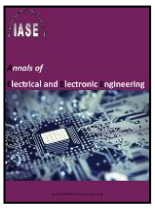
Annals of Electrical and Electronic Engineering, 2(4) 2019, Pages: 8-13

Published by IASE

Annals of Electrical and Electronic Engineering

Journal homepage: www.aeeej.org



Control chart pattern identification using a synergy between neural networks and bees algorithm

P. K. Wong*, A. Chua

Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119260, Singapore

ARTICLE INFO

Article history:
Received 5 November 2018
Received in revised form
20 February 2019
Accepted 27 February 2019

Keywords:
Artificial neural network
Control chart patterns
Bees algorithm
Shape feature
Feature selection

ABSTRACT

In the recent years, as an alternative of the traditional process quality management methods, such as Shewhart statistical process control (SPC), artificial neural networks (ANN) have been widely used to recognize the abnormal pattern of control charts. A common problem of existing approaches to control chart patterns (CCPs) recognition is false classification between different types of CCPs that share similar features in a real-time process-monitoring scenario, in which only limited pattern points are available for recognition. This study presents an automatic recognition system for control chart patterns recognition based on bees algorithm (BA) and artificial neural networks. In this study, BA is used for reducing the dimension of CCPs database and ANN is used for intelligent classification. The proposed BA +ANN system performance is compared with ANN model. The dimension of input feature space is reduced from nine to four by using BA. The proposed method (BA+ANN) uses a multiplayer perceptrons (MLP) neural network as pattern recognizer. The MLP architecture has been successfully applied to solve some difficult and diverse problems in modeling, prediction and pattern classification. Simulation results show that the proposed method (BA+ANN) has very high recognition accuracy. This high efficiency is achieved with only little features, which have been selected using BA.

1. Introduction

Control charts are the one of the simplest monitoring tools and by the implementations of this tool the information obtained about the process is either is in control or out of control. If the process is in control, the operator or user can precedes the process under the same conditions. Conversely, the operator or user must identify the root causes of events happened in the process.

CCPs can exhibit six types of pattern: normal (NR), cyclic (CC), upward trend (UT), downward trend (DT), upward shift (US) and downward shift (DS) (Montgomery, 2007). Except for normal patterns, all other patterns indicate that the process being monitored is not functioning correctly and requires adjustment. Fig. 1 shows six pattern types of control chart.

In recent years, several studies have been performed for recognition of the unnatural patterns. Some of the researchers used the expert systems (Swift and Mize, 1995; Evans and Lindsay, 1988). The advantage of an expert system or rule-based system is that it contains the information explicitly. If required, the rules can be modified and updated easily. However, the use of rules based on statistical properties has the difficulty that similar statistical properties may be derived for some patterns of different classes, which may create problems of incorrect recognition.

Also, artificial neural networks (ANNs) have been widely applied for classifiers. ANNs can be simply categorized into two groups comprising supervised and unsupervised. Most researchers (Le et al., 2004; Pharm and Oztemel, 1995; Cheng and Ma, 2008; Sağıroğlu et al., 2000; Pham and Oztemel, 1994) have used supervised ANNs, such as multi-layer perceptron (MLP), radial basis function (RBF), and learning vector quantization (LVQ), to classify different types of CCPs. Furthermore, unsupervised methods, e.g. self-organized maps (SOM) and adaptive resonance theory (ART) have been applied to fulfill the same objective in other studies (Wang et al., 2007). The advantage with neural network is that it is capable of handling noisy measurements requiring no assumption about the statistical distribution of the monitored data. It learns to recognize patterns directly through typical example patterns during a training phase.

Some of the researchers used the support vector machine to CCP recognition. The accuracy of an SVM is dependent on the choice of kernel function and the parameters (e.g., cost parameter, slack variables, margin of the hyper plane, etc.). Failure to find the optimal parameters for an SVM model affects its prediction accuracy (Campbell and Cristianini, 1998).

Most the existing techniques used the unprocessed data as the inputs of CCPs recognition system. The use of unprocessed CCP data has further problems such as the amount of data to be processed is large. On the other hand, the approaches which use features are more flexible to deal with a complex process problem, especially when no prior information is available. If the features represent the characteristic of patterns explicitly and if their components are reproducible with the process conditions, the classifier recognition accuracy (RA) will increase (Pacella et al., 2004). Features could be obtained in various forms, including shape features (Wani and Rashid, 2005; Gauri and Chakraborty, 2009; Pham and Wani, 1997), multi-resolution wavelet analysis and statistical features (Hassan et al., 2003).

