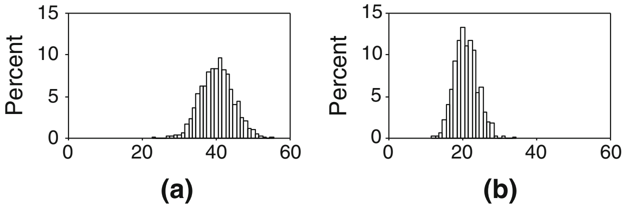


Fig. 3 The distributions of the values of feature a_{pm}

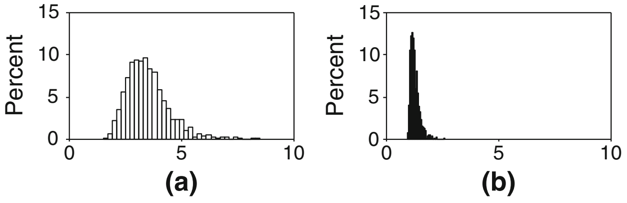


Fig. 4 The distributions of the values of feature RVPEPE

proposed pattern features computed from correlation analysis in Sect. 3. Subsequently, a description of the neural network-based recognizer is presented in Sect. 4. Section 5 provides the experimental results and discussions on the comparison between the raw data-based recognizer and feature-based recognizer. Finally, we conclude the paper in Sect. 6.

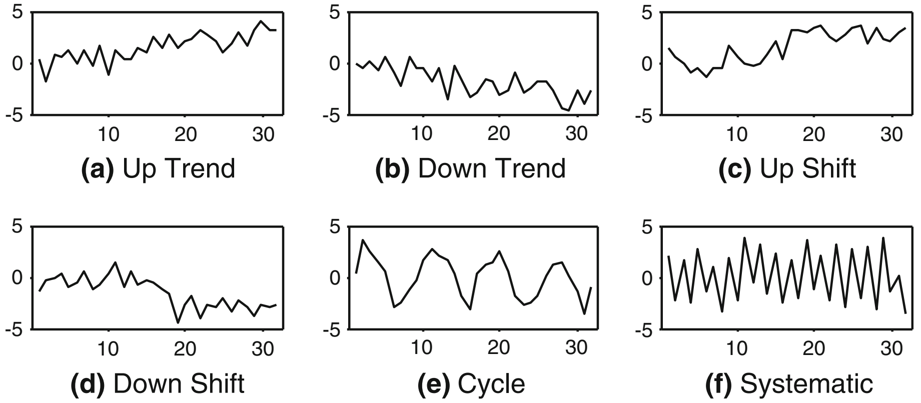
2 Patterns generation

In the research described here, six commonly found nonrandom patterns on control charts were considered, namely, upward trend (UT), downward trend (DT), upward shift (US), downward shift (DS), cyclic pattern (CYC), and systematic pattern (SYS). Figure 5 gives an example for each pattern class. Ideally, sample patterns should be developed from a real process. In practical situations, sufficient training samples of nonrandom patterns may not be readily available. One common approach adopted by previous studies was to generate training samples based on predefined mathematical model and Monte Carlo simulation. The previous research work usually used the mathematical models proposed in Swift [24] and Cheng [7]. The generator is expressed in a mathematical formula and includes an in-control mean, a random noise, and a disturbance. The pattern generator can be expressed as:

$$x_t = \mu + n_t + d_t \tag{1}$$

where x_t is the observation collected at time t, μ represents a known in-control process mean (can be estimated from historical data), n_t is a random noise, and d_t is a special disturbance due to assignable cause. By manipulating the value of d_t, a pattern can then be simulated. For simplicity and with no loss of generality, we assume $\mu = 0$ and n_t has a $N(0,1)$ distribution as in most of the previous studies. The level of disturbance is usually expressed in terms of process standard deviation, σ. Note that a normal pattern (NORM), indicating the process is in the normal operating condition, can be expressed as $x_t = \mu + n_t$. In this paper, a set of mathematical expressions are proposed to address the pattern displacement issue. The mathematical models are slightly different from those of Swift [24] and Cheng [7], with the intention to simulate the diversity of patterns that may be identified by an abnormality detector. The parameters along with the equations used for simulating the CCPs are given in Table 1.

A set of training samples is required in the development of a pattern recognizer. In this research, the training samples for each pattern were generated according to the parameters described in Table 2. There were a total of 7,000 samples (1,000 for each pattern class, including normal pattern) in the training dataset. This dataset is marked as DS_1. The value of $t*$ for systematic patterns was set to 0.



