Computational Research Progress in Applied Science & Engineering
©PEARL publication, 2016
ISSN 2423-4591

CRPASE Vol. 02(02), 71-80, April 2016

Control Chart Pattern Recognition Using Associated Rules and Optimized Classifier

Abdoljalil Addeh*

Faculty of Electrical and Computer Engineering, Babol University of Technology, Babol, Iran

Keywords

Spread,
RBF,
BA,
CCP,
Association rules.

Abstract

Control chart pattern (CCP) recognition is an important issue in statistical process control because unnatural control chart patterns exhibited on control charts can be associated with specific causes that adversely affect the manufacturing processes. In this paper, a hybrid method is introduced to CCPs recognition. In the proposed method, radial basis function neural network (RBFNN) is used as intelligent classifier and shape features are used as effective features. In each pattern recognition problem, the dimension of input data has vital role in classifier performance. Therefore, in the proposed method, the association rule (AR) technique is used to select the effective features and remove the redundant features. Also, in RBF neural network, free parameters such as spread and number of radial basis function have high effect on network performance. Therefore in the proposed bee's algorithm (BA) is used to select the best values of these parameters. The proposed method is tested on real world data and the obtained results show that the proposed method has excellent recognition accuracy.

1. Introduction

Quality process engineering provides the necessary mathematical tools to determine whether a product satisfies some predetermined quality standards and is acceptable for market release. Statistical process control is essential for process improvement, safety assurance, and reliability analysis [1]. Sometimes, quality control is also used for early diagnosis of machinery malfunctions. Sequential production of items that do not comply with the quality standards is an indicator of potential machine failure [2, 3]. In such cases, production process should stop, and the faulty part should be repaired or replaced. Thus, early detection of abnormal quality control patterns is essential for protecting expensive equipment and lowering maintenance costs. Manual quality inspection is a time consuming and tedious task often requiring attention of trained personnel. For this reason, many automated quality process monitoring methods, often termed control chart pattern recognition algorithms, have been proposed in the literature [4, 5].

CCP recognition algorithms provide tools for automated detection of control patterns that demonstrate characteristics that differ significantly compared to the normal process patterns. Over the years, several abnormal patterns have been reported in real industrial problems, each of them reflecting a different underlying fault mechanism. CCPs can exhibit six types of pattern: normal (NR), cyclic (CC), upward trend (UT), downward trend (DT), upward shift (US) and downward shift (DS) [5]. Except for normal patterns, all other patterns indicate that the process being monitored is not functioning correctly and requires adjustment. Figure 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 [6, 7]. 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, ANNs have been widely applied for classifiers. Most researchers [8- 14] 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 [15]. 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. Unfortunately, there is no systematic way to select the topology and

* Corresponding Author:
E-mail address: jalil_addeh@yahoo.com

Received: 27 November 2015; Accepted: 28 March 2016

architecture of a neural network. In general, this has to be found empirically, which can be time consuming.

Some of the researchers used the support vector machine to CCP recognition [16-17]. Using SVMs is the method that is receiving increasing attention, with remarkable results recently. However, 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 [18].

Based on the published papers, there exist some important issues in the design of automatic CCP recognition system which if suitably addressed, lead to the development of more efficient recognizers. One of these issues is the selection of the features [19-23]. The better selection of features usually results in higher retrieval accuracy. In this paper, AR technique is used as feature selection technique. AR learning is a powerful and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness [24].

Another issue is related to the choice of the classification approach to be adopted. The developed system uses RBFNN for recognition. RBFNN represents the promising new generation of information processing systems. RBFNN is good at tasks such as pattern matching and classification, function approximation and optimization [25]. In RBFNN training process, the number of RBFs and their respective spreads have very important role in its performance. Therefore, BA is proposed for selecting appropriate parameters of the classifier. This technique will improve the RBFNN performance.

The details of the proposed method are presented in the following sections. The second section briefly describes the shape feature. In the third section, the needed concepts including AR, RBF neural network and optimization algorithm is presented. The details of proposed method are presented in forth section. The simulation results are presented in fifth section and finally the sixth section concludes the paper.