Based on the published articles, there exist some important issues in the design of automatic CCPs recognition system which if suitably addressed, lead to the development of more efficient recognizers. One of these issues is the extraction of the features. In this article for obtaining the compact set of features which

* *Corresponding Author.*
Email Address: Pohkamwong@nus.edu.sg (P. K. Wong)
https://doi.org/10.21833/AEEE.2019.04.002

capture the prominent characteristics of the CCPs, a proper set of the shape features and are proposed.

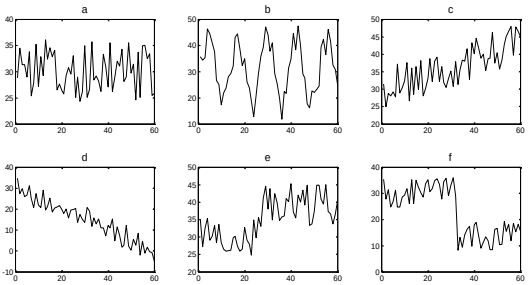


Fig. 1. Six various basic patterns of control charts: (a) normal pattern, (b) cyclic pattern, (c) upward trend, and (d) downward Trend, (e) upward shift, (f) downward shift.

Unnecessary features increase the size of the search space and make generalization more difficult. Irrelevant and redundant features increase the search space size, making patterns more difficult to detect and making it more difficult to capture rules necessary for forecasting or classification, whether by machine or by hand. In addition, the more the features, the higher the risk of over fitting. The probability that some features will coincidentally fit the data increases, unless the sample size grows exponentially with the number of features. Furthermore, in most practical applications, we want to know the collection of core variables that are most critical in explaining an event. For this respect, in this study, Bees algorithm (BA) (Pham et al., 2006) is chosen as an optimization technique to optimize the input feature subset selection. This technique will improve the MLP performance.

Another issue is related to the choice of the classification approach to be adopted. The developed method uses a multiplayer perceptrons (MLP) as pattern recognizer. The MLP architecture has been successfully applied to solve some difficult and diverse problems in modeling, prediction and pattern classification (Haykin, 1999).

This article is organized as follows. Section 2 describes the data and features. Section 3 describes the MLP and BA. Section 4 shows some simulation results and finally Section 5 concludes the article.

2. Data description and features

2.1. Data description

For this study, each pattern was taken as a time series of 60 data points. The following equations were used to create the data points for the various patterns (Wani and Pham, 1999):

- Normal patterns:

$$p(t) = \eta + r(t)\sigma \quad (1)$$

- Cyclic patterns:

$$p(t) = \eta + r(t)\sigma + a\sin(2\pi t/T) \quad (2)$$

- Increasing trend patterns:

$$p(t) = \eta + r(t)\sigma + gt \quad (3)$$

- Decreasing trend patterns:

$$p(t) = \eta + r(t)\sigma - gt \quad (4)$$

- Upward shift patterns:

$$p(t) = \eta + r(t)\sigma + bs \quad (5)$$

- Downward shift patterns:

$$p(t) = \eta + r(t)\sigma - bs, \quad (6)$$

where η is the nominal mean value of the process variable under observation (set to 80), σ is the standard deviation of the process variable (set to 5), a is the amplitude of cyclic variations in a cyclic pattern (set to 15 or less), g is the gradient of an increasing trend pattern or a decreasing trend pattern (set in the range 0.2 to 0.5), b indicates the shift position in an upward shift pattern and a downward shift pattern ($b = 0$ before the shift and $b = 1$ at the shift and thereafter), s is the magnitude of the shift (set between 7.5 and 20), $r(.)$ is a function that generates random numbers normally distributed between -3 and 3, t is the discrete time at which the monitored process variable is sampled (set within the range 0 to 59), T is the period of the cycle (set between 4 and 12 sampling intervals) and $p(t)$ is the value of the sampled data point at time t.