(a) Up Trend **(b)** Down Trend **(c)** Up Shift **(d)** Down Shift **(e)** Cycle **(f)** Systematic

Fig. 5 Various types of control chart patterns considered in this research

Table 1 The formulae and parameters used for CCPs

Pattern	Parameters	Formula for d_t	Remarks
Trend	θ is the trend slope	$d_t = \theta(t - t_0)$, if $t \geq t_0$; else $d_t = 0$	t_0 determines the starting point of a trend; the sign of θ determines the direction of trends
Shift	δ is the magnitude of the shift	$d_t = \delta$, if $t \geq t_0$; else $d_t = 0$	t_0 determines the point of the shifting; the sign of δ determines the direction of shifts
Cycle	κ is the amplitude of the cyclic patterns; Ω is the period	$d_t = \kappa \cdot \sin(2\pi(t + t_0)/\Omega)$	t_0 determines the phase difference of a cyclic pattern
Systematic	Δ is the offset from the process mean	$d_t = (-1)^{(t-t_0-t*)}\Delta$, if $t \geq t_0$; else $d_t = 0$	t_0 determines the starting point of a systematic pattern; $t* \in \{0,1\}$controls the location of the first observation

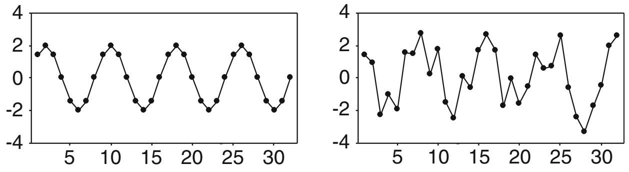
Table 2 Parameters for simulating training samples

Control chart pattern	Parameter values	t_0
Up trend	θ: 0.1 to 0.15	0
Down trend	θ: -0.15 to -0.1	0
Up shift	δ: 1 to 3	13, 17
Down shift	δ: -3 to -1	13, 17
Cycle	κ: 1 to 3, $\Omega = 8$	0
Systematic	Δ: 1 to 3	1

We have to mention here that a number of previous studies assumed that the magnitude of random noise associated with nonrandom patterns is less than that of a normal pattern (e.g., [20, 25]). The implication is that they considered the cases when signal to noise ratios of patterns are very high. The reported results may be too optimistic and misleading. It can be shown that the pattern recognizer performs worse if more noise is added to the data series. In the generation of nonrandom patterns, some authors [20, 25–28] maintain that the observations of nonrandom patterns should be kept within control limits. The remark could be ignored, however, due to the reasons that the existence of n_t; the control chart pattern recognition is for analyzing long-term behavior of process data.

3 Feature extraction

This section describes the feature extraction based on the concept of correlation analysis. There have been some applications of correlation analysis in control chart pattern recognition [26, 28]. Basically, the approach involves computing correlation coefficient between the input data and various reference patterns. The determination of reference patterns is an influential design issue in correlation coefficient-based pattern recognizer. It is easy to show that the reference pattern determined in Yang and Yang [28] is in essence a nearly noise-free (or highly smoothed) version of the nonrandom pattern. The recognition method introduced in Yang and Yang [28] will sometimes yield an incorrect classification due to pattern displacement. As an example, consider data sequences shown in Fig. 6. A low correlation coefficient (-0.074) indicates that the data sequence in the analysis window is different from the reference pattern even though they have the same pattern characteristics. The procedure proposed in Wang et al. [8] suffered the same problem. Certainly, if the data sequence in analysis window resembles the reference pattern, a high correlation coefficient could be obtained after several window movements.



(a) a reference pattern **(b)** data in monitoring window

Fig. 6 Incorrect recognition due to pattern displacement

In order to address the aforementioned problems, this research proposes a feature extraction approach without the need for defining the reference patterns. The details of feature extraction are described in the following. Let $\mathbf{X} = \{x_1, x_2, \ldots, x_m\}$ denote the most recent m observations when an abnormality detector determines that the collected observations are coming from an unstable process, where the subscript i in x_i indicates simply the order in which the observations were recorded. To implement the proposed approach, the window size m has to be determined first. For the purpose of avoiding computational complexity and excessive computation time, the parameter m is determined empirically as 32. Let $\mathbf{X} = \{\mathbf{Y}, \mathbf{Z}\}$, where $\mathbf{Y} = \{y_1, y_2, \ldots, y_{16}\} = \{x_1, x_2, \ldots, x_{16}\}$ and $\mathbf{Z} = \{z_1, z_2, \ldots, z_{16}\} = \{x_{17}, x_{18}, \ldots, x_{32}\}$. Some variants of data vectors **X** and **Z** are defined as follows.

W a circle shift of **X**, denoted as $\mathbf{W} = \{x_{32}, x_1, x_2, \ldots, x_{31}\}$.$\mathbf{Z}^a$ a set of order statistics obtained by sorting the $\{z_i\}$ in ascending order. The resultant is denoted as $\mathbf{Z}^a = \{z_{(1)}, z_{(2)}, \ldots, z_{(16)}\}$.$\mathbf{Z}^d$ a set of order statistics obtained by sorting the $\{z_i\}$ in descending order. The resultant is denoted as $\mathbf{Z}^d = \{z_{(16)}, z_{(15)}, \ldots, z_{(1)}\}$.

