Int J Adv Manuf Technol (2015) 76:311–327
DOI 10.1007/s00170-014-6236-2

ORIGINAL ARTICLE

On artificial neural networking-based process monitoring under bootstrapping using runs rules schemes

Babar Zaman · Muhammad Riaz · Shabbir Ahmad · Saddam Akber Abbasi

Received: 14 February 2014 / Accepted: 4 August 2014 / Published online: 30 August 2014

Abstract Control charts are popular tools in statistical process control (SPC) and artificial neural network (ANN) technique is an attractive alternative for efficient monitoring of process parameters. This study uses the artificial neural network technique with back propagation method to process control system for dispersion parameter. We have trained an artificial neural network to be used in statistical control charts using varying runs rules schemes. By investigating the performance of trained artificial neural network under normal and bootstrapping environments we have made comparisons of the usual ANN and three runs rules-based schemes for ANN to gain the precision of process. We have used average run length (ARL), extra quadratic loss (EQL), relative ARL (RARL), and performance comparison index (PCI) measures and explored the said structures of trained artificial neural network under bootstrapping by implementing runs rules schemes. We have also suggested a modification in the trained ANN for variance change detection. An example with real data is also given for practical considerations.

Keywords Artificial neural network (ANN) · Average run length (ARL) · Bootstrapping · Extra quadratic loss (EQL) · Performance comparison index (PCI) · Process spread · Quality control · Relative ARL (RARL) · Runs rules

B. Zaman
King Khaled Eye Specialist Hospital, Riyadh, Kingdom of Saudi Arabia

M. Riaz · S. A. Abbasi
Department of Mathematics and Statistics, King Fahad University of Petroleum and Minerals, Dhahran 31261, Kingdom of Saudi Arabia

S. Ahmad (✉)
Department of Mathematics, COMSATS Institute of Information Technology, Wah Cantt 47040, Pakistan
e-mail: shabbirahmad786@yahoo.com

1 Introduction

Statistical process control (SPC) charts is a collection of very powerful tools to monitor and ensure stability of the parameters of a process (manufacturing or non-manufacturing). Control chart is the most important and widely used technique in SPC tool kit. There are mainly three types of control charts namely Shewhart, EWMA, and CUSUM. The major limitation of the Shewhart type control charts is that these are not very efficient in detecting small process shifts. To increase the sensitivity of Shewhart control charts for small shifts, additional supplementary runs rules, in literature, have been suggested at the cost of inflated false alarm rate, however some remedial measures have also been proposed by different researchers. Moreover, to address particularly the smaller shifts, EWMA and CUSUM type charts serve the purpose efficiently.

Artificial neural network (ANN) is also currently a technique popular to be used for monitoring process parameters as an alternative to SPC control charts due to its application and superior performance. These days ANNs are widely used in different fields like industry, banking, marketing, medical, etc. They are considered as efficient successors of charting methods in modern era. A lot of work has been done in the field of artificial neural network for process monitoring. Pugh [35] provided comparisons of neural networking with SPC control charting methodology. Guo and Dooly [17, 18] used the neural network approach for process monitoring and quality improvement. Velasco and Rowe [46] used the back propagation artificial neural networks in the analysis of quality control charts to recognize patterns that indicate out-of-control situations as specified by the Western Electric Company [47] rules. Smith [45] trained ANN to discriminate between samples from probability distributions considered within control limits and those which have shifted in both location and variance. Neural networks are also trained to recognize samples and to predict future points from processes which exhibit long-term or cyclical drift. In Bayesian framework, de Freitas [14] used the Bayesian statistics to find the proper structure of neural network. Chang and Ho [10] used two-stage artificial neural network approaches with back propagation method for variance change detection and classification. Sagiroglu et al. [42] used neural networks for control charts pattern recognition.

Junsub et al. [21] used the neural network method for a non-normal data to detect the process control shifts for location parameter. Perry et al. [34] used back propagation artificial neural networks for the purpose of control chart pattern recognition. Perry and Pignatiello Jr [33] used the technique of neural network to recognition of pattern behavior of on-line automated process analysis. Sigut et al. [44] used the artificial neural network method to test the normality of data rather than classical method. Muammer et al. [28] used the neural network application for regression modeling purposes. Fioramanti [16] used a comparative approach and predicted sovereign debt crises using artificial neural networks: the papers by Hwarng [20], Chiu et al. [13], Feipeng and Amirkhanian [15], Mahmoudi et al. [25], Shaban et al. [43], Natarajan et al. [29], Mohanty et al. [27] and the references therein may also be seen for further literature. Rowley et al. [40] used the idea of bootstrapping for face detection system, Zio [51] applied bootstrap method for quantifying the uncertainties in the output of supervised neural networks in nuclear industry. Matchenko and Dube [26] used ANN and bootstrapping techniques to test the significance of single nucleotide polymorphisms (SNPs) in the categorization of case control status in genetic association. Kaunga et al. [22] explained about modeling chemical durability of high-level waste glass for nuclear waste processing using bootstrap aggregated neural networks. Raviv and Intrator [36] showed that noisy bootstrap performs best in conjunction with weight-decay regularization and ensemble averaging. On these lines, we may find applications of ANN to identify the pattern of process like increasing trend, decreasing trend, etc., and ANN with bootstrapping technique in literature.

Our study contributes in the direction of ANN using bootstrapping under runs rules schemes to monitor process spread parameter for small, moderate, and large shifts. More specifically, we will train an ANN with back propagation technique for dispersion parameter by implementing runs rules schemes under bootstrapping environments. The said bootstrapped evaluation will help in determining the precision of the ANN structures by taking into account the behavior of consecutive points (in the form of runs rules schemes such as 2/2, 2/3, 3/3).

The organization of the rest of the study is as follows: In Section 2, we provide the procedural details of our trained ANN; Section 3 evaluates the performance of the trained ANN and provides comparisons among the traditional and runs rules-based ANN schemes using bootstrapping technique; Section 4 provides an illustrative example to justify the practicability of proposals with real-life data set while Section 5 presents the conclusions of our study and gives some suggestions for future research.

2 Artificial neural networking approach

Neural networks are used in modeling the relationships among certain variables of interests. They are commonly used in modeling complex relationships between inputs and outputs and also finding the patterns in datasets. There are mainly three types of ANNs namely supervised, unsupervised, and reinforcement learning, which are used for any type of network architecture. Here, we are mainly concerned with monitoring the process variability for both large and small shifts and we have adopted the supervised technique of ANN. For our study purposes, the supervised technique of ANN with back propagation namely multi-layer perception (MLP) network is the most suitable choice since we require different desired outputs according to different inputs. Here, the main problem is to find out the numbers of hidden layers and nodes for each layer, for the minimum error and the choice of learning rate that gives optimal weights for the network. The mathematical details regarding the choice of optimal weights may be seen in Bishop [8, 9].

The general form of the neural network structure is $[n-m-k-O]$ (we will call it topology of ANN in this article) where n is number of inputs, m is number of nodes of first hidden layer, k is number of nodes of second hidden layer, and O is the final output. For the selection of hidden layers and nodes we adapted the rule as described by Chang and Ho [10] and selected the ANN with the ANN topology $[n-12-12-1]$ for our study purposes. An improved monitoring of process parameters is always desirable for all types of manufacturing/non-manufacturing processes and runs rules schemes meet this objective very efficiently. The papers by Klein [24], Khoo [23], Antzoulakos and Rakitzis [6, 7], Abbas et al. [1], Riaz et al. [38, 39], and the references there in may be consulted. The runs rules idea can be applied in the structure of neural network techniques for an efficient process monitoring. Taking inspiration from the above-mentioned works, we suggest here three runs rules schemes to be applied with neural network to monitor the variability parameter. Also, we investigate the performance of runs rules schemes of ANN under bootstrapping environment for practical considerations when limited data is available for study purposes instead of full information on population distribution.

2.1 Runs rules-based ANN

Now, we use the trained ANN (cf. Chang and Ho [10]) for variance change detection to implement runs rules-based schemes for an efficient monitoring of variance. The trained ANN works with q normal distributions consisting of a total of $120\times q$ observations from $N(0,\ \rho^2\sigma_0^2)$ where ρ represents different amounts of variance/shifts. We assume here that x_{ji} (for $j=1,2,\ldots,120$ and $i=1,2,\ldots,n$) for each choice of ($\rho=1,2,\ldots,q$) is the raw data from the said normal distribution. It is to be noted that we have used $q=5$ (for our study purposes) and each distribution with a different variance/shifts will generate 120 observations which results in 600 observations for our trained ANN. Here, we will consider $\sigma_0=1$ throughout our study without loss of generality. The data x_{ji} for each group are transformed to $Z_{ji}=\left|x_{ji}-\overline{x}\right|$ (where $\overline{x}=\frac{1}{n}\sum_{i=1}^{n}x_{ji}$) values which are chosen as input to ANN due to its robustness in statistical tests for equality of variances.

Figure 1 provides the layout of the suggested runs rules-based ANN structure for detecting out-of-control signals for the trained ANN which deals with the five inputs at a time with two hidden layers having 12 nodes for each layer and which produces a single output.

The ANN algorithm can be decomposed in the following four steps.

- Feed–forward computation
- Back propagation to the output layer
- Back propagation to the hidden layer
- Weights updated

In our derivations here x_k is the output from the kth input $z_{1k},z_{2k},z_{3k},z_{4k},z_{5k}$. Based on the output n_{lk} from the lth node in the second hidden layer and weights u_l between −1 and 1 are randomly assigned to them we have the following expression.

$$x_k=f\left(\sum_{l=1}^{12}u_l n_{lk}\right)=\frac{1}{\left(1+e^{-\sum_{l=1}^{12}u_l n_{lk}}\right)},\tag{1}$$

where x^d denoting the desired output, the loss square function is given by:

$$E=\frac{1}{2}\sum_{k=1}^{120}\left(x^d-x_k\right)^2=\frac{1}{2}\sum_{k=1}^{120}e_k^2$$

where $e_k=x^d-x_k$.

Let T denotes the whole set of weights w, v, and u. Given initial set of weights T, which have been randomly set to values between −1 and 1, we find a better set of weights T from the following:

$$T(t+1)=T(t)-h(dE/dT),\tag{2}$$

where dE/dT refers to the first order derivative. This is in fact the well-known substitution method. An alternative method the Newton's method is the following.

$$T(t+1)=T-\left(d^2E/dT^2\right)^{-1}(dE/dT),\tag{3}$$

This is more efficient but requires a second-order derivative.