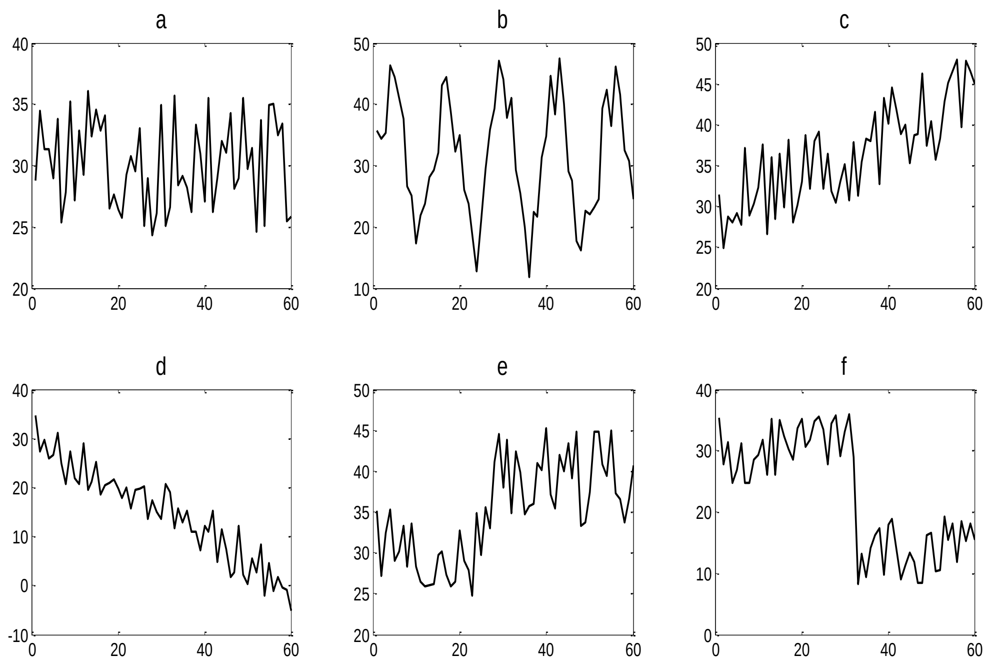

Figure 1. Six various basic patterns of control charts: (a) Normal pattern, (b) Cyclic pattern, (c) Upward trend, and (d) Downward Trend, (e) Upward shift, (f) Downward shift

2. Shape Feature

Features represent the format of the CCPs. As we know, different types of CCP have different properties; therefore finding the suitable features in order to identify them (especially in higher-order and/or non-square cases) is a difficult task. In the signal recognition area, choosing good features not only enables the classifier to distinguish more and higher CCPs, but also helps reduce the complexity of the classifier. In this paper, for the feature extraction module we have used a suitable set of shape features of the CCPs. These features are summarized in Table 1. More details regarding the feature can be found in [21, 22].

3. Needed Concepts

3.1. Feature Selection Algorithm

AR is one of the most popular and useful feature selection techniques that was first introduced in [24]. The goal of AR is to draw out interesting correlations, repeated patterns, associations or casual structures among sets of items in the transaction dataset or other data repositories. The pseudo code of AR is shown in Figure 2.

Table 1. Shape features

No	Ref.	Abbreviation	Description
1	[21]	AB	Absolute slope of the least square (LS) line representing the overall pattern
2	[21]	SB	Sign of slope of the LS line representing the overall pattern
3	[21]	RVE	Ratio between variance of the data points in the observation window and mean sum of squares of errors of the LS line representing the overall pattern
4	[21]	ACLPI	Area between the overall pattern and mean line per interval in terms of SD^2
5	[21]	ACLMLC	Area between the overall pattern and mean line per mean line crossover in terms of SD^2
6	[21]	ALSPI	Area between the overall pattern and LS line per interval in terms of SD^2
7	[21]	ALSLSC	Area between the pattern and least square line per LS line crossover in terms of SD^2
8	[21]	RACLALS	Ratio of area between the pattern and mean line and area between the pattern and LS line
9	[21]	PMLC	Proportion of the number of crossovers to mean line
10	[21]	PLSC	Proportion of the number of crossovers to least square line
11	[21]	PSMLSC	Proportion of the sum of number of crossovers to mean line and LS line
12	[21]	ADIST	Average distance between the consecutive points in terms of SD
13	[21]	AASBP	Average absolute slope of straight lines passing through the consecutive points
14	[21]	AASL	Absolute average slope of the straight lines passing through six pair wise combinations of midpoints of four equal segments
15	[21]	SASL	Sign of average slope of the straight lines passing through six pair wise combinations of midpoints
16	[21]	SRANGE	Range of slopes of straight lines passing through six pair wise combinations of midpoints of four equal segments
17	[21]	AABL	Absolute average slope of the LS lines fitted to six subsets of N/2 data points
18	[21]	SABL	Sign of average slope of the LS lines fitted to six subsets of N/2 data points
19	[21]	BRANGE	Range of slopes of the LS lines fitted to six subsets of N/2 data points
20	[21]	REAE	Ratio of mean sum of squares of errors (MSE) of the LS line fitted to overall data and average MSE of the LS lines fitted to six subsets of N/2 data points
21	[21]	RVAE	Ratio of variance of the observations (SD^2) and average MSE of the LS lines fitted to six subsets of N/2 data points
22	[21]	ADABL	Absolute average slope of the LS lines fitted to four subsets of observations
23	[21]	SDABL	Sign of average slope of the LS lines fitted to four subsets of observations
24	[21]	DBRANGE	Range of slopes of the LS lines fitted to four subsets of observations
25	[21]	AABPE	Absolute average slope of the LS lines fitted to two segments
26	[21]	ABDPE	Absolute slope difference between the LS line representing the overall pattern and the line segments representing the patterns within the two segments
27	[21]	SASDPE	Sum of absolute slope difference between the LS line representing the overall pattern and the individual line segment
28	[21]	SASPE	Sum of absolute slopes of the two line segments
29	[21]	REPEPE	Ratio of MSE of the LS line representing the overall pattern and PMSE of the LS lines fitted to two segments
30	[21]	RVPEPE	Ratio of variance of observations SD^2 and PMSE of the LS lines fitted to two segments
31	[22]	S	The slope of the least-square line representing the pattern.
32	[22]	NC1	The number of mean crossings, i.e. the crossings of the pattern with the mean line
33	[22]	NC2	The number of least-square line crossings.
34	[22]	AS	the average slope of the line segments.
35	[22]	SD	The slope difference between the least-square line and the line segments representing a pattern.
36	[22]	APML	The area between the pattern and the mean line.
37	[22]	APSL	The area between the pattern and its least-square line
38	[22]	ASS	The area between the least-square line and the line segments.
39	[22]	AMEMBER	Cyclic membership