T and **U** are two additional vectors whose elements are time indices, defined as $\mathbf{T} = \{1, 2, \ldots, 32\}$ and $\mathbf{U} = \{17, 18, \ldots, 32\}$. Let $\rho(\bullet, \bullet)$ denote the product–moment correlation coefficient or Pearson's correlation coefficient of a set of paired observations, measuring the strength of a relationship between two variables. The mathematical expressions for extracted features are given below:

$r_1 = \rho(\mathbf{Z}^a, \mathbf{Z})$, $r_2 = \rho(\mathbf{Z}^d, \mathbf{Z})$, $r_3 = \rho(\mathbf{Y}, \mathbf{Z})$, $r_4 = \rho(\mathbf{X}, \mathbf{W})$, $r_5 = \rho(\mathbf{X}, \mathbf{T})$, $r_6 = \rho(\mathbf{Z}, \mathbf{U})$,

Regarding the application of correlation analysis to extracting features, one important point to be mentioned is that a reference pattern is not required in our proposed method. This is a uniqueness compared to other studies in the literature. The proposed method computes the correlation coefficient between the input vector and its variants with the intention to retain the shape characteristics of a specific nonrandom pattern. One minor restriction imposed on the proposed method is that the size of analysis window is divisible by 2. Two additional features are the commonly used summary statistics: mean and standard deviation. The eight features described above establish the input vector for the feature-based recognizer.

Before describing the discriminating capability of each feature, it is useful to look at the distributions of the proposed features. An ideal set of features should maximize the inter-class variability while minimizing the intra-class variations. Figure 7 shows the distributional properties of the selected features. The subplots in each column, arranged from left to right, represent various $r_i^{\prime}s$, calculated mean, and standard deviation, respectively. The subplots