Optimum weights between output and second hidden layer Following Eq. (1) for optimum weights between output and second hidden layer, we have

$$u_l(t+1)=u_l(t)-h(dE/du_l),\tag{4}$$

The differential of E in Eq. (4) w. r. t weight u_l is given by

$$\begin{aligned}\frac{dE}{du_l}&=-\sum_{k=1}^{120}e_k(dx_k/du_l)\\&=-\sum_{k=1}^{120}e_k f'\left(\sum_{l=1}^{12}u_l n_{lk}\right)\frac{d}{du_l}\left(\sum_{l=1}^{12}u_l n_{lk}\right)\\&=-\sum_{k=1}^{120}e_k f\left(\sum_{l=1}^{12}u_l n_{lk}\right)\left(1-f\left(\sum_{l=1}^{12}u_l n_{lk}\right)\right)n_{lk}\frac{dE}{du_l}\\&=-\sum_{k=1}^{120}e_k x_k(1-x_k)n_{lk}\end{aligned}\tag{5}$$

Now for updating the weight of u_l we put (5) in (4) we get

$$u_l(t+1)=u_l(t)+h\sum_{k=1}^{120}e^k x_k(1-x_k)n_{lk}\tag{6}$$

Optimum weights between first and second hidden layers Following Eq. (1) for optimum weights between first and second hidden layers, we have

$$v_{lj}(t+1)=v_{lj}(t)-h\left(dE/dv_{lj}\right)\tag{7}$$

The output from node l with $k$thinput(i.e.,$z_{1k},z_{2k},z_{3k},z_{4k},z_{5k}$) in the second hidden layers is given by

$$n_{lk}=f\left(\sum_{l=1}^{12}v_{lj}g_{jk}\right)=\frac{1}{\left(1+e^{-\sum_{l=1}^{12}v_{lj}g_{jk}}\right)}\tag{8}$$

The differential of E in Eq. (7) w. r. t weight vlj the differential of E in Eq. (7) w. r. t weight v_{lj} is given as:

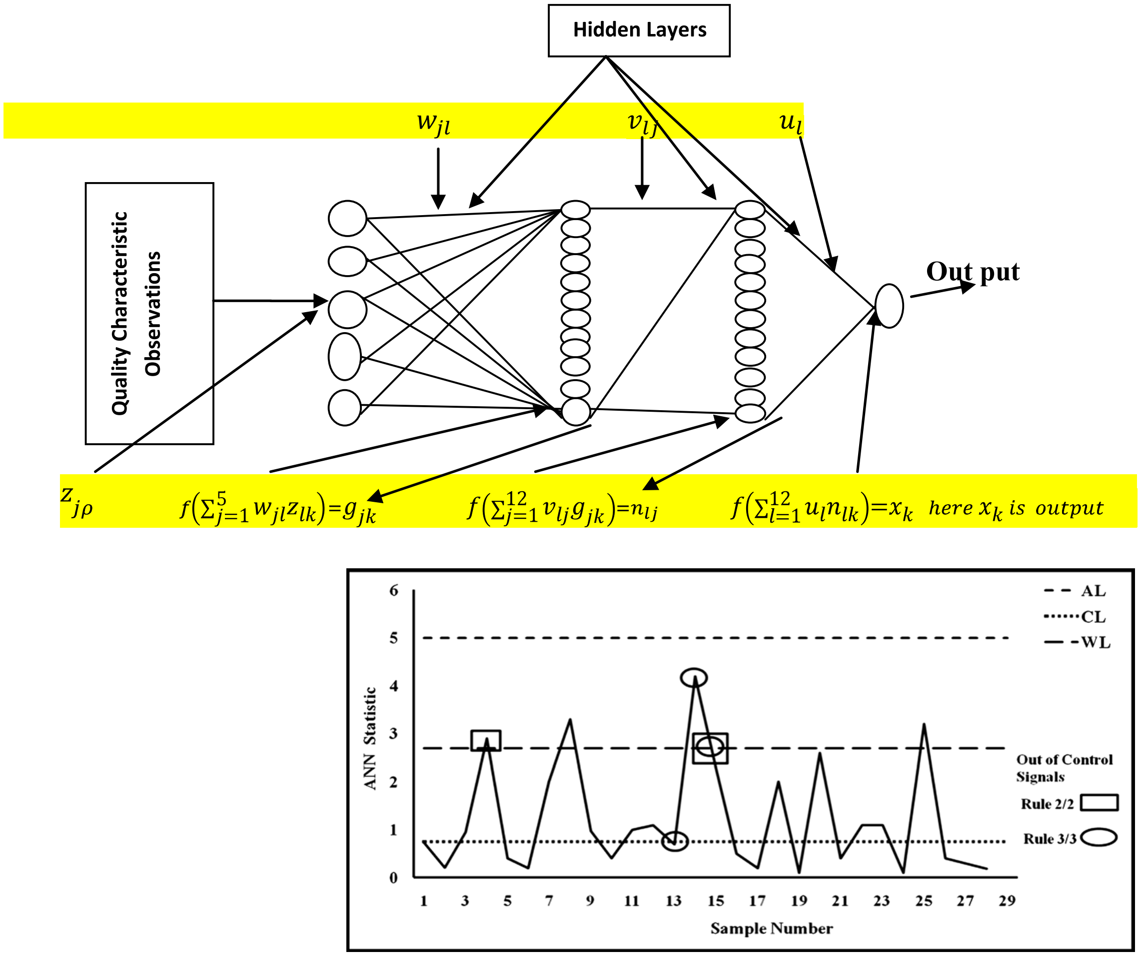

Fig. 1 Runs rules-based ANN structure for variance shifts detection

is given as

$$
\begin{aligned}
dE/dv_{ab} &= \sum_{k=1}^{120} e^k (dx_k/dv_{ab}) \\
&= \sum_{k=1}^{120} e_k f'\left(\sum_{l=1}^{12} u_l n_{lk}\right) \frac{d}{dv_{ab}} \left(\sum_{l=1}^{12} u_l n_{lk}\right) \\
&= \sum_{k=1}^{120} e_k f\left(\sum_{l=1}^{12} u_l n_{lk}\right) \left(1 f\left(\sum_{l=1}^{12} u_l n_{lk}\right)\right) \sum_{l=1}^{12} u_l \frac{d}{dv_{ab}} n_{lk} \\
&= \sum_{k=1}^{120} e_k x_k (1 x_k) \sum_{l=1}^{12} u_l f'\left(\sum_{j=1}^{12} v_{lj} g_{jk}\right) \frac{d}{dv_{ab}} \left(\sum_{j=1}^{12} v_{lj} g_{jk}\right)
\end{aligned}
$$

$$
\text{Since} \frac{d\left(\sum_{j=1}^{12} v_{jk} g_{jk}\right)}{dv_{ab}} = \begin{cases} g_{bk} & if\ l = a \\ 0 & if\ l\ \ a \end{cases}
$$

$$
\text{So we have} \frac{dn_{lk}}{dv_{ab}} = \begin{cases} f'\left(\sum_{j=1}^{12} v_{aj} g_{jk}\right) g_{bk} & if\ l = a \\ 0 & if\ l\ \ a \end{cases}
$$

and we get

$$
dE/dv_{ab} = \sum_{k=1}^{120} e_k x_k (1 x_k) \sum_{l=1}^{12} u_l f\left(\sum_{j=1}^{12} v_{aj} g_{jk}\right) \left(1 f\left(\sum_{j=1}^{12} v_{aj} g_{jk}\right)\right) g_{bk}
$$

$$
dE/dv_{ab} = \sum_{k=1}^{120} e_k x_k (1 x_k) u_a n_{ak} (1 n_{ak}) g_{bk} \tag{9}
$$

Put Eq. (8) in (7), we get the following:

$$v_{ab}(t+1) = v_{ab}(t) + h\sum_{k=1}^{120} e_k x_k(1-x_k) u_a n_{ak}(1-n_{ak}) g_{bk} \tag{10}$$

Optimum weights between inputs and first hidden layers Following Eq. (1) for optimum weights between inputs and first hidden layers, we have

$$w_{lj}(t+1) = w_{lj}(t) - h\left(dE/dw_{lj}\right) \tag{11}$$

The output from node l with input $kth(i.e. z_{1k}, z_{2k}, z_{3k}, z_{4k}, z_{5k})$ in the first hidden layer is

$$g_{jk} = f\left(\sum_{j=1}^{5} w_{jl} z_{jk}\right) = \frac{1}{\left(1 + e^{-\sum_{j=1}^{5} w_{jl} z_{jk}}\right)}, \tag{12}$$

The differential of E in Eq. (11) w. r. t weight w_{ac} is given as:

$$\begin{aligned}
dE/dw_{ac} &= -\sum_{k=1}^{120} e^k \left(dx_k/dw_{ac}\right) \\
&= -\sum_{k=1}^{120} e^k f'\left(\sum_{l=1}^{12} u_l n_{lk}\right) \frac{d}{dw_{ac}} \left(\sum_{l=1}^{12} u_l n_{lk}\right) \\
&= -\sum_{k=1}^{120} e^k f\left(\sum_{l=1}^{12} u_l n_{lk}\right)\left(1 - f\left(\sum_{l=1}^{12} u_l n_{lk}\right)\right) \sum_{l=1}^{12} u_l \frac{d}{dw_{ac}} n_{lk} \\
&= -\sum_{k=1}^{120} e_k x_k(1-x_k) \sum_{l=1}^{12} u_l f'\left(\sum_{j=1}^{12} v_{lj} g_{jk}\right) \frac{d}{dw_{ac}} \left(\sum_{j=1}^{12} v_{lj} g_{jk}\right) \\
&= -\sum_{k=1}^{120} e_k x_k(1-x_k) \sum_{l=1}^{12} u_l f\left(\sum_{j=1}^{12} v_{lj} g_{jk}\right)\left(1 - f\left(\sum_{j=1}^{12} v_{lj} g_{jk}\right)\right) \frac{d}{dw_{ac}} \left(\sum_{j=1}^{12} v_{lj} g_{jk}\right. \\
&= -\sum_{k=1}^{120} e_k x_k(1-x_k) \sum_{l=1}^{12} u_l n_{ak}(1-n_{ak}) \frac{d}{dw_{ac}} \left(\sum_{j=1}^{12} v_{lj} g_{jk}\right) \\
&= -\sum_{k=1}^{120} e_k x_k(1-x_k) \sum_{l=1}^{12} u_l n_{ak}(1-n_{ak}) \sum_{j=1}^{12} v_{lj} \frac{d}{dw_{ac}} \left(g_{jk}\right) \\
&= -\sum_{k=1}^{120} e_k x_k(1-x_k) \sum_{l=1}^{12} u_l n_{ak}(1-n_{ak}) \sum_{j=1}^{12} v_{lj} f'\left(\sum_{j=1}^{5} w_{lj} z_{jk}\right) \frac{dw_{lj}}{dw_{ac}} \\
&\text{We have } \frac{dg_{lk}}{dw_{ac}} = \begin{cases} f'\left(\sum_{j=1}^{5} w_{aj} z_{jk}\right) z_{ck} & if\ l = a \\ 0 & if\ l \neq a \end{cases} \\
dE/dw_{ac} &= -\sum_{k=1}^{120} e_k x_k(1-x_k) g_{ak}(1-g_{ak}) z_{ck} \sum_{l=1}^{12} u_l n_{lk}(1-n_{lk}) v_{la},
\end{aligned} \tag{13}$$