Input:
Database D
Mini Support ε
Mini Confidence ξ
Output:
R_t All association rules
Method:
1- L_1 = large 1-itemsets;
2- for (k=2; $L_{k-1} \neq \varnothing$; k++) do begin
3- C_k = apriori-gen (L_{k-1}); //
generate naw candidates from L_{k-1}
4-for all transactions $T \in D$ do bagin
5- C_t = subset (C_k, T); //
candidates contained in T.
6- for all candidates $C \in C_t$ do
7- Count (C)=Count (C)+1; //
increase support count of C by 1
8- *End*
9- $L_k = \{C \in C_t | \text{count(C)} \geq \varepsilon \times |D|\}$
10-*End*
11-$L_f = \cup_k L_k$;
12-R_t = GenerateRules (L_f, ξ)

Figure 2. Pseudo code of AR

3.2. RBF Neural Network

RBF network is a type of feed-forward neural network composed of three layers, namely the input layer, the hidden layer and the output layer. Each of these layers has different tasks. A general structure of a RBF network is illustrated in Figure 3 [26].

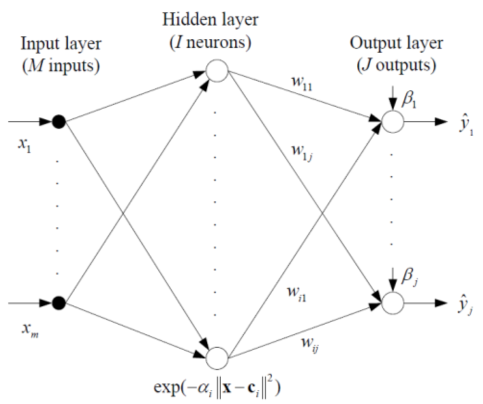

Figure 3. Network architecture of the RBF

3.3. Bee's Algorithm

Bee's Algorithm is an optimization algorithms inspired by the natural foraging behavior of honey bees to find the optimal solution. Fig 4 shows the pseudo code for the algorithm in its simplest form.

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 neighborhood 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

Figure 4. Pseudo code [27]

4. Proposed System

In this paper an intelligent system is proposed for CCP recognition. This system consists of two-stages: feature selection and optimized classification. The main structure of proposed classification system has been shown in Figure 5.

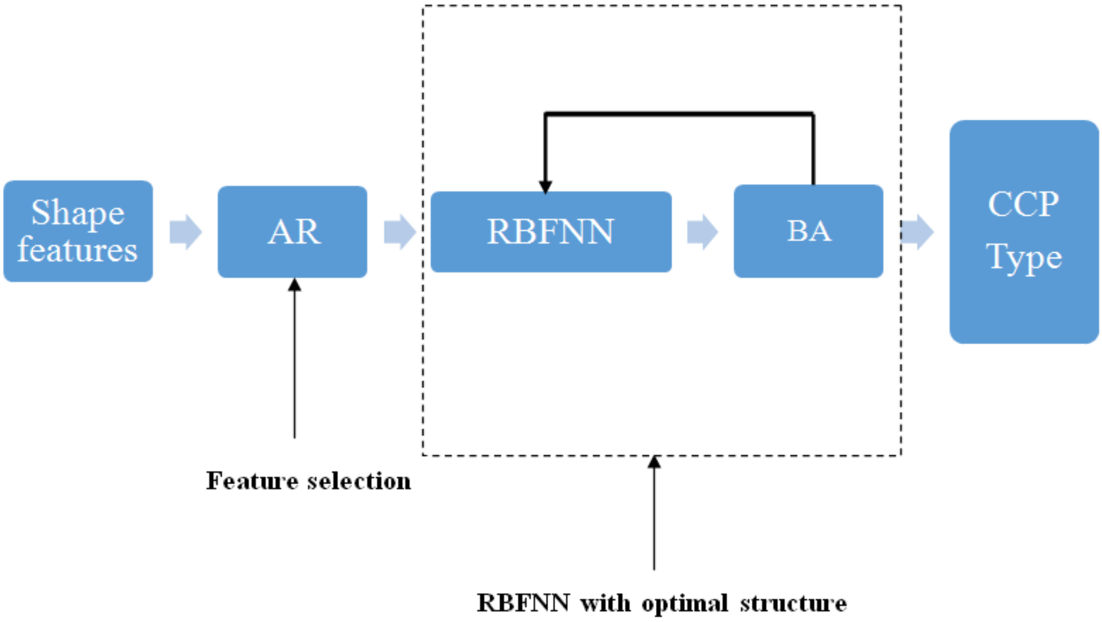

Figure 5. The main structure of proposed system

In the first stage, the input feature vector dimension is reduced and effective features selected by using AR techniques. The AR is used for simplification of models to make them easier to interpret by researchers/ users, shorter training times and enhanced generalization by reducing over fitting.

