Research article

Control chart pattern recognition using RBF neural network with new training algorithm and practical features

Abdoljalil Addeh\*, Aminollah Khormali\*, Noorbakhsh Amiri Golilzar\*

\* Faculty of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran
\* Faculty of Electrical Engineering, K. N. Toosi University of Technology, Iran
\* Department of Electrical and Electronic Engineering, Eastern Mediterranean University, Cyprus

ARTICLE INFO

Keywords:
CCP
RBFNN
Shape features
Statistic features
AR

ABSTRACT

The control chart patterns are the most commonly used statistical process control (SPC) tools to monitor process changes. When a control chart produces an out-of-control signal, this means that the process has been changed. In this study, a new method based on optimized radial basis function neural network (RBFNN) is proposed for control chart patterns (CCPs) recognition. The proposed method consists of four main modules: feature extraction, feature selection, classification and learning algorithm. In the feature extraction module, shape and statistical features are used. Recently, various shape and statistical features have been presented for the CCPs recognition. In the feature selection module, the association rules (AR) method has been employed to select the best set of the shape and statistical features. In the classifier section, RBFNN is used and finally, in RBFNN, learning algorithm has a high impact on the network performance. Therefore, a new learning algorithm based on the bees algorithm has been used in the learning module. Most studies have considered only six patterns: Normal, Cyclic, Increasing Trend, Decreasing Trend, Upward Shift and Downward Shift. Since three patterns namely Normal, Stratification, and Systematic are very similar to each other and distinguishing them is very difficult, in most studies Stratification and Systematic have not been considered. Regarding to the continuous monitoring and control over the production process and the exact type detection of the problem encountered during the production process, eight patterns have been investigated in this study. The proposed method is tested on a dataset containing 1600 samples (200 samples from each pattern) and the results showed that the proposed method has a very good performance.

1. Introduction

The control chart as a basic tool of the statistical process control plays an important role in the quality control of various production processes. Based on the principle of statistical hypothesis testing, the control chart is used to record and monitor the volatility of some key quality characteristics [1,2]. With the in-depth study, some researchers found out that it is difficult to determine whether a control chart is going to deteriorate rapidly by observing the fluctuations. After a long period of research and review, most control chart patterns (CCPs) can be classified as normal patterns (NOR) and seven abnormal patterns including Stratification (STR), Systematic (SYS), Cyclic (CYC), Increasing Trend (IT), Decreasing Trend (DT), Upward Shift (US) and Downward Shift (DS) [3]. The CCPs are shown in Fig. 1 and their assignable causes are listed below [2,3]. Utilizing the formulas presented in Ref. [3], 200 samples are generated for each pattern.

(1) Trend patterns: A trend can be defined as a continuous movement in either positive or negative direction. Possible causes are tool wear, operator fatigue and equipment deterioration.
(2) Shift patterns: A shift can be defined as a sudden change above or below the average of the process. This change may be caused by an alternation in process setting, replacement of raw materials, minor failure of machine parts and introduction of new workers.
(3) Cyclic patterns: Cyclic behaviors can be observed by a serial of peaks and troughs occurred in the process. Typical causes are the periodic rotation of operators, systematic environmental changes or fluctuation in the production equipment.
(4) Systematic patterns: The characteristic of systematic patterns is that a point-to-point fluctuation can be occurred systematically. It means that a low point is always followed by a high point and vice versa. Possible causes include differences between the test sets and also between the production lines where the product is sampled in the rotation.
(5) Stratification pattern: In statistical process control, “stratification” refers to a situation in which the data being charted does not represent a single process but comes from two or more different processes.

In recent years, machine learning algorithms such as support vector machine (SVM), fuzzy systems and artificial neural networks (ANNs) have been implemented for the efficient CCPs recognition. Some of the researchers used the SVM for CCPs recognition [4–7]. Recently, using SVMs is attracting increasing attention among the researchers because of its performance in obtaining remarkable results. However, the accuracy of an SVM is dependent on the choice of the kernel function and the parameters (e.g. cost parameter, slack variables and the margin of the hyper plane.) Failure to find the optimal parameters for an SVM model affects its recognition accuracy [8]. Another drawback of the SVM is its computational cost [9].

In Refs. [10–12] fuzzy approach was used to recognize CCPs. The fuzzy systems have a strong inference engine containing fuzzy rules that can detect hidden relations which is unrecognized by the human expert. On the other hand, the error rate of fuzzy system is high and requires a cyclic membership, the average slope of the line segments, slope difference and three different measures for the area. The scheme was aimed at improving the performance of the pattern recognizer by presenting a smaller input vector (features). Gauri and Chakraborty [19] also presented a set of seven most useful features that were selected from a large number of potentially useful features using a CART-based systematic approach. Based on these selected features, eight most commonly observed CCPs were recognized using heuristic and ANN techniques.

Chen et al. [20] presented a hybrid approach by integrating wavelet method and neural network for on-line recognition of concurrent CCPs. In the hybrid system, concurrent CCPs are firstly preprocessed by a wavelet transform to decompose the concurrent patterns into different levels or patterns. Then the corresponding features are fed into backpropagation ANN classifiers for pattern recognition. The wavelet transform was studied as the input of SVM for CCP recognition in Ref. [21]. Hassan et al. [22] conducted an experimental study to use Backpropagation Neural Network (BPN) for identifying six types of basic SPC patterns, in which they compared the performances of two BPN recognizers using statistical features and raw data as input feature respectively. The results indicated that the BPN using statistical features as input vectors had better performance than the other BPN using raw data as input vectors.

Based on the published papers about the automatic CCPs recognition, there are some facts which should be considered during the design of recognizer. One of these issues is the feature extraction. In this paper, the statistical and shape features are applied for obtaining the compact set of features which capture the prominent characteristics of the CCPs in a relatively small number of the components. These features are presented in Section 2. Another issue is related to the choice of the classification approach to be adopted. The developed method uses radial basis function neural network (RBFNN) for the recognition task. RBFNN is good at tasks such as pattern recognition and classification [23,24], function approximation [25] and nonlinear time series prediction [26].

The CCPs recognition using machine learning methods basically...
consists of two phases: training and testing. In the training stage, weights are calculated according to the chosen learning algorithm. The issue of the learning algorithm and its speed is very important for the RBFNN. Different methods have been used to train RBFNN. This paper proposed to use the Bees-Inspired RBFNN introduced in Ref. [27] as a learning algorithm.

The rest of the paper is organized as follows. Section 2 explains the feature extraction. Section 3 presents the needed concepts including classifier and feature selection method. Section 4 presents the proposed method. Section 5 shows some simulation results and finally section 6 concludes the paper.

2. Feature extraction

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 is a difficult task. In the pattern recognition area, choosing good features not only enables the classifier to distinguish more and higher CCPs but also helps to reduce the complexity of the classifier. In recent years, multiple shape and statistical features have been presented for recognition of CCPs. The remains of this section present these features briefly.