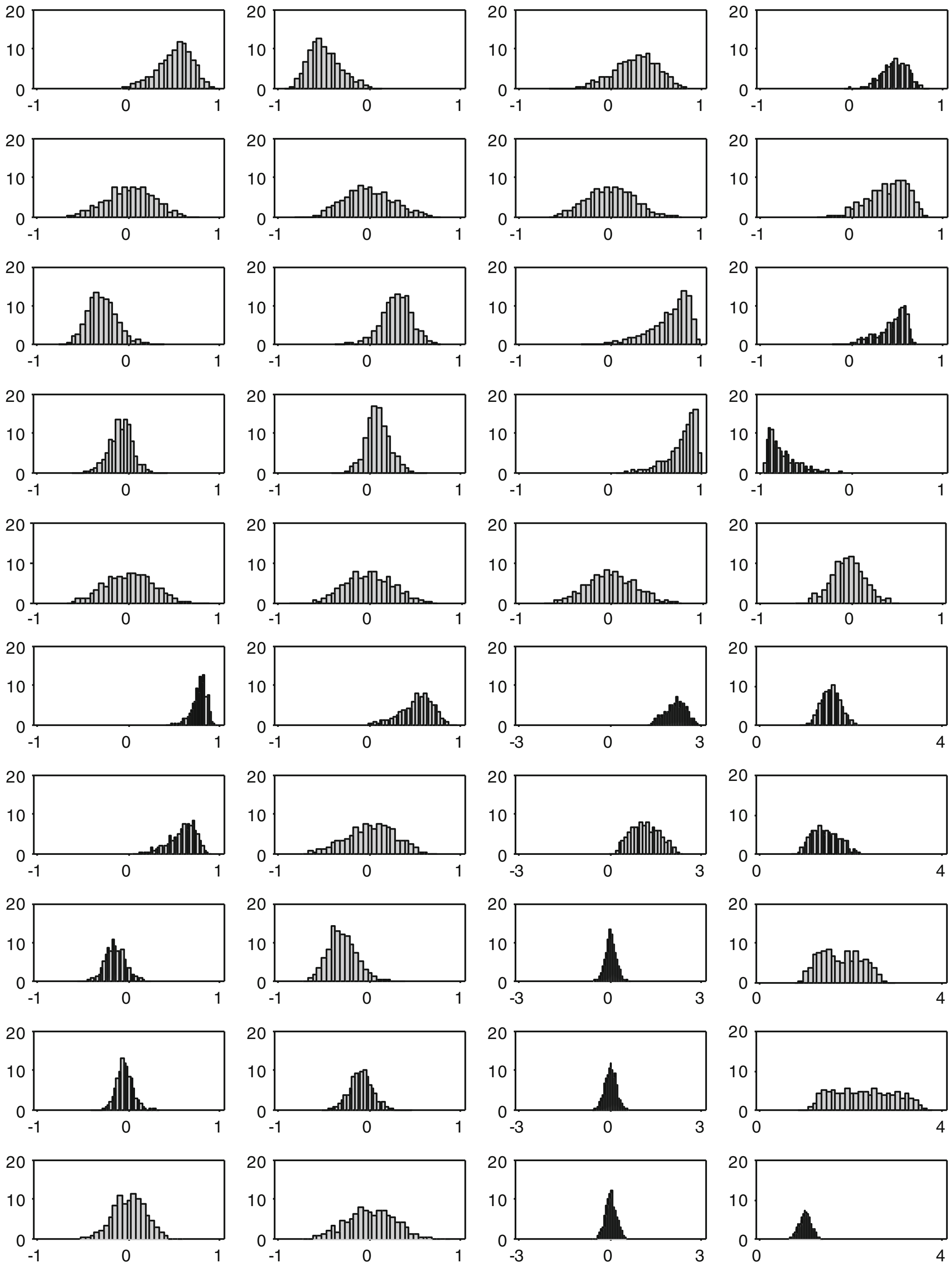


Fig. 7 Distributional properties of the selected features

from a specific pattern class are arranged in the same row. The rows in the figure, from top to bottom, represent trend, shift, cycle, systematic pattern, and normal pattern, respectively. Note that only the upward versions of trend and shift patterns are given in order to conserve space. The vertical axis of each plot in Fig. 7 is a percent scale. The following is a detailed explanation of the aforementioned features.

First, it is not surprisingly to see that the $r_i's$ of normal pattern are all cluster around zero. This is due to the fact that normal data vector and its variants are random in nature. We can see from Fig. 7 that the magnitude of r_1 for trend and cyclic patterns is different from zero. For all other patterns, i.e., shift, systematic, and normal, the value of r_1 is distributed around zero. The magnitude of r_1 can differentiate trend and cyclic patterns from other patterns. The sign (±) of r_1 can assist in distinguishing between upward and downward trends. From Fig. 7, we observe that the magnitudes of r_2 for shift, systematic, and normal patterns will vary in the same fashion as r_1. The sign of r_2 for trend and cyclic patterns is opposite to that of r_1. The magnitude of r_3 will be highest for cyclic and systematic patterns, intermediate for trend patterns, and around zero for shift and normal patterns. The values of r_3 for cyclic and systematic patterns show a strong positive correlation. Therefore, we conclude that the magnitude of r_3 is a distinguishing feature to differentiate cyclic and systematic patterns from other patterns.

It is observed that the magnitude of r_4 will be highest for systematic patterns, intermediate for trend, shift, and cyclic patterns, and around zero for normal patterns. The values of r_4 for systematic patterns show a strong negative correlation. The systematic patterns can be differentiated more efficiently from other patterns by the magnitude of r_4. The magnitude of r_5 will be highest for trend patterns, intermediate for shift patterns, and around zero for other patterns. We conclude that the magnitude of r_5 is an important distinguishing feature used to differentiate trend and shift patterns from other patterns. The sign of r_5 can help distinguish between upward and downward trends and upward and downward shifts. The magnitude of r_6 will be highest for trend patterns, intermediate for cyclic patterns, and around zero for other patterns. Thus, this feature differentiates trend patterns from other patterns. The sign of r_6 is helpful in distinguishing between upward and downward trends.

The plots in Fig. 7 show that the magnitude of the calculated mean will be highest for trend patterns, intermediate for shift patterns, and around zero for all other patterns. The magnitude of the calculated mean can be used to differentiate trend and shift patterns from other patterns. In addition, the sign of the calculated mean can distinguish between increasing and decreasing trends and upward and downward shifts. The value of the standard deviation will be highest for systematic patterns, followed by cyclic patterns, intermediate for trend and shift patterns, and least for normal patterns. The magnitude of standard deviation is a distinguishing feature useful to differentiate normal patterns from other patterns.

Determining a pattern recognition algorithm is the second step of the feature-based recognizer. The following section is devoted to the description of designing a neural network-based control chart pattern recognizer.

4 Neural network-based pattern recognizer

Recently, artificial neural networks have received increasing attention in the area of control chart pattern recognition, due to their noteworthy generalization performance. The most frequently used neural network is feed-forward multi-layer perceptrons (MLPs). The typical MLP-type neural network is composed of an input layer, one or more hidden layers, and an output layer of nodes. The MLP neural networks have been successfully applied to diverse fields. The details of neural networks are well described in standard text books (e.g., [29]). This section provides a brief summary of MLP networks.