In the second stage, the selected features are fed into RBFNN to be classified. The output of the network is a scalar function of the input vector and is given by

$$\hat{y}_j = \sum_{i=1}^{I} w_{ij}\ \phi(\| x - c_i \|) + \beta_j \qquad (1)$$

The norm is usually taken to be the Euclidean distance and the radial basis function is also taken to be Gaussian function and defined as follows

$$\varphi(r) = \exp(-\alpha_i . \| x - c_i \|^2) \qquad (2)$$

where *i* is the number of neurons in the hidden layer, J is the number of neurons in the output layer,   is the weight of the ith neuron and jth output,   is the radial basis function,   is the spread parameter of the ith neuron, X is the input data vector,   is the center vector of the ith neuron,   is the bias value of the output jth neuron and   is the network output of the *j*th neuron.

The design procedure of the RBF neural network includes determining the number of neurons in the hidden layer. Then, in order to obtain the desired output of the RBF neural network w, α, c and β parameters might be adjusted properly. Error expression for the RBF network may be defined as follows

$$E^{SSE}(w, \alpha, c, \beta) = \sum_{j=1}^{J} (y_j - \hat{y}_j)^2 \qquad (3)$$

Here y_i indicates the desired output and $\hat{y}_i$ indicates the RBF neural network output. The training procedure of the RBF neural network involves minimizing the error function. In this paper we used back propagation (BP) algorithm as training algorithm.

In RBFNN training, the number of RBFs and their respective spread have very important role in its performance. Therefore, BA is proposed for selecting appropriate parameters of the classifier. The algorithm requires a number of parameters to be set. These parameters and their symbols are listed in Table 2.

Table 2. BA parameters and symbols

Parameter	Symbol
number of scout bees	n
number of sites selected for neighborhood searching (out of n visited sites)	m
number of top-rated (elite) sites among m selected sites	e
number of bees recruited for the best e sites	nep
number of bees recruited for the other selected sites (m-e)	nsp
initial size of neighborhood	ngh

The algorithm starts with an initial population of n scout bees. Each bee represents the number of RBFs and spread value. This initialization process is applied each time new bees are to be created.

Then, the fitness computation process is carried out for each site visited by a bee. In the proposed method we used recognition accuracy (RA) as fitness function. For this purpose we formed confusion matrix. Let $\aleph$ be either a countable set, or a complete separable metric space equipped with the standard Borel $\sigma - a\lg ebra$ of measurable sets. Let $X \epsilon \aleph$ and $Y \epsilon \{0,1\}$ represent input and output matrixes respectively. Further, let Θ epresent the set of all classifiers $\Theta = \{\theta: \aleph \mapsto [0,1]\}$. We assume the existence of a fixed unknown distribution $\mathbb{p}$. Samples $(X, Y) \sim \mathbb{p}$. Define the quantities: $\pi = \mathbb{p}\,(Y = 1)$ and $\gamma(\theta) = \mathbb{p}\,(\theta = 1)$.

The components of the confusion matrix are the fundamental population quantities for binary classification. Table 3 shows the confusion matrix. The entries in the confusion matrix have the following meaning in the context of our study:

- TP is the number of correct predictions that an instance is positive.
- FN is the number of incorrect predictions that an instance is negative,
- FP is the number of incorrect of predictions that an instance positive
- TN is the number of correct predictions that an instance is negative.

These components are defined as follow

$$TP(\theta,\mathbb{P}) = \mathbb{P}(Y=1\ ,\ \theta=1) \tag{4}$$

$$FP(\theta,\mathbb{P}) = \mathbb{P}(Y=0\ ,\ \theta=1) \tag{5}$$

$$FN(\theta,\mathbb{P}) = \mathbb{P}(Y=1\ ,\ \theta=0) \tag{6}$$

$$TN(\theta,\mathbb{P}) = \mathbb{P}(Y=0\ ,\ \theta=0) \tag{7}$$

These quantities may be further decomposed as

$$FP(\theta,\mathbb{P}) = \gamma(\theta) - TP(\theta) \tag{8}$$

$$FN(\theta,\mathbb{P}) = \pi - TP(\theta) \tag{9}$$

$$TN(\theta,\mathbb{P}) = 1-\gamma(\theta) - \pi + TP(\theta) \tag{10}$$

Table 3. Confusion matrix

Data class	Classified as positive	Classified as negative
Positive	TP	FN
Negative	FP	TN

As mentioned, we used RA as fitness function. The RA is the proportion of the total number of predictions that were correct. It is determined using Eq. (11) as

$$\text{Fitness function:RA}=\frac{\text{TP+TN}}{\text{TP+FN+FP+TN}} \tag{11}$$

In the next step, the m sites with the highest fitnesses are designated as "selected sites" and chosen for neighborhood search. Then, the algorithm conducts searches around the selected sites, assigning more bees to search in the vicinity of the best e sites. Selection of the best sites can be made directly according to the fitnesses associated with them. Searches in the neighborhood of the best e sites – those which represent the most promising solutions - are made more detailed. This is done by recruiting more bees for the best e sites than for the other selected sites. The size $a = \{a_1, \dots, a_n\}$ of the flower patches is initially set to a large value. For each variable a_i , it is set as follows

$$a_i = ngh(t)\times(\max_i - \min_i)$$
$$ngh(t+1) = 0.8\times ngh(t) \tag{12}$$

where $\min_i$ and $\max_i$ are the ith optimization variable boundaries.