2.2. Shape features

The shape features used by the CCP recognizer in this study are such that they facilitate recognition of CCPs quickly and accurately. The six types of CCP considered in this work have different forms, which can be characterized by a number of shape features. In Pham and Wani (1997), the authors have introduced nine shape features for discrimination of the CCPs. These features are as follows:

1) S: The slope of the least-square line representing the pattern. The magnitude of S of this line for normal and cyclic patterns is approximately zero, while that for trend and shift patterns is greater than zero. Therefore S may be a good candidate to differentiate natural and cyclic patterns from trend and shift patterns.
2) NC1: The number of mean crossings, i.e. the crossings of the pattern with the mean line. NC1 is small for shift and trend patterns. It is highest for normal patterns. For cyclic patterns, the number of crossings is intermediate between those for normal patterns and shift or trend patterns. This feature differentiates normal patterns from cyclic patterns. It also differentiates normal and cyclic patterns from trend and shift patterns.
3) NC2: The number of least-square line crossings. NC2 is highest for normal and trend patterns and lowest for shift and cyclic patterns. Thus it can be used for separation of natural and trend patterns from others.
4) Cyclic membership (cmember): This feature indicates how closely a pattern resembles a cyclic pattern. It is possible that if complete cycles are not present then the slope of a cyclic pattern may not be equal to zero but may lie in the range for trend and shift patterns. For such situations, a cyclic membership function is defined which gives a measure of how closely a pattern resembles a cycle. The membership function, detailed later, produces a positive value for cyclic patterns and a negative value for all other patterns. This feature therefore differentiates cyclic patterns from other patterns.
5) AS: The average slope of the line segments. In addition to the least-square line which approximates the complete pattern, each pattern also has two line segments which fit the data starting from either end of the pattern. The average slope of the line segments for a trend pattern will be higher than for normal, cyclic and shift patterns. This feature therefore differentiates trend patterns from other patterns.
6) SD: The slope difference between the least-square line and the line segments representing a pattern. The SD value is obtained by subtracting the average slope as of the two line segments from the slopes of the least-square line. For normal, cyclic and trend patterns, the least-square line and the line

segments will be different. Thus, the SD will have a high value for a shift pattern and small values for normal, cyclic and trend patterns. This feature therefore differentiates a shift pattern from other patterns.

7) APML: The area between the pattern and the mean line. The APML is lowest for a normal pattern. Thus, this feature differentiates between normal and other patterns.
8) APSL: the area between the pattern and its least-square line. Cyclic and shift patterns have a higher APSL value than normal and trend patterns and therefore the APSL can be used to differentiate cyclic and shift patterns from normal and trend patterns.
9) ASS: The area between the least-square line and the line segments. The value of this feature is approximately zero for a trend pattern and is higher for a shift pattern. This feature thus differentiates trend patterns from shift patterns.

3. Needed concepts

3.1. Multi-layer perceptron (MLP) neural networks

An MLP neural network consists of an input layer (of source nodes), one or more hidden layers (of computation nodes) and an output layer. The recognition basically consists of two phases: training and testing. In the training stage, weights are calculated according to the chosen learning algorithm. The issue of learning algorithm and its speed is very important for the MLP model. It is very difficult to know which training algorithm will be the fastest for a given problem. It depends on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, the error goal, and whether the network is being used for pattern recognition (discriminant analysis) or function approximation (regression). In this study the following learning algorithms are considered.

3.1.1. Back-propagation with momentum (BP with momentum)

The BP algorithm makes use of gradient descent with a momentum term to smooth out oscillation (Haykin, 1999). Eq. (21) gives the weight update for BP with momentum:

$$\Delta W_{ij}(t+1) = -\varepsilon \frac{\delta E}{\delta W_{ij}}(t) + \mu \frac{\delta E}{\delta W_{ij}}(t-1) \tag{7}$$

where W_{ij} represents the weight value from neuron j to neuron i, ε is the learning rate parameter, and E represents the error function. It adds an extra momentum parameter, μ, to the weight changes.

