International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 10, No. 5 (2017), pp. 13-28
http://dx.doi.org/10.14257/ijsip.2017.10.5.02

Control Chart Pattern Classification Based on Optimized Intelligent Approach: A Comparative Study

Esmaeil Atoon[1] and Milad Azarbad[2]

[1]Faculty of Electrical and Computer Engineering, Islamic Azad University, Gonabad, Iran
[2]Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran
[2] azarbadmilad@yahoo.com

Abstract

The most common statistical process control (SPC) tools employed for monitoring process changes are control charts. A control chart demonstrates that the process has altered by generating an out-of-control signal. This study investigates the design of an accurate system for the control chart patterns (CCPs) recognition in two aspects. First, an efficient system is introduced that includes two main modules: feature extraction module and classifier module. In the feature extraction module, a proper set of shape features and statistical feature are proposed as the efficient characteristics of the patterns. In the classifier module, several neural networks, such as multilayer perceptron(MLP), adaptive neuro-fuzzy inference system (ANFIS) and radial basis function (RBF) are investigated. Based on an experimental study, the best classifier is chosen in order to recognize the CCPs. Second, a hybrid heuristic recognition system is introduced based on improved bee algorithm (IBA) to improve the generalization performance of the classifier. The simulation results show that the proposed algorithm has high recognition accuracy.

Keywords*: Control chart patterns; Neuro-fuuzy; IBA; Shape feature; Statistical feature*

1. Introduction

In modern industrial and service organization, control charts have been utilized extensively. Different kinds of control charts have been recently improved according to various quality characteristics and control goals. For the first time, Shewhart proposed using control charts for monitoring process fluctuation in 1924. It is supposed that in a process fluctuation, assignable causes make abnormal changes, while non-assignable causes lead to normal changes. Consequently, automatically recognizing Control Chart Pattern (CCP) is a necessary subject to identify efficiently the process fluctuation [1].

In recent years, several studies have been performed for recognition of the unnatural patterns. Some of the researchers used the expert systems [2, 3]. The advantage of an expert system or rule-based system is that it contains the information explicitly. If required, the rules can be modified and updated easily. However, the use of rules based on statistical properties has the difficulty that similar statistical properties may be derived for some patterns of different classes, which may create problems of incorrect recognition. Also, Artificial Neural Networks (ANNs) have been widely applied for classifiers. ANNs can be simply categorized into two groups comprising supervised and unsupervised. Most researchers [4- 6] 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 [7, 8]. 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. One disadvantage with neural network is the difficulty in understanding how a particular classification decision has been reached and also in determining the details of how a given pattern resembles with a particular class. In addition, there is no systematic way to select the topology and architecture of a neural network. In general, this has to be found empirically, which can be time consuming.

Most the existing techniques used the unprocessed data as the inputs of CCPs recognition system. The use of unprocessed CCP data has further many problems such as the amount of data to be processed is large. On the other hand, the approaches which use features are more flexible to deal with a complex process problem, especially when no prior information is available. If the features represent the characteristic of patterns explicitly and if their components are reproducible with the process conditions, the classifier recognition accuracy will increase [9]. Further, if the feature is amenable to reasoning, it will help in understanding how a particular decision was made and thus makes the recognition process a transparent process. Features could be obtained in various forms, including principal component analysis shape features [10, 11], multi-resolution wavelet analysis [12, 13] and statistical features [14]. Pham and Wani [10] introduced feature based control chart pattern recognition. Nine geometric features were proposed: slope, number of mean crossings, number of least-square line crossings, cyclic membership, average slope of the line segments, slope difference, and three different measures for area. The scheme was aimed at improving the performance of the pattern recognizer by presenting a smaller input vector (features). Gauri and Chakraborty [11] also present a set of seven most useful features that are 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 are recognized using heuristic and ANN techniques. Chen *et al*. [12] 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 first preprocessed by a wavelet transform to decompose the concurrent patterns into different levels or patterns, and then the corresponding features are fed into back-propagation ANN classifiers for pattern recognition. The wavelet transform was studied as the input of SVM for CCP recognition in [13]. A Genetic Algorithm (GA) was utilized to improve the recognition performance of SVM. Hassan *et al*. [14] conducted an experimental study to use BPNs for identifying six types of basic SPC patterns, where the performances of two BPN recognizers using statistical features and raw data as input feature, respectively, were compared. The results indicated that the BPN using statistical features as input vectors has better performance than those of the other BPN using raw data as input vectors. Based on the published papers, there exist some important issues in the design of automatic CCPs recognition system which if suitably addressed, lead to the development of more efficient recognizers. One of these issues is the extraction of the features. In this paper for obtaining the compact set of features which capture the prominent characteristics of the CCPs in a relatively small number of the components, the statistical and shape features are applied. These features are presented in Section 2.

This study investigates the design of an accurate system for the control chart patterns (CCPs) recognition in two aspects. First, an efficient system is introduced that includes two main modules: feature extraction module and classifier module. In the feature extraction module, a proper set of shape features and statistical feature are proposed as the efficient characteristics of the patterns. In the classifier module, several neural networks, such as multilayer perceptron, probabilistic neural network and radial basis function are investigated. Based on an experimental study, the best classifier is chosen in order to recognize the CCPs. Second, a hybrid heuristic recognition system is introduced based on cuckoo optimization algorithm (COA) algorithm to improve the generalization performance of the classifier.

The rest of paper is organized as follows. Section 2 explains the feature extraction. Section 3 describes neural networks. Section 4 presents the COA. Section 5 describes the proposed method. Section 6, shows simulation results and finally Section 7 concludes the paper.