In the next step, for each patch, only the bee that has found the site with the highest fitness (the "fittest" bee in the patch) will be selected to form part of the next bee population. Then, the remaining bees in the population are assigned randomly around the search space to scout for new potential solutions. At the end of per iteration, the colony will have two parts to its new population: representatives from the selected patches, and scout bees assigned to conduct random searches. These steps are repeated until a stopping criterion is met. Figure 6 shows the pseudo code of proposed method.

1. Shape features
2. Select the best features using AR.
3. Set bee's algorithm parameters and $i=0$.
4. Generate the initial population of the bees randomly in search space.
No. RBFs is an integer optimization variable in [1- 400], and spread is a continuous optimization variable in [0.1- 10].
5. Evaluate the fitness of the initial population based on Eq. (11).
6. Sort the initial population based on fitness results.
7. While $i \le$ Max Iteration
8. $i=i+1$
9. Select the elite patches and non-elite best patches for neighborhood search.
10. Recruit the forager bees to the elite patches and non-elite best patches.
11. Evaluate the fitness of each patch (Eq. (11)).
12. Sort the results based on their fitness.
13. Allocate the rest of the bees for global search to the non-best locations.
14. Evaluate the fitness of non-best patches.
15. Sort the overall results based on their fitness.
16. Run the algorithm until termination criteria met.

Figure 6. Pseudo code of proposed method

5. Results

In this section the results of proposed method are presented. For this purpose we have used the practical and real world data [28]. This dataset contains 600 examples of control charts. For this study, we have used 50% of data for training the classifier and the rest for testing. The easiest way to assess the performance rate is to choose a test set independent of the training set and validation set to classify its examples, count the examples that have been correctly classified and divide by the size of the test set. The proportion of test-set examples that are classified correctly to the total samples, estimates the performance of recognizer for each pattern. In order to achieve the recognition accuracy of system, one needs to compute the average value of the performances of the CCPs.

The computational experiments for this section were performed on Intel core 2 Duo with 4 GB RAM using ASUS computer. The proposed algorithm was implemented on

MATLAB 2014 package by using the neural networks toolbox. Furthermore traditional RBF neural network was also implemented to compare with the proposed algorithm. In the traditional RBF neural network, the number of RBFs and value of spread was selected by trial and error.

5.1. RBFNN Performance Without Optimization

In this section we used all shape features as input of network. In this scenario, number of RBFs was selected equal to training data and the value of spread is tested for various values. The obtained results are listed in Table 4. It may be seen that the network with 300 RBFs and spread equal to 3.5, leads to best recognition accuracy equal 89.63%. It can be seen that there is no linear relation between the value of spread and performance of RBFNN. Therefore the value of spread should be obtained through trial and error and based on extensive simulations. This manner of network topology selection is very time consuming. Figure 7 shows the effect of spread on RBFNN's performance. In this figure, each node indicates the accuracy of RBF neural network with 300 RBFs and assigned spread valued in the Figure 7.

Table 4. The performance of RBFNN without optimization and selected features (No. of RBFs=300)

Spread	RA (%)	Spread	RA (%)
0.5	88.52	5.5	88.22
1	89.23	6	85.21
1.5	87.44	6.5	89.15
2	85.59	7	87.29
2.5	87.29	7.5	87.38
3	88.25	8	86.29
3.5	**89.63**	8.5	88.04
4	89.18	9	88.23
4.5	85.10	9.5	88.12
5	87.26	10	83.78

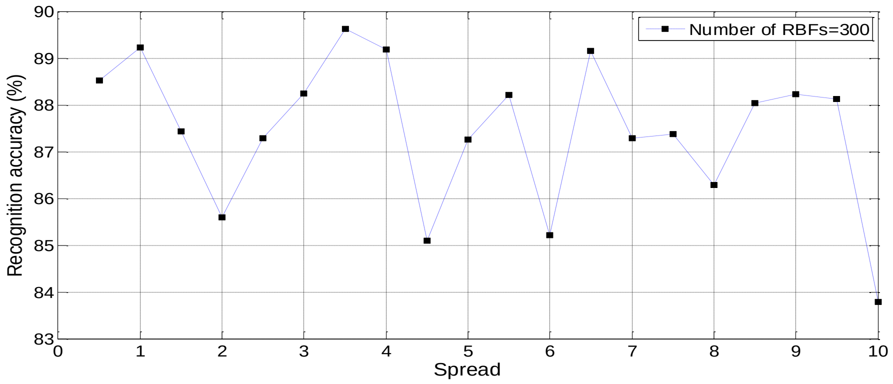

Figure 7. The effect of spread on RBFNN performance (300 RBFs)

In the next experiment, the RBFs number is selected less than training data. For more investigations, various numbers of RBFs and various spread values are considered. In Figure 8, the RBFNN with 50, 100 and 300 RBFs were built and the value of spread is changed from 0.5 to 10. It may be seen that the performance of the network with 50 RBFs is better than the network with 100 and 300 RBFs. The obtained results show that the changing of RBFs number improves the recognition accuracy significantly.