Two different types of MLP-based recognizers are constructed in this paper. For feature-based recognizer, the input vector is composed of the extracted features described in Sect. 3. On the other hand, the input vectors of raw data-based recognizer are data sequences each comprising 32 data points. The parameters selection is critical to the success of the neural network-based recognizer. The determination of the number of nodes in each layer is described as follows. In the proposed methodology, the size of the input vector determines the number of input nodes (n_{I}). Hence, the values of n_{I} are set as 32 and 8 for the raw data-based and feature-based recognizers, respectively. There are no general guidelines for determining the optimal number of nodes (n_{H}) required in the hidden layer. The number of hidden layer nodes was chosen through experimentation by varying the number of hidden nodes. The appropriate value of the hidden layer nodes should be chosen to limit the computational requirements and achieve a reasonable recognition performance. The determination of the values of n_{H} will be discussed later in this section. The number of nodes in the output layer (n_{O}) of both the two types of networks is set according to the number of pattern classes, i.e., seven, each representing a particular pattern class.

There are various algorithms available to train the MLP in a supervised manner. Gradient descent (GD) is a first order optimization algorithm that attempts to move incrementally to successively lower points in search space in order to locate a minimum. The basic gradient descent method adjusts the weights in the steepest descent direction. This is the direction in which the performance function is decreasing most rapidly. Although the function decreases most quickly along the negative of the gradient, this approach does not always generate the fastest convergence. Many advanced versions of training algorithms have been proposed. Conjugate gradient (CG) is a fast training algorithm for MLP that proceeds by a series of line searches through error space. Succeeding search directions are selected to be conjugate (non-interfering). A search along conjugate directions usually produces faster convergence than steepest descent directions. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (or Quasi-Newton) is a powerful second order training algorithm with very fast convergence but high memory requirements.

In the following, we describe the development of the neural network-based pattern recognizer. In this study, a BFGS algorithm, built in the MATLAB, was selected for training the MLP neural networks. The BFGS training algorithm was adopted since it provides good performance and more consistent results. The input data were scaled to the interval $[-1,1]$ using a simple linear transformation due to the reason that the data points include both positive and negative values. In this research, the transfer function used was a hyperbolic tangent for hidden nodes and a sigmoid for the output nodes. The maximum number of the iterations to train was 200. Some early stopping criteria were set to achieve better generalization performance. The performance was minimized to the goal 10^{-7}, the performance gradient was 10^{-6}. Other parameters were adopted as default in MATLAB.

The trained neural network stores the implicit decision rules through a set of interconnection weights used to recognize the nonrandom patterns. Given an input vector, the neural network produces an output vector. In this application, the value of each output layer node is a real-valued variable (between 0 and 1). The neural network selects the pattern class corresponding to the output node having the maximum value.

We now describe the determination of the number of n_{H}. We investigated the recognition rate (expressed as a percentage) of the MLP neural networks by varying the number of hidden nodes. The recognition rate is one of the major performance measurements of a recognizer. It is defined as the ratio of correctly recognized patterns to the whole set of patterns. Since the trained MLP networks produced different results according to the initial learning condition, we averaged 10-trial results of the average performance for each size selection. The simulation results are reported in Figs. 8 and 9. It can be observed the value of hidden layer nodes has a significant impact on the performance measurement. The performance of the MLP neural network improves as the number of hidden nodes increases. However, the results also indicate that the number of hidden layer nodes has a diminishing return in recognition accuracy. From Figs. 8 and 9, it is inferred that best performances for raw data-based and feature-based recognizers are 21 and 8, respectively. The final neural network structure for both types of recognizers is shown in Fig. 10. The values in parenthesis indicate the number of nodes for the feature-based recognizer.

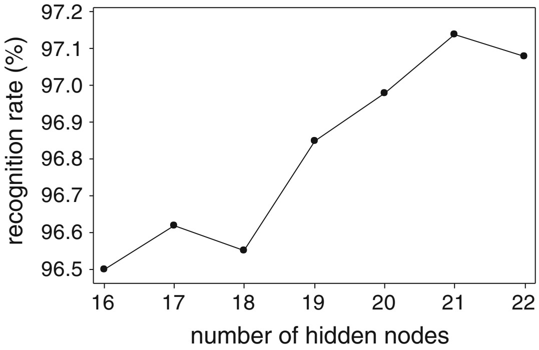


Fig. 8 The performance of the raw data-based recognizer under different hidden nodes

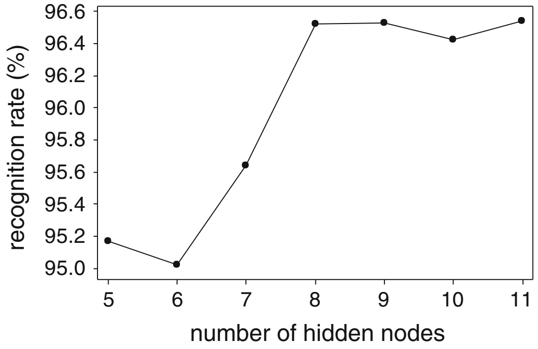


Fig. 9 The performance of the feature-based recognizer under different hidden nodes

5 Computational results and discussion

This section describes the results and comparisons of performance between the feature-based and raw data-based recognizers. The performance of the recognizer using features proposed by Wang et al. [8] was used as a benchmark. The set of features proposed by Wang et al. [8] were chosen simply because they addressed the same patterns considered in this research. The benchmark pattern recognizer was optimized in the manner described above.