2.1. 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 eight types of CCP considered in this work have different forms which can be characterized by a number of shape features. In Ref. [18], the authors have introduced several shape features for discrimination of the CCPs. These features are as follows:

(1) S: the slope of the least-square line representing the pattern. If your data shows a linear relationship between the X and Y variables, you need to find the best line that fits this linear relationship. That line is called a Regression Line ($\hat{y} = a + bx$). The Least Squares Line is the line that makes the vertical distance from the data points to the regression line as small as possible. The magnitude of S of this line for normal, stratification, systematic and cyclic patterns is approximately zero while 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. The value of this feature for training data is shown in Fig. 2

(2) NC1: the number of mean crossings, i.e. the crossings of the pattern with the mean line. NC1 is small for cyclic, shift and trend patterns. It is highest for normal, stratification and systematic patterns. This feature differentiates normal, stratification and systematic patterns from cyclic, shift and trend patterns. The value of this feature for training data is shown in Fig. 2

(3) NC2: the number of least-square line crossings. NC2 is highest for normal, stratification, systematic and trend patterns and lowest for shift and cyclic patterns. Thus it can be used for separation of natural and trend patterns from others. The value of this feature for training data is shown in Fig. 2

(4) 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 normal, stratification, systematic, cyclic and shift patterns. This feature, therefore, differentiates trend patterns from other patterns. The value of this feature for training data is shown in Fig. 2

(5) 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, stratification, systematic, 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, stratification, systematic, cyclic and trend patterns. This feature, therefore, differentiates a shift pattern from other patterns. The value of this feature for training data is shown in Fig. 3

(6) APML: the area between the pattern and the mean line. The APML is lowest for stratification pattern. Thus, this feature differentiates between stratification and other patterns. The value of this feature for training data is shown in Fig. 3

(7) APSL: the area between the pattern and its least-square line. Cyclic and shift patterns have a higher APSL value than normal, stratification, systematic and trend patterns and therefore the APSL can be used to differentiate cyclic and shift patterns from normal and trend patterns. The value of this feature for training data is shown in Fig. 3

(8) 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. The value of this feature for training data is shown in Fig. 3

2.2. Statistical features

In Ref. [22] statistical features are employed for CCPs recognition. The results of this study reveal that mean, standard deviation, skewness and kurtosis statistical features can lead to higher recognition accuracy. Therefore, these features are considered in the proposed method.

The mean value is approximately equal for a normal, stratification, systematic and cyclic pattern and different for the remaining patterns. This feature thus differentiates normal, stratification, systematic and cyclic patterns from other patterns.

The value of standard deviation for a stratification pattern will be lower than other patterns. Therefore, this feature differentiates stratification pattern from other patterns. Skewness provides the information regarding the degree of asymmetry. Additionally, kurtosis measures the relative peakedness or flatness of its distribution [22]. The value of these feature for training data are shown in Fig. 4.

3. Needed concepts

3.1. Classifier

RBFNN is one of the most important ANN paradigms in machine learning. It is a feed forward network with a single layer of hidden units, called radial basis functions (RBFs). RBF outputs show the maximum value at its center point and decrease its output value as the input leaves the center. Typically, the Gaussian function is used for the activation function [27].

The RBF network is constructed with three layers: input layer, hidden layer and output layer as shown by Fig. 5.

In input layer, the number of neurons is the same with the number of input dimension. The neurons of input layer will transmit data to the hidden layer and calculates a value of the RBFs received from the input layer. These values will be transmitted to the output layer which calculates the values of linear sum of the hidden neuron. In this study, the Gaussian function is used as RBF. Let $h_i(\cdot)$ be the $i\text{th}$ radial basis function. The output of each radial basis function is:

$$H_j = h_i(\|x - c_j\|, \sigma_j), \quad j = 1, 2, ..., m$$

Here, $x = (x_1, x_2, ..., x_d)^T$ is the input vector, $\cdot$ is a norm, usually Euclidean, defined on the input space, $c_j = (c_{j1}, c_{j2}, ..., c_{jd})^T$ and $\sigma_j^2$ are the $j\text{th}$ center vector and the width parameter, respectively. The output of RBF network $y$ which is the linear sum of radial basis function, is given as follows:

$$H_j = h_i(\|x - c_j\|, \sigma_j), \quad j = 1, 2, ..., m$$

Here, $x = (x_1, x_2, ..., x_d)^T$ is the input vector, $\cdot$ is a norm, usually Euclidean, defined on the input space, $c_j = (c_{j1}, c_{j2}, ..., c_{jd})^T$ and $\sigma_j^2$ are the $j\text{th}$ center vector and the width parameter, respectively. The output of RBF network $y$ which is the linear sum of radial basis function, is given as follows:
\[ y = w_i^T H = \sum_{j=1}^{m} w_{ij} H_j, \quad i = 1, 2, \ldots, o \]  \hspace{1cm} (2)

where \( y \) is the output of the RBF network, \( w_i = [w_{i1}, w_{i2}, \ldots, w_{io}]^T \), \( i = 1, 2, \ldots, o \) are the network weight vectors for each output neuron \( i \), \( H = [H_1, H_2, \ldots, H_m]^T \) is the vector of basis functions and \( o \) is the number of network output units \([27]\).

To construct RBF network, the number of the hidden layer neuron \( m \) must be set. Moreover, the centers \( c_j \), the widths \( \sigma_j \) and the weights \( w_i \) must be estimated. In RBF typical learning, the network structure will be determined based on prior knowledge or the experiences of experts. The parameters are estimated using either the clustering or the least mean squared method.

### 3.2. Bees algorithm

The bees algorithm is an optimization algorithm inspired by the natural foraging behavior of honey bees to find the optimal solution \([28]\). Fig. 6 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 e 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 fitness of the sites visited by the scout bees is evaluated in step 2. In step 4, bees that have the highest fitness 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 fitness 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. The size of the flower patches $a = [a_1, a_2, ..., a_K]$ is initially set to a large value, where K shows the number of variables. For each variable $a_i$, it is set as follows:

$$a_i(t) = ngh(t) \times (max_i - min_i)$$

$$ngh(0) = 1.0$$

(3)

where $t$ denotes the $t^{th}$ iteration of the bees algorithm main loop. The size of a patch is kept unchanged as long as the local search procedure yields higher points of fitness. If the local search fails to bring any improvement in fitness, the size $a$ is decreased. The updating of the neighborhood size follows the following heuristic formula:

$$ngh(t+1) = 0.8 \times ngh(t)$$

(4)

Thus, following this strategy, the local search is initially defined over a large neighborhood, and has a largely explorative character. As the algorithm progresses, a more detailed search is needed to refine the current local optimum. Hence, the search is made increasingly exploitative, and the area around the optimum is searched more thoroughly. In step 6, for each patch only the bee with the highest fitness will be selected to form the next bee population.

In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions.

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

Fig. 6. Pseudo code of bees algorithm.
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. More details regarding to bees algorithm can be found in Ref. [28].

3.3. Bees-RBF

The RBF neural network training process can be divided into two steps. The first involves the determination of the radial basis functions features to be used in the hidden layer of the network and the second one involves determining the weights of the output neurons. Different learning strategies can be used in the design of an RBF network depending on how the centers of the radial basis functions network are specified, such as fixed centers selection and self-organized selection. A RBFN uses radial basis as its activation function and presents some main free parameters to be adjusted during training:

1) The number and location of the basis functions in the hidden layer;
2) The widths or spreads of these basis functions;
3) The weights in the output layer of the network.

