Int J Adv Manuf Technol (2008) 36:1191–1201
DOI 10.1007/s00170-006-0925-4

ORIGINAL ARTICLE

Improved recognition of control chart patterns using artificial neural networks

Susanta Kumar Gauri · Shankar Chakraborty

Received: 29 July 2006 / Accepted: 29 December 2006 / Published online: 22 February 2007

Abstract Recognition of abnormal patterns in control charts provides clues to reveal potential quality problems in the manufacturing processes. One potentially popular approach for recognizing different control chart patterns (CCPs) is to develop heuristics based on various shape features of the patterns. The advantage of this approach is that the users can easily understand how a particular pattern is identified. However, consistency in the recognition performance is found to be considerably poor in the heuristics approach. Since shape features represent the main characteristics of the patterns in a condensed form, artificial neural network (ANN) with features extracted from the process data as input vector representation can facilitate efficient pattern recognition with a smaller network size. In this paper, a set of seven shape features is selected, whose magnitudes are independent of the process mean and standard deviation under a special representation of the sampling interval in the control chart plot. Based on these features, the CCPs are recognized using a multilayered perceptron neural network trained by back-propagation algorithm. The recognizer can recognize all the eight commonly observed CCPs. Extensive performance evaluation of this recognizer is carried out using simulated pattern data. Numerical results indicate that the developed ANN recognizer can perform well in real time process control applications with respect to both recognition accuracy and consistency.

S. K. Gauri
SQC & OR Unit, Indian Statistical Institute,
203, B. T. Road,
Kolkata 700108, India

S. Chakraborty (✉)
Department of Production Engineering, Jadavpur University,
Kolkata, West Bengal 700032, India
e-mail: s_chakraborty00@yahoo.co.in

Keywords Control chart patterns · Features · Pattern recognition · Neural network · Recognition performance

1 Introduction

Control charts, predominantly in the form of $\overline{\text{X}}$ chart, are important tools in statistical process control (SPC). They are useful in determining whether a process is behaving as intended or there are some unnatural causes of variation. A process is out of control if a point falls outside the control limits or a series of points exhibit an unnatural pattern. One of the eight types of patterns, e.g., normal (NOR), stratification (STA), systematic (SYS), cyclic (CYC), upward shift (US), downward shift (DS), increasing trend (UT) and decreasing trend (DT) [1] are usually observed in control charts. Only the normal pattern is indicative of the process continuing to operate under the chance causes, all other patterns are unnatural. Recognition of the abnormal patterns is an important aspect of SPC. Identification of the unnatural patterns can greatly narrow down the set of possible causes that must be investigated and thus the diagnostic search process can be effectively reduced in length.

Over the years, numerous supplementary rules known as zone tests or run tests [2] have been proposed to analyze control charts. Interpretation of the process data still remains difficult because it involves pattern recognition tasks. It often relies on the skill and experience of the quality control personnel to identify the existence of an unnatural pattern in the process. An efficient automated control chart pattern (CCP) recognition system can compensate this gap and ensure consistent and unbiased interpretation of CCPs leading to lesser number of false alarms and better implementation of control charts. Aiming this, some researchers [3–5] have developed expert systems for CCP recognition. Several techniques have been deployed in the knowledge base design of the expert systems, such as template matching, statistical testing, run rules and heuristic algorithms/rules. Although the results are promising, a common problem as reported in the previous studies is that of false recognition.

Development in the computing technology has motivated many researchers [6–11] to explore the use of artificial neural networks (ANNs) for control chart pattern recognition. The use of neural network techniques has overcome some of the drawbacks encountered in the conventional expert system approaches. Most of the existing ANN-based control chart pattern recognition schemes as reported in the literature, have used normalized or scaled raw data as input vectors to the recognizer. These data representations normally produce large ANN structures and are not very effective and efficient for complicated recognition problems. A smaller ANN size can lead to faster training with better recognition performance. The limitations of using pre-processed raw data as input vectors can be overcome with the use of extracted features from control charts for representing the data [10, 12]. Whereas Pham and Wani [12] have used nine shape features, Hassan et al. [10] have used six statistical features for recognition of six principal control chart patterns, i.e., NOR, CYC, UT, US, DT and DS. Since extracted features represent the main characteristics of the original data in a condensed form, the feature-based neural network approaches can facilitate accurate and efficient pattern recognition.

One limitation in extraction of statistical features is that it requires considerably large number of observations. Moreover, the statistical features lose information on the order of the data. On the other hand, each type of control chart pattern has its own geometric shape and various features can represent this shape. The advantage of shape features is that those can be extracted from lesser number of observations without losing order of the data. However, extraction of some of the shape features considered by Pham and Wani [12] requires user's inputs and consequently, their CCP recognition system is not truly automated.