Put Eq. (13) in (12) got the following expression

$$w_{jl}(t+1) = w_{jl}(t) + h\sum_{k=1}^{120} e_k x_k(1-x_k) g_{ak}(1-g_{ak}) z_{ck} \sum_{l=1}^{12} u_l n_{lk}(1-n_{lk}) v_{la} \tag{14}$$

The aim is to obtain the minimum error to get the optimum weights for the whole network. The optimum weights between output and second hidden layer (cf., Eq. 6), optimum weights between first and second hidden layers (cf., Eq. 10) and optimum weights between inputs and first hidden layers (cf., Eq. 14).

where A, B, C, D, and E the representatives of different groups of distributions as are mentioned in Table 1 which will use for our study proposes. If the proposed ANN is not able to produce the desired outputs (cf., Table 1) for all the shifts, this means that there is need of necessary corrections to train ANN by back propagation method for optimal weights.

$$\text{Loss function} = E = \frac{1}{2}\sum_{A,B,C,D,E} (\text{desired output} - \text{actual output})^2 \tag{15}$$

Table 1 Input data for training and desired output from ANN

Group	Input data for training ANN			Desired output from ANN	
	No. of samples	Distribution $N(0,(\rho\sigma_0)^2)$	Shift $\rho=\sigma/\sigma_0$	Desired outputs$=x^d$	Interpretation
A	120	$N(0,1^2)$	1	0.050	$\rho=1$ no shift
B	120	$N(0,2^2)$	2	0.275	$\rho=2$ shift
C	120	$N(0,3^2)$	3	0.500	$\rho=3$ shift
D	120	$N(0,4^2)$	4	0.725	$\rho=4$ shift
E	120	$N(0,5^2)$	5	0.950	$\rho=5$ shift

The ANN algorithm is stopped when the value of the loss function defined above has become sufficiently small (cf., Fig. 2a).

The plot of each percentage data is shown in Fig. 2b. As it is advocated that the outputs of ANN scatter around the desired values for a particular process, a good estimate of variance change magnitude cannot be obtained. This issue may be addressed by looking at grouping in the form of clusters and monitoring their patterns in variance, if any, for possible detection (Table 2).

The said runs rules-based ANN schemes are based on the following terms and definitions.

Actions Limit (AL) This is a threshold value of neural network statistic (output in Fig. 1). If values of the neural network exceed the AL, the process is called out of control. The AL is greater than

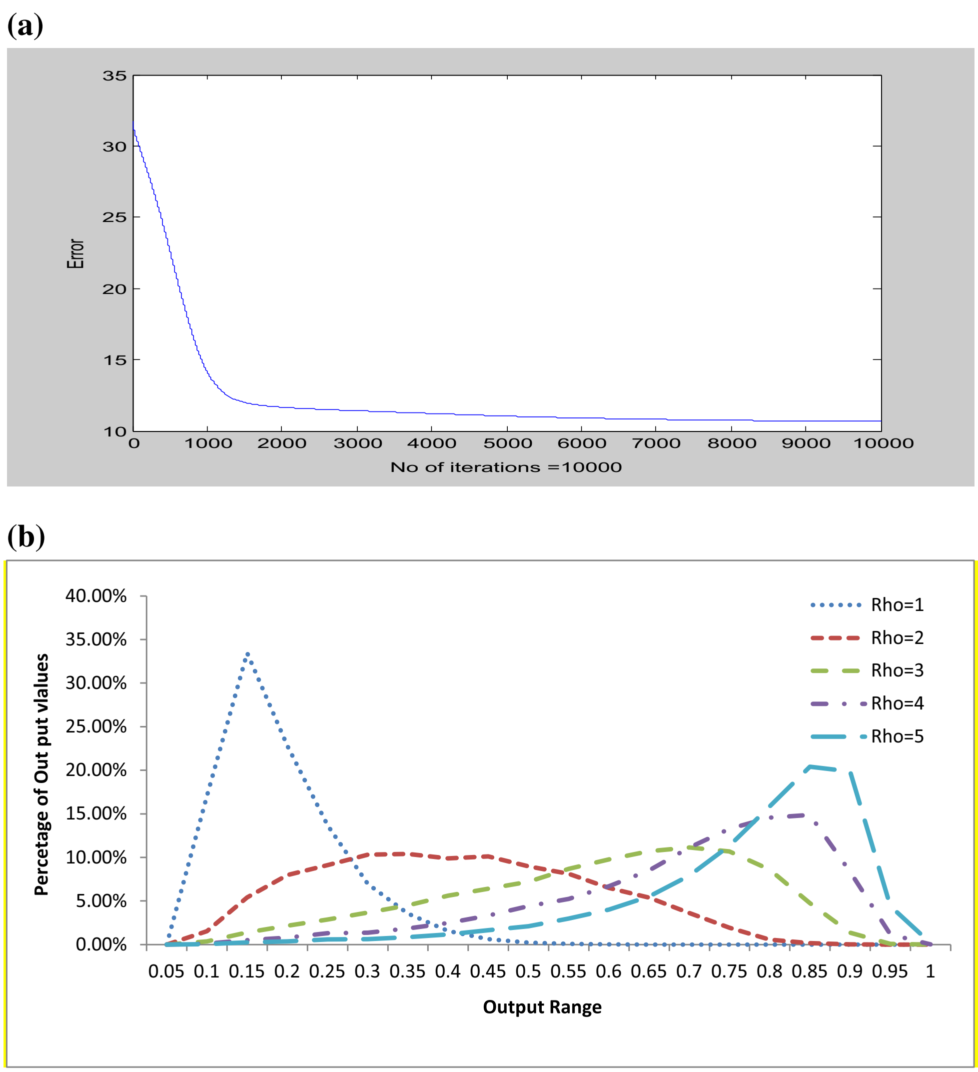

Fig. 2 **a** Error behavior with 10,000 iterations. **b** Behavior of percentage ANN Outputs for different shifts

Table 2 Percentages of ANN outputs falling into specified output range

Output range	Input data ρ=1 vs target=0.05 (%)	Input data ρ=2 vs target=0.275 (%)	Input data ρ=3 vs target=0.5 (%)	Input data ρ=4 vs target=0.725 (%)	Input data ρ=5 vs target=0.95 (%)
0.00–0.05	0.00	0.00	0.00	0.00	0
0.05–0.10	17.04	1.53	0.35	0.12	0
0.10–0.15	33.47	5.42	1.40	0.51	0.24
0.15–0.20	22.80	7.96	2.17	0.80	0.36
0.20–0.25	13.68	9.12	2.88	1.31	0.59
0.25–0.30	6.96	10.32	3.68	1.36	0.61
0.30–0.35	3.55	10.41	4.51	1.84	0.85
0.35–0.40	1.57	9.87	5.60	2.48	1.17
0.40–0.45	0.61	10.10	6.42	3.32	1.65
0.45–0.50	0.23	8.98	7.20	4.40	2.10
0.50–0.55	0.07	8.12	8.69	5.25	2.98
0.55–0.60	0.02	6.50	9.76	6.70	4.00
0.60–0.65	0.00	5.39	10.72	8.52	5.51
0.65–0.70	0.00	3.61	11.13	11.15	7.91
0.70–0.75	0.00	1.93	10.68	13.29	11.33
0.75–0.80	0.00	1.00	8.60	14.55	15.85
0.80–0.85	0.00	0.16	4.75	14.88	20.39
0.85–0.90	0.00	0.01	1.37	8.35	19.87
0.90–0.95	0.00	0.00	0.09	1.16	4.40
0.95–1.00	0.00	0.00	0.00	0.01	0.15

usual critical limit of neural network for a given ARL_0.

Warning Limit (WL) This is a critical limit for the value of the neural network statistic beyond which (but not crossing the AL) some pattern of consecutive points indicates an out-of-control situation. The value of the WL would be greater than Central Line (CL) and smaller than the usual neural network statistic for a fixed ARL_0.

With the help of the above information, we propose three runs rules schemes for our trained neural network technique.

Scheme I (2 out of 2): A process is called out of control if one of the following conditions is true.

1. One point of neural network statistic falls outside the AL
2. Two consecutive points of neural network statistic fall between WL and AL

Scheme II (2 out of 3): A process is called out of control if one of the following conditions is true.

1. One point of neural network statistic falls outside the AL
2. Two out of three consecutive points of neural network statistic fall between WL and AL

Scheme III (3 out of 3): A process is called out of control if one of the following conditions is true.

1. One point of neural network statistic falls outside the AL
2. Three consecutive points of neural network statistic fall between WL and AL

Usual Scheme (1 out of 1) A process is called out of control if one point of neural network statistic falls outside the usual control limit of Shewhart chart.