The performance of the RBFNN strongly depends upon the number and positions of the basis functions composing the network hidden layer. The traditional methods to determine the centers are: random selection of the input vectors from the training dataset; obtaining prototypes based on unsupervised learning algorithms, such as k-means clustering; or using the supervised learning to train the network. Using the fixed or self-organized centers in RBFNN have the main drawback of working with an arbitrary number of RBF centers whose positions and spreads are either chosen randomly or self-organized, respectively [27].

Many approaches have been proposed in the literature with the goal of overcoming these limitations. In Ref. [27], a new learning algorithm, named Bees-RBF is introduced that utilizes the bees algorithm (BA) inspired clustering algorithm to obtain the number and location of radial basis function centers (prototypes) automatically to be used in an RBFNN. Then, the spread of each RBF center found by algorithm is dynamically determined based on the distribution of the clustered input data. The goal of this method is to guarantee that each basis function is sufficiently spread so as to cover all the data points that lie within its radius. Using this approach, the most important parameter of RBFNN including RBF centers and spread of RBFs will be found automatically and optimally.

In the Bees-RBF method, the clustering performed in the first layer of the RBFNN is done by the bees algorithm. In this method, each bee represents the centers of the clusters, and the number of clusters is determined by the algorithm. If the number of m clusters is determined, a spread must be determined for each cluster center. To do so, each vector in the data set is grouped to the nearest cluster center based on the Euclidean distance. After dividing the data between the clusters, the distance from the farthest data is determined from the center of the same cluster. For example, for the jth cluster, the spread amount $\sigma_j$ is calculated as follows:

$$\sigma_j = 1.1 \times d_{j_{\text{max}}}$$

where $d_{j_{\text{max}}}$ is the farthest distance from the center of jth cluster (in the same cluster (cluster $j$)).

After determining the cluster centers and spread of each cluster by the bee algorithm, the output of the first layer 1 is calculated by Eq. (1). Based on the output of the first layer, the final output of the network is obtained using Eq. (2). More details regarding this algorithm can be found in Ref. [27].

3.4. Feature selection algorithm

3.4.1. AR

Feature selection is the process of calculating importance of each feature and then selecting the most discriminative subset of features. AR is the search for relationship of the event or frequent pattern that has the potential to be applied in the analysis or predicting the future events. This relationship is in the form “IF condition Then consequence” [29]. To limit the search space, the discovered relationship has to satisfy these two criteria:

Support: It is the frequency of the occurring event. The occurrence frequency can be computed as the probability that two events (A and B) occur in the same transaction (Eq. (6)). The minimum threshold of support value is normally specified by users.

$$\text{Support}(A \rightarrow B) = \frac{\text{Support}(A \cap B)}{\text{Support}(A)}$$

Confidence: It is the proportion of frequency of co-occurring events (A and B) to the frequency of antecedent event (A). The computation is in equation (2). The minimum confidence is the threshold used to screen only interesting relationships.

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \rightarrow B)}{\text{Support}(A)}$$

3.4.2. Apriori algorithm

Apriori is a well-known algorithm for association rule mining. The algorithm finds frequent itemsets from database, but reduces the unnecessary search by deleting the itemset that its frequency is lower than minimum support. Fig. 7 shows the Apriori algorithm, which consists of five steps.

Step 1: scan database to count items and calculate support, and then generate 1-itemset frequent pattern ($L_1$) containing only items having support value higher than the minimum support.

Step 2: from line 1 to 2, generate candidate itemset ($C_{k+1}$) with $k = 1$,

| $C_k$: Candidate itemset of size k |
| $L_k$: frequent itemset of size k |
| $L_1 = \{\text{frequent items}\}$ |
| 1. for (k = 1; $L_k$=!\emptyset; k++) do begin |
| 2. $C_{k+1}$ = candidates generated from $L_k$; |
| 3. for each transaction t in database do |
| 4. increment the count of all candidates in $C_{k+1}$ that are contained in t |
| 5. end |
| 6. $L_{k+1}$ = candidates in $C_{k+1}$ with min_support |
| 7. end |
| 8. return $\bigcup L_k$ |

Fig. 7. pseudo code of AR [29].
2, 3, ..., n) from frequent itemset \((L_k\) with \(k = 1, 2, 3, \ldots\)).

Step 3: from line 3 to 4, scan database to count support values of all candidates in \(C_{k+1}\).

Step 4: from line 6, generate frequent itemset \(L_{k+1}\) from \(C_{k+1}\) with more than minimum support.

Step 5: repeat steps 2 to 5 until \(L_k\) is empty. The algorithm then returns all frequent itemsets of the given database. After that the association rules, having confidence higher than the minimum threshold, can be generated from these frequent itemsets.

### 4. Proposed method

This section belongs to describing the details of the proposed method. The proposed method includes four main modules namely feature extraction module, feature selection module, classifier module and learning algorithm module. Fig. 8 shows the main structure of proposed pattern recognition method in the general state. This method can be applied for any pattern recognition problem. In this figure, \(Input, (i = 1, 2, \ldots, Q)\) indicates the \(i\)th array of the input vector. The value of \(Q\) depends on the problem. In this figure, \(Fe_i, (i = 1, 2, \ldots, N)\), \(Fs_i, (i = 1, 2, \ldots, M)\), \(RBFNN\), \(BA\), \(P_{t-1}\), \(P_t\), \(P_{t+1}\), and \(P_i\) are representative of \(i\)th extracted feature, number of the extracted features, \(i\)th selected feature, number of the selected features, \(i\)th pattern and number of the patterns, respectively.

The dimension of the raw control chart patterns is 60. Thus \(Q = 60\) in the time domain as it is illustrated in Fig. 1. Number of the extracted features are 12 \((N = 12)\) and there are eight different patterns, \(T = 8\).

Fig. 8. The main structure of proposed pattern recognition method in general state.

Fig. 9 shows the traditional RBFNN based CCP pattern recognition system. In this system, raw data is fed to RBFNN.

Most of the recent instances of pattern recognition problems (whether in computer vision, image understanding, biology, text interpretation or spam detection) involve highly complex data sets with a huge number of possible explanatory variables. This abundance of variables significantly compromises classification or recognition tasks for several reasons. Weak informative features act as artificial noise in data and limit the accuracy of the classification algorithms. In addition, the variance of a statistical model is typically an increasing function of the number of variables, whereas the bias is a decreasing function of this quantity [30–32]. Reducing the dimension of the feature space is necessary to infer reliable conclusions. In some cases, the application of classification algorithms like RBFNN or q-nearest neighbors on the full feature space is not possible or realistic due to the time needed to apply the decision rule. Furthermore, there are many applications for which detecting the pertinent explanatory variables is critical, as important as correctly performing classification tasks.

The CCPs time domain signals composed of 60 components. In Fig. 1, the studied CCPs are shown. It can be observed that these patterns are very similar to each other, specially increasing trend with upward shift and decreasing trend with downward shift. This fact is investigated in Fig. 10 by depicting the box plots of CCPs.