Gauri and Chakraborty [13] have studied the usefulness of 32 possible shape features and presented a set of heuristics based on an optimal set of 13 features using classification and regression tree (CART) algorithm [14]. Their heuristics can recognize all the eight types of CCPs. The main advantage of their proposed approach is that extraction of the shape features does not require user's inputs in any form and so the CCP recognizer developed using those features is truly automated. However, a rigorous study on the recognition performance of the heuristic-based recognizer on multiple sets of simulated test samples reveals that its performances vary widely. In other words, the generalization of the heuristic-based recognizer is found to be quite poor. This is because that in the process of automatic selection of features under CART algorithm, a few correlated features have been selected in the optimal set and as pointed out by Montgomery and Peck [15], prediction based on correlated variables can lead to prediction instability. The number of patterns in different pattern classes are widely different (e.g., number of upward shift patterns are nine times more than normal patterns) in the learning samples used by Gauri and Chakraborty [13] for selection of the optimal set of features and the related heuristics. This may be the possible reason why such correlated features have been selected automatically.

On the other hand, although the magnitudes of all the features in the optimal set [13] are independent of the process mean, magnitudes of some features in the optimal set (e.g., *AASBP, ASL*, *SRANGE*, *BRANGE*, *DABL*, *DBRANGE, ALSPI*, *ABDPE*, *SASDPE* and *SASPE*) are dependent on the process standard deviation. Consequently, the heuristics based on these features will only be applicable to a specific process from where the learning samples are collected/simulated. From a preliminary study, it is observed that the magnitudes of all these features become independent of the process standard deviation if the mathematical expressions for two features *AASBP* and *ALSPI* are slightly modified and each sampling interval is represented by a constant linear distance, $c=1\sigma$, where σ is the standard deviation of the underlying process. Consequently, the CCP recognizers developed using the shape features extracted under this representation of sampling interval will be applicable to any general process.

In this paper, a set of seven shape features is considered and extracted under the above-said representation of sampling interval. Based on these features, CCPs are recognized using ANN techniques and the performance of this recognizer is extensively studied using synthetic pattern data.

2 Sample patterns

Ideally, sample patterns for developing/validating a CCP recognizer should be collected from a real process. Since, a large number of patterns are required for developing/validating a CCP recognizer and as those are not economically available, simulated data are often used. This is a common approach adopted by other researchers also.

Various control chart patterns are generated considering different pattern parameters as shown in Table 1. The window size (N) is taken to be 32, i.e., each observation window consists of 32 data points. The values of different pattern parameters are varied randomly in a uniform

Table 1 Parameters for simulating control chart patterns

Control chart patterns	Pattern parameters	Parameter values	Pattern equations
Normal	Mean (μ)	80	$y_i = \mu + r_i\sigma$
	Standard deviation (σ)	5	
Stratification	Random noise (σ')	0.2σ to 0.4σ	$y_i = \mu + r_i\sigma'$
Systematic	Systematic departure (d)	1σ to 3σ	$y_i = \mu + r_i\sigma + d \times (-1)^i$
Cyclic	Amplitude (a)	1.5σ to 2.5σ	$y_i = \mu + r_i\sigma + a \sin(2\pi i/T)$
	Period (T)	8 and 16	
Increasing trend	Gradient (g)	0.05σ to 0.1σ	$y_i = \mu + r_i\sigma + ig$
Decreasing trend	Gradient (g)	−0.1σ to −0.05σ	$y_i = \mu + r_i\sigma - ig$
Upward shift	Shift magnitude (s)	1.5σ to 2.5σ	$y_i = \mu + r_i\sigma + ks$;
	Shift position (P)	9, 17, 25	$k = 1$ if $i \geq P$, else $k = 0$
Downward shift	Shift magnitude (s)	−2.5σ to −1.5σ	$y_i = \mu + r_i\sigma - ks$;
	Shift position (P)	9, 17, 25	$k = 1$ if $i \geq P$, else $k = 0$

manner between the limits shown. For the generation of training and test patterns, 300 and 250 time series of standard normal data are used respectively. Since there are eight pattern classes considered in this study, a total of 2400 (300×8) and 2000 (250×8) sample patterns are simulated for training and validation/verification phases respectively. It may be noted that the training set contains equal number of samples for each pattern class. This is so done because if a particular pattern type is trained more number of times, the network will become biased towards that pattern.

3 Shape features

The pair-wise correlation coefficients among the thirteen shape features in the optimal set of features as derived by Gauri and Chakraborty [13] are estimated from the learning samples. It is noted that some pair-wise correlation coefficient values are very high, as given in Table 2. This implies that the optimal set of features includes some highly correlated features. For example, *ASL* is highly correlated with *DABL;SRANGE* with *BRANGE*, *REAE* and *DBRANGE*; and *ABDPE* with *SASDPE* and *SASPE*. Consequently, the stability (or consistency) of the recognition performance becomes considerably poor when the CCPs are recognized using those features. In this study, therefore, only those features, which are having fairly low correlation among themselves, are chosen (see Table 3). These features are listed below:

Table 2 Pair-wise correlation coefficients between some selected features

Features	ASL	SRANGE	ABDPE
AASBP	0.01	−0.04	0.30
ASL	1.00	−0.2	−0.11
SRANGE	−0.20	1.00	0.38
RVE	0.01	0.10	−0.09
REAE	−0.19	**0.91**	0.26
BRANGE	−0.21	**0.97**	0.46
DABL	**0.87**	−0.33	−0.30
DBRANGE	−0.15	**0.89**	0.36
ALSPI	0.03	−0.32	−0.46
ABDPE	−0.11	0.38	1.00
SASDPE	−0.11	0.41	**0.97**
SASPE	−0.13	0.32	**0.82**
REPEPE	−0.12	0.65	0.28

a) Ratio between the variance of the observations and mean sum of squares of errors of the least square (LS) line representing the overall pattern (*RVE*)
b) Average absolute slope of the straight lines passing through the consecutive points *(AASBP)*
c) Area between the overall pattern and LS line per interval in terms of SD^2 (*ALSPI*)
d) Average of slopes of straight lines passing through six pair-wise combinations of midpoints in four equal segments (*ASL*)
e) Range of slopes of straight lines passing through six pair-wise combinations of midpoints in four equal segments (*SRANGE*)
f) Ratio of mean sum of squares of errors of the LS line representing the overall pattern and pooled mean sum of squares of errors of the LS lines fitted to two segments that minimize the pooled mean sum of squares of errors (*REPEPE*)
g) Absolute slope difference between the LS line representing the overall pattern and line segments representing the patterns within the two segments that minimize the pooled mean sum of squares of errors (*ABDPE*)

The mathematical expressions of the above-mentioned seven features are shown in Appendix. It may be noted that

Table 3 Pair-wise correlation coefficients between considered features

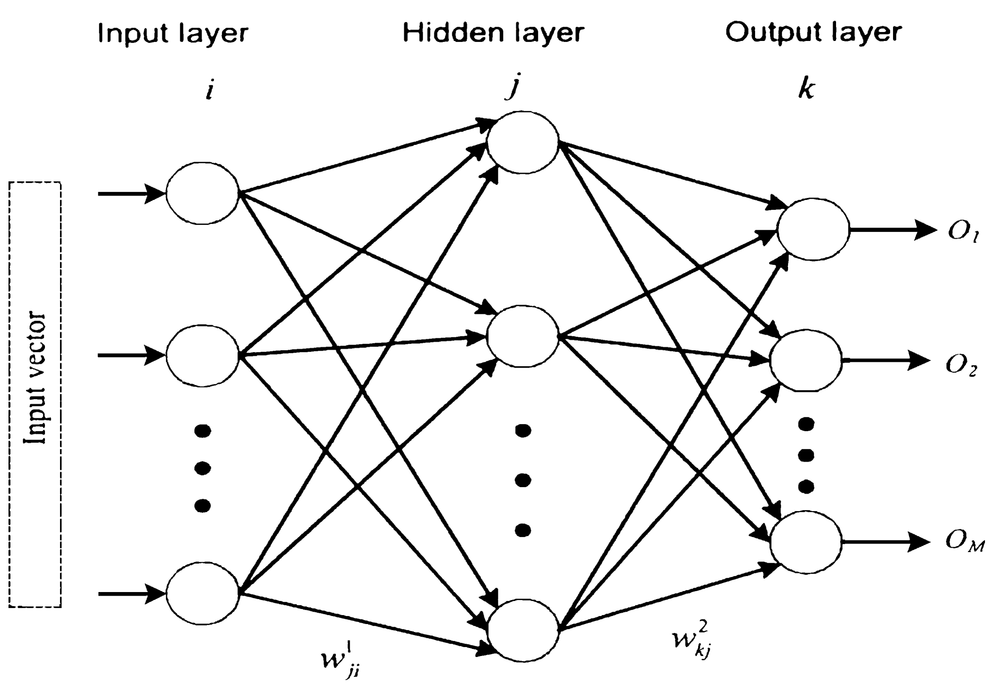
Features	AASBP	ASL	SRANGE	RVE	ALSPI	ABDPE	REPEPE
AASBP	1.00	0.01	−0.04	−0.22	−0.46	0.30	−0.27
ASL	0.01	1.00	−0.20	0.01	0.03	−0.11	−0.12
SRANGE	−0.04	−0.20	1.00	0.10	−0.32	0.38	0.65
RVE	−0.22	0.01	0.10	1.00	−0.30	−0.09	0.15
ALSPI	−0.46	0.03	−0.32	−0.30	1.00	−0.46	−0.12
ABDPE	0.30	−0.11	0.38	−0.09	−0.46	1.00	0.28
REPEPE	−0.27	−0.12	0.65	0.15	−0.12	0.28	1.00

the expressions for the features *AASBP* and *ALSPI* are marginally different from that used by Gauri and Chakraborty [13]. This is so done to ensure that the magnitudes of these two features become independent of the process standard deviation. The remaining features are independent of the process standard deviation due to the particular representation of the sampling interval in the form of c=1σ.

4 Pattern recognizer design using artificial neural network

The structure of a neural network can be characterized by the interconnection architecture among the processing elements, the transfer function for conversion of inputs into outputs and the learning algorithm. There exists a variety of different structures and learning algorithms useful for neural network applications, e.g., multilayer perceptron (MLP), counter propagation network, probabilistic neural network (PNN), learning vector quantization (LVQ), modular neural network (MNN) and others. Since a multilayer perceptron with back propagation learning rule has been successfully used by many researchers [6, 7, 9, 10] to solve pattern classification problems, this pattern recognizer is also developed based on MLP architecture. This type of neural network is simple and ideally suited for pattern recognition tasks. Its basic structure comprises an input layer, one or more hidden layer(s) and an output layer. The input layer receives numerical values from the outside world and the output layer sends information to the users or external devices. The processing elements in the hidden layer are used to create internal representations. Each processing element in a particular layer is fully connected to every processing element in the succeeding layer. There is no feed back to any of the processing elements. Figure 1 shows an MLP neural network architecture comprising these layers and their respective weight connections, w^1_{ji} and w^2_{kj}.