3 Performance measures and comparisons

In this section, we evaluate the performance of runs rules-based schemes for the trained ANN under normal and

Table 3 Error and critical values of the trained ANN for ARL_0=175

n	Error	CV usual	CV 2 out of 2	CV 3 out of 3	CV 2 out 3
2	24.1039	0.5629	0.4705	0.4303	0.4850
3	17.5137	0.5435	0.4210	0.3668	0.4400
4	12.6862	0.4766	0.3250	0.2606	0.3472
5	10.6862	0.4248	0.2860	0.2288	0.3065
8	5.6828	0.3724	0.1963	0.1372	0.2215
10	3.9601	0.3066	0.1730	0.1760	0.1924

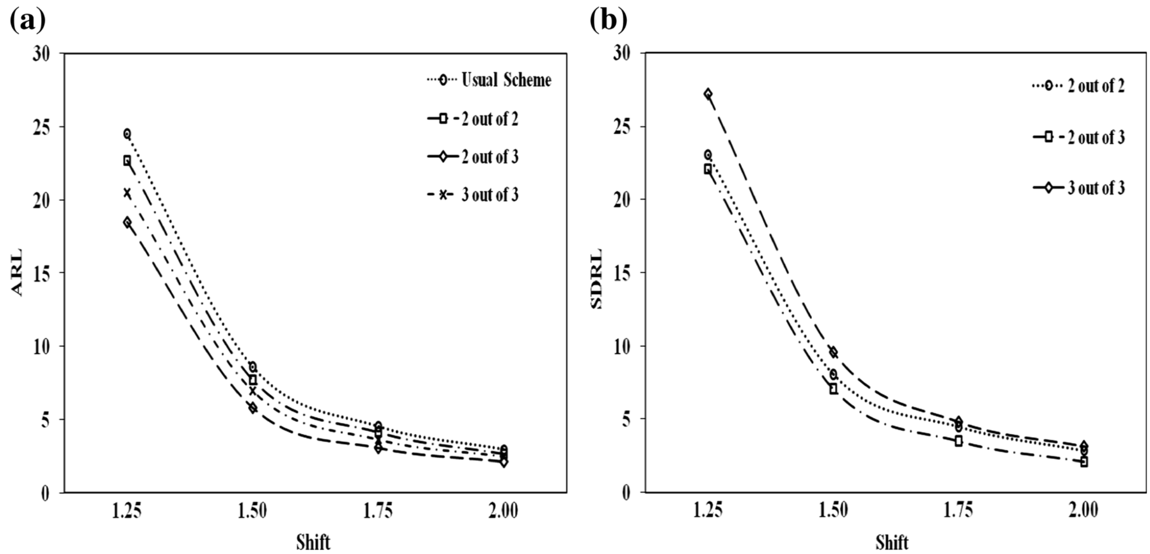

Fig. 3 **a** ARL performance of usual scheme and runs rules-based ANN using ARL_0=175 n=4 and E=12.6862. **b** SDRL performance of runs rules-based ANN using ARL_0=175 n=4 and E=12.6862

bootstrapping environments to detect shifts in variability parameter of the quality characteristic of interest. The performances evaluations are based on vary common performance measure in quality control literature namely average run length (ARL). The ARL is a famous tool and is widely used by researchers for measuring the performance of memory type control charts. The performance is assessed by two types of ARL2, i.e. ARL_0 and ARL_1. ARL_0 is the expected number of samples before an out-of-control point is detected when the process is actually in control. ARL_1 is the expected number of samples before an out-of-control signal is received when the process is actually shifted to an out-of-control state. For a fixed value of ARL_0, a chart is considered to be more effective than other charts if it has a smaller ARL_1 (see Wu et al. [48]). Some other effective performance measures are also incorporated in order to have more efficient picture of performance of proposals. The details of these measures as:

Extra Quadratic Loss (EQL): The ARL evaluates the performance of a charting structure at a specific shift point. The EQL is an alternative performance measure, which describes the overall effectiveness of a control chart. The EQL is defined as a weighted average ARL over the whole process shift domain $\rho_{\min}<\rho<\rho_{\max}$ using the square of shift (ρ^2) as a weight. Mathematically the EQL is given as

$$EQL = \frac{1}{\rho_{\max}-\rho_{\min}} \int_{\rho_{\min}}^{\rho_{\max}} \rho^2 \, ARL(\rho) d\rho . \qquad (16)$$

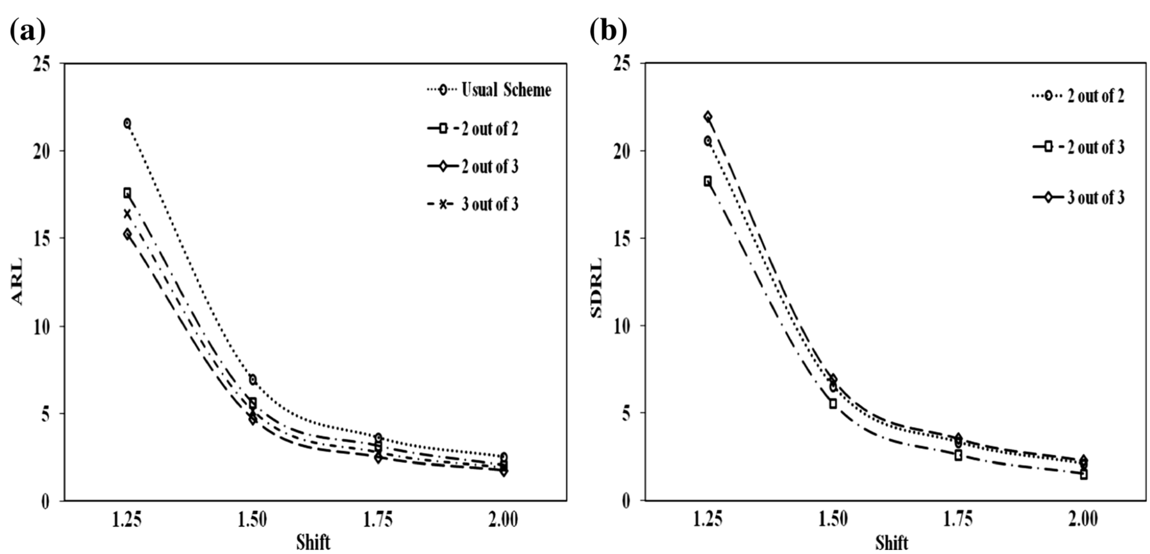

Fig. 4 **a** ARL performance of usual scheme and runs rules-based ANN using ARL_0=175 n=5 and E=10.6862. **b** SDRL performance of runs rules-based ANN using ARL_0=175 n=5 and E=10.6862

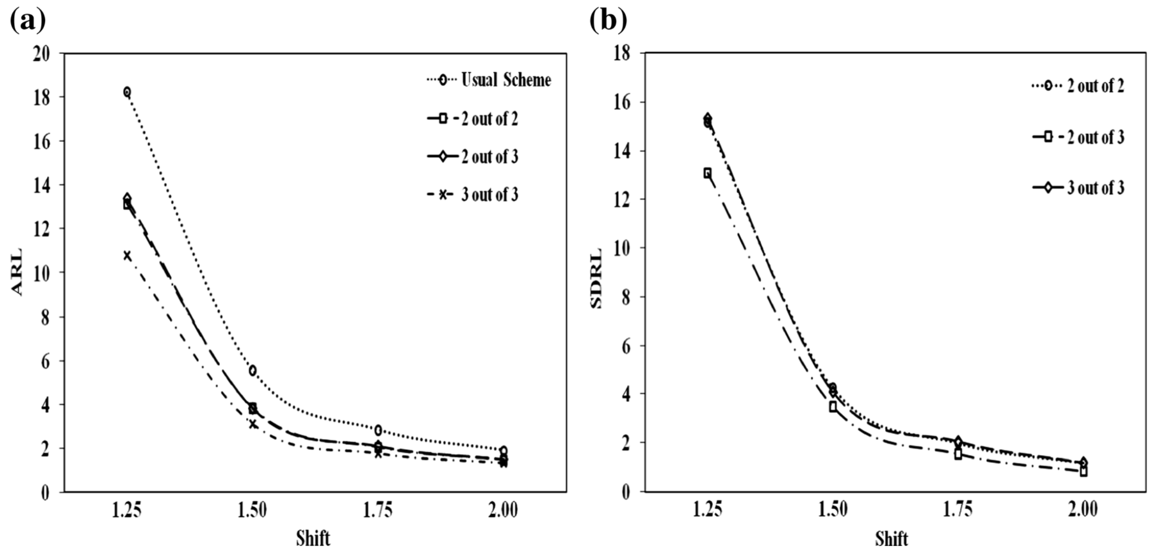

Fig. 5 **a** ARL performance of usual scheme and runs rules-based ANN using $ARL_0=175$ $n=10$ and $E=3.9601$. **b** SDRL performance of runs rules-based ANN using $ARL_0=175$ $n=10$ and $E=3.9601$

Relative Average Run Length (RARL): The RARL describes the overall effectiveness of particular a charting structure relative to the benchmark chart. It examines that how close a particular chart performs to the benchmark chart for each shift in terms of ARL (cf. Wu et al. [48] and Ryu et al. [41]). The RARL measure can be defined mathematically as:

$$\mathrm{RARL} = \frac{1}{\rho_{\max}-\rho_{\min}} \int_{\rho_{\min}}^{\rho_{\max}} \frac{\mathrm{ARL}(\rho)}{\mathrm{ARL}_{\mathrm{bmk}}(\rho)} d\rho , \quad (17)$$

where ρ is the amount of shift in the process variance σ_y, $ARL(\rho)$ and $\mathrm{ARL}_{\mathrm{bmk}}(\rho)$ are the average run lengths of a particular chart and the benchmark chart at ρ respectively. The chart having the minimum value of EQL is considered to be benchmark chart. The RARL may be observed as RARL=1 (for benchmark chart) and RARL>1 (for the other charts). The RARL value above 1 shows the inferiority in performance of a particular chart relative to the benchmark chart. Zhao et al. [50] and Han et al. [19] also considered the uniform distribution of ρ in their studies.

Performance comparison index (PCI): According to Ou et al. [32], it is the ratio between the EQL of a chart and the EQL of the best chart under the same conditions. This index facilitates the performance comparison and a ranking based on the EQL. The chart with the lowest EQL has a PCI value equal to one, and the PCI values of all other charts are larger than one.