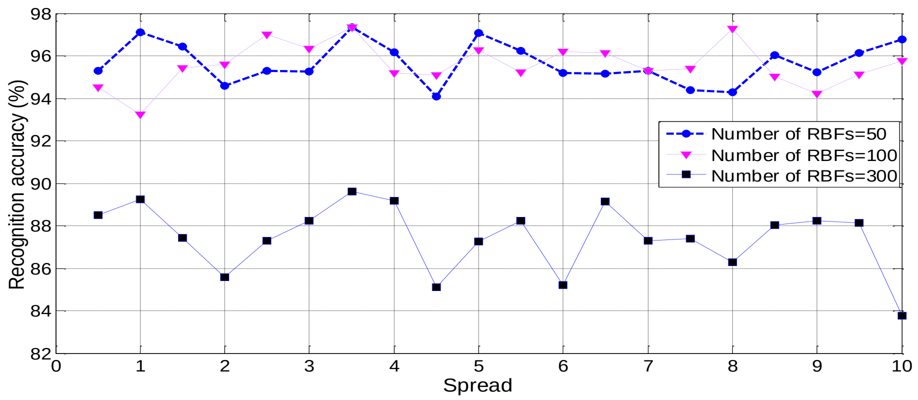

Figure 8. The effect of spread on RBFNN performance with various RBF numbers

In previous experiments the effect of spread on RBFNN performance was investigated and it was seen that there were no linear relation between the spread value and network performance. In another test, the effect of RBF numbers has been investigated. For this purpose, the number of RBFs was changed from one to 300 and the value of spread is fixed. In Figure 9, the value of spread was hold on 3.5 and the number of RBFs is changed from 1 to 300

(300 is training data number). It can be seen that the performance of RBFNN has been highly dependent on the number of RBFs.

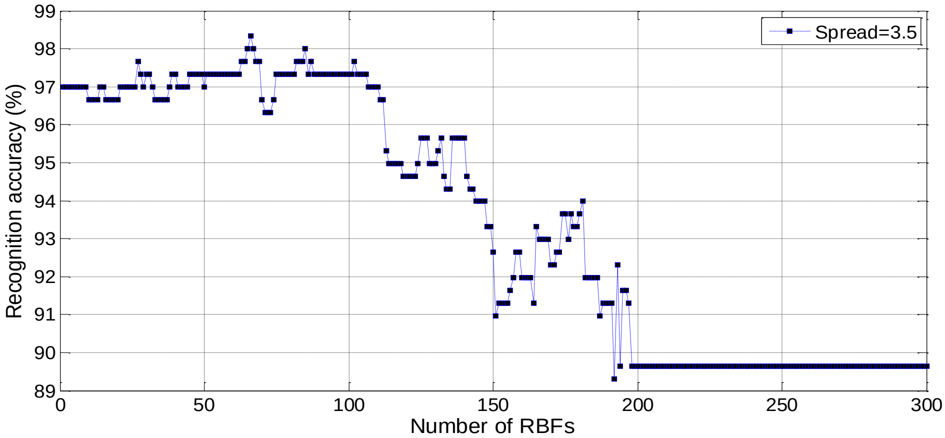

Figure 9. The effect of RBF number on network performance (spread= 3.5)

5.2. The Performance of Proposed Method

Feature selection is the key for pattern recognition and classification. The AR can enhance the classification accuracy of CCP recognition. Therefore in this study we used AR as feature selection technique. Using the AR, the 6th, 18th, 20th, 24th, 29th, 30th and 35th features were selected as effective features and the other features were removed.

In the proposed method, RBF numbers and spread value were selected by BA. The obtained results have been listed in Table 5. It may be seen that the proposed system has recognized the CCPs with 99.61% accuracy which is a significant increase in recognition accuracy. Improvement in recognition accuracy by proposed method shows the feature selection and optimization importance. Furthermore, for comparison, the performance of optimized classifier with row data and all shape features are presented in Table 5.

In order to indicate the details of the recognition for each pattern, the confusion matrix of the recognizer is shown by Table 6. As we know, the values in the diagonal of confusion matrix show the correct performance of recognizer for each pattern. In other words, these value show that how many of considered pattern are recognized correctly by the system. The other values show the mistakes of system. For example, look at the fourth row of confusion matrix. The value of 97.7 % shows the percentage of correct recognition of downward trend pattern and the value of 2.3 % shows that this type of pattern is wrongly recognized with downward shift pattern. In order to achieve the recognition accuracy of system, it is needed to compute the average value of that appears in diagonal.

Table 5. Performance of optimized classifier

Input	Input size	Parameters	RA (%)
Row data	60	Spread= 4.764, No. RBFs= 128	97.56
All shape feature	39	Spread= 2.529, No. RBFs= 104	99.24
Selected features	**7**	**Spread= 1.163, No. RBFs= 62**	**99.61**

Table 6. Confusion matrix for best result (99.61%)

	NR	CC	UT	DT	US	DS
NR	100	0	0	0	0	0
CC	0	100	0	0	0	0
UT	0	0	100	0	0	0
DT	0	0	0	97.7	0	2.3
US	0	0	0	0	100	0
DS	0	0	0	0	0	100

5.3. Comparing Performances of the Feature Selection Techniques

The performance of the proposed feature selection method has been compared with other feature selection methods. In this respect, filter method [29], Sequential backward selection (SBS) [30], Sequential forward selection (SFS) [31], sequential floating forward selection (SFFS) [32] and sequential floating backward selection (SFBS) [33] are considered.