Learning algorithms generally perform better in lower dimensional space. Thus, it is important that the patterns can be transformed to lower dimensional space so that the learning can be performed well. This enables us to view the pattern recognitions tasks as consisting of two parts, namely, a feature extraction part and a pattern association part. Feature extraction is problem-dependent. The performance of the classifier depends on how well the features are selected by the designer.

The results of the similar studies have shown that the usage of shape and statistical features leads to higher recognition accuracy in CCPs [3,18,22]. Therefore, these features are employed to reduce the input dimension and also to improve the accuracy. Features that may have the same discrimination property should be eliminated. Since, redundant features will affect the complexity, computational time and consequently the classifier’s performance. Thus, AR has been employed to select the best set of the shape and statistical features. AR selects a reasonable subset of features then it improves both the recognition accuracy and the complexity of the problem. The algorithm for feature selection based on AR for classification is given in Fig. 11. Our algorithm consists of four phases.

The first phase is at line 1. This phase is for conventional association
rule mining with Apriori algorithm from training data set (D). This phase requires three parameters, which are minimum support threshold \((\text{minsup})\), minimum confidence threshold \((\text{minconf})\), maximum number of attributes that can be appeared at the conditional (or antecedent) of the association rules \((\text{maxlen})\), and minimum frequency of features or attributes that appeared in association rules \((\text{minfrequent})\). This last threshold is for discarding features with low importance.

Phase 2 is the part from lines 2 to 6. This phase is the rule pruning,

```plaintext
//Input: D, training data set.  
\text{minsup}, \text{minimum support threshold.}  
\text{minconf}, \text{minimum confidence threshold.}  
\text{maxlen}, \text{maximum number of conditional attributes.}  
C, \text{class attribute.}  
\text{minfrequent}, \text{minimum frequency of attributes in set of association rules.}  

//Output: F, a set of frequent features.
1. R = Apriori(D, \text{minsup}, \text{minconf, maxlen})  
2. For each rule r \in R do  
3. \quad If \ consequent(r) \neq C \ Then  
4. \quad delete r from R  
5. \quad End  
6. End  
7. For each attribute Attr from D do  
8. \quad For each rule r \in R do  
9. \quad \quad If \ condition(r) = \text{Attr} \ Then  
10. \quad \quad count_{\text{Attr}} ++  
11. \quad End  
12. End  
13. add Attr and count_{\text{Attr}} to F  
14. End  
15. For each feature f \in F do  
16. \quad If \ \text{FrequentFeature}(f) < \text{minfrequent} \ Then  
17. \quad \quad delete f from F  
18. \quad End  
19. End  
20. Return F
```

Fig. 10. Box plots of different patterns.

Fig. 11. Feature selection in the proposed method based on AR.
which is the deletion of rules that have attributes in their consequence part disagree with the specified subset of class attribute (C). This phase is for selecting from association rules the frequent patterns containing both predictive features and class attribute.

Phase 3 is the part from lines 7 to 14. The operation of this phase is to count the frequency of features appeared in association rules that are obtained from phase 2. We design these steps to iterate over each attribute and count the appearance frequency of attributes that appear in the conditional part of the association rules.

Phase 4 is the part from line 15 to 19. This phase is the features selection from subset of frequent features of rules (F) by deleting features that have percentage of frequency appearance in the set of association rules lower than the specified minimum frequency threshold. Finally, the algorithm returns the subset of features that has been considered high importance to class attribute prediction based on the analysis of their appearance in the set of association rules induced from the training data set.

By extracting the new features and selecting of the best collection of those features, the dimension of input data to RBFNN will decrease from 60 (in unprocessed data state) to lower value (using selected features). Therefore, training period of the RBFNN will be decreased as it is shown in Fig. 12. In the proposed method, the learning algorithm introduced in section 3.3 is used to train the RBFNN. Fig. 13 shows the main structure of proposed method.

The main steps of the proposed method are as follows:

Step 1: Generate raw data using the relationships provided in reference [3].
Step 2: Extraction of the shape and statistical features introduced in the second part
Step 3: Select the effective features using the AR method
Step 4: Setting the parameters of the bee algorithm like number of scout bees (n), number of sites selected out of n visited sites (m) and ... [32]
Step 5: Generate the initial population. In the proposed method, each bee represents the center of the cluster. Note that the clustering algorithm does not use the class labels, only the input patterns x_i, ∀i. The optimal number of clusters will be determined by the bee algorithm. In the proposed method, each bee will be as follows:

\[ \text{Bee}=\{N_c, C_1, C_2, \ldots, C_{N_c}\} \]
\[ C_i = [x_1, x_2, \ldots, x_{\text{dim}}], \quad i = 1, 2, \ldots, N_c \]  

In this regard, \( N_c \) represents the number of clusters, \( C_i (i=1,2,\ldots,N_c) \) represents the center of the cluster \( i \), \( \text{dim} \) represents the number of input vector arrays and \( x_j (j=1,2,\ldots,\text{dim}) \) represents the arrays forming each cluster center. The number of clusters \( N_c \) is variable and its optimal value will be determined by the bee algorithm. The maximum number of clusters should be determined at the beginning of the algorithm. In this study, the maximum number of clusters \( (N_{c_{\text{max}}}^c) \) is considered to be 30. Therefore, the number of clusters is in the interval of \( 1 \leq N_c \leq N_{c_{\text{max}}}^c \).

Step 6: Calculate the fitness function for the bees population.
Step 6-1: After bees population formation and determination the centers of the clusters, the input data is clustered. To do this, each data in the cluster is linked to the closest cluster center. With respect to the performed clustering, the optimal distribution of each cluster is determined by equation (5).
Step 6-2: The output of each radial basis function is calculated by equation (1).
Stage 6-3: The second layer’s weights of the neural network are determined by equation (2).
Step 6-4: The accuracy of the neural network is calculated on the train data and is considered to be a fitness function.
Step 7: After calculating the fitness function of the bees (Recognition accuracy of RBFNN on training data), the bees are arranged and a local search for \( m \) number of the best bees will be done.
Step 8: Remaining bees are removed and new bees are produced randomly instead of them.
Step 9: Check the condition of the stop. If it is satisfied, go to the next step, otherwise go to step six. In this method, reaching the maximum allowed repetition is set as the condition of the stop.
Step 10: End of the algorithm

5. Simulation results

In this section we evaluate the performance of the proposed recognizer. The formulas presented in Ref. [3] are employed to generate 200 samples for each pattern. These relationships are shown in Table 1. In this table, \( P \) represents the moment at which the shift occurs.

For this study, we have used 50% of data for training the classifier and the rest for testing. In order to evaluate the performance of the classifier, the confusion matrix is used. The values in the diagonal of confusion matrix show the correct performance of recognizer for each pattern. These values show that how many of considered pattern are recognized correctly by the classifier. The other values show the
mistakes of the system. In order to achieve the recognition accuracy (RA) of system, it is needed to compute the average value of that appears in diagonal [34].