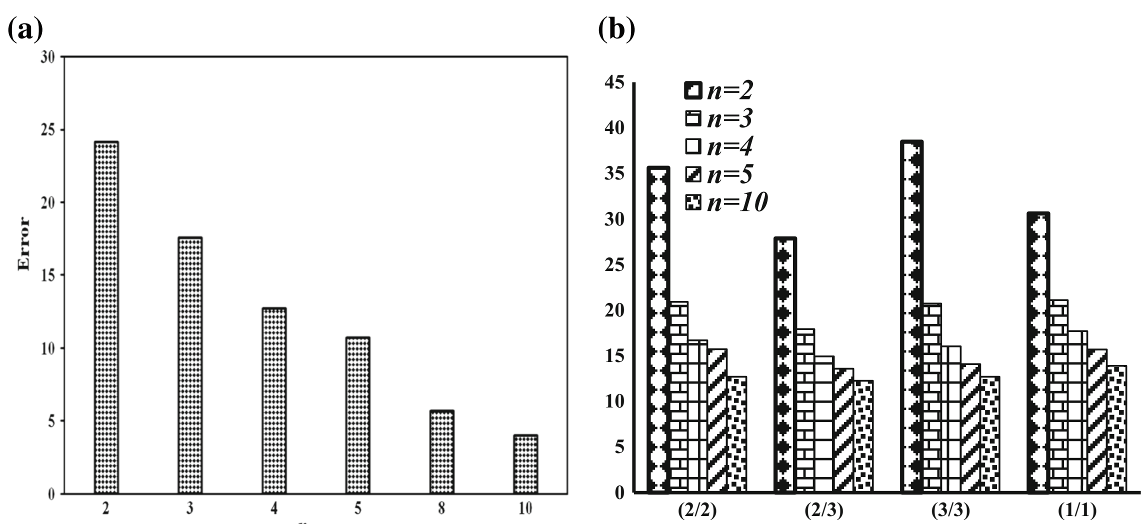

Fig. 6 **a** Errors of the trained ANN for different n. **b** Errors of the trained ANN schemes for different n

Table 4 EQL, RARL, and PCI values of different schemes at different sample sizes

N	Usual scheme			2 out of 2			2 out of 3			3 out of 3		
	EQL	RARL	PCI	EQL	RARL	PCI	EQL	RARL	PCI	EQL	RARL	PCI
2	30.60	1.00	1.09	35.61	1.17	1.27	27.88	0.90	1.00	38.48	1.26	1.38
3	21.11	1.00	1.17	20.91	0.98	1.16	17.91	0.83	1.00	20.73	0.97	1.15
4	17.69	1.00	1.18	16.70	0.93	1.11	14.92	0.83	1.00	16.02	0.90	1.07
5	15.70	1.00	1.15	15.73	1.00	1.15	13.59	0.85	1.00	14.07	0.88	1.03

$$\mathrm{PCI} = \frac{\mathrm{EQL}}{\mathrm{EQL}_{\mathrm{benchmark}}} \tag{18}$$

Some modifications of these measures may be seen in Zhang and Wu [49], Wu et al. [48], Ryu et al. [41], Ou et al. [31, 32], Ahmad et al. [2], Ahmad et al. [3, 5]. The values of WL and AL are proportional to the value of ARL for a given shift, e.g., if the value of ARL is higher than value of WL then AL is also high and vice versa. There are many pairs of WL and AL that give the desired ARL_0 and we look for the optimum pair of WL and AL (i.e., it minimizes the ARL_1 for fixed ARL_0). We have

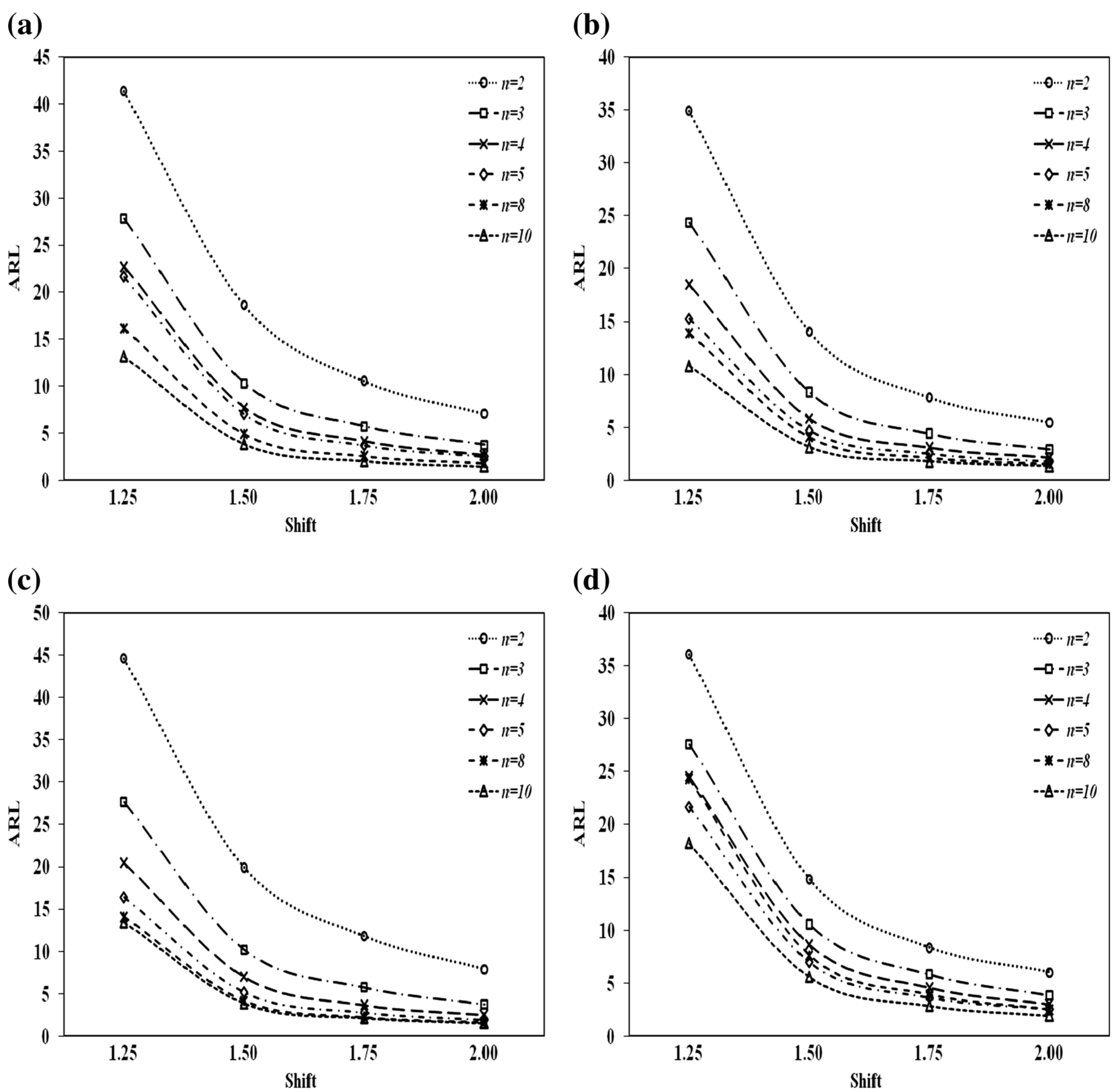

Fig. 7 **a** ARL curves of ANN scheme I for varying *n*. **b** ARL curves of ANN scheme II for varying *n*. **c** ARL curves of ANN scheme III for varying *n*. **d** ARL curves of ANN the usual scheme for varying *n*

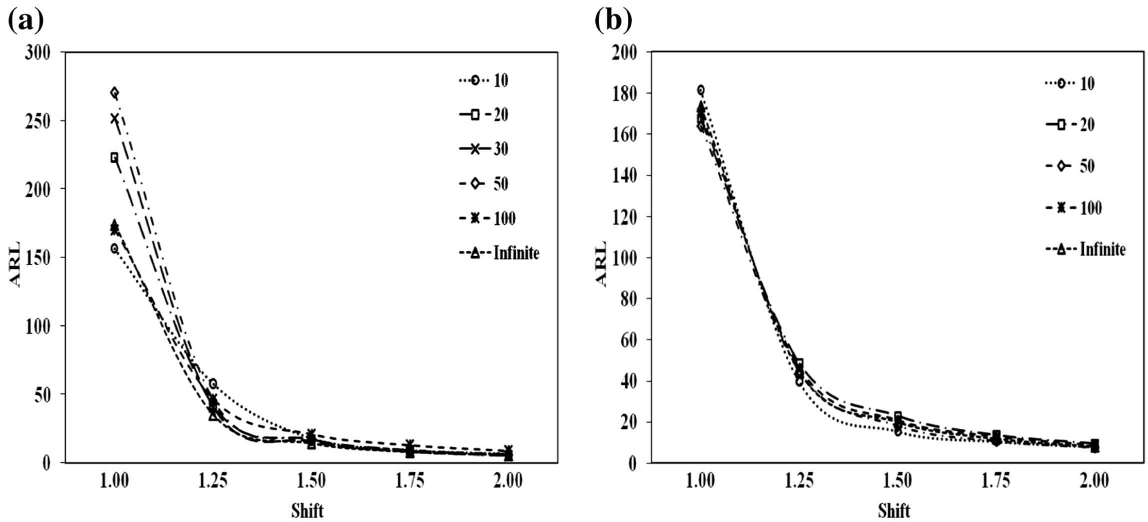

Fig. 8 **a** Bootstrapped ARL performance of scheme II for Trained ANN at $n=2$. **b** Bootstrapped ARL performance of scheme III for trained ANN at $n=2$

evaluated the ARL performance of the three proposed runs rules-based ANN schemes and also the usual ANN scheme.

3.1 Comparisons for mean-based ANN schemes

We use here the trained ANN based on $Z_{j\rho} = \left|x_{j\rho} - \bar{\bar{x}}_{\rho}\right|$ using the topology $[n-m-k-O]=[n-12-12-1]$ for varying number of inputs n and compute their ARL measures with and without bootstrapping. We also investigate the effect of replacing $\bar{\bar{x}}_{\rho}$ by $\widetilde{x}_{j\rho}$ in $Z_{j\rho}$ on ARL performance taking inspiration from Riaz and Saghir [37]. According to the percentage output of ANN a cut point/s is selected as critical value for out-of-control signal for different schemes. This means if the output of ANN greater than cut point, the process is considered to be out of control (cf., Table 3).