2. Feature Extraction

Automatically recognizing control chart patterns (CCPs) is an essential issue for identifying the process fluctuation effectively. CCPs can exhibit six common types of pattern: normal (NOR), cyclic (CYC), increasing trend (IT), decreasing trend (DT), upward shift (US), and downward shift (DS). Except for normal patterns, all other patterns indicate that the process being monitored is not functioning correctly and requires adjustment. Figure 1 shows these six types of patterns [1].

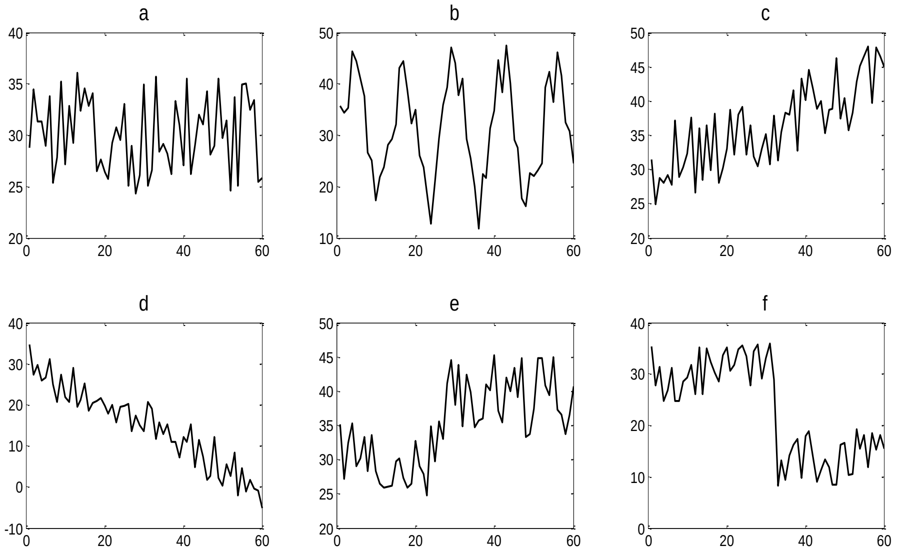

Figure 1. Six Various Basic Patterns of Control Charts: (a) Normal, (b) Cyclic, (c) Upward Trend, and (d) Downward Trend, (e) Upward Shift, (f) Downward Shift

In this study, the real Control Chart Pattern database was used and analyzed. They have been collected in the Mehr Cartoon Company. This is located in the north of Iran as one of the cartoon producer. We used the six common types of records for recognition. 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 signal recognition area, choosing the 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 features that consists of both shaping and statistical information of the CCPs. Figures 2-8 show these features. These features are briefly described as follows.

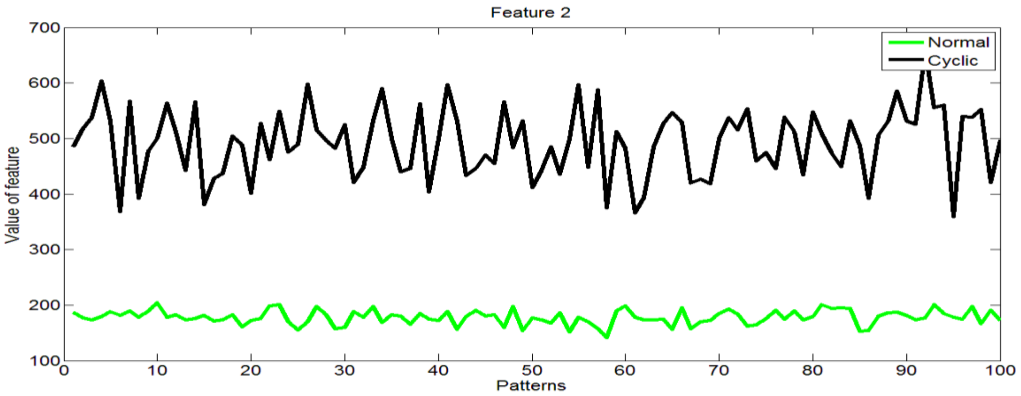

Figure 2. APML of Normal and Cyclic Patterns. The Area between the Pattern and the Mean Line. The APML is Lowest for a Normal Pattern. Thus, this Feature Differentiates between Normal and Other Patterns

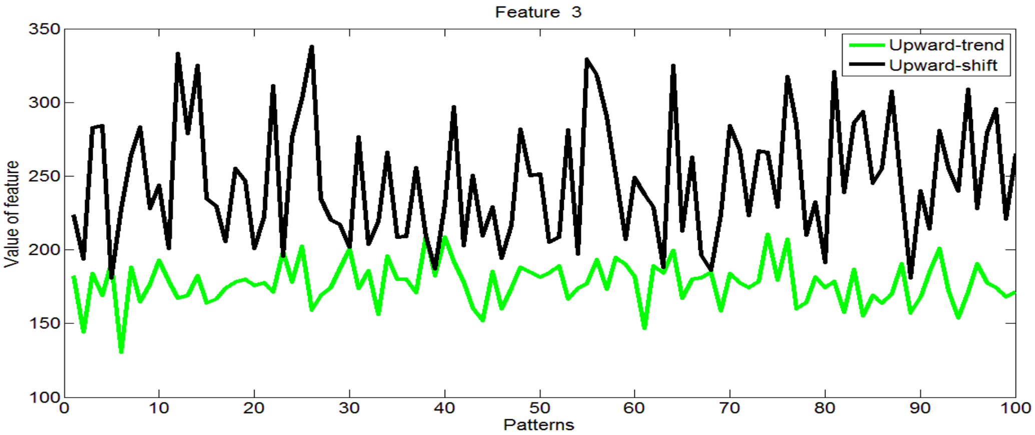

Figure 3. APSL of Upward-Trend and Upward-Shift Patterns. The Area between the Pattern and its Least-Square Line. Cyclic and Shift Patterns Have a Higher APSL Value than Normal and Trend Patterns and therefore the APSL can be used to Differentiate Cyclic and Shift Patterns from Normal and Trend Patterns