3.1.2. Resilient back-propagation (RPROP) algorithm

RPROP considers the sign of derivatives as the indication for the direction of the weight update (Riedmiller and Braun, 1993). In doing so, the size of the partial derivative does not influence the weight step. The following equation shows the adaptation of the update values of Δ_{ij} (weight changes) for the RPROP algorithm. For initialization, all are set to small positive values:

$$\Delta_{ij}(t) = \begin{cases} \eta^{+} \times \Delta_{ij}(t-1); if \frac{\delta E}{\delta W_{ij}}(t-1) \frac{\delta E}{\delta W_{ij}}(t) > 0 \\ \eta^{-} \times \Delta_{ij}(t-1); if \frac{\delta E}{\delta W_{ij}}(t-1) \frac{\delta E}{\delta W_{ij}}(t) < 0 \\ \eta^{0} \times \Delta_{ij}(t-1); otherwise \end{cases} \tag{8}$$

where $\eta^0 = 0, 0 < \eta^- < 1 < \eta^+, \eta^{-,0,+}$ are known as the update factors. Whenever the derivative of the corresponding weight changes its sign, this implies that the previous update value is too large and it has skipped a minimum. Therefore, the update value is then reduced (η^-), as shown above. However, if the derivative retains its sign, the update value is increased (η^+). This will help to accelerate convergence in shallow areas. To avoid over-acceleration, in the epoch following the application of (η^+), the new update value is neither increased nor decreased (η^0) from the previous one. Note that the values of Δ_{ij} remain non-negative in every epoch. This update value adaptation process is then followed by the actual weight update process, which is governed by the following equations:

$$\Delta W_{ij}(t) = \begin{cases} -\Delta_{ij}; if \frac{\delta E}{\delta W_{ij}}(t) > 0 \\ +\Delta_{ij}; if \frac{\delta E}{\delta W_{ij}}(t) < 0 \\ 0; otherwise \end{cases} \tag{9}$$

The values of the training parameters adopted for the algorithms were determined empirically.

3.1.3. Levenberg–Marquardt (LM) algorithm

The LM algorithm (Hagan and Menhaj, 1994) uses the approximation to the Hessian matrix in the following Newton-like update:

$$w_{ij}(t+1) = w_{ij}(t) - [J^T J + \mu I]^{-1} J^T e \tag{10}$$

where J is the Jacobian matrix, e a vector of network errors and μ a constant.

3.1.4. scaled conjugate gradient algorithm (SCG)

The scaled conjugate gradient algorithm (SCG), developed by Moller (1990), was designed to avoid the time-consuming line search. This algorithm combines the model-trust region approach (used in the Levenberg-Marquardt algorithm, described in Levenberg-Marquardt), with the conjugate gradient approach. See Moller (1990) for a detailed explanation of the algorithm.

3.1.5. Conjugate gradient backpropagation with Fletcher-Reeves updates (CGBFR)

More detail regarding the Conjugate gradient backpropagation with Fletcher-Reeves updates can be found in Scales (1985).

3.1.6. BFGS quasi-Newton backpropagation (BFGSQB)

Newton's method is an alternative to the conjugate gradient methods for fast optimization. In optimization, quasi-Newton methods (a special case of variable metric methods) are algorithms for finding local maxima and minima of functions. Quasi-Newton methods are based on Newton's method to find the stationary point of a function, where the gradient is 0. Newton's method assumes that the function can be locally approximated as a quadratic in the region around the optimum, and uses the first and second derivatives to find the stationary point. In higher dimensions, Newton's method uses the gradient and the Hessian matrix of second derivatives of the function to be minimized.

In quasi-Newton methods the Hessian matrix does not need to be computed. The Hessian is updated by analyzing successive gradient vectors instead. Quasi-Newton methods are a generalization of the secant method to find the root of the first derivative for multidimensional problems. In multi-dimensions the secant equation is under-determined, and quasi-Newton methods differ in how they constrain the solution, typically by adding a simple low-rank update to the current estimate of the

Hessian. More detail regarding the BFGS quasi-Newton backpropagation can be found in NAG (2012).

3.1.7. One-step secant backpropagation (OSS)

The one step secant (OSS) method is an attempt to bridge the gap between the conjugate gradient algorithms and the quasi-Newton (secant) algorithms. This algorithm does not store the complete Hessian matrix; it assumes that at each iteration, the previous Hessian was the identity matrix. This has the additional advantage that the new search direction can be calculated without computing a matrix inverse.

More detail regarding the OSS can be found in Battiti (1992).