Runs rules-based ANN performance We have evaluated the ARL performance of the trained ANN using the three proposed runs rules schemes and also the usual ANN scheme (the usual one—we call it 1/1 scheme) for varying number of inputs n. We have trained the ANN for each n with minimum error E for the above-mentioned topology $[n-12-12-1]$ of ANN. The ARL results, along with standard deviation of run length (SDRL), at $ARL_0=175$ are provided (for different runs rules schemes of ANN) in Figs. 3a, 4a, and 5a for different values on n with their respective errors (E) which is actually the loss function as defined in Chang and Ho [10]. Table 3

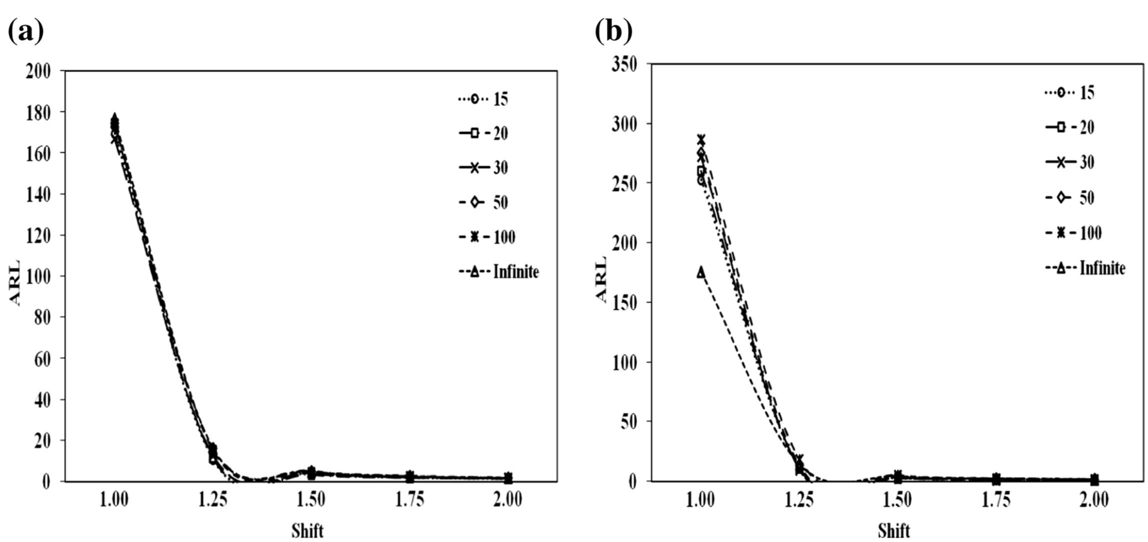

Fig. 9 **a** Bootstrapped ARL performance of scheme I for trained ANN at $n=8$. **b** Bootstrapped ARL performance of scheme II for trained ANN at $n=8$

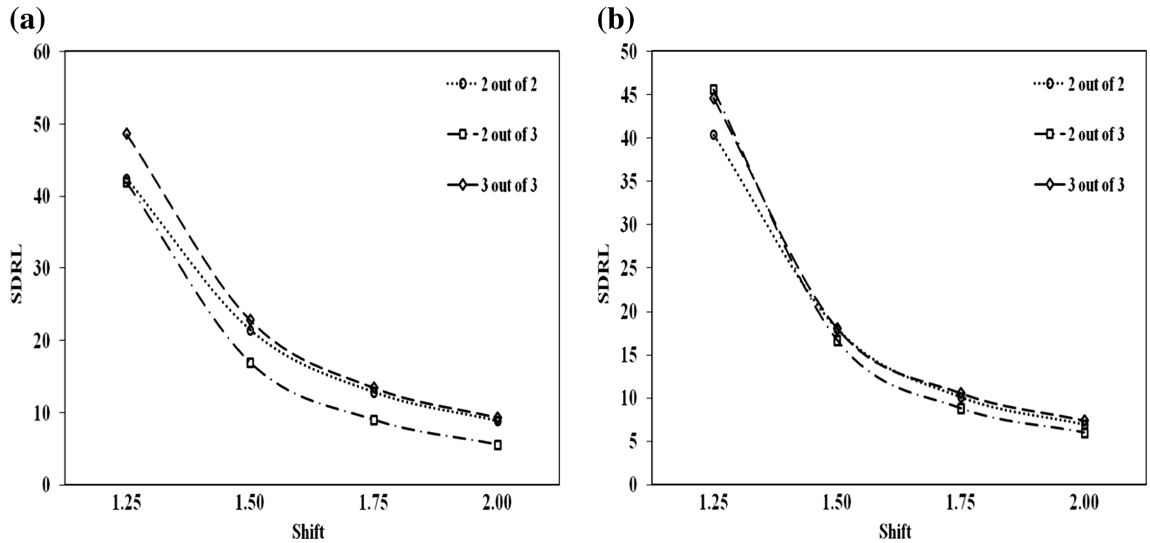

Fig. 10 **a** Bootstrapped ARL performance of different schemes for $n=2$ and population size=20. **b** Bootstrapped ARL performance of different schemes for $n=2$ and population size=50

contains the information of all Critical Values (CV) of usual and proposed schemes and errors of the different sample sizes to fix $ARL_0=175$. It is to be noted that the error for all schemes reduce with the increase in sample size for all schemes (cf. Fig. 6a–b).

The ARL comparisons indicate that all the three proposals for ANN are superior to the usual ANN scheme (cf. Figs. 3a, 4a, and 5a). Also the related SDRL curves indicate the precision of outcomes for different schemes, which is not varying significantly from each other (cf. Figs. 3b, 4, and 5b). From among the three proposals, Scheme II is the best in terms of smallest ARL_1 and EQL in all the cases, as can be seen from Figs. 3a, 4a, and 5a and Table 4, respectively. Also schemes I and III have very close performance ability in shift detection. For larger shifts all the schemes behave almost same as the usual ANN scheme (cf. Figs. 3a, 4a, and 5a). Moreover, with an increase in the size of ANN topology the detection ability of all the schemes keeps improving (cf. Figs. 7a–d). It is to be mentioned here that for our study purposes, we fixed the size of the hidden layers and size of nodes of hidden layers, but a variation in these quantities may also affect the performance of ANN.

Bootstrapped performance of trained ANN with runs rules schemes We investigate the performance of the runs rules schemes for the trained ANN under bootstrapped environments for the practical considerations. For this purpose, we have considered limited population data of sizes 10, 15, 20, 30, 50 and 100. For all these data sets of different sizes (from normal setup with the same parameters as used for Figs. 3a, 4a, and 5a results), we have evaluated ARLs EQL, RARL and PCI for ANN using repeated samples of size $n=2$ and 8 by fixing ARL_0 at 175 again for comparison and validity purposes. The same we have also done for all the schemes including usual and runs rule-based schemes, for the same choices of different quantities.

The resulting bootstrapped ARL curves for different data sizes, as mentioned above, are provided in Figs. 8a, b and 9a, b for runs rules-based ANN schemes. The bootstrapped pattern of ARL curves has revealed that runs rules-based schemes for ANN may work satisfactorily with the limited data to maintain its ability closer to the actual one (i.e., based on whole population behavior) as obvious from Figs. 8a, b and 9a, b. It is observed that the suggested scheme II needs comparatively more data as compared to other schemes to converge to the true ARL performance for better detection of out-of-control signals. It is to be noted that the established superiority of runs rules-based ANN schemes, as claimed based on Figs. 3a, 4a, and 5a, also holds in all bootstrapped scenarios (cf. Fig. 10a, b) (Tables 5 and 6).

Table 5 Error and critical values of the modified ANN for $ARL_0=175$

N	Error	CV usual	CV 2 out of 2	CV 3 out of 3	CV 2 out 3
2	24.1068	0.5656	0.4703	0.4291	0.4926
3	18.6355	0.5034	0.4062	0.3510	0.4357
4	13.7801	0.4351	0.3231	0.2733	0.3358
5	12.6100	0.4446	0.3162	0.2613	0.3357
6	8.9870	0.3705	0.2530	0.2048	0.2802
7	6.7041	0.4093	0.2068	0.1446	0.2320
10	4.6500	0.4061	0.1826	0.1375	0.2035

Table 6 EQL and RARL values of different modified schemes at different sample sizes

N	Usual scheme			2 out of 2			2 out of 3			3 out of 3		
	EQL	RARL	PCI	EQL	RARL	PCI	EQL	RARL	PCI	EQL	RARL	PCI
2	28.53	1.00	1.12	33.24	1.16	1.30	25.38	0.88	1.00	36.32	1.28	1.43
3	17.92	1.00	1.20	18.73	1.04	1.26	14.86	0.82	1.00	18.63	1.03	1.25
4	14.67	1.00	1.29	13.78	0.93	1.29	11.34	0.77	1.00	13.39	0.91	1.18
5	13.21	1.00	1.40	11.21	0.85	1.19	9.41	0.72	1.00	10.41	0.79	1.10

In order to justify the performance of proposed and usual schemes explore in this study, we provide EQL, RARL and PCI values at different sample sizes in Table 6. It is evident from Table 6 that the proposed scheme 2 out of 3 has exhibited best performance in terms of EQL, RARL and PCI (smallest EQL, RARL, and PCI values) that is in accordance with the graphical presentation of ARL analysis in this section. It is also observed that the performances of understudy schemes improve with the increment of sample size (Table 7).

3.2 Comparisons for median-based ANN (modified) schemes

Riaz and Saghir [37] proposed the use of average absolute deviation from median in control charting setup for dispersion parameter, Ahmad et al. [4] suggested the use of median control charting, Ning and Wu [30] proposed quantile-based control charting structures, Chao-Yu et al. [11] investigated different dispersion estimators for process monitoring. Taking inspiration from the said literature, we intend here to do the same in ANN setup for an efficient monitoring of dispersion parameter using different runs rules schemes. We have studied, till now, the ANN performance trained for the data $x_{j\rho}$ for each group by transforming to $Z_{ji} = \left|x_{ji} - \overline{x}_j\right|$ values which are chosen as input to ANN, where $\overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ji}$ for a sample of size n using j=1,2,…120, referring to index of the observation within a sample. Now, we suggest a modification for the trained ANN by replacing by $\widetilde{x}_j$ (median estimator) instead of $\overline{x}$ (the mean estimator) in input $Z_{ji} = \left|x_{ji} - \overline{x}_j\right|$ of ANN (cf., Riaz and Saghir [37]). Now $\widetilde{Z}_{ji} = \left|x_{ji} - \widetilde{x}_j\right|$ is the input estimator for ANN and now we again train the neural network for different topologies [$n-12-12-1$] for varying number of inputs "n". We implement the same runs rules schemes and investigate their performance for the newly trained modified ANN for $\widetilde{Z}_{ji} = \left|x_{ji} - \widetilde{x}_j\right|$ for different choices of n and also compare the performance of the modified ANN with the original ANN based on to $Z_{ji} = \left|x_{ji} - \overline{x}_j\right|$ as we did previously.