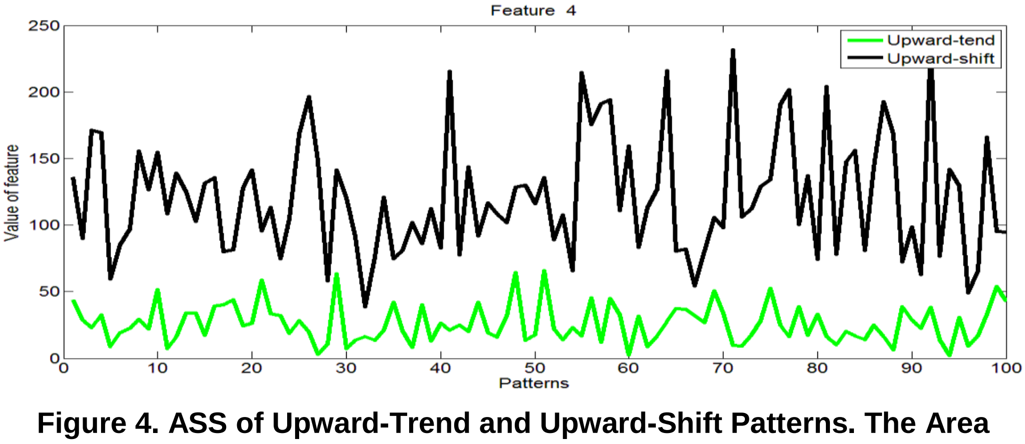

Figure 4. ASS of Upward-Trend and Upward-Shift Patterns. The Area between the Least-Square Line and the Line Segments. The Value of this Feature is Approximately Zero for a Trend Pattern and is Higher for a Shift Pattern. This Feature thus Differentiates Trend Patterns from Shift Patterns

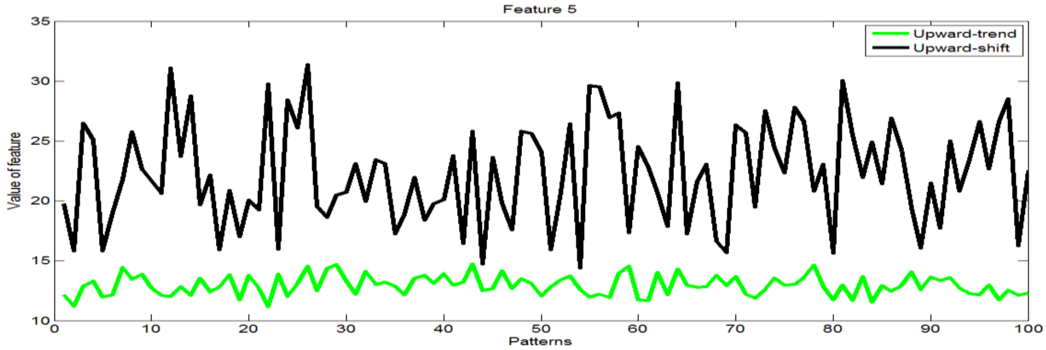

Figure 5. MVSASTI of Upward-Trend and Upward-Shift Patterns. The Maximum Value of Variation in Signal Amplitude in a Short Time Interval

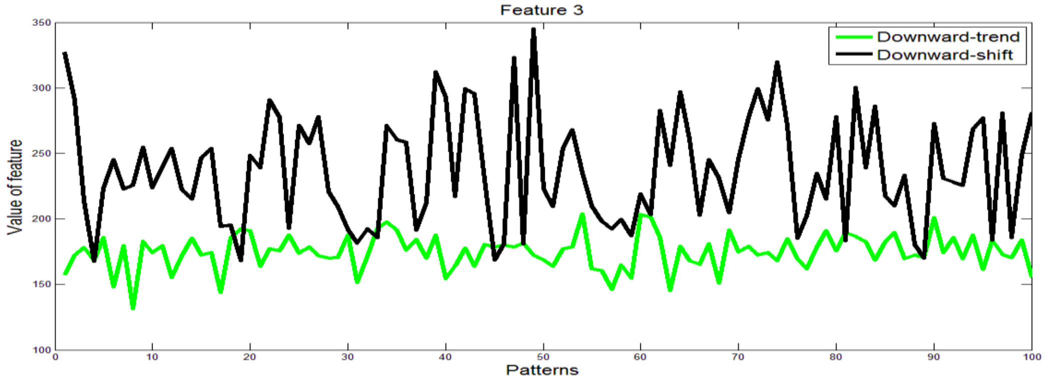

Figure 6. APSL of Downward-Trend and Downward-Shift Patterns. The Area between the Pattern and its Least-Square Line. Cyclic and Shift Patterns Have a Higher APSL Value than Normal and Trend Patterns and therefore the APSL can be used to Differentiate Cyclic and Shift Patterns from Normal and Trend Patterns