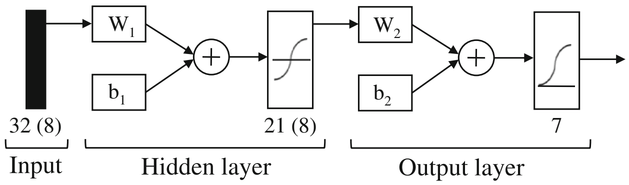


Fig. 10 The structure of MLP neural network

Table 3 Parameters for simulating testing samples

Control chart pattern	Parameter values	t_0
Up trend	θ: 0.1 to 0.15	2 to 6
Down trend	θ: −0.15 to −0.1	2 to 6
Up shift	δ: 1 to 3	8, 9, 15, 18, 19
Down shift	δ: −3 to −1	8, 9, 15, 18, 19
Cycle	κ: 1 to 3, $\Omega = 8$	1 to 4
Systematic	Δ: 1 to 3	5

With a series of experiments, we show that the recognizer with the proposed features as input vectors substantially improve the recognition performance. In the first experiment (experiment I), the parameters used for simulating testing patterns are given in Table 3. A total of 35,000 samples (5,000 × 7), termed as DS_2, were generated for the testing dataset. The value of $t*$ for systematic patterns was randomly set to 0 or 1. It is evident that the testing dataset is quite different from that of the training dataset. The purpose is to test the generalization capability of the feature-based pattern recognizer.

The overall recognition results for the raw data-based and feature-based pattern recognizers are displayed using confusion matrices. Entries (in boldface) along the diagonal indicate correct recognitions, whereas off-diagonal entries would indicate the misrecognitions. Considering the effects of training and testing data as well as the initial learning condition of neural networks, we performed ten independent simulation runs to obtain more reliable conclusions. Entries of the confusion matrices presented below are averages over ten independent runs. Tables 4 and 5 give the confusion matrices showing the recognition results of the raw data-based and feature-based pattern recognizers, respectively. The results for recognition of normal patterns in Tables 4 (95.9 %) and 5 (94.8 %) suggest that both types of recognizers perform equally well in terms of type I error. Valuable improvement obtained by the proposed features can be noticed by comparing Tables 4 and 5. The feature-based recognizer gives an overall correct recognition rate of 88.47 %; on the other hand, raw data-based recognizer gives a poor value of 67.84 %. Table 4 shows that cycles and systematic patterns are the hardest to be recognized correctly by the raw data-based recognizer. This happens because the shape characteristics of nonrandom patterns observed from raw data are different between training dataset and testing dataset. Another explanation is that the raw data-based recognizer treats the value of each time point as one absolute attribute. The raw data-based recognizer identifies a pattern by matching the attribute at each position in the monitoring window. As can be seen from these tables, the performance of the feature-based pattern recognizer is affected very little by the displacement of the nonrandom pattern, whereas the performance of the raw data-based recognizer worsens as the displacement varies. The results also imply that the proposed features can retain the pattern characteristics even when pattern displacement occurs.

In the second experiment (experiment II), we swapped training dataset and testing dataset. That is, the recognizers were trained with DS_2 and tested with DS_1. In this experimental setting, the training samples are more diversified than testing samples. The determination of the value of n_H is exactly similar to the first experiment. Based on our numerical experiments, the optimal values for n_H were set as 8 for feature-based recognizer and 22 for raw data-based recognizer. The recognition results are summarized in Tables 6 and 7. Examining the results in Tables 6 and 7 demonstrates the superiority of the feature-based recognizer. The feature-based recognizer achieves a 93.8 % recognition rate, while raw data-based recognizer yields a poorer value of 87.28 %. It is noted here that both recognizers have higher recognition rates than in the first experiment. The increase of recognition rate can be

Table 4 Recognition results for raw data-based recognizer (I)

Actual	Predicted pattern class						
pattern class	UT	DT	US	DS	CYC	SYS	NORM
UT	**80.0**	0.0	20.0	0.0	0.0	0.0	0.0
DT	0.0	**80.9**	0.0	19.1	0.0	0.0	0.0
US	30.2	0.0	**67.4**	0.0	0.2	0.0	2.2
DS	0.0	30.0	0.0	**67.8**	0.1	0.0	2.2
CYC	0.0	0.0	1.1	2.1	**34.4**	2.0	60.4
SYS	0.1	0.0	11.3	0.5	23.5	**48.5**	16.2
NORM	0.0	0.0	1.3	1.0	1.4	0.3	**95.9**

Table 5 Recognition results for feature-based recognizer (I)