Fig. 1 MLP neural network architecture



Before this recognizer can be put into application, it needs to be trained and tested. In the supervised training approach, sets of training data comprising input and target vectors are presented to the MLP. The learning process takes place through calculating the error, propagating the error back through the network, and modifying/adjusting the weight connections between the input and hidden layers (w_{ji}^1) and between the hidden and output layers (w_{kj}^2) to reduce the error. These weight connections are adjusted according to the specified performance and learning functions of the neural network.

4.1 Neural network configuration

The general rule is that the network size should be as small as possible to allow efficient computation. In the present neural network application, the number of nodes in the input layer is set according to the actual number of features used, i.e., seven. The number of output nodes is set corresponding to the number of pattern classes, i.e., eight, each representing a particular pattern class. On the other hand, the number of nodes in the hidden layer is selected based on the results of many experiments conducted by varying the number of nodes from 10 to 20. All those experiments are coded in MATLAB® using its ANN toolbox [16]. The neural network is trained using a particular set of learning samples, and then two different sets of test samples are subjected to classification by the trained network. The resulting average misclassification percentage value is, then, estimated. The transfer functions, data representation scheme and training algorithms mentioned in Section 4.2, 4.3 and 4.4, respectively, are used for the network during those experimentations. The average recognition performances achieved under different number of hidden nodes are shown in Fig. 2, which indicates that the recognition performance of the neural network for the two sets of test samples is the maximum when the number of nodes in the hidden layer is 16. The selected ANN architecture is, therefore, 7×16×8.

4.2 Transfer function

The transfer functions used are hyperbolic tangent (*tansig*) for the hidden layer and sigmoid (*logsig*) for the output layer. The hyperbolic tangent function transforms the layer inputs to output range from −1 to +1 and the sigmoid function transforms the layer inputs to output range from 0 to 1 [17].

4.3 Data representation

Various shape features from different control chart patterns are extracted. The feature values are then mapped to an interval of (−1, 1) using a simple linear transformation,

$$s_i = [2 \times (f_i - f_{\min})/(f_{\max} - f_{\min}) - 1] \tag{1}$$

where, f_i and s_i are the actual and scaled values, respectively, for the i^{th} feature (i=1,2,..,7).

Since this study uses the supervised training approach, each pattern presentation is tagged with its respective label.

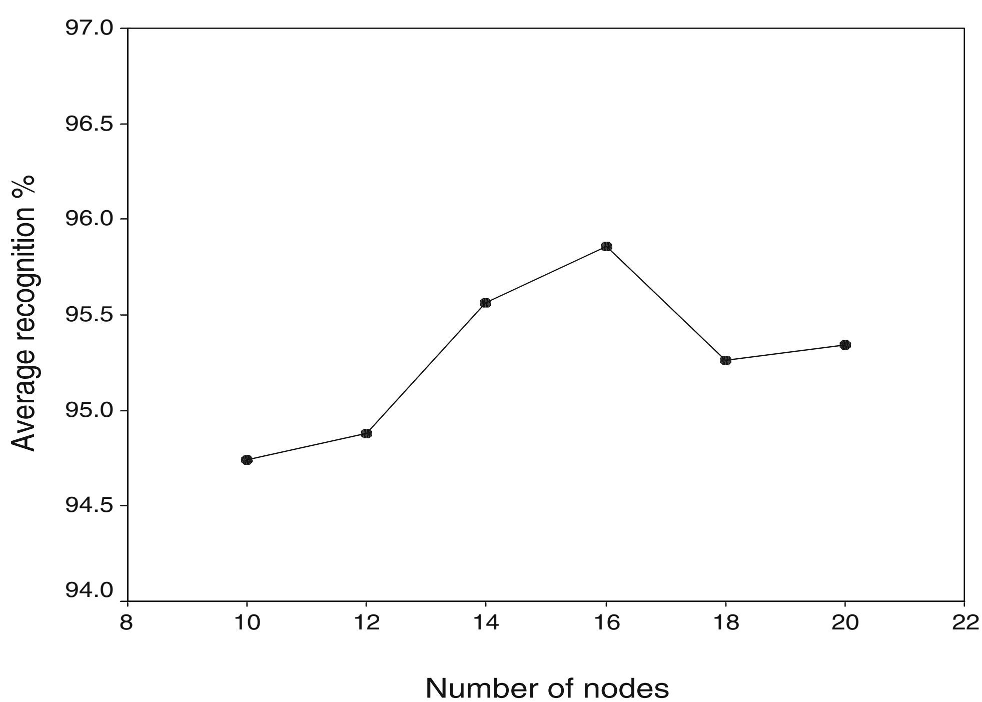


Fig. 2 Effect of number of nodes in the hidden layer on average recognition percentage