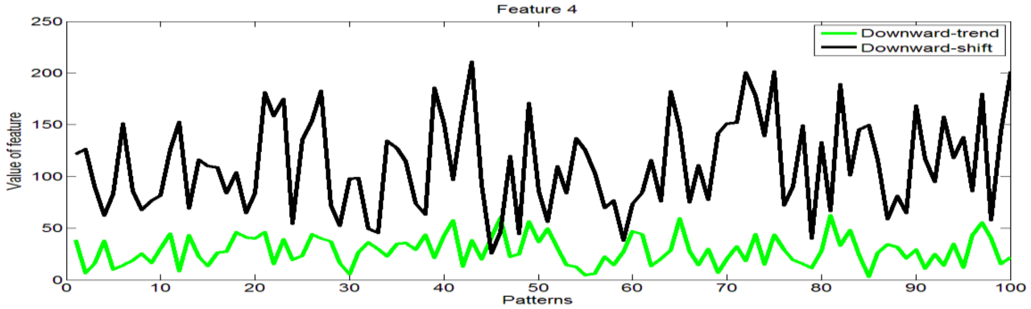

Figure 7. ASS of Downward-Trend and Downward-Shift Patterns. The Area between the Least-Square Line and the Line Segments. The Value of this Feature is Approximately Zero for a Trend Pattern and is Higher for a Shift Pattern. This Feature thus Differentiates Trend Patterns from Shift Patterns

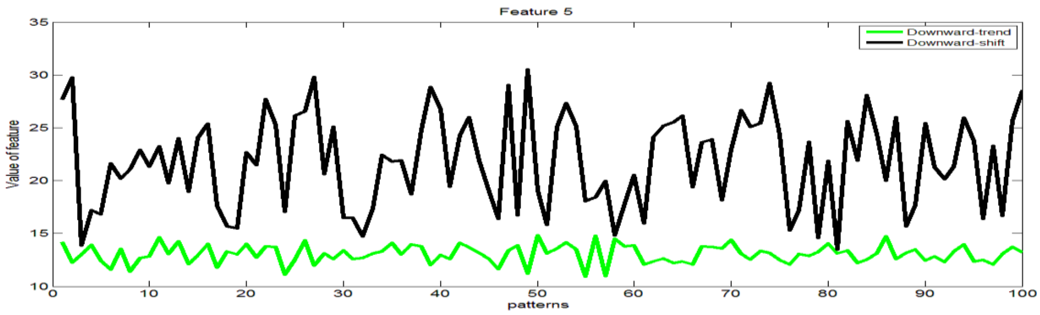

Figure 8. MVSASTI of Downward-Trend and Downward-Shift Patterns. The Maximum Value of Variation in Signal Amplitude in a Short Time Interval (Proposed Shape Feature)

2.1. Statistical Feature

Some statistical features are mean, standard deviation, skewness, kurtosis, and autocorrelation. In this paper we have used mean as feature1. Its mathematical form is shown below:

$$mean = \frac{\sum_{i=1}^{n} X_i}{n} \tag{1}$$

Where X_i represent the input (reference) vector and n is the total length of the observing window.

2.2. Shape Features

In [10], the authors have introduced nine shape features for discrimination of the CCPs. In this paper, three of these features are considered as features 2, 3 and 4. Feature 5 is proposed in this paper.

Feature 2 (APML): the area between the pattern and the mean line. The APML is lowest for a normal pattern. Thus, this feature differentiates between normal and other patterns.

Feature 3, 6 (APSL): the area between the pattern and its least-square line. Cyclic and shift patterns have a higher APSL value than normal and trend patterns and therefore the APSL can be used to differentiate cyclic and shift patterns from normal and trend patterns.

Feature 4, 7 (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.

Feature 8 (MVSASTI): the maximum value of variation in signal amplitude in a short time interval (proposed shape feature). As it is depicted in Figure 1, the maximum variation of signal amplitude in a time interval of 60 second is approximately identical in both shift and trend patterns. But in shift pattern, there is a large amount of variation in signal amplitude in a part of signal and in a short time interval which is not existed in the trend pattern. This difference could be used for separating these two patterns. So, the maximum of signal variation is calculated in a short time interval like T. beginning from t=0, the maximum value of signal variation is calculated, from t=0 to t= t+T. This process is proceeded for t=1,…, 60-T. Finally the maximum variation in different time interval is selected as a desired feature. T could be any number between 5 to 10. This shape feature is new in this area. In this study t=7 sec is selected.

3. The Classifiers

An MLP neural network consists of an input layer (of source nodes), one or more hidden layers (of computation nodes) and an output layer. The recognition basically consists of two phases: training and testing. In the training stage, weights are calculated according to the chosen learning algorithm. The issue of learning algorithm and its speed is very important for the MLP model. In this study the following learning algorithms are considered [4].

3.1.1. Back-Propagation with Momentum (BP with Momentum)

The BP algorithm makes use of gradient descent with a momentum term to smooth out oscillation [4]. Eq. (2) gives the weight update for BP with momentum:

$$\Delta w_{ij}(t+1) = \varepsilon \frac{\partial E(t)}{\partial w_{ij}(t)} - \mu \frac{\partial E(t-1)}{\partial w_{ij}(t-1)} \tag{2}$$

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

3.1.2. Resilient Back-propagation (RPROP) Algorithm

RPROP considers the sign of derivatives as the indication for the direction of the weight update [18-19]. In doing so, the size of the partial derivative does not influence the weight step. The following equation shows the adaptation of the update values of Δ_{ij} (weight changes) for the RPROP algorithm. For initialization, all are set to small positive values:

$$\Delta_{ij}(t) = \begin{cases} \eta^{+} \times \Delta_{ij}(t-1); & if \ \frac{\delta E}{\delta W_{ij}}(t-1)\frac{\delta E}{\delta W_{ij}}(t) > 0 \\ \eta^{-} \times \Delta_{ij}(t-1); & if \ \frac{\delta E}{\delta W_{ij}}(t-1)\frac{\delta E}{\delta W_{ij}}(t) < 0 \\ \eta^{0} \times \Delta_{ij}(t-1); & otherwise \end{cases} \tag{3}$$