We have carried out the ARL analysis on the similar lines as previous and ARL curves are presented in Figs. 11a–d and 12a–b for different runs rules and varying values of n. In Fig. 12 the symbol "M" in labels indicate modified ANN runs rules scheme otherwise simple ANN runs rules scheme. The CVs of usual and proposed schemes and errors of the different sample

Table 7 ARL values of different non modified and modified schemes at n=5 when ARL_0=175

Shift	Schemes				Modified schemes			
	Usual scheme	2 out of 2	2 out of 3	3 out of 3	Usual scheme	2 out of 2	2 out of 3	3 out of 3
1.25	21.61	21.60	15.27	16.40	22.09	18.67	14.90	17.34
1.50	6.94	7.01	4.68	5.17	7.35	5.96	4.56	5.15
1.75	3.61	3.68	2.49	2.79	3.83	3.05	2.49	2.82
2.00	2.51	2.49	1.74	1.93	2.53	2.10	1.75	1.95
2.25	1.90	1.91	1.44	1.55	1.93	1.64	1.44	1.55
2.50	1.58	1.61	1.28	1.39	1.63	1.45	1.27	1.37
2.75	1.41	1.40	1.18	1.24	1.43	1.30	1.18	1.25
3.00	1.29	1.30	1.14	1.16	1.31	1.20	1.13	1.19

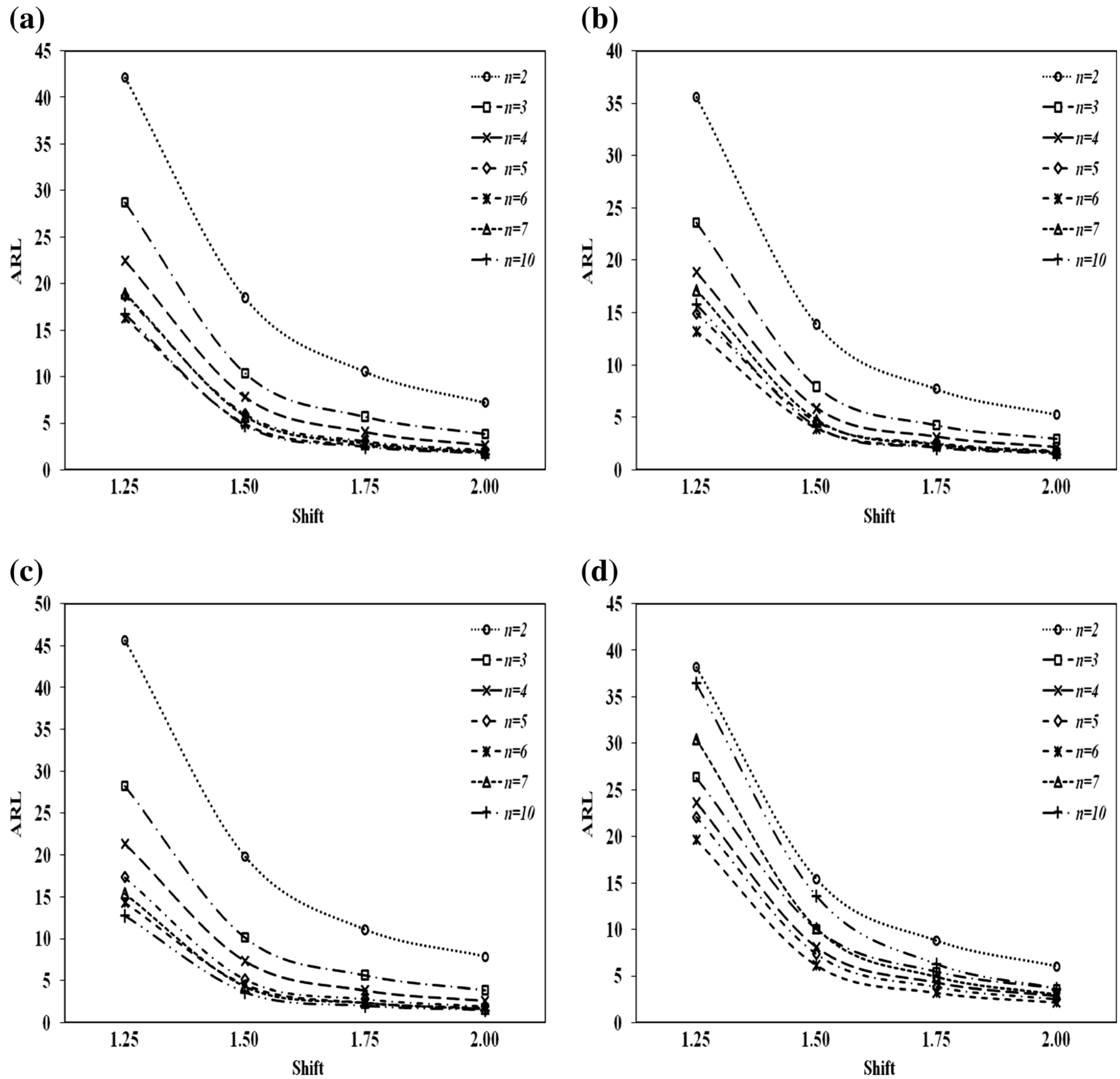

Fig. 11 **a** ARL curves of modified ANN with scheme I for varying *n*. **b** ARL curves of modified ANN with scheme II for varying *n*. **c** ARL curves of modified ANN with scheme III for varying *n*. **d** ARL curves of modified ANN with the usual scheme for varying *n*

sizes to fix $ARL_0=175$ are provided in Table 2. The ARL comparisons advocate that: (i) all the runs rules-based schemes perform superior as compared to the usual scheme for the modified ANN; (ii) modified Scheme II is the best in terms of smallest ARL_1 in all the cases and schemes I and III have very close performance ability in shift detection; (iii) with an increase in the size "n" of ANN topology the detection ability of all the schemes for the modified ANN keeps improving (cf. Fig. 11a–d) same as that of simple ANN; (iv); modified and simple ANN has very close performance in normal setup for different choices of *n* (however, in contaminated and/or non-normal environments modified ANN may have an edge because of its robustness (cf. Riaz and Saghir [37])).

It is to be mentioned that the runs rule-based ANN structures may have comparable performance as that of runs rules-based SPC charts like *R*, *S*, and EWMA (like those given in Abbas et al. [1] and Riaz et al. [39]) charts with an added advantage of estimating the variance with bootstrapping, as indicated by Chang and Ho [10]. Table 6 provided EQL, RARL, and PCI analysis for different modified schemes at different sample sizes. The modified scheme 2 out of 3 has exhibited best performance in terms of EQL, RARL and PCI measures similarly as observed in Table 6. The performance of modified schemes keep improves with the increment of sample size as expected (cf. Table 6).

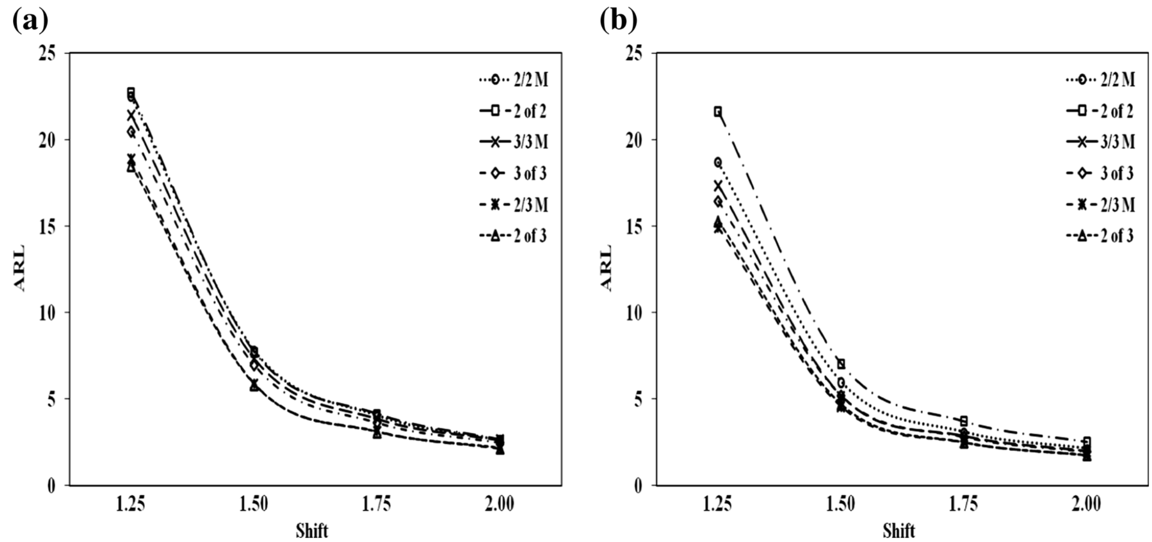

Fig. 12 **a** ARL curves of modified and simple ANN with the different runs rules schemes for n=4 and ARL_0=175. **b** ARL curves of modified and simple ANN with the different runs rules schemes for n=5 and ARL_0=175

4 An example and application with real date set

In this section, an illustrative example is provided to demonstrate how the application in practice using a real dataset. The data used in this example is taken from Chen et al. [12] that contains the information of the inside diameter of cylinder bores in an engine block (cf. Table 8). We have used 1/1 and 2/3 run rules with the trained ANN and summary for the out-of-control signals is provided here. It is to be mentioned that the subsample values in Table 8 are recorded in the last digits of its actual measurement as indicated by Chen et al. [12]. The actual measurements are of the 3.5205, 3.5202, and 3.5204 and so on. It is evident from the summary of diagnosis for out-control signals that our proposed runs rules-based schemes of

Table 8 Cylinder diameter data. Only the last three digits of the raw data are displayed here; the rest of the digits are the same for all data points

$Sample_i$	x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	$Sample_i$	x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}
1	205	202	204	207	205	19	207	206	194	197	201
2	202	196	201	198	202	20	200	204	198	199	199
3	201	202	199	197	196	21	203	200	204	199	200
4	205	203	196	201	197	22	196	203	197	201	194
5	199	196	201	200	195	23	197	199	203	200	196 s
6	203	198	192	217	196	24	201	197	196	199	197
7	202	202	198	203	202	25	204	196	201	199	197
8	197	196	196	200	204	26	206	206	199	200	203
9	199	200	204	196	202	27	204	203	199	199	197
10	202	196	204	195	197	28	199	201	201	194	200
11	205	204	202	208	205	29	201	196	197	204	200
12	200	201	199	200	201	30	203	197	199	197	201
13	205	196	201	197	198	31	203	197	199	197	201
14	202	199	200	198	200	32	197	194	199	200	199
15	200	200	201	205	201	33	200	201	200	197	200
16	201	187	209	202	200	34	199	199	201	201	201
17	202	202	204	198	203	35	200	204	197	197	199
18	201	198	204	201	201						

ANN has better diagnostic ability relative to the usual ANN scheme.