Table 4 Target recognizer outputs

Pattern class	Description	Recognizer outputs							
		Node							
		1	2	3	4	5	6	7	8
1	NOR	0.9	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	STA	0.1	0.9	0.1	0.1	0.1	0.1	0.1	0.1
3	SYS	0.1	0.1	0.9	0.1	0.1	0.1	0.1	0.1
4	CYC	0.1	0.1	0.1	0.9	0.1	0.1	0.1	0.1
5	UT	0.1	0.1	0.1	0.1	0.9	0.1	0.1	0.1
6	US	0.1	0.1	0.1	0.1	0.1	0.9	0.1	0.1
7	DT	0.1	0.1	0.1	0.1	0.1	0.1	0.9	0.1
8	DS	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.9

These labels, shown in Table 4, are the target values for the recognizers' output nodes. The maximum value in each row (0.9) identifies the corresponding node expected to secure the highest output for a pattern considered to be correctly classified. The output values are denoted as O_1, O_2,...,O_M in Fig. 1.

4.4 Training algorithm

Preliminary investigations are conducted to choose a suitable training algorithm. Three types of back propagation training algorithms, e.g., gradient descent with momentum and adaptive learning rate (*traingdx*), BFGS quasi-Newton (*trainbfg*), and Levenberg-Marquardt (*trainlm*) algorithms are evaluated, based on some experiments coded in MATLAB® using its ANN toolbox [16]. *Traingdx*, *trainbfg* and *trainlm* are various codes available in MATLAB® for different training algorithms. The *traingdx* is adopted here for training of the network, since it provides reasonably good performance and more consistent results. This result is in conformity with that of Hassan et al. [10]. This training algorithm is also more memory-efficient compared with *trainlm*. *Trainlm* gives the fastest convergence with the least number of epochs, but it requires too much memory. *Trainbfg* gives much faster convergence compared to *traingdx*, but the results are relatively less consistent. The network performance is measured using the mean squared error (MSE) value.

4.5 Training and verification

Recognition performance and generalization are the two critical issues for acceptance of a pattern recognizer for real time applications. With the aim to obtain a rigorous evaluation of the developed ANN-based pattern recognizer, three new sets of training samples of size 2400 each and four new sets of verification samples of size 2000 each are generated. Each set of the training data comprising the input and target vectors is presented to the MLP with the following training parameters:

– Maximum number of epochs=2500
– Learning rate=0.1
– Ratio to increase learning rate=1.05
– Error goal=0.01
– Momentum constant=0.5
– Ratio to decrease learning rate=0.7

The training is stopped whenever either the error goal has been achieved or the maximum allowable number of training epochs has been met. In this process, three different ANN-based recognizers are developed. All these recognizers have the same architecture and differ only in the training data sets used.

A good idea about the generalization capability of a recognizer can be obtained by exposing it to multiple sets of test samples for classification. The recognition performance of all these three ANN-based recognizers are, therefore, tested using different sets of test samples. The procedures for training and verification are coded in MATLAB® using its ANN toolbox [17]. The recognition performance of the ANN-based pattern recognizers at the training and verification stages are provided in Table 5.

4.6 Implementation

The trained network stores the implicit decision rules through a set of connection weights used to recognize the unnatural patterns. Given with an input vector, the neural network will always produce an output vector. In this application, the value of each processing element in the output layer is a real-valued variable (between 0 and 1).

Table 5 Training and verification performance of the ANN recognizers

Recognizer number	Training phase	Verification phase			
	Correct classification (%)	Correct classification (%)			
		Mean	Min.	Max.	Range
R1	95.63	95.69	95.46	96.01	0.55
R2	96.12	95.76	95.13	96.18	1.05
R3	95.89	95.81	95.34	96.29	0.95
Overall mean	95.88	95.75	Overall range		1.16

The recognizer selects the pattern corresponding to the output node having the maximum value. The proposed pattern recognizer involves the following steps:

Step 0. Collect the most recent 32 sample means from the monitored process.

Step 1. Extract the magnitudes of seven shape features, e.g., *RVE*, *AASBP*, *ALSPI*, *ASL*, *SRANGE*, *REPEPE* and *ABDPE*. Scale these values to (−1, 1) interval. Denote the coded values by v_i (i=1,2,..,7). The coded data form an input vector, V_I, to the neural network, where $V_I=\{v_i\}$.

Step 2. Present the input vector V_I to the trained neural network and compute the output vector $V_o=(o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8)$.

Step 3. Let $o^*=\max\ [o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8]$ corresponds to the *ith* node. Then the pattern corresponding to the *ith* output node will be the identified pattern by the recognizer.

Step 4. Go to step 0.

5 Results and discussions

Table 5 shows the training and verification performance of the three ANN-based recognizers (coded as R1, R2 and R3). It is noted that for all these three recognizers, the recognition performance at the training and verification phases do not differ significantly. The overall mean percentage values of correct recognition at the training and verification phases are 95.88% and 95.75% respectively, and the overall range for percentage of correct classification in the verification phase is 1.16% only. This implies that the proposed ANN-based pattern recognizer can produce highly reliable and consistent recognition performance.