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

$$\Delta W_{ij}(t)=\begin{cases}-\Delta_{ij}; & if\ \dfrac{\delta E}{\delta W_{ij}}(t)>0\\ +\Delta_{ij}; & if\ \dfrac{\delta E}{\delta W_{ij}}(t)<0\\ 0\quad ; & otherwise\end{cases}\tag{4}$$

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

3.2. Radial Basis Function Neural Network (RBFNN)

RBF neural networks with their structural simplicity and training efficiency are a good candidate to perform a nonlinear mapping between the input and the output vector space [5]. RBFNN is fully connected feed forward structure and consist of three layers namely, an input layer, a single layer of nonlinear processing units, and an output layer. Input layer is composed of input nodes that are equal to the dimension of the input vector *x*. The output of the *j* th hidden neuron with Gaussian transfer function can be calculated as:

$$h_j=\exp(-\left\|x-c_j\right\|^2 \big/ \sigma^2)\tag{5}$$

Where is h_j the output of the *Jth* neuron, $x\in\Re^{n\times1}$ is an input vector, $c_j\in\Re^{n\times1}$ is the *jth* RBF center, σ the center spread parameter, which controls the width of the RBF. The output of any neuron at the output layer of RBF network is calculated as:

$$y_i=\sum\nolimits_{j=1}^{k} w_{ij}h_j\tag{6}$$

Where w_{ij} is the weight connecting hidden neuron *j* to output neuron i and k the number of hidden layer neurons.

3.3. Adaptive Neuro-fuzzy Inference System (ANFIS)

Fuzzy if-then rules that contain the parts premise and consequent of Sugeno type fuzzy inference system has been applied in the structure of ANFIS [15]. The assumption in this system is that the interference system has two inputs x and y and one output f. For a first order Sugeno fuzzy model the rule basis including two fuzzy if-then rules is as follows:

$$If\_x\_is\_A\_and\_y\_is\_B\_then\_Z\_is\_f(x,y)\tag{7}$$

Where *A* and *B* are the fuzzy sets in the records and at the consequent, $Z=f(x,y)$ is a crisp function. Generally for the input variables *x* and *y*, $f(x,y)$ is a polynomial. But as ascertained formerly, also there could be any other function that approximately explains the system within the fuzzy region. If $f(x,y)$ is a constant, there will be a zero order Sugeno fuzzy model, considered as a special case of Mamdani fuzzy inference system. When $f(x,y)$ is a first order polynomial, there will be a first order Sugeno fuzzy model. For a first order two-rule Sugeno fuzzy inference system, the two rules are as follows:

$$Rule1: If\_x\_is\_A_1\_and\_y\_is\_B_1\_then\_f_1=p_1x+q_1y+r_1\tag{8-1}$$

$$Rule2: If\_x\_is\_A_2\_and\_y\_is\_B_2\_then\_f_2=p_2x+q_2y+r_2\tag{8-2}$$

where within the fuzzy zone, x and y are the inputs, A_1 , A_2 and B_1, B_2 are the fuzzy sets, f_1 , f_2 are the outputs which is characterized by the fuzzy rule and p_i , r_i , q_i are linear and concerning as consequent parameters of the node noted and specified within the training operation. Then, weighted average of each rule's output will be the final output. There is the relevant equivalent ANFIS structure shown in Figure 9.

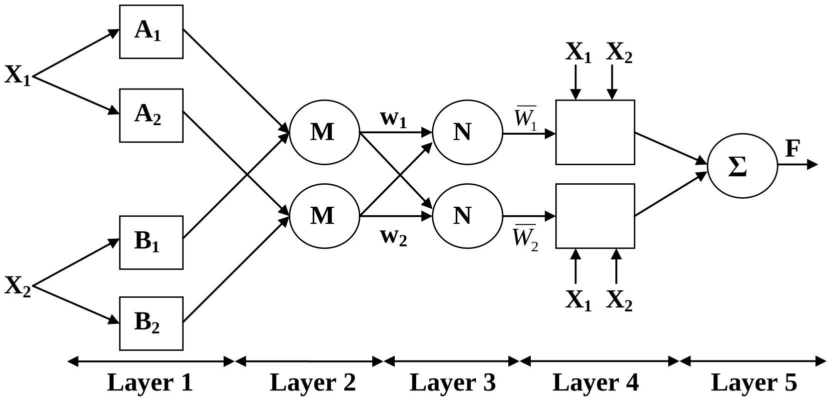

Figure 9. The Structure of ANFIS

Here is the description of particular layers of this ANFIS structure as follows:

Layer 1: in this layer there is a suitable node function for neuron ith. The node function which describes the relationship between the input and output functions is as below:

$$O_i^1 = \mu_{A_i}(x) \tag{9}$$

where x is the input to neuron *ith*, A_i the linguistic variable correlated with this node function, and μ_{A_i} is approximately the membership function of A_i and Usually is given by:

$$\mu_{A_i}(x) = \frac{1}{1+[(x-c_i/a_i)^2]^{b_i}} \tag{10}$$

Or

$$\mu_{A_i}(x) = \exp\left\{-(\frac{x-c_i}{a_i})^2\right\} \tag{11}$$

where x is the input and $\{a_i, b_i, c_i\}$ is parameter set for the premise.