Actual	Predicted pattern class						
pattern class	UT	DT	US	DS	CYC	SYS	NORM
UT	**79.4**	0.0	20.6	0.0	0.0	0.0	0.0
DT	0.0	**81.4**	0.0	18.6	0.0	0.0	0.0
US	11.3	0.0	**87.0**	0.0	0.1	0.0	1.6
DS	0.0	12.0	0.0	**86.3**	0.4	0.0	1.3
CYC	0.0	0.0	0.2	0.3	**93.2**	0.1	6.2
SYS	0.0	0.0	0.1	0.1	0.2	**97.2**	2.4
NORM	0.0	0.0	1.3	0.9	2.5	0.5	**94.8**

Table 6 Recognition results for raw data-based recognizer (II)

Actual	Predicted pattern class						
pattern class	UT	DT	US	DS	CYC	SYS	NORM
UT	**98.5**	0.0	1.5	0.0	0.0	0.0	0.0
DT	0.0	**98.0**	0.0	2.0	0.0	0.0	0.0
US	23.2	0.0	**75.7**	0.0	0.0	0.1	1.1
DS	0.0	21.6	0.0	**77.1**	0.0	0.1	1.2
CYC	0.0	1.1	0.2	6.1	**67.8**	1.5	23.3
SYS	0.0	1.4	0.1	0.0	0.1	**97.5**	1.0
NORM	0.0	0.0	0.9	0.8	1.3	0.5	**96.5**

Table 7 Recognition results for feature-based recognizer (II)

Actual	Predicted pattern class						
pattern class	UT	DT	US	DS	CYC	SYS	NORM
UT	**96.8**	0.0	3.2	0.0	0.0	0.0	0.0
DT	0.0	**96.4**	0.0	3.6	0.0	0.0	0.0
US	10.8	0.0	**87.7**	0.0	0.2	0.0	1.3
DS	0.0	11.1	0.0	**87.5**	0.3	0.1	1.0
CYC	0.0	0.0	0.0	0.5	**93.8**	0.0	5.7
SYS	0.0	0.0	0.0	0.1	0.2	**98.9**	0.8
NORM	0.0	0.0	1.1	1.0	1.5	0.9	**95.5**

attributed to the fact that pattern recognizers learned more diversified samples. Also note that the performance of the feature-based recognizer can still be improved by the identification of new features that will be more useful in discriminating trend from shift patterns.

In the final experiment (experiment III), we investigated the performance of both types of recognizers using traditional evaluation approach. That is, the starting position (relative to the analysis window) of a specific pattern is the same for training and testing tasks. Training and testing samples were generated according to the pattern parameters and t_0 described in Table 2. The results of both recognizers are comparable in terms of recognition rate. The overall recognition rate is 98.14 % for raw data-based recognizer and 97.81 % for feature-based recognizer. The recognition rates are predictably higher than those of previous experiments due to the fact that the starting position of nonrandom patterns is the same for training and testing. The results of this experiment confirm that the recognition performance deteriorates significantly with pattern displacement.

To further demonstrate the advantage of the proposed features over previous work, a comparative study was conducted. Table 8 summarizes the performances of the neural network-based control chart pattern recognizers

Table 8 Performances of pattern recognizers using different sets of features

Experiment\Method	Wang et al.'s Features	Proposed Features
I	87.22/84.23 (0.52)	97.38/88.47 (0.45)
II	86.52/87.26 (0.80)	93.10/93.80 (0.51)
III	87.68/87.57 (0.28)	98.05/97.81 (0.25)

using the features proposed by the current research and those of Wang et al. [8]. The reported performance is the average recognition rate over 10 independent simulation runs. The number in parentheses next to the average recognition rate indicates the standard deviation of recognition rates. In all cases considered, our approach performs better than the pattern recognizer using the features proposed by Wang et al. [8]. The pattern recognizer using the proposed features has a higher average recognition rate and a lower standard deviation than that using Wang et al.'s features in each experiment. Even for traditional evaluation approach (experiment III), the average recognition rate of the proposed approach is about 10 % higher than that of the recognizer using Wang et al.'s features. This comparison clearly demonstrates the benefits of the proposed features in control chart pattern recognition.

Through simulation experiments, we have confirmed empirically that the feature-based recognizer performs better than the raw data-based recognizer. The proposed feature-based recognizer produces highly reliable and consistent recognition performance. One further advantage of the proposed approach can be noticed. It is found that the performance of the feature-based recognizer is robust against the variation of training samples in comparison with the raw data-based recognizer. This property is very attractive in terms of stability of the pattern recognizer.