3.2. Bees algorithm

Bees Algorithm is an optimization algorithm inspired by the natural foraging behavior of honey bees to find the optimal solution. Fig. 2 shows the pseudo code for the algorithm in its simplest form. The algorithm requires a number of parameters to be set, namely: number of scout bees (*n*), number of sites selected out of n visited sites (*m*), number of best sites out of m selected sites (*e*), number of bees recruited for best sites (*nep*), number of bees recruited for the other (*m-e*) selected sites (*nsp*), initial size of patches (*ngh*) which includes site and its neighborhood and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitnesses of the sites visited by the scout bees are evaluated in step 2.

1. Initialize the solution population.
2. Evaluate the fitness of the population.
3. While (stopping criterion is not met) //Forming new population.
4. Select sites for neighbourhood search.
5. Recruit bees for selected sites (more bees for the best e sites) and evaluate fitnesses.
6. Select the fittest bee from each site.
7. Assign remaining bees to search randomly and evaluate their fitnesses.
8. End While

Fig. 2. Pseudo code.

In step 4, bees that have the highest fitnesses are chosen as "selected bees" and sites visited by them are chosen for neighborhood search. Then, in steps 5 and 6, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the best e sites. The bees can be chosen directly according to the fitnesses associated with the sites they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Searches in the neighborhood of the best e sites which represent more promising solutions are made more detailed by recruiting more bees to follow them than the other selected bees. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm.

However, in step 6, for each patch only the bee with the highest fitness will be selected to form the next bee population. In nature, there is no such a restriction. This restriction is introduced here to reduce the number of points to be explored. In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts to its new population representatives from each selected patch and other scout bees assigned to conduct random searches (Pham et al., 2006).

4. Simulation results

In this section we evaluate the performance of proposed recognizer. For this purpose we have used the generated patterns (see section 2.1). This dataset contains 600 examples of control charts. For this study, we have used 60% of data for training the classifier and the rest for testing.

4.1. Performance comparison of different training algorithms with row data

First we have evaluated the performance of the recognizer with row data. The training parameters and the configuration of the MLP used in this study are shown in Table 1. The MLP classifiers were tested with various neurons for a single hidden layer and the best networks are selected.

Table 1
MLP architecture and training parameter.

The number of layers	2
Number of output neurons	6
Learning algorithm	Back-propagation with momentum RProp LM SCG CGBFR BFGSQB OSS
The initial weights and basis	Random
Activation function (Hidden layer)	Tangent-sigmoid
Activation function (Output layer)	Linear

Table 2 shows the recognition accuracy (RA) of different systems. In this table, NNHL means the number neurons in the hidden layers. The obtained results are the average of 50 independent runs. As it is depicted in Table 2, using various training algorithms and raw data, the highest accuracy is 98.55%, which is achieved by SCG training algorithms.

Table 2
Recognition accuracy different systems using row data.

Training algorithm	RA (%)	NNHL	Run time (Sec)	Standard
Back-propagation with momentum	95.03	17	5	6.8
RPROP	97.67	20	2	2.3
LM	96.43	24	18	9.5
SCG	98.55	20	2	1.6
CGBFR	95.64	18	3	12.8
BFGSQB	96.34	32	118	6.76
OSS	97.02	19	2	4.9

4.2. Performance comparison of different training algorithms with shape features

Table 3 shows the recognition accuracy of different systems. As it is depicted in Table 3, using various training algorithms and shape feature as input of MLP, the highest accuracy is 99.28%, which is achieved by SCG training algorithms.

Table 3
Recognition accuracy different systems using shape feature.

Training algorithm	RA (%)	NNHL	Run time (Sec)	Standard
Back-propagation with momentum	98.67	19	3	2.9
RPROP	*99.06*	20	*1*	*1.4*
LM	*98.52*	16	*6*	*2.7*
SCG	*99.28*	20	*1*	*0.9*
CGBFR	*98.84*	18	2	7.4
BFGSQB	*98.94*	24	*45*	*3.87*
OSS	*98.54*	16	*1*	*2.8*

4.3. Performance of proposed method (BA+ANN)

In this experiment to evaluating the performance of the proposed method (BA+ANN), ten different runs have been performed. The BA finds the best features to gain the fitness function maximum. Table 4 shows the coefficient values in the BA algorithm. A more detail information about the selected features by proposed algorithm (BA+ANN) is shown in Tables 5- 11. It can be seen that features 2th, 5 th, 7th and 9 th produced the best

accuracy of 99.58%. This result repeated in multiple runs of the program and shows that, these features have very good discrimination ability for our classes.