Layer 2: In this layer, each node is a fixed node and computes the firing strength w_i for a rule. The output of each node would be the product of all the incoming signals to it as shown below:

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(x) \tag{12}$$

Layer 3: In this layer every node is a fixed node. Each node computes the ratio firing strength of the *ith* rule to the sum of firing strengths of all. The output from the node *ith* is the normalized firing strength which is given by:

$$O_i^3 = \overline{w} = \frac{w_i}{w_1 + w_2} \tag{13}$$

Layer 4: In this layer every node is suitable with a node function that is given by:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \tag{14}$$

where $\overline{w}_i$ is the output of Layer 3 and $\{p_i, q_i, r_i\}$ is the consequence parameter set.

Layer 5: This layer includes only one specified node. The overall output would be computed as the summation of all incoming signals as shown below:

$$O_i^5 = Overall\_output = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{15}$$

The fact that the output is the weighted sum of the results of the rules is remarkable. The amount of fuzzy sets is specified by the number of nodes in layer 1. Besides, the dimension of layer 4 specifies the number of fuzzy rules which is used in the architecture and exhibits the complication and flexibility of the ANFIS architecture.

3.3.1. Learning Algorithm

The proposed ANFIS structure shows that if the values of premise parameters are given, the final output would be shown as a linear combination of the consequent parameters. In Figure 4, the output f can be written as:

$$\begin{gathered} f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_1}{w_1 + w_2} f_2 = \bar{w}_1 f_1 + \bar{w}_2 f_2 = \\ \bar{w}_1 (p_1 x + q_1 y + r_1) + \bar{w}_2 (p_2 x + q_2 y + r_2) \end{gathered} \tag{16}$$

where f is linear in the consequent parameters $\{p_i, q_i, r_i\}$. In the forward procedure of the training algorithm, the least squares estimate is applied to identify the consequent parameters. But in the backward pass, the derivative of the squared error with respect to each node output, named error signals, duplicate backward from the output layer to the input layer. In this pass, the premise parameters would be updated by the gradient descent algorithm [15-16].

4. Optimization of the Classifier

Bees Algorithm (BA) is an optimization algorithm inspired by the natural behavior of honey bees to find food. The structure of BA is design to find the right solution in a search space. Table 1 shows the pseudo code for the algorithm in its simplest form. More details about bees algorithm and its relations described in [17].

Table 1. The Procedure of the Bees' Algorithm

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 and evaluate fitness
6. Select the fittest bee from each site
7. Assign remaining bees to search randomly and evaluate their fitness
8. End While

4.1. Improved Bees Algorithm (IBA)

We are proposed an improved bee algorithm (IBA) to expedite find the considered answer and enhance the performance of the original bee algorithm. In the BA, *"ngh"* defines the initial size of the neighborhood in which follower bees are placed. For example, if x is the position of an elite bee in the ith dimension, follower bees will be placed randomly in the interval $X_{ie} \pm ngh$ in that dimension at the beginning of the optimization process. Also, the Size of a patch will remain unchanged until the local research process has the higher values of fitness. If the fitness value obtained in the local investigation has not improved, in this case, the size of a $a_1,\ldots,a_n$ will reduce. To update the values of the neighborhood size, we use the following equation:

$$ngh(t+1) = 0.8 \times ngh(t) \tag{17}$$

where t refers to the tth iteration of main loop of the BA. Therefore, succeeding this strategy, the local search is initially determined over a great neighborhood and has a mostly explorative feature. During the execution of the algorithm, to refine the current local optimum, a more detailed probe is required. Hence, the search is made increasingly exploitative, and the area around the optimum is searched more meticulously.

Due to non-linearity and high complexity of the BA, reducing the size of the cemetery without getting feedback from elite bee’s fitness cannot show exactly the actual performance of the search process. At the start of the search, since bees are relatively large distance with optimal patches and so for faster to converge to the optimum solutions, a large size of patch is intended for globally search in the search space. Subsequently, after several iterations when the best point discovered by the elite bees highly improves and the bees reached a near optimum solution, just small motions are required and patch size must change the size to smaller values. Accordingly, in this paper, we proposed an improved bee algorithm (IBA) in which unlike the origin BA are subject to change and according to the conditions will change. The patch sizes consider as a function of elite bees’ fitness during as follows:

$$ngh_i^t = \frac{1}{t} \times \frac{ngh}{1+\exp(-F(elite_i^t))} \tag{18}$$

Where the function $F(elite_i^t)$ describes the fitness of *i*th elite bee in *t*th iteration and ngh_i^t is the *i*th elite bee's size of the neighborhood in *t*th iteration. Here, the rate of elite bee fitness improvement will change to fit the patch size. As expressed in formula (18), during the search of IBA, due to have a strong global search abilities and locating the promising search areas, the patch size will be large while the fitness of a elite bee is far away from the real global optimal. Subsequently, the patch size becomes smaller the closer to the optimum fitness of an elite bee, depending on the nearness of its fitness to the optimal value, to facilitate a finer local explorations and hence accelerate convergence.

The major difference between the improved algorithm and the origin is in defining the size of patch. In the BA, The patch size is equal for all elite bees; meanwhile in IBA, the patch size is related to fitness value and each elite bee has its own patch size which is evaluated by Equation (18).

5. The Proposed Method

The ANFIS model was developed using MATLAB Fuzzy Logic Toolbox (R2014a). In this study, the Sugeno-type fuzzy inference system was implemented to obtain a concise representation of a system's behavior with a minimum number of rules. The linear least square estimation was used to determine each rule's consequent equation. A radius value was given in the MATLAB program to specify the cluster center's range of influence to all data dimensions of both input and output. If the cluster radius was specified a small number, then there will be many small clusters in the data that results in many rules. In contrast, specifying a large cluster radius will yield a few large clusters in the data resulting in fewer rules. In this study IBA-ANFIS is proposed to find the optimum vector of radius. Figure 10 shows a sample bee. In this figure, *p* denotes the number of input-output variables.