5.1 Confusion matrix

The confusion matrix is a table summarizing the tendency of the recognizer to classify a recognized pattern into a correct class or into any of the other seven possible (wrong) classes. The confusion matrix, as given in Table 6, provides the overall mean percentage of confusions among different pattern classes for the recognizer. In other words, these are the mean of scores from 12 such matrices (3 recognizers × 4 test sets).

Table 6 shows that there is confusion in the classification process by the recognizer. There is a tendency for cyclic patterns to be mostly confused with normal patterns, shift patterns with trend patterns, and trend patterns with shift patterns. Cyclic patterns are the hardest to be classified (93.67%), followed by downward shift (94.00%) and downward trend (94.33%) patterns. This indicates that the performance of the recognizer can still be improved by the identification of new features that will be more useful in

Table 6 Confusion matrix for the ANN-based recognizers

True pattern class	Identified pattern class							
	NOR	STA	SYS	CYC	UT	US	DT	DS
NOR	94.78	0.00	0.00	3.11	1.44	0.00	0.67	0.00
STA	0.11	99.89	0.00	0.00	0.00	0.00	0.00	0.00
SYS	0.00	0.00	99.89	0.11	0.00	0.00	0.00	0.00
CYC	3.00	0.00	0.11	93.67	0.33	0.00	1.11	1.78
UT	1.00	0.00	0.00	0.44	94.89	2.67	1.00	0.00
US	0.44	0.00	0.00	0.44	3.56	94.56	0.44	0.56
DT	1.56	0.00	0.00	0.44	0.00	0.22	94.33	3.44
DS	1.33	0.00	0.00	0.56	0.00	0.33	3.78	94.00

discriminating cyclic from normal patterns, and shift from trend patterns. On the other hand, the results for classification of normal patterns in Table 6 (94.78%) suggest that the type I error performance of the recognizer does not seem to be quite good. This is possibly due to the unpredictable nature of the random data streams that make them relatively more difficult to be recognized compared to other unnatural patterns.

5.2 Relative importance of the selected features

Use of too many features as input vector will lead to a more complex neural network due to increase in the network size, which is undesirable. Understanding the relative importance of various features on the recognition performance can be helpful in identifying the unimportant feature, if any, in the chosen set. This will be valuable because the network size can be reduced by eliminating the unimportant feature(s) or the unimportant feature(s) can be replaced by more powerful new feature(s), if can be identified and thus, the recognition performance can be improved restricting the network size to the bare minimum.

A possible approach for assessing the relative importance of various features can be described as follows:

a) Select a particular set of verification samples.
b) From the selected set of verification samples, prepare seven artificial sets of test samples. In each artificial set of test samples, replace all the individual values of a feature corresponding to different patterns by the mean value of the feature, i.e., make the values of one feature invariant artificially keeping the values of all other features as those are in the original set of verification samples.
c) Recognize the control chart patterns from the artificial sets of test samples using the proposed CCP recognizer.
d) Compute the amount of increase in misclassification error (MCE) due to making various features invariant in the artificial sets of test samples.
e) The feature whose invariance leads to the maximum increase in MCE may be considered as the most important feature for correct recognition of CCPs, and the feature whose invariance does not affect MCE significantly may be considered as an unimportant feature.

As noted from Table 5, the range for percentage of correct classification is the minimum (0.55%) for the recognize R1 and therefore, it is taken for the purpose of assessing the relative importance of various features. This CCP recognizer has resulted in an average 95.69% of correct recognition, i.e., 4.31% of misclassification at the verification phase. Seven artificial sets of test samples are prepared from a set of verification samples and then subjected to classification using the recognizer R1. The observed misclassification values in seven artificial sets of test patterns, arranged in a descending order, are given in Table 7. It can be observed from the table that all the chosen features have some important contribution in recognition of various CCPs. However, the most important and the least important features in the chosen set are *RVE* and *SRANGE* respectively. It can be noted from Table 3 that the feature *SRANGE* is quite highly correlated with feature *REPEPE*, which is the second most important feature. This may be the possible reason for the least importance of the feature *SRANGE*. If a new feature, which will be more powerful in pattern discrimination and whose degree of association with the chosen set of features will be fairly low, can be identified, it should replace the feature *SRANGE* to achieve better recognition performance.

Table 7 Effects on pattern misclassification (%) due to various invariant features

Set of test patterns	Invariant feature	Observed misclassification	Amount of increase in misclassification
1	RVE	48.91	44.60
2	REPEPE	41.42	37.11
3	AASBP	30.70	26.39
4	ABDPE	28.17	23.86
5	ALSPI	24.45	20.14
6	ASL	22.37	18.06
7	SRANGE	9.67	5.36

6 Conclusions

A set of seven shape features is selected so that their magnitudes will be independent of the process mean and standard deviation. Based on these features, all the eight commonly observed CCPs are recognized using a multilayered perceptron artificial neural network trained by backpropagation algorithm. The performance of the recognizer is extensively studied using synthetic pattern data. The numerical results reveal that the developed ANN-based recognizer can perform well in real time process control applications in terms of reliability and consistency of recognition performance. Analysis of the confusion matrix indicates that the recognizer has a general tendency of confusing cyclic patterns with normal, shift patterns with trend, and trend patterns with shift. This is indicative of the fact that the performance of the recognizer can be improved further by identification of new features that will help in discriminating cyclic patterns from normal, and shift patterns from trend. The least contributing feature is identified based on analysis of the relative importance of various features. If a new feature, which will be more