In this section, several experiments have been done for evaluating of the proposed method. The computational experiments for this section were done on Intel core i7 with 16 GB RAM using ASUS computer. The computer program was performed on MATLAB (version 8.5.0.197613 [R2015a], Massachusetts, USA) environment.

5.1. Effect of RBF numbers and spread value on RBFNN performance

In this experiment, the raw data is used as input to RBFNN. The number of RBFs is equal to the number of the training data. Besides, the value of spread is tested for varying different values. The obtained results are listed in Table 2. In this table, RA stands for recognition accuracy. It can be seen that the network with 800 RBFs and spread equals to 9, leads to the best performance (93.85%) which is shown by bold pen. It can be seen that there is no linear relation between the value of spread and recognition accuracy of RBFNN. Therefore the value of spread must be obtained through trial and error and based on extensive simulations. This manner of network topology selection is very time consuming. Fig. 14, shows the effect of spread on RBFNN’s performance.

In the next experiment, the RBFs number is selected less than training data. The obtained results have been shown by Fig. 15. In Fig. 15 the RBFNN with 40 RBFs is built and the value of spread is changed from 0.5 to 20. It can be seen that the performance of the network with 40 RBFs is better than the network with 800 RBFs. In the network with 800 RBFs, the best recognition accuracy was 93.85%, whereas in the network with 40 RBFs and width equal to 13, the highest

Table 1

<table>
<thead>
<tr>
<th>Control Chart patterns</th>
<th>Pattern parameters</th>
<th>Parameter values</th>
<th>Pattern equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOR</td>
<td>Mean (μ)</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard deviation (σ)</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>STR</td>
<td>Random noise (σ')</td>
<td>0.2σ ≤ σ' ≤ 0.2σ</td>
<td></td>
</tr>
<tr>
<td>SYS</td>
<td>Systematic departure (d)</td>
<td>1σ ≤ d ≤ 3σ</td>
<td></td>
</tr>
<tr>
<td>CYC</td>
<td>Amplitude (a)</td>
<td>1.5σ ≤ a ≤ 2.5σ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Period (T)</td>
<td>T ≤ 16</td>
<td></td>
</tr>
<tr>
<td>IT</td>
<td>Gradient (g)</td>
<td>0.05σ ≤ g ≤ 0.1σ</td>
<td></td>
</tr>
<tr>
<td>DT</td>
<td>Gradient (g)</td>
<td>−0.1σ ≤ g ≤ −0.05σ</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>Shift magnitude (s)</td>
<td>1.5σ ≤ s ≤ 2.5σ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Shift position (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DS</td>
<td>Shift magnitude (s)</td>
<td>−2.5σ ≤ s ≤ −1.5σ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Shift position (P)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The best results are shown by bold.

Table 2

The performance of RBFNN with raw data (No. of RBFs is equal to training data).

<table>
<thead>
<tr>
<th>Width</th>
<th>No. of RBFs</th>
<th>RA (%)</th>
<th>Width</th>
<th>No. of RBFs</th>
<th>RA (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>800</td>
<td>92.11</td>
<td>10.5</td>
<td>800</td>
<td>92.86</td>
</tr>
<tr>
<td>1</td>
<td>800</td>
<td>92.18</td>
<td>11</td>
<td>800</td>
<td>93.53</td>
</tr>
<tr>
<td>1.5</td>
<td>800</td>
<td>92.65</td>
<td>11.5</td>
<td>800</td>
<td>93.14</td>
</tr>
<tr>
<td>2</td>
<td>800</td>
<td>93.14</td>
<td>12</td>
<td>800</td>
<td>93.17</td>
</tr>
<tr>
<td>2.5</td>
<td>800</td>
<td>93.35</td>
<td>12.5</td>
<td>800</td>
<td>93.25</td>
</tr>
<tr>
<td>3</td>
<td>800</td>
<td>93.67</td>
<td>13</td>
<td>800</td>
<td>93.44</td>
</tr>
<tr>
<td>3.5</td>
<td>800</td>
<td>93.66</td>
<td>13.5</td>
<td>800</td>
<td>92.41</td>
</tr>
<tr>
<td>4</td>
<td>800</td>
<td>93.06</td>
<td>14</td>
<td>800</td>
<td>93.21</td>
</tr>
<tr>
<td>4.5</td>
<td>800</td>
<td>93.33</td>
<td>14.5</td>
<td>800</td>
<td>93.11</td>
</tr>
<tr>
<td>5</td>
<td>800</td>
<td>93.12</td>
<td>15</td>
<td>800</td>
<td>92.67</td>
</tr>
<tr>
<td>5.5</td>
<td>800</td>
<td>93.38</td>
<td>15.5</td>
<td>800</td>
<td>92.97</td>
</tr>
<tr>
<td>6</td>
<td>800</td>
<td>92.43</td>
<td>16</td>
<td>800</td>
<td>92.67</td>
</tr>
<tr>
<td>6.5</td>
<td>800</td>
<td>93.55</td>
<td>16.5</td>
<td>800</td>
<td>92.55</td>
</tr>
<tr>
<td>7</td>
<td>800</td>
<td>92.63</td>
<td>17</td>
<td>800</td>
<td>92.65</td>
</tr>
<tr>
<td>7.5</td>
<td>800</td>
<td>91.76</td>
<td>17.5</td>
<td>800</td>
<td>92.32</td>
</tr>
<tr>
<td>8</td>
<td>800</td>
<td>91.56</td>
<td>18</td>
<td>800</td>
<td>92.33</td>
</tr>
<tr>
<td>8.5</td>
<td>800</td>
<td>91.88</td>
<td>18.5</td>
<td>800</td>
<td>92.17</td>
</tr>
<tr>
<td>9</td>
<td>800</td>
<td>91.85</td>
<td>19</td>
<td>800</td>
<td>92.11</td>
</tr>
<tr>
<td>9.5</td>
<td>800</td>
<td>92.77</td>
<td>19.5</td>
<td>800</td>
<td>92.04</td>
</tr>
<tr>
<td>10</td>
<td>800</td>
<td>92.93</td>
<td>20</td>
<td>800</td>
<td>91.31</td>
</tr>
</tbody>
</table>

Fig. 13. The main structure of proposed method.

The best results are shown by bold.
recognition accuracy is 95.21%. The obtained results show that the changing of RBFs number improves the recognition accuracy significantly.

In previous experiments the effect of spread on RBFNN performance was investigated and it was seen that there were no linear relation between the width (spread) value and network performance. In the new experiment, the effect of RBF numbers will be investigated. For this purpose, the number of RBFs is changed from one to 800 and the value of spread is fixed. In Fig. 16, the value of spread is fixed on 5 and the number of RBFs is changed from 1 to 800 (800 is the number of the training data). It can be seen that the performance of RBFNN is highly dependent on the number of RBFs. The best recognition accuracy (97.33%) is obtained by network with 14 RBFs and width equal to 5. Based on these experiments it is clear that the RBFNN performance is highly dependent on RBFs number and spread value.