Bee= [radius1 , radius2 ,? radiusp]

Figure 10. A Sample Vector of Bee

Based on the above descriptions, the flowchart of the IBA-ANFIS algorithm used in this paper is shown in Figure 11. The proposed Sugeno-type ANFIS structure includes: "Gaussian" for the input membership function, "linear" for the output membership function, Mamdani product implication and weighted average is used as the defuzzifier, experimentally.

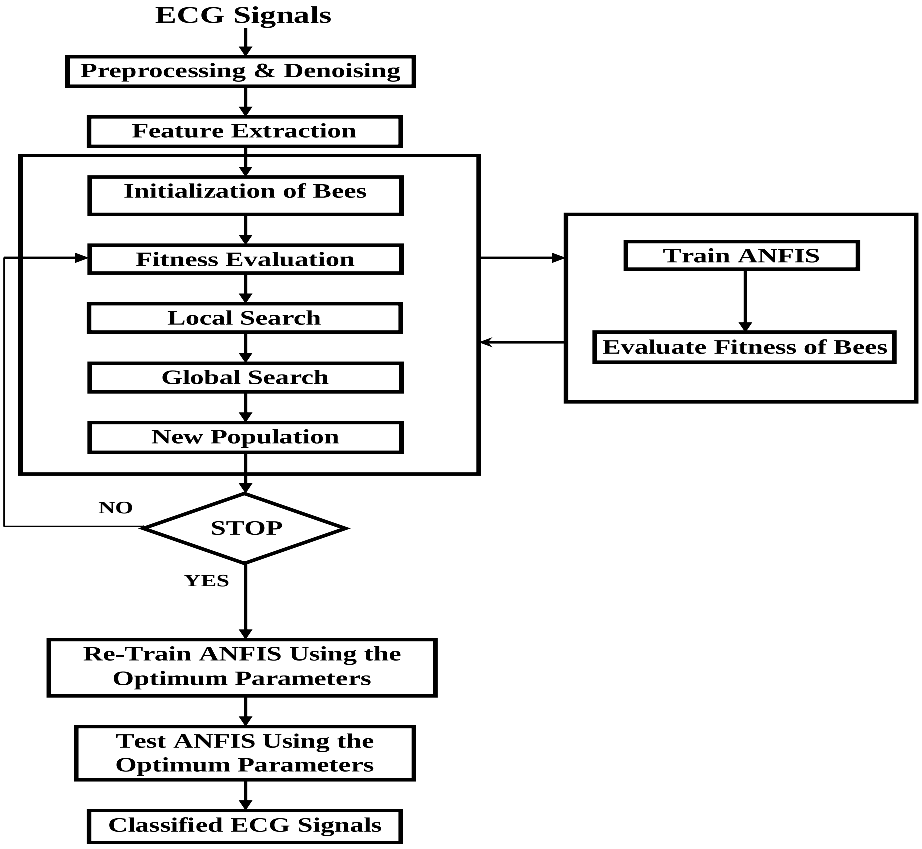

Figure 11. Flowchart of the Proposed Method

6. Simulation Results

In this section we evaluate the performance of proposed recognizer. For this purpose we have used the practical and real data. This dataset contains 600 examples of control charts. For this study, we have used 60% of data for training the classifier and the rest for testing. 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 (RA) of system, one needs to compute the average value of the performances of the CCPs. All the obtained results are the average of 50 independent runs.

In the next step, the IBA is applied to find the optimal parameters of the ANFIS. Table 2 shows the coefficient values in the IBA algorithm. Table 3 shows the recognition accuracies obtained for the proposed algorithm. According to the Table, it can be found that the RA is improved to 99.93% and the recognizer performance has been become significantly better. As can be found, the best performance is related to the feature 5 separately with around 99.71%. When we used features 3, 4 and 5 together, the greatest recognition accuracy was achieved.

Table 2. Set Default Values of the Parameters Used in the Proposed Improved Bees Algorithm

IBA Parameters	Values
Number of scout bees, n	26
Number of sites selected for neighborhood search, m	4
Number of best “elite” sites out of m selected sites, e	6
Number of bees recruited for best e sites, nep	2
Number of bees recruited for the other (m-e) selected sites, nsp	2
Maximum number of iterations, R	150

Table 3. Performance of the Proposed System without Optimization

Classifier	RA (%) with different features (features 3, 4 and 5)						
	3	4	5	3, 4	3,5	4,5	3,4,5
MLP (BP with momentum)	94.43	95.66	95.03	96.54	96.71	96.84	96.12
MLP (RPROP)	95.26	96.54	95.87	96.86	97.82	97.21	96.28
RBFNN	96.11	96.43	96.54	96.76	97.67	96.77	97.52
ANFIS	97.40	97.19	97.35	98.03	97.94	97.72	98.36
The Proposed ANFIS-IBA	98.04	98.23	98.71	99.36	99.53	99.57	99.72

In order to indicate the details of the recognition for each pattern, the confusion matrix of the recognizer is shown by Table 4. 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. For example, look at the third row of this matrix. The value of 99.61% shows the percentage of correct recognition of upward trend pattern and the value of 1.6% shows that this type of pattern is wrongly recognized with upward shift pattern. In order to achieve the recognition accuracy (RA) of system, it is needed to compute the average value of that appears in diagonal.