6 Conclusion

In this paper, we have described a neural network-based pattern recognizer for control chart pattern analysis. Relevant features of the control chart patterns were extracted from the raw process data using statistical correlation analysis. The experimental results demonstrate that the recognition performance is markedly improved after using the proposed pattern features. The proposed features can retain the shape characteristics even when pattern displacement occurs. That is, the features proposed in this paper are robust to time-varying nature of control chart patterns. When adopting a machine learning-based pattern recognition approach, one advantage of the proposed features extraction is that it does not require a set of reference patterns. The proposed method computes the correlation coefficient between the input vector and its variants with the intention to retain the shape characteristics of a specific nonrandom pattern. More importantly, the performance of the proposed approach is robust against the effect of pattern displacement.

The approach proposed in the paper contributes to process monitoring and interpretation. It can be used as a diagnostic supplement to the traditional control charts. The performance of the recognizer can still be improved by the identification of discriminative features that will be more useful in separating trend from shift patterns. Further work can also include defining and extracting features suitable for those patterns not addressed in this paper.

References

1. Nelson LS (1984) The Shewhart control chart–test for special causes. J Qual Technol 16:237–239
2. Nelson LS (1985) Interpreting shewhart. X control chart. J Qual Technol 17:114–117
3. Western Electric Company (1956) Statistical Quality Control Handbook. Western Electric Company, Indianapolis
4. Cheng CS (1997) A neural network approach for the analysis of control chart patterns. Int J Prod Res 35:667–697
5. Cheng CS, Hubele NF (1996) A pattern recognition algorithm for an x-bar control chart. IIE Trans 29:215–224
6. Jiang P, Liu D, Zeng Z (2009) Recognizing control chart patterns with neural network and numerical fitting. J Intell Manuf 20:625–635
7. Cheng CS (1989) Group technology and expert systems concepts applied to statistical process control in small-batch manufacturing. Arizona State University, Dissertation
8. Wang CH, Guo RS, Chiang MH, Wong JY (2008) Decision tree based control chart pattern recognition. Int J Prod Res 46:4889–4901
9. Barghash MA, Santarisi NS (2004) Pattern recognition of control charts using artificial neural networks-analyzing the effect of the training parameters. J Intell Manuf 15:635–644
10. Guh RS, Tannock JDT (1999) A neural network approach to characterize pattern parameters in process control charts. J Intell Manuf 10:449–462
11. Pacella M, Semeraro Q, Anglani A (2004) Adaptive resonance theory-based neural algorithms for manufacturing process quality control. Int J Prod Res 42:4581–4607
12. Wang CH, Dong TP, Kuo W (2009) A hybrid approach for identification of concurrent control chart patterns. J Intell Manuf 20:409–419
13. Yang MS, Yang JH (2002) A fuzzy-soft learning vector quantization for control chart pattern recognition. Int J Prod Res 40:2721–2731
14. Das P, Banerjee I (2010) An hybrid detection system of control chart patterns using cascaded SVM and neural network-based detector. Neural Comput Appl 20:287–296
15. Zorriassatine F, Tannock JDT (1998) A review of neural networks for statistical process control. J Intell Manuf 9:209–224
16. Barghash MA, Santarisi NS (2007) Literature survey on pattern recognition in control charts using artificial neural networks. Conference on 37th Computers and Industrial Engineering, pp 20–23
17. Masood I, Hassan A (2010) Issues in development of artificial neural network-based control chart pattern recognition schemes. Eur J Sci Res 39:336–355
18. Al-Assaf Y (2004) Recognition of control chart patterns using multi-resolution wavelets analysis and neural network. Comput Ind Eng 47:17–29
19. Gauri SK, Chakraborty S (2009) Recognition of control chart patterns using improved selection of features. Comput Ind Eng 56:1577–1588
20. Hassan A, Baksh MSN, Shaharoun AM, Jamaluddin H (2003) Improved SPC chart pattern recognition using statistical features. Int J Prod Res 41(7):1587–1603

21. Pham DT, Wani MA (1997) Feature-based control chart pattern recognition. Int J Prod Res 35:1875–1890
22. Ebrahimzadeh A, Addeh J, Rahmani Z (2012) Control chart pattern recognition using K-MICA clustering and neural networks. ISA Trans 51:111–119
23. Hassan A, Baksh MSN (2008) An improved scheme for on-line recognition of control chart patterns. In: Proceedings 4th I*PROMS Virtual Conference
24. Swift JA (1987) Development of a knowledge based expert system for control chart pattern recognition and analysis. Oklahoma State University, Dissertation
25. Jang KY, Yang K, Kang C (2003) Application of artificial neural network to identify non-random variation pattern on the run chart in automotive assembly process. Int J Prod Res 41:1239–1254
26. Al-Ghanim AM, Ludeman LC (1997) Automated unnatural pattern recognition on control charts using correlation analysis techniques. Comput Ind Eng 32:679–690
27. Guh RS (2005) A hybrid learning-based model for on-line detection and analysis of control chart patterns. Comput Ind Eng 49:35–62
28. Yang JH, Yang MS (2005) A control chart pattern recognition system using a statistical correlation coefficient method. Comput Ind Eng 48:205–221
29. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, New York