Filter methods are fast but lack robustness against interactions among features and feature redundancy. In addition, it is not clear how to determine the cut-off point for rankings to select only truly important features and exclude noise. SFS and SBS can easily be trapped into local minima. The problem with wrapper method is its single-track search.

Recent empirical studies demonstrate that SFFS is not superior to SFS and SFBS is not feasible for feature sets of more than about 100 features [34]. The problem with sequentially adding or removing features is that the utility of an individual feature is often not apparent on its own, but only in combinations including just the right other features.

It can be seen from Table 7 that the proposed method has better recognition accuracy than other classifiers. In all cases the RBF neural network parameters are optimized by BA.

Table 7. Comparison the performance of different feature selection methods with optimized classifier

Feature selection algorithm	Selected features	Parameters	RA (%)
SFS	1, 3, 5, 10, 18, 24, 29, 31, 33, 36, 39	Spread= 1.829 No. RBFs= 88	99.10
SBS	2, 3, 5, 9, 10, 20, 21, 29, 31, 33, 35, 37	Spread= 1.054 No. RBFs= 92	99.22
SFFS	3, 5, 9, 10, 19, 20, 24, 39	Spread= 2.764 No. RBFs= 114	99.18
SFBS	1, 3, 4, 9, 10, 16, 20, 33, 34, 39	Spread= 1.655 No. RBFs= 93	99.36
AR	6, 18, 20, 24, 29, 30, 35	**Spread= 1.163** **No. RBFs= 62**	**99.61**

5.4. Comparison and Discussion

For comparison purposes, Table 8 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 excellent classification accuracy.

Table 8. A summary of different classification algorithms together with their reported results used measures of the accuracy

Ref. no	Year	Classifier	RA (%)
[11]	1992	MLP	94.30
[13]	1992	MLP	93.73
[14]	1994	LVQ	97.70
[22]	1997	MLP	99.00
[35]	1999	MLP(SPA)	96.38
[23]	2003	MLP	97.18
[36]	2006	MLP	97.20
[10]	2008	MLP(RSFM)	97.46
[12]	2008	PNN	95.58
[21]	2009	MLP	97.22
This work	-	**Optimized RBF**	**99.61**

6. Conclusions

In this paper, an intelligent system proposed for CCP recognition based on shape features. In the proposed system, AR applied to remove the redundant features and improve the recognition accuracy. Also bee's optimization algorithm is used to find the optimal structure of radial basis function neural network. To demonstrate the advantages of proposed system some experiments done.

In the first experiment, the number of RBFs fixed and the value of spread changed from 0.5 to 10. The simulation results showed that the performance of RBFNN is highly dependent to spread value. The important notice was that there are no linear relations between the performance of RBFNN and spread value. Therefore in RBFNN applications, the value of this parameter must be selected based on trial and error.

In the next experiment, the value of spread fixed and the number of RBFs changed from 1 to 300. The simulation results show that the performance of RBFNN is highly dependent to RBFs number. Similar to previous case, it is observed that there are no linear relation between the RBF numbers and network performance. This experiment reveals that the network with large number of RBFs reduce the generalization capability of network and therefore reduce the recognition accuracy.

When the determination of the optimal neural network structure proved, we used BA algorithm to find the best structure of network. Also we used AR to find the best features and remove the redundant features from original dataset. The best classifier will perform poorly if the features are not chosen well. A feature selection algorithm should reduce the feature vector to a lower dimension, which contains most of the useful information from the original vector. The RBFNN with optimal structure and selected feature classified the CCPs with 99.61% accuracy. The proposed system has high recognition accuracy and it is recommended for CCP recognition.

References

[1] W. Yang, W. Zhou, W. Liao, Y. Guo, Identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extreme learning machines, Neurocomputing 147 (2015) 260–270.

[2] S. Haghtalab, P. Xanthopoulos, K. Madani, A robust unsupervised consensus control chart pattern recognition

Framework, Expert Systems with Applications 42 (2015) 6767–6776.

[3] P. Veiga, L. Mendes, L. Lourenço, A retrospective view of statistical quality control research and identification of emerging trends: A bibliometric analysis, Quality & Quantity 50 (2016) 673–692.

[4] A. Nikpey, S. Mirzaei, M. Pourmandi, Identification of the Control Chart Patterns Using the Optimized Adaptive Neuro-Fuzzy Inference System, I.J. Modern Education and Computer Science 7 (2014) 16–24.

[5] B.C. Franco, G. Celano, P. Castagliola, A. Fernando, B. Costa, Economic design of Shewhart control charts for monitoring autocorrelated data with skip sampling strategies, International Journal of Production Economics 151 (2014) 121–130.

[6] J.A. Swift, J.H. Mize, Out-of-control pattern recognition and analysis for quality control charts using lisp-based systems, Computers and Industrial Engineering 28 (1995) 81–91.