5.2. Performance of Bees-RBF with raw data

The importance of optimal selection of RBF numbers and their spread is proved in the previous experiment. In this experiment, for optimal selection of RBFs center and spread of RBFs, Bees-RBF is used. The results were compared with those obtained by k-means and random center selection mechanism. To allow a fair comparison of the methods, the value of k for k-means and the number of random centers were chosen as the same value found by the Bees-RBF approach. Table 3 presents the recognition accuracy and number of centers. The number of centers was automatically found by Bees-RBF and this value used in k-means and the random selection approach. In this experiment, we used raw data.

From Table 3, it can be seen that using Bees-RBF method, the optimal number of RBFs and optimal value of spread of RBFs are determined automatically. Using Bees-RBF provides us with increasing the recognition accuracy up to 98.36%. The recognition accuracy enhancement proves the effectiveness of learning algorithm.

5.3. Performance of the proposed method

In the first experiment, the effect of RBFs number and their relative spread have been investigated. In the second experiment, the effectiveness of Bees-RBF was introduced and proved. In this subsection, the performance of proposed method is investigated. For this purpose, we used AR to determine the best feature collection. Based on the AR method, 6 features including first shape feature (S), fourth shape feature (AS), fifth shape feature (SD), eighth shape feature (ASS), first statistical feature (Mean) and second statistical feature (STD) are selected as the best feature collection. The obtained results are listed in
Table 4. It can be seen that 99.11% recognition accuracy is achieved by Bees-RBF network and shape feature, 98.75% recognition accuracy is acquired by Bees-RBF network and statistical feature, 98.86% recognition accuracy is achieved by Bees-RBF network and both shape and statistical features and 99.63% recognition accuracy is obtained by the proposed method.

5.4. Comparison with different classifier

The performance of the proposed classifier has been compared with other classifiers for investigating the capability of the proposed classifier, as indicated in Table 5. In this respect, adaptive neuro-fuzzy inference system (ANFIS), probabilistic neural networks (PNN), Multi layered Perceptron (MLP) neural network with different training algorithm such as Back propagation (BP) learning algorithm, with Resilient propagation (RP) learning algorithm and Levenberg Marquardt (LM) are considered. In this experiment, we used six selected shapes and statistical features. It can be seen from Table 4 that the proposed method has better recognition accuracy than other classifiers.

In this sub-section, for evaluating the robustness and convergence speed of the proposed method, five different runs have been performed. Fig. 17 shows a typical increase of the fitness (recognition accuracy) of the best individual fitness of the population obtained from the proposed system for different runs. As it indicated in this figure, its fitness curves

<table>
<thead>
<tr>
<th>Clustering method</th>
<th>Input dimension</th>
<th>Spread</th>
<th>No. RBFs</th>
<th>RA (%)</th>
<th>Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centers randomly selected</td>
<td>60</td>
<td>5.329</td>
<td>11</td>
<td>96.45</td>
<td>2</td>
</tr>
<tr>
<td>Centers selected by k-means</td>
<td>60</td>
<td>5.329</td>
<td>11</td>
<td>97.17</td>
<td>4</td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>60</td>
<td>5.329</td>
<td>11</td>
<td>98.36</td>
<td>3</td>
</tr>
</tbody>
</table>

The best results are shown by bold.

Table 5

Comparison the performance of proposed classifier with other classifiers.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Parameters</th>
<th>Recognition accuracy (%)</th>
<th>Standard deviation</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNN</td>
<td>Spread = 0.75</td>
<td>97.54 ± 2.11</td>
<td>1.26</td>
<td>2.76</td>
</tr>
<tr>
<td>ANFIS</td>
<td>Radii = 0.5</td>
<td>99.12 ± 0.36</td>
<td>0.94</td>
<td>15.76</td>
</tr>
<tr>
<td>MLP (BP)</td>
<td>Number of hidden layer = 2</td>
<td>96.62 ± 2.75</td>
<td>15.76</td>
<td></td>
</tr>
<tr>
<td>MLP (RP)</td>
<td>Number of hidden layer = 1</td>
<td>98.55 ± 1.05</td>
<td>6.54</td>
<td></td>
</tr>
<tr>
<td>MLP (LM)</td>
<td>Number of hidden layer = 1</td>
<td>99.14 ± 0.73</td>
<td>8.42</td>
<td></td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>No. RBFs = 9</td>
<td>99.63 ± 0.27</td>
<td>0</td>
<td>0.27</td>
</tr>
</tbody>
</table>

The best results are shown by bold.

Fig. 16. The effect of RBF number on network performance (spread = 5).

Table 3

Performance of Bees-RBF with raw data.

<table>
<thead>
<tr>
<th>Clustering method</th>
<th>Input dimension</th>
<th>Spread</th>
<th>No. RBFs</th>
<th>RA (%)</th>
<th>Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centers randomly selected</td>
<td>8</td>
<td>3.965</td>
<td>9</td>
<td>98.45</td>
<td>0.31</td>
</tr>
<tr>
<td>Centers selected by k-means</td>
<td>8</td>
<td>3.965</td>
<td>9</td>
<td>98.78</td>
<td>0.58</td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>8</td>
<td>3.965</td>
<td>9</td>
<td>99.11</td>
<td>0.34</td>
</tr>
<tr>
<td>Centers randomly selected</td>
<td>4</td>
<td>3.274</td>
<td>9</td>
<td>97.87</td>
<td>0.29</td>
</tr>
<tr>
<td>Centers selected by k-means</td>
<td>4</td>
<td>3.274</td>
<td>9</td>
<td>98.56</td>
<td>0.51</td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>4</td>
<td>3.274</td>
<td>9</td>
<td>98.75</td>
<td>0.32</td>
</tr>
<tr>
<td>Centers randomly selected</td>
<td>12</td>
<td>3.442</td>
<td>9</td>
<td>98.14</td>
<td>0.33</td>
</tr>
<tr>
<td>Centers selected by k-means</td>
<td>12</td>
<td>3.442</td>
<td>9</td>
<td>98.31</td>
<td>0.65</td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>12</td>
<td>3.442</td>
<td>9</td>
<td>98.86</td>
<td>0.38</td>
</tr>
<tr>
<td>Centers randomly selected</td>
<td>6</td>
<td>2.762</td>
<td>9</td>
<td>98.73</td>
<td>0.20</td>
</tr>
<tr>
<td>Centers selected by k-means</td>
<td>6</td>
<td>2.762</td>
<td>9</td>
<td>99.26</td>
<td>0.48</td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>6</td>
<td>2.762</td>
<td>9</td>
<td>99.63</td>
<td>0.27</td>
</tr>
</tbody>
</table>

The best results are shown by bold.

Table 4

Performance of proposed method.