Table 4
Parameters of BA.

Number of scout bees, n	20
Number of sites selected for neighborhood search, m	8
Number of best "elite" sites out of m selected sites, e	4
Number of bees recruited for best e sites, nep	4
Number of bees recruited for the other (m-e) selected sites, nsp	4
Number of iterations, R	100

Table 5
Selected features for different runs (Back-propagation with momentum).

Run	Features	Size of features	Correct feature	Best fitness
#1	[2, 5, 7, 9]	4	Yes	99.16
#2	[1, 2, 5, 7, 8, 9]	6	No	99.02
#3	[2, 5, 7, 9]	4	Yes	99.16
#4	[2, 5, 7, 9]	4	Yes	99.16
#5	[2, 3, 5, 6, 7, 9]	6	No	98.96
#6	[2, 5, 7, 9]	4	Yes	99.16
#7	[2, 5, 7, 9]	4	Yes	99.16
#8	[2, 5, 7, 9]	4	Yes	99.16
#9	[2, 5, 7, 9]	4	Yes	99.16
#10	[2, 5, 7, 9]	4	Yes	99.16

Table 6
Selected features for different runs (RPROP).

Run	Features	Size of features	Correct feature	Best fitness
#1	[2, 5, 7, 9]	4	Yes	99.16
#2	[1, 2, 5, 6, 7, 8, 9]	7	No	99.04
#3	[2, 5, 7, 9]	4	Yes	99.16
#4	[2, 4, 5, 7, 9]	5	No	99.11
#5	[1, 2, 3, 5, 6, 7, 9]	7	No	99.12
#6	[2, 5, 7, 9]	4	Yes	99.16
#7	[2, 5, 7, 9]	4	Yes	99.16
#8	[2, 5, 7, 9]	4	Yes	99.16
#9	[2, 5, 7, 9]	4	Yes	99.16
#10	[2, 5, 7, 8, 9]	5	No	99.10

Table 7
Selected features for different runs (LM).

Run	Features	Size of features	Correct feature	Best fitness
#1	[2, 5, 7, 9]	4	Yes	99.16
#2	[2, 5, 7, 9]	4	Yes	99.16
#3	[2, 5, 7, 9]	4	Yes	99.16
#4	[2, 5, 7, 9]	4	Yes	99.16
#5	[1, 2, 3, 6, 7, 9]	6	No	99.15
#6	[2, 3, 5, 7, 9]	5	No	99.04
#7	[2, 5, 7, 9]	4	Yes	99.16
#8	[2, 5, 7, 9]	4	Yes	99.16
#9	[2, 5, 7, 9]	4	Yes	99.16
#10	[2, 5, 7, 8, 9]	5	No	99.07

Table 8
Selected features for different runs (SCG).

Run	Features	Size of features	Correct feature	Best fitness
#1	[2, 5, 7, 9]	4	Yes	99.58
#2	[2, 5, 6, 7, 9]	5	No	99.31
#3	[2, 5, 7, 9]	4	Yes	99.58
#4	[2, 5, 7, 9]	4	Yes	99.58
#5	[1, 2, 3, 6, 7, 9]	6	No	99.21
#6	[2, 5, 7, 9]	4	Yes	99.58
#7	[2, 5, 7, 9]	4	Yes	99.58
#8	[2, 4, 5, 7, 9]	5	No	99.37
#9	[2, 5, 7, 9]	4	Yes	99.58
#10	[2, 5, 7, 9]	4	Yes	99.58

Table 9
Selected features for different runs (CGBFR).