Table 4. Confusion Matrix for 50 Independent Runs

	NR	CC	UT	DT	US	DS
NR	100	0	0	0	0	0
CC	0	100	0	0	0	0
UT	0	0	99.26	0	0.74	0
DT	0	0	0	98.4	0	1.6
US	0	0	0	0	100	0
DS	0	0	0	0	0	100

For comparison purposes, Table 5 gives a comparison between the classification accuracies of our method and previous methods applied to the same database. As can be seen from the results, proposed method obtains an excellent accuracy.

Table 5. A Summary of Different Classification Algorithms Together with their Reported Results Used Measures of the Accuracy

Ref. no	Classifier	RA (%)
[18]	PNN	95.58
[9]	WSVM	98.95
[9]	SVM	98.37
[19]	MLP-BP	92
[9]	PSO-SVM	95.58
[20]	Wavelet-SVM-GA	99.37
[21]	Hachicha and Ghorbel	95.59
The proposed method	ANFIS-IBA	99.72

7. Conclusion

The Control Chart Patterns (CCPs) are important statistical process control tools to determine whether a process is run in its intended mode or in the presence of unnatural patterns. This study has investigated the design of an automatic and accurate system for the recognition of CCPs. In this study, a hybrid system which is composed of two decision levels is proposed for the recognition of the control chart patterns. The complexity of the proposed system is much less than others works. Using the two-level decision system and different kinds of neural networks as a classifier, the obtained highest accuracy is 98.36%. Then, in order to increase the recognition accuracy, the important parameters of the Adaptive Neuro-fuzzy Inference System (ANFIS), *i.e.* the number of the radial basis functions, centers and spread of these functions, are optimized by using IBA. The simulation results showed that the proposed method increases the recognition accuracy to 99.72 %, using the neural network structure optimization.

References

[1] D. C. Montgomery, “Introduction to Statistical Quality Control”, 5th ed., John Wiley, Hoboken, NJ, USA, **(2005)**.

[2] 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, vol. 28, no. 1, **(1995)**, pp. 81–91.

[3] J. R. Evans, W. M. Lindsay, “A framework for expert system development in statistical quality control”, Computers and Industrial Engineering, vol. 14, no. 3, **(1988)**, pp. 335-343.

[4] A. Ebrahimzadeh, V. Ranaee, “Control chart pattern recognition using an optimized neural network and efficient features”, ISA Transactions, vol. 49, **(2010)**, pp. 387–393.

[5] V. Ranaee, A. Ebrahimzadeh, “Control chart pattern recognition using neural networks and efficient features: a comparative study”, Pattern Analysis & Applications, **(2011)**.

[6] A. Ebrahimzadeh, J. Addeh, Z. Rahmani, “Control chart pattern recognition using K-MICA clustering and neural networks”, ISA Transactions, vol. 51, **(2012)**, pp. 111-119.

[7] R. S. Guh, Y. R. Shiue, “Online identification of control chart patterns using self-organized approaches”, International Journal of Production Research, vol. 43, **(2005)**, pp. 1225-1254.

[8] 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, vol. 45, no.1, **(2007)**, pp. 227-244.

[9] V. Ranaee, A. Ebrahimzadeh, R. Ghaderi, “Application of the PSO_SVM model for recognition of control chart patterns”, ISA Transactions, vol. 49, no. 4, **(2010)**, pp. 577-586.

[10] D. T. Pham, M. A. Wani, “Feature-based control pattern recognition”, International Journal of Production Research, vol. 35, no. 7, **(1997)**, pp. 1875-1890.

[11] S. K. Gauri., S. Chakraborty, “Improved recognition of control chart patterns using artificial neural networks”, Int J Adv Manuf Technol, vol. 36, **(2008)**, pp. 1191-1201.

[12] Z. Chen, S. Lu, S. Lam, “A hybrid system for SPC concurrent pattern recognition”, Advanced engineering informatics, vol. 21, **(2007)**, pp. 303-31.

[13] V. Ranaee, , A. Ebrahimzadeh, “Control chart pattern recognition using a novel hybrid intelligent method”, Applied Soft Computing, vol. 11, no. 2, **(2011)**, pp. 2676-2686.

[14] A. Hassan, M. S. Nabi Baksh., A. M. Shaharoun, H. Jamaluddin, “Improved SPC chart pattern recognition using statistical features”, International Journal of Production Research, vol. 41, no. 7, **(2003)**, pp. 1587—1603.

[15] M. Buragohain, C. Mahanta, “A novel approach for ANFIS modeling based on full factorial design”, Applied Soft Computing, vol. 8, **(2008)**, pp. 609–625.

[16] M. Azarbad, A. Ebrahimzade, M. Yousefi, “A comparison of clustering algorithm for recognition of radio communication signals based on the HOS”, Proceedings of the IASTED Signal Processing, Pattern Recognition, and Applications, vol. 721, **(2011),** 221-228.

[17] D.T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, M. Zaidi, “The Bees Algorithm, a novel tool for complex optimization problems”, Intelligent Production Machines and Systems, vol. 12, **(2006)**, pp.454-459.

[18] Z. Cheng, Y. Ma, “A research about pattern recognition of control chart using probability neural network”, in: Proc. ISECS, **(2008)**, pp. 140–145.

[19] P. Xanthopoulos, T. Razzaghi, “A weighted support vector machine method for control chart pattern recognition”, Computers & Industrial Engineering.

[20] vol. 70, **(2014)**, pp. 134-149.

[21] V. Ranaee, A. Ebrahimzadeh, “Control Pattern Chart Recognition Using a Novel Intelligent Method”, vol. 49, **(2011)**, pp. 577-586.

[22] W. Hachicha, A. Ghorbel, “A survey of control-chart pattern-recognition literature based on a new conceptual classification scheme”, Computers & Industrial Engineering, vol. 63, **(2012)**, pp. 204-222.