<table>
<thead>
<tr>
<th>Input type</th>
<th>Clustering method</th>
<th>Input dimension</th>
<th>Spread</th>
<th>No. RBFs</th>
<th>RA (%)</th>
<th>Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shape feature</td>
<td>Centers randomly selected</td>
<td>8</td>
<td>3.965</td>
<td>9</td>
<td>98.45</td>
<td>0.31</td>
</tr>
<tr>
<td>Statistical feature</td>
<td>Centers selected by k-means</td>
<td>8</td>
<td>3.965</td>
<td>9</td>
<td>98.78</td>
<td>0.58</td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>8</td>
<td>3.965</td>
<td>9</td>
<td>99.11</td>
<td>0.34</td>
<td></td>
</tr>
<tr>
<td>Shape &amp; Statistical features</td>
<td>Centers randomly selected</td>
<td>4</td>
<td>3.274</td>
<td>9</td>
<td>97.87</td>
<td>0.29</td>
</tr>
<tr>
<td>Centers selected by k-means</td>
<td>4</td>
<td>3.274</td>
<td>9</td>
<td>98.56</td>
<td>0.51</td>
<td></td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>4</td>
<td>3.274</td>
<td>9</td>
<td>98.75</td>
<td>0.32</td>
<td></td>
</tr>
<tr>
<td>Selected shape &amp; Statistical feature</td>
<td>Centers randomly selected</td>
<td>12</td>
<td>3.442</td>
<td>9</td>
<td>98.14</td>
<td>0.33</td>
</tr>
<tr>
<td>Centers selected by k-means</td>
<td>12</td>
<td>3.442</td>
<td>9</td>
<td>98.31</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>12</td>
<td>3.442</td>
<td>9</td>
<td>98.86</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>Centers randomly selected</td>
<td>6</td>
<td>2.762</td>
<td>9</td>
<td>98.73</td>
<td>0.20</td>
<td></td>
</tr>
<tr>
<td>Centers selected by k-means</td>
<td>6</td>
<td>2.762</td>
<td>9</td>
<td>99.26</td>
<td>0.48</td>
<td></td>
</tr>
<tr>
<td>Bees-RBF</td>
<td>6</td>
<td>2.762</td>
<td>9</td>
<td>99.63</td>
<td>0.27</td>
<td></td>
</tr>
</tbody>
</table>

The best results are shown by bold.
gradually improved from iteration 0 to 100, and exhibited no significant improvements after iteration 54 for the five different runs. The optimal stopping iteration to get the highest validation accuracy for the five different runs was around iteration 50.

### 5.5. Comparison and discussion

Considering the importance of the continuous monitoring and control of the production process, in the recent years extensive studies have been conducted to recognize CCPs successfully. Direct comparison with other works is difficult for control chart pattern recognition problems. This is mainly because of the fact that there is no single unified data set available. A different setup of patterns (for example, the number of training and testing samples and the number of patterns) will lead to different performance. Besides, there are many different kinds of benchmarking system used for system quality. This causes difficulties for direct numerical comparison. Table 6 compares some different methods in case of: the recognition accuracy, the used classifier and the used inputs.

In Ref. [18], The MLP neural network is used to classify CCPs. The authors have considered only six patterns including NOR, CYC, IT, DT, US and DS. They have used shape features as inputs of the neural network whose parameters are selected based on the authors’ experience. Their approach was able to recognize the patterns with the accuracy rate of 94.30% and 99% while using the raw data and shape features, respectively. Neural networks are also used in Ref. [35] to recognize CCPs. Like [18], only six features are considered and the data is generated using existing relationships. In this study, raw data was used as input for the MLP neural network. To improve the performance of the neural network, modifications have been made to the learning algorithm of the neural network. The presented method was able to correctly identify patterns with accuracy rate of 97.73%. Furthermore, in Ref. [22] neural network is employed for CCP recognition. Similar to previous studies six features are considered and 2160 data is generated using existing relationships. To improve the accuracy of the recognition, six statistical features have been extracted. The proposed method was able to identify the patterns with accuracy rate of 94.10% and 97.18% using raw data and extracted statistical features, respectively. In Ref. [36], only Normal, Cyclic, Shift and Trend patterns are used for recognition of CCPs and existing relationships are used for generating 3000 data. The ID-DT and US-DS patterns are considered as one pattern. Frequency features (first-order wavelet approximation coefficients) have been used to improve the recognition accuracy from 85.50% to 91.15%.

In Ref. [37], only four patterns are considered, and frequency features (fourth-order wavelet approximation coefficients) are used as inputs of the neural network. Their analysis showed that this method is able to identify the patterns with accuracy rate of 80.44% and 97.22% using raw data and frequency features, respectively. In this study, the effects of the number of hidden layers and their neurons have been considered.

<table>
<thead>
<tr>
<th>Ref. no</th>
<th>Year</th>
<th>No. Patterns</th>
<th>Input type</th>
<th>Input size</th>
<th>Classifier</th>
<th>RA (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18]</td>
<td>1997</td>
<td>6</td>
<td>Shape feature</td>
<td>9</td>
<td>ANN-MLP</td>
<td>99.00</td>
</tr>
<tr>
<td>[35]</td>
<td>2000</td>
<td>6</td>
<td>Raw data</td>
<td>60</td>
<td>ANN-MLP</td>
<td>97.73</td>
</tr>
<tr>
<td>[22]</td>
<td>2003</td>
<td>6</td>
<td>Statistical feature</td>
<td>6</td>
<td>ANN-MLP</td>
<td>97.18</td>
</tr>
<tr>
<td>[36]</td>
<td>2004</td>
<td>4</td>
<td>Frequency features</td>
<td>32</td>
<td>ANN-MLP</td>
<td>90.15</td>
</tr>
<tr>
<td>[27]</td>
<td>2005</td>
<td>4</td>
<td>Frequency features</td>
<td>7</td>
<td>ANN-MLP</td>
<td>97.22</td>
</tr>
<tr>
<td>[3]</td>
<td>2009</td>
<td>8</td>
<td>Shape feature</td>
<td>30</td>
<td>ANN-MLP</td>
<td>98.54</td>
</tr>
<tr>
<td>[38]</td>
<td>2011</td>
<td>6</td>
<td>Shape feature</td>
<td>8</td>
<td>ANN-MLP</td>
<td>99.21</td>
</tr>
<tr>
<td>[40]</td>
<td>2011</td>
<td>6</td>
<td>Euclidean distance</td>
<td>6</td>
<td>ANN-RBF</td>
<td>99.26</td>
</tr>
<tr>
<td>[41]</td>
<td>2013</td>
<td>6</td>
<td>Frequency features</td>
<td>30</td>
<td>ANFIS</td>
<td>99.32</td>
</tr>
<tr>
<td>[42]</td>
<td>2016</td>
<td>6</td>
<td>Shape feature</td>
<td>8</td>
<td>ANN-MLP</td>
<td>99.21</td>
</tr>
<tr>
<td>[43]</td>
<td>2017</td>
<td>6</td>
<td>Shape and Statistical feature</td>
<td>6</td>
<td>SVM</td>
<td>99.42</td>
</tr>
<tr>
<td>This work</td>
<td>-</td>
<td>8</td>
<td>Shape and Statistical feature</td>
<td>6</td>
<td>Bees-RBF</td>
<td>99.63</td>
</tr>
</tbody>
</table>

The best results are shown by bold.

![Fig. 17. Evolution of fitness function for different runs.](image-url)
investigated and ultimately the best structure has been chosen. The results highlighted the high impact of the neural network parameters. In Ref. [3], all of the eight patterns are considered and data is generated using existing relationships. They have extracted 30 shape features as inputs of the neural network. Employing the shape features, the CCP are recognized by 98.54% accuracy rate.