Run	Features	Size of features	Correct feature	Best fitness
#1	[1, 2, 5, 7, 9]	5	No	99.01
#2	[2, 5, 6, 7, 9]	5	No	99.04
#3	[2, 5, 7, 9]	4	Yes	99.16
#4	[2, 5, 7, 9]	4	Yes	99.16
#5	[1, 2, 3, 6, 7, 9]	6	No	99.12
#6	[2, 5, 7, 9]	4	Yes	99.16
#7	[2, 5, 7, 9]	4	Yes	99.16
#8	[2, 4, 5, 7, 9]	5	No	99.11
#9	[2, 5, 7, 9]	4	Yes	99.16
#10	[2, 5, 7, 9]	4	Yes	99.16

Table 10
Selected features for different runs (BFGSQB).

Run	Features	Size of features	Correct feature	Best fitness
#1	[2, 5, 7, 9]	4	Yes	99.16
#2	[2, 3, 5, 6, 7, 9]	6	No	99.10
#3	[2, 5, 7, 9]	4	Yes	99.16
#4	[2, 5, 7, 9]	4	Yes	99.16
#5	[1, 2, 3, 6, 7, 9]	6	No	99.11
#6	[2, 5, 7, 9]	4	Yes	99.16
#7	[2, 5, 7, 9]	4	Yes	99.16
#8	[2, 4, 5, 7, 9]	5	No	99.14
#9	[1, 2, 3, 4, 5, 7, 9]	7	No	99.06
#10	[2, 5, 7, 9]	4	Yes	99.16

Table 11
Selected features for different runs (OSS).

Run	Features	Size of features	Correct feature	Best fitness
#1	[2, 5, 7, 8, 9]	5	No	99.11
#2	[2, 3, 5, 6, 7, 9]	6	No	99.14
#3	[2, 5, 7, 9]	4	Yes	99.16
#4	[2, 5, 7, 9]	4	Yes	99.16
#5	[2, 5, 7, 9]	4	Yes	99.16
#6	[2, 5, 7, 9]	4	Yes	99.16
#7	[2, 5, 7, 9]	4	Yes	99.16
#8	[2, 4, 5, 7, 9]	5	No	99.10
#9	[1, 2, 3, 4, 5, 9]	6	No	99.13
#10	[2, 5, 7, 9]	4	Yes	99.16

4.4. Comparison and discussion

For comparison purposes, Table 12 gives the classification accuracies of our method and previous methods applied to the same database. As can be seen from the results, proposed method obtains an excellent classification accuracy.

Table 12
A summary of different classification algorithms together with their reported results used measures of the accuracy.

Ref. no	Year	Classifier	RA (%)
Pharm and Oztemel (1995)	1992	MLP	94.30
Sağıroğlu et al. (2009)	1992	MLP	93.73
Pham and Oztemel (1994)	1994	LVQ	97.70
Pham and Wani (1997)	1997	MLP	99.00
Hassan et al. (2003)	2003	MLP	97.18
Le et al. (2004)	2008	MLP(RSFM)	97.46
Cheng and Ma (2008)	2008	PNN	95.58
Gauri and Chakraborty (2009)	2009	MLP	97.22
Perry et al. (2001)	2011	MLP	99.21
This work	-	MLP (SCG)	99.58

5. Conclusion

With the widespread usage of automatic data acquisition system for computer charting and analysis of manufacturing process data, there exists a need to automate the analysis of process data with little or no human intervention. This study presents methods for improving ANN performance in three aspects: feature extraction, feature selection and ANN training algorithm. The highest level of accuracy obtained by MLP with SCG training algorithm using unprocessed data was 98.55%. The proposed method improves the accuracy up to 99.58% by using selected shape features as the classifier inputs.

References

Battiti R (1992). First-and second-order methods for learning: Between steepest descent and Newton's method. Neural Computation, 4(2): 141-166. https://doi.org/10.1162/neco.1992.4.2.141

Campbell C and Cristianini N (1998). Simple learning algorithms for training support vector machines. University of Bristol, Bristol, UK.