[7] J. R. Evans, W.M.A. Lindsay, framework for expert system development in statistical quality control, Computers and Industrial Engineering 14 (1988) 335–343.

[8] M.H.A. Awadalla, M. Abdellatif Sadek, Spiking neural network-based control chart pattern recognition, Alexandria Engineering Journal 51 (2012) 27–35.

[9] T.T. El-Midany, M.A. El-Baz, M.S. Abd-Elwahed, A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks, Expert Systems with Applications 37 (2010) 1035–1042

[10] Q. Le, X. Goal, L. Teng, M. Zhu, A new ANN model and its application in pattern recognition of control charts. In: Proceedings of the Fifth World Congress on Intelligent Control and Automation (WCICA 2004), vol. 2, June 15–19 (2004).

[11] D.T. Pham, E. Oztemel, Control chart pattern recognition using neural networks, Journal of Systems Engineering 2 (1992) 256–262.

[12] Z. Cheng, Y. Ma, A research about pattern recognition of control chart using probability neural network. In: Proceedings of 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, vol. 2, August 3–4 (2008).

[13] S. Sagiroujlu, E. Besdoc, M. Erler, Contro chart pattern recognition using artificial neural networks, Turkish Journal of Electrical Engineering 8 (2000) 137–147.

[14] D.T. Pham, E. Oztemel, Control chart pattern recognition using linear vector quantization networks, International Journal of Production Research 32 (1994) 721–729.

[15] C. H. Wang, W. Kuo, H. Qi, An integrated approach for process monitoring using wavelet analysis and competitive neural network, International Journal of Production Research 45 (2007) 227–244 .

[16] P. Xanthopoulos, T. Razzaghi, A weighted support vector machine method for control chart pattern Recognition, Computers & Industrial Engineering 70 (2014) 134–149.

[17] S. Du, D. Huang, J. Lv, Recognition of concurrent control chart patterns using wavelet transform decomposition and multiclass support vector machines, Computers & Industrial Engineering 66 (2013) 683–695.

[18] C. Campbell, N. Cristianini, Simple learning algorithms for training support vector machines, CiteSeerXbeta, 1998.

[19] M. Pacella, Q. Semeraro, A. Anglani, Adaptive resonance theory-based neural algorithms for manufacturing process quality control, International Journal of Production Research 43 (2004) 4581–4607.

[20] M.A. Wani, S. Rashid, Parallel algorithm for control chart pattern recognition. In: Proceedings of the Fourth International Conference on Machine Learning and Applications (ICMLA'05), December 15–17 (2005).

[21] S. Gauri, S. Chakraborty, Recognition of control chart patterns using improved selection of features, Computers & Industrial Engineering 56 (2009) 1577–1588.

[22] D.T. Pham, M. A. Wani, Feature-based control chart pattern recognition, International Journal of Production Research 35 (1997) 1875–1890.

[23] A. Hassan, M. Shariff Nabi Bakhsh, A.M. Shaharoun, H. Jamaluddin, Improved SPC chart pattern recognition using statistical features, International Journal of Production Research 41 (2003) 1587–1603.

[24] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, D.C., May (1993).

[25] Y. Fang, J. Fei, K. Ma, Model reference adaptive sliding mode control using RBF neural network for active power filter, International Journal of Electrical Power & Energy Systems 73 (2015) 249–258.

[26] S. Haykin, Neural networks: a comprehensive foundation. New York, MacMillan, 1999.

[27] D.T. Pham, M. Castellani, The Bees Algorithm: modeling foraging behavior to solve continuous optimization problems, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 223 (2009) 2919–2938,.

[28] http://archive.ics.uci.edu/ml/databases/syntheticcontrol/synthetic control. data.html

[29] J. Biesiada, W. Duch, Feature selection for high-dimensional data–a Pearson redundancy based filter, Advances in Soft Computing 45 (2008) 242–249.

[30] S.F. Cotter, K. Kreutz-Delgado, B.D. Rao, Back ward sequential elimination for sparse vector selection, Signal Processing 81 (2001)1849–1864.

[31] S. Colak, C. Isik, Feature subset selection for blood pressure classification using orthogonal forward selection. In: Proceedings of 2003 IEEE 29th Annual Bioengineering Conference, March 22–23 (2003).

[32] P. Pudil, J. Novovicov, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (1994) 1119–1125.

[33] M. Bensch, M. Schröder, M. Bogdan, W. Rosenstiel, P. Czerner, R. Montino, G. Soberger, P. Linke, R. Schmidt, Feature selection for high-dimensional industrial data. In: Proceedings of European Symposium on Artificial Neural Networks (ESANN'2005), Bruges (Belgim), April 27–29 (2005).

[34] H.T. Ng, W.B. Goh, K.L. Low, Feature selection, perceptron learning, and a susability case study for text categorization. In: Proceedings of 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, July 27–31 (1997).

[35] R.S. Guh, J.D.T. Tannock, A neural network approach to characterize pattern parameters in process control charts, Journal of Intelligent Manufacturing 10 (1999) 449–462.

[36] S. Gauri, S. Chakraborty, Feature-based recognition of control chart patterns, Computers & Industrial Engineering 51 (2006) 726–742.