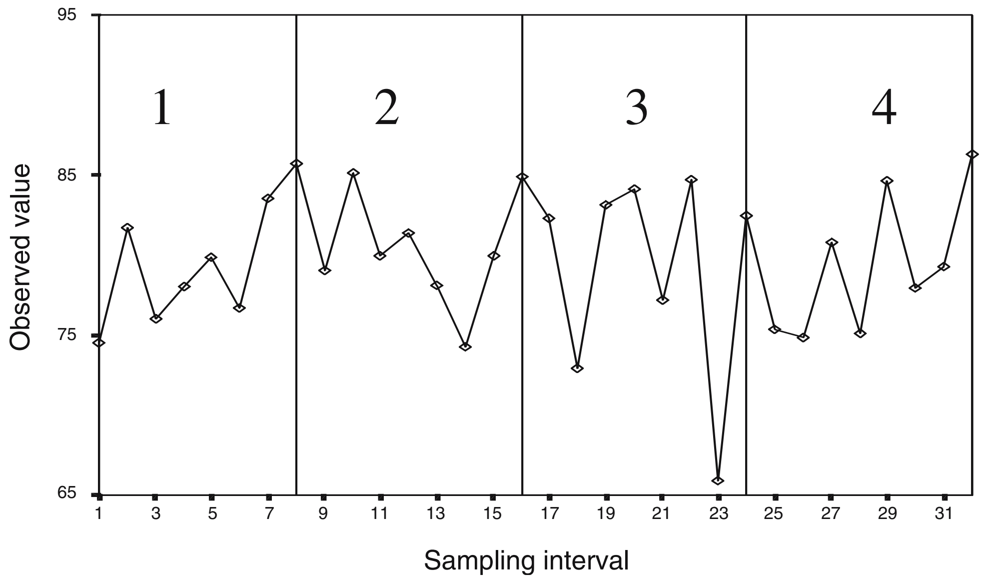


Fig. 3 Four equal segments in a pattern

powerful in pattern discrimination, can be identified, it should be used in place of the least contributing feature to achieve better pattern recognition performance without increasing the neural network size.

Appendix: Mathematical expressions for various shape features

(a) Ratio between variance of the observations and mean sum of squares of errors of the least square (LS) line representing overall pattern (*RVE*):

$$RVE = \left[\sum_{i=1}^{N}(y_i-\overline{y})^2 \Big/ (N-1)\right] \Big/ \left[\left\{\sum_{i=1}^{N}(y_i-\overline{y})^2 - \left(\sum_{i=1}^{N}y_i(t_i-\overline{t})\right)^2 \Big/ \sum_{i=1}^{N}(t_i-\overline{t})^2\right\} \Big/ (N-2)\right] \tag{2}$$

where, $t_i = ic\,(i = 1, 2, .., N)$ is the distance of *ith* time point of observation from the origin, y_i is the observed value of a quality characteristic at *ith* time point, and N is the total number of observations in the window.

b) Average absolute slope of the straight lines passing through the consecutive points *(AASBP)*:

$$AASBP = \sum_{i=1}^{N-1}|(y_{i+1}-y_i)/(t_{i+1}-t_i)| \Big/ (N-1) \tag{3}$$

c) Area between the pattern and LS line per interval in terms of SD^2 (*ALSPI*):

$$ALSPI = [ALS/(N-1)] \Big/ SD^2 \text{ and } SD^2 = \sum_{i=1}^{N}(y_i-\overline{y})^2 \Big/ (N-1) \tag{4}$$

where, *ALS* is the area between the pattern and the fitted LS line. The value of *ALS* can be easily computed by summing the areas of the triangles and trapeziums that are formed by the LS line and overall pattern.

d) Average slope of the straight lines passing through six pair-wise combinations of midpoints in four equal segments (*ASL*):

The total length of the data plot is divided into four equal segments (see Fig. 3) and the behavior of the process in a quarter time period, i.e., within a segment, is represented by the midpoint of the segment, which can be estimated as

$$\left\{\left[\sum_{i=n_1}^{n_1+7}t_i \Big/ 8\right], \left[\sum_{i=n_1}^{n_1+7}y_i \Big/ 8\right]\right\}$$

where n_1=1, 9, 17 and 25 for the first, second, third and fourth segment, respectively. A combination of two midpoints can be obtained in $C_2^4 = 6$ ways implying that six straight lines can be drawn passing through the midpoints of these four segments. Thus,

$$ASL = \sum_{\substack{j,k \\ j<k}} s_{jk} \Big/ 6; (j = 1, 2, 3; k = 2, 3, 4) \tag{5}$$

where s_{jk} is the slope of the straight line passing through the midpoints of *jth* and *kth* segments.

e) Range of slopes of straight lines passing through six pair-wise combinations of midpoints (*SRANGE*):

$$SRANGE = \text{maximum}(s_{jk}) - \text{minimum}(s_{jk}); (j = 1, 2, 3; k = 2, 3, 4; j < k) \quad (6)$$