Cheng Z and Ma Y (2008). A research about pattern recognition of control chart using probability neural network. In The 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, IEEE, Guangzhou, China, 2: 140-145. https://doi.org/10.1109/CCCM.2008.168

Evans JR and Lindsay WM (1988). A framework for expert system development in statistical quality control. Computers and Industrial

Engineering, 14(3): 335-343. https://doi.org/10.1016/0360-8352(88)90011-3

Gauri SK and Chakraborty S (2009). Recognition of control chart patterns using improved selection of features. Computers and Industrial Engineering, 56(4): 1577-1588. https://doi.org/10.1016/j.cie.2008.10.006

Hagan MT and Menhaj MB (1994). Training feedforward networks with the Marquardt algorithm. IEEE transactions on Neural Networks, 5(6): 989-993. https://doi.org/10.1109/72.329697

Hassan A, Baksh MSN, Shaharoun AM, and Jamaluddin H (2003). Improved SPC chart pattern recognition using statistical features. International Journal of Production Research, 41(7): 1587-1603. https://doi.org/10.1080/0020754021000049844

Haykin S (1999). Neural network: A comprehensive foundation. 2nd Edition, Prentice-Hall, Upper Saddle River, USA.

Le Q, Gao X, Teng L, and Zhu M (2004). A new ANN model and its application in pattern recognition of control charts. In The Fifth World Congress on Intelligent Control and Automation, IEEE, Hangzhou, China, 2: 1807-1811.

Moller MF (1990). A scaled conjugate gradient algorithm for fast supervised learning PB-339 reprint. Computer Science Department, University of Aaurhus, Aarhus, Denmark. https://doi.org/10.7146/dpb.v19i339.6570

Montgomery DC (2007). Introduction to statistical quality control. John Wiley and Sons, Hoboken, USA.

NAG (2012). Keyword Index: Quasi-Newton. The Numerical Algorithms Group Library Manual, Oxford, UK.

Pacella M, Semeraro Q, and Anglani A (2004). Adaptive resonance theory-based neural algorithms for manufacturing process quality control. International Journal of Production Research, 42(21): 4581-4607. https://doi.org/10.1080/00207540410001715706

Perry MB, Spoerre JK, and Velasco T (2001). Control chart pattern recognition using back propagation artificial neural networks. International Journal of Production Research, 39(15): 3399-3418. https://doi.org/10.1080/00207540110061616

Pham DT and Oztemel E (1994). Control chart pattern recognition using learning vector quantization networks. The International Journal of Production Research, 32(3): 721-729. https://doi.org/10.1080/00207549408956963

Pham DT and Wani MA (1997). Feature-based control chart pattern recognition. International Journal of Production Research, 35(7): 1875-18. https://doi.org/10.1080/002075497194967

Pham DT, Ghanbarzadeh A, Koç E, Otri S, Rahim S, and Zaidi M (2006). The bees algorithm—A novel tool for complex optimisation problems. In: Pham DT, Eldukhri EE and Soroka AJ (Eds.), Intelligent production machines and systems: 454-459. Elsevier Science Ltd., Amsterdam, Netherlands. https://doi.org/10.1016/B978-008045157-2/50081-X

Pharm DT and Oztemel E (1995). Control chart pattern recognition using neural networks. Journal of System Engineering, 2(4): 256-262.

Riedmiller M and Braun H (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In the IEEE International Conference on Neural Networks, San Francisco, CA, USA, 1993: 586-591. https://doi.org/10.1109/ICNN.1993.298623

Sağıroğlu Ş, Beşdok E, and Erler M (2000). Control chart pattern recognition using artificial neural network. Turkish Journal of Electrical Engineering, 8(2): 137-147.

Scales LE (1985). Introduction to non-linear optimization. Springer-Verlag, Berlin, Germany. https://doi.org/10.1007/978-1-349-17741-7

Swift JA and Mize JH (1995). Out-of-control pattern recognition and analysis for quality control charts using LISP-based systems. Computers and Industrial Engineering, 28(1): 81-91. https://doi.org/10.1016/0360-8352(94)00028-L

Wang CH, Kuo W, and Qi H (2007). An integrated approach for process monitoring using wavelet analysis and competitive neural network. International Journal of Production Research, 45(1): 227-244. https://doi.org/10.1080/00207540500442393

Wani MA and Pham DT (1999). Efficient control chart pattern recognition through synergistic and distributed artificial neural networks. Journal of Engineering Manufacture, 213(2): 157-169. https://doi.org/10.1243/0954405991517335

Wani MA and Rashid S (2005). Parallel algorithm for control chart pattern recognition. In the Fourth International Conference on Machine Learning and Applications, IEEE, Los Angeles, USA. https://doi.org/10.1109/ICMLA.2005.51