In Ref. [38], real data that includes six patterns, including NOR, CYC, IT, DT, US and DS is used [39]. Also, 8 shape features are used as inputs of the neural network. The proposed method was able to identify the patterns using shape features with a precision of 99.21%. In Ref. [40], the RBF neural network is applied to data [30]. Fuzzy C-mean clustering algorithm is used to extract the new features. The presented method was able to recognize the patterns by 99.26% of accuracy rate. The authors in Ref. [41] have applied ANFIS on the dataset [39]. They have employed frequency features (first-order wavelet approximation coefficients) to improve the performance. The proposed method was able to identify the patterns by accuracy of 99.32%. In Ref. [42], neural network is applied on the dataset [39]. In this approach, shape features are used and the structure of the neural network is optimized using the Cuckoo optimization algorithm. The proposed method was able to correctly recognize the patterns using frequency characteristics with accuracy rate of 99.42%.

In Ref. [43], SVM is used to recognize CCPs. In this study, the data was generated using existing relationships, and only six patterns NOR, CYC, IT, DT, US, and DS were considered. Also, they have used 12 shape and statistical features including six shape features and six statistical features as inputs. SVM parameters are selected using the genetic algorithm. Additionally, improved supervised locally linear embedding algorithm has been used to select effective features. The results show that the feature has a higher impact on the accuracy of the classifier. The proposed method was able to correctly identify the patterns using the six selected features with a precision of 99.42%.

In the most studies, only six patterns including NOR, CYC, IT, DT, US and DS have been considered. Due to the fact that the three patterns NOR, STR and SYS are very similar and their resolution is very difficult, in most studies STR and SYS patterns are not considered. Regarding the importance of the continuous monitoring and control of the production process and precise identification of the encountered fault, eight patterns are considered in this study. In the most published approaches, the selection of the appropriate features has not been made and only new features have been extracted.

Obviously, only extracting features and using all of them would not lead to the best result. In addition, in most of the works which have been done, the structure of the classifier including number of the neurons in the hidden layer, learning rate, and etc. in neural networks, penalty factor, kernel function and etc. in SVM, and membership function, fuzzifier, defuzzifier and etc. in fuzzy systems are selected based on trial and error. Due to the complex relationship between the structure of these classifiers and their accuracy, these parameters are not optimal and their performance is reduced. In this study, the bee based learning algorithm is used to overcome this problem. This method is able to identify the classes by 99.63% accuracy rate. Comparison of the results shows that the proposed method is superior to other methods along with the fact that the proposed method is being applied on (8) patterns.

5.6. Performance of the proposed method on the other pattern recognition problems

In order to evaluate the performance of the proposed method in dealing with other pattern recognition problems, the proposed method has been used for fault detection in power transformers.

One of the most important components in power system networks is power transformer which changes the voltage level of system from one magnitude to another magnitude. Faults in transformer will cause power system outage, resulting in loss of revenue and hardship to consumers. Phenomena which can contribute to transformer faults are corona, partial discharge, sparking, arcing and overheating. Therefore, transformer condition diagnosis, monitoring and maintenance are important. Transformer fault identification is important in reducing the maintenance cost and repair time to restore the power supply. The most widely used transformer fault identification method is dissolved gas analysis (DGA). DGA is an analysis based on the amount of dissolved gas in transformer oil which is taken directly from the transformer tank.

In Ref. [44], an MLP neural network optimized with particle swarm algorithm is used for fault detection in power transformers. In Ref. [44], hybrid modified evolutionary particle swarm optimization-time varying acceleration coefficient (MEPSO-TVAC)-artificial neural network (ANN) was proposed for transformer fault diagnosis based on dissolved gas data. In this work, the input and output data were obtained from the actual dissolved gas analysis from a power transformer diagnosis of an electrical utility. The input data are the dissolved gas from the power transformer oil, which are hydrogen (H2), methane (CH4), acetylene (C2H2), ethylene (C2H4), ethane (C2H6) and carbon monoxide (CO). The output data are high intensity discharge, low intensity discharge, thermal fault and no fault. 400 data of input-output were used. The performance of the proposed method in Ref. [44] is listed in Table 7. They have employed all of the six features as inputs of the classifier. Furthermore, parameters of the MLP neural network have been determined using MEPSO-TVAC, which are shown in the second column of the table.

In the proposed method, AR is used to select the more efficient set of features. The analysis shows that the fifth feature, ethane, is redundant and can be discarded, since the information is provided by other features. It should be noticed that excluding redundant features can increase computations speed and can prevent classification errors. The results are shown in Table 7. As it can be seen in this table, the proposed method has higher accuracy. The important point is the robustness of the proposed method. The proposed method successfully identifies the faults with high accuracy rate (99.52%) in all experiments; however, the method presented in Ref. [44] is not robust and fails in some experiments. In the last column, the standard deviation of the obtained results by both proposed method and the method presented in Ref. [44] are given.

6. Conclusion

Unnatural patterns in the control charts can be associated with a specific set of assignable causes for process variation. Hence, pattern recognition is very useful in identifying the process problems. In this study a fast intelligent and accurate method was proposed for control chart pattern recognition. In the proposed method, shape and statistical features have been extracted so that the recognition accuracy would be better and the volume of computation would be reduced. A new learning algorithm based on the bee algorithm was used to train the classifier. Several experiments were performed to evaluate the

Table 7  Evaluation of the performance of the proposed method on other pattern recognition problems - fault detection in power transformers.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Size of Inputs</th>
<th>Parameters</th>
<th>RA (%)</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEPSO-TVAC-ANN</td>
<td>6</td>
<td>Number of layers = 2</td>
<td>99</td>
<td>± 0.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Neurons on the hidden layer = [5 6]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learning rate = 0.095</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>momentum constant = 0.4752</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bees-RBF (Proposed method)</td>
<td>5</td>
<td>No. RBFs = 11</td>
<td>99.52</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spread = 4.652</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The best results are shown by bold.
performance of the proposed method.

In the first experiment, the number of RBFs was fixed and the amount of spread varied from 0.5 to 20. The simulation results showed that the performance of RBFNN is strongly dependent on the spread value. The main result of this experiment was that it showed no linear relationship between the spread and RBFNN function. Therefore, when using RBFNN, the value of this parameter must be determined by trial and error.

In next experiment, the value of spread was kept constant and the number of RBFs was changed from 1 to 800. The simulation results showed that the performance of RBFNN is highly dependent on RBFs number. Similar to the previous test, it is observed that there is no linear relationship between the RBF numbers and RBFNN performance. The experiment also showed that the network with high number of RBFs has a low generalization capability and, as a result, reduces network accuracy.

By identifying the importance of network structure and the effect of parameters on its performance, we used the Bees-RBF algorithm to find the optimal network structure. Also, shape and statistical features were used as inputs of the network. The best network will have poor performance if input features are not properly selected. A feature selection algorithm should select the best and most effective features and minimize the number of input dimensions to the lowest and most optimal. RBFNN, with optimal structure and the use of selected features as input, was able to correctly identify the patterns with a 99.63% accuracy. The proposed system has a high RA and therefore we recommend the proposed system for CCPs recognition.

References