In case of a shift pattern, the total observations can be divided into two windows (before and after the occurrence of a shift) and two LS lines (each approximately horizontal to the X-axis) can be fitted well in these windows. However, the time point of occurrence of shift cannot be known exactly. Therefore, a criterion-based segmentation, where the window sizes may vary in order to satisfy the desired criterion, is taken into account. Two LS lines that lead to the minimum pooled mean sum of squares of errors (*PMSE*) are considered here as the two best fitted lines within the overall pattern. Let the limiting time point of segmentation is m ($8 \leq m \leq 24$). The *PMSE* value of the two LS lines is, then, given by the following expression:

$$\frac{\left[\sum_{i=1}^{m}(y_i-\bar{y})^2-\left(\sum_{i=1}^{m}y_i(t_i-\bar{t}_1)\right)^2\Big/\sum_{i=1}^{m}(t_i-\bar{t}_1)^2\right]+\left[\sum_{i=m+1}^{N}(y_i-\bar{y})^2-\left(\sum_{i=m+1}^{N}y_i(t_i-\bar{t}_2)\right)^2\Big/\sum_{i=m+1}^{N}(t_i-\bar{t}_2)^2\right]}{N-4}$$

where

$$\bar{t}_1 = \sum_{i=1}^{m} t_i \Big/ m, \text{ and } \bar{t}_2 = \sum_{i=m+1}^{N} t_i \Big/ (N-m).$$

Assuming that at least eight observations are required for fitting a LS line, we fit the LS lines to all possible two windows and compute the corresponding *PMSE* values. Then the value of m that leads to the minimum *PMSE* is considered as the limit point of the first window and (m+1) becomes the starting point of the second window. Using this segmentation, the following two features are extracted.

f) Ratio of mean sum of squares of errors of the LS line representing the overall pattern and pooled mean sum of squares of errors of the LS lines fitted to the two segments (*REPEPE*):

$$REPEPE = MSE/PMSE \quad (7)$$

where *MSE* is the mean sum of squares of errors of the LS line fitted to overall pattern and is given by the following equation:

$$\left[\sum_{i=1}^{N}(y_i-\bar{y})^2-\left(\sum_{i=1}^{N}y_i(t_i-\bar{t})\right)^2\Big/\sum_{i=1}^{N}(t_i-\bar{t})^2\right]\Big/(N-2)$$

g) Absolute slope difference between the LS line representing the overall pattern and line segments representing the patterns within the two segments (*ABDPE*):

$$ABDPE = \left|B - \left(\sum_{j=1}^{2} B_j \Big/ 2\right)\right|; (j = 1, 2) \quad (8)$$

where B_j is the slope of the LS line fitted to *jth* window and B is the slope of the LS line fitted to overall pattern, and can be given by the expression:

$$\left[\sum_{i=1}^{N} y_i(t_i-\bar{t})\right]\Big/\left[\sum_{i=1}^{N}(t_i-\bar{t})^2\right]$$

References

1. Montgomery DC (1985) Introduction to statistical quality control. McGraw-Hill, New York
2. Western Electric (1958) Statistical quality control handbook. American Telephone and Telegraph Company, Chicage, Illinois
3. Evans JR, Lindsay WM (1988) A framework for expert system development in statistical quality control. Comput Ind Eng 14 (3):335–343
4. Pham DT, Oztemel E (1992) XPC: an on-line expert system for statistical process control. Int J Prod Res 30(12):2857–2872
5. Swift JA, Mize JH (1995) Out-of-control pattern recognition and analysis for quality control charts using LISP-based systems. Comput Ind Eng 28(1):81–91
6. Hwarng HB, Hubele NF (1993) Back-propagation pattern recognisers for $\overline{\mathrm{X}}$ control charts: methodology and performance. Comput Ind Eng 24(2):219–235
7. Cheng CS (1997) A neural network approach for the analysis of control chart patterns. Int J Prod Res 35(3):667–697
8. Guh RS, Tannock JDT (1999) Recognition of control chart concurrent patterns using a neural network approach. Int J Prod Res 37(8):1743–1765
9. Perry MB, Spoerre JK, Velasco T (2001) Control chart pattern recognition using back propagation artificial neural networks. Int J Prod Res 39(15):3399–3418
10. Hassan A, Nabi Baksh MS, Shaharoun AM, Jamaluddin H (2003) Improved SPC chart pattern recognition using statistical features. Int J Prod Res 41(7):1587–1603

11. Guh RS, Shiue YR (2005) On-line identification of control chart patterns using self-organizing approaches. Int J Prod Res 43 (6):1225–1254
12. Pham DT, Wani MA (1997) Feature-based control chart pattern recognition. Int J Prod Res 35(7):1875–1890
13. Gauri SK, Chakraborty S (2006) A study on the various features for effective control chart pattern recognition. Int J Adv Manuf Technol (available online on 8th June, 2006)
14. Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth and Brooks, Monterey, CA
15. Montgomery DC, Peck EA (1982) Introduction to linear regression analysis. John Wiley, New York
16. Demuth H, Beale M (1998) Neural network toolbox user's guide. Math Works, Natick, MA
17. Smith M (1993) Neural networks for statistical modeling. Van Norstrand Reinhold, New York