Diagnostic summary of out-of-control signals:		
Sample size=35	Subgroup size=5	False alarm rate=0.0057
Run rules	Control limits	Diagnose ability
(1/1)	0.4248	13
(2/3)	0.3065	21

5 Conclusions and recommendations

The study has trained an ANN (with back propagation method) for process variance change detection and tested its performance using different measures like ARL, EQL, RARL, and PCI. The study also suggested the runs rules implementation with the trained ANN (with back propagation method). The runs rules-based ANN schemes have shown an efficient signaling ability as compared to the usual ANN scheme (under normality as well as bootstrapped environment. More specifically, scheme II (i.e., 2 out of 3) has performed the best followed by I and III and then comes the usual (1 out of 1) for variance change detection using the trained ANN with the implementation of runs rules schemes. Moreover, a modification in the trained ANN structure is also suggested. The scope of study may be extended by training ANN for other input estimators of location and dispersion with more runs rules under different distributional environments like Gamma, Weibull, Log-Normal and Rayleigh probability models. The runs rules-based ANN may also be trained under EWMA and CUSUM structures. Moreover, multivariate generalizations and Bayesian analysis may be potential dimensions for further exploration in this direction.

Acknowledgments The authors are thankful to the referees for their valuable comments that helped to improve the initial version of the article. The authors Muhammad Riaz and Saddam Akber Abbasi are indebted to King Fahd University of Petroleum and Minerals Dhahran Saudi Arabia respectively for providing excellent research facilities.

References

1. Abbas N, Riaz M, Does RJMM (2011) Enhancing the performance of EWMA charts. Qual Reliab Eng Int 27(6):821–833
2. Ahmad S, Riaz M, Abbasi SA, Lin Z (2013) On efficient median control charting. J Chin Inst Eng.
3. Ahmad S, Riaz M, Abbasi SA, Lin Z (2012) On median control charting under double sampling scheme. Eur J Ind Eng.
4. Ahmad S, Riaz M, Abbasi SA, Lin Z (2013) On efficient median control charting. J Chin Inst Eng. doi:10.1080/02533839.02532013.02781794
5. Ahmad S, Riaz M, Abbasi SA, Lin Z (2013) On monitoring process variability under double sampling scheme. Int J Prod Econ 142(2): 388–400
6. Antzoulakos DL, Rakitzis AC (2008) The modified R out of M control chart. Commun Stat Theory Methods 37(2):396–408
7. Antzoulakos DL, Rakitzis AC (2010) Runs rules schemes for monitoring process variability. J Appl Stat 37:1231–1247
8. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford
9. Bishop CM (2006) Pattern recognition and machine learning (information science and statistics). Springer Science and Business Media, New York
10. Chang SI, Ho ES (1999) A two stage neural network approach for process variance change detection and classification. Int J Prod Res 37:1581–1599
11. Chao-Yu C, Pin-Hao W, Jiang BC (1999) A comparative study on the estimators of standard deviation in statistical process control. J Chin Inst Eng 22(1):109–116
12. Chen G, Cheng SW, Xie H (2001) Monitoring process mean and variability with one EWMA chart. J Qual Technol 33:223–233
13. Chiu CC, Chen MK, Lee KM (2001) Shifts recognition in correlated process data using a neural network. Int J Syst Sci 32(2):137–143
14. de Freitas JFG (1999) Bayesian methods for neural networks. Trinity College, University of Cambridge
15. Feipeng X, Amirkhanian SN (2009) Effects of binders on resilient modulus of rubberized mixtures containing rap using artificial neural network approach. J Test Eval 37(2):129–138
16. Fioramanti M (2008) Predicting sovereign debt crises using artificial neural networks: a comparative approach. J Financ Stab 4:149–164
17. Guo Y, Dooly KJ (1992) The application of neural networks to a diagnostic problem in quality control. In: Proceeding of ASME-WAM: monitoring and control for manufacturing processes. pp 111–122
18. Guo Y, Dooly KJ (1992) Identification of change structure in statistics process control. Int J Prod Res 30:1655–1669
19. Han D, Tsung F, Li Y (2007) A CUSUM chart with local signal amplification for detecting a range of unknown shifts. Int J Reliab Qual Saf Eng 14(2):81–97
20. Hwarng HB (1997) A neural network approach to identifying cyclic behavior on control charts: a comparative study. Int J Syst Sci 28(1): 99–112
21. Junsub Y, Victor RP, Howard RC (2001) ARL comparisons between neural network models and control charts for quality characteristics that are non normally distributed. Econ Qual Control 16(1):5–15
22. Kaunga DL, Zhang J, Ferguson K, Steele C (2013) Reliable modeling of chemical durability of high level waste glass using bootstrap aggregated neural networks. Natl Comput 178–183
23. Khoo MBC (2004) Designs of runs rules schemes. Qual Eng 16(1): 27–43
24. Klein M (2000) Two alternatives to the Shewhart X control chart. J Qual Technol 32:427–431
25. Mahmoudi AH, Nourbakhsh S, Amali R (2012) An alternative approach to determine material characteristics using spherical indentation and neural networks for bulk metals. J Test Eval 40(2):211–219
26. Matchenko SN, Dube MP (2006) Bootstrap inference with neural-network modeling for gene-disease association testing. Computational Intelligence and Bioinformatics and Computational Biology. 1–7
27. Mohanty JR, Verma BB, Ray PK, Parhi DRK (2010) Application of artificial neural network for fatigue life prediction under interspersed mode-I spike overload. J Test Eval 38(2):177–187
28. Muammer N, Hasan G, Ihsan T (2007) Comparison of regression and artificial neural network for surface roughness with the cutting parameter in Cnc turning. Modeling and Simulation Engineering, Article ID 92717

29. Natarajan C, Muthu S, Karuppuswamy P (2011) Investigation of cutting parameters of surface roughness for a non-ferrous material using artificial neural network in Cnc turning. J Mech Eng Res 3(1): 1–14
30. Ning X, Wu C (2011) Improved design of quantile-based control charts. J Chin Inst Ind Eng 28(7):504–511
31. Ou YJ, Wu Z, Goh TN (2011) A new SPRT chart for monitoring process mean and variance. Int J Prod Econ 132(2):303–314
32. Ou YJ, Wu Z, Tsung F (2012) A comparison study of effectiveness and robustness of control charts for monitoring process mean. Int J Prod Econ 135(1):479–490
33. Perry M, Pignatiello JJ Jr (2002) A review of artificial neural network applications in control chart pattern recognition. In: Proceedings of the Industrial Engineering Research Conference, Orlando, FL
34. Perry MB, Spoerre JK, Velasco T (2001) Control chart pattern recognition using back propagation artificial neural networks. Int J Prod Res 39(15):3399–3418
35. Pugh GA (1991) A comparison of neural networks to SPC charts. Comput Ind Eng 21:253–255
36. Raviv Y, Intrator N (2010) Bootstrapping with noise: an effective regularization technique. Connect Sci 8:355–372
37. Riaz M, Saghir A (2009) A mean deviation-based approach to monitor process variability. J Stat Comput Simul 79:1173–1193
38. Riaz M, Abbas N, Does RJMM (2011) Improving the performance of Cusum charts. Qual Reliab Eng Int 27(4):415–424
39. Riaz M, Mehmood R, Does RJMM (2011) On the performance of different control charting rules. Qual Reliab Eng Int 27(8):1059–1067
40. Rowley HA, Baluja S, Kanade T (1998) Neural network based face detection. Pattern Anal Mach Intell 20(1):23–38
41. Ryu JH, Wan H, Kim S (2010) Optimal design of a Cusum chart for a mean shift of unknown size. J Qual Technol 42(3):311–326
42. Sagiroglu S, Besdok E, Erler M (2000) Control chart pattern recognition using artificial neural networks. Turk J Electr Eng 8:137–147
43. Shaban A, Shalaby M, Abdelhafiez E, Youssef AS (2010) Automated identification of basic control charts patterns using neural networks. J Softw Eng Appl 3:208–220
44. Sigut J, Piñeiro J, Est vez J, Toledo P (2006) A neural network approach to normality test. Intell Data Anal 10:509–519
45. Smith AE (1994) X-bar and R control charts interpretation using neural computing. Int J Prod Res 32:309–320
46. Velasco T, Rowe MR (1993) Back propagation artificial neural networks for the analysis of quality control charts. Comput Ind Eng 25(1–4):397–400
47. Western Electric Company (1956) Statistical quality control handbook. Indianapolis
48. Wu Z, Jiao JX, Yang M, Liu Y, Wang ZJ (2009) An enhanced adaptive Cusum control chart. IIE Trans 41(7):642–653
49. Zhang S, Wu Z (2006) Monitoring the process mean and variance by the WLC scheme with variable sampling intervals. IIE Trans 38(4): 377–387
50. Zhao Y, Tsun F, Wang Z (2005) Dual Cusum control schemes for detecting a range of mean shifts. IIE Trans 37(11):1047–1057
51. Zio E (2006) A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes. Nucl Sci 53(3):1460–1478