JJMIE

Volume 5, Number 4, Aug. 2011
ISSN 1995-6665
Pages 291 - 300

Jordan Journal of Mechanical and Industrial Engineering

A Diverse Neural Network Ensemble Team for Mean Shift Detection in X-Bar and CUSUM Control Charts

M. A. Barghash *,[a]

[a]Industrial Engineering Department, Faculty of Engineering and Technology, University of Jordan, 11942, Amman, Jordan

Abstract

In manufacturing processes, maintaining quality is associated with proper process mean parameters and product quality metrics. The early detection of mean changes is important to reduce the number of defectives or non-conformities in the production. In this work, a diverse ensemble of Artificial Neural Networks (ANNs) with a leader network have been used to achieve this purpose, then a performance comparison was conducted on two types of control charts: X-bar and CUSUM in addition to comparing it to individual neural network performance. The traditional and individual neural network performances were obtained from published literature. It was found that, the diverse ensemble of ANNs detects small shift in process mean far earlier (Shorter Average Run Length (ARL)) than individual ANN's, traditional X-bar and CUSUM control charts. The postulated reason for this improvement is that the ensemble ANN system analyses more than one sample point, rather, it considers the points pattern and overcomes the instabilities in individual ANN's.

Keywords: Neural networks ensemble; Artificial intelligence; Pattern recognition; team of networks; diversified artificial neural networks; X-bar control charts; CUSUM control charts

1. Introduction

1.1. Quality control and supervisory process control:

Quality control charts are graphical tools commonly used to discover patterns and assignable causes in the manufacturing processes. They help manufacturers to discover the root cause of the problem and to reduce the number of defective products. X-bar charts and CUSUM control charts play an important role in maintaining quality.

Within the scope of mean shift detection, the basic idea for quality process control analysis is to continuously sample the manufacturing process, where for each new sample collected, one of two binary decisions can be made: the process mean did not shift "Normal or no-shift" or the mean had changed "process not normal or shift existing". Also, two types of mistakes can be made, particularly, no-shift decision while the process mean had changed and a shift decision with no change in the process mean existing (false alarm). If the decision is no-shift, the sampling process is continued however, if the decision is a shift then the process is stopped and an attempt is made to discover the reason for the change in the process mean.

1.2. Traditional control charts development:

Shewart developed the basic X-bar control charts in the early 1920s. Among the earliest work in X-bar control charts was devoted to heuristics and zone testing [1-2]. The developed rules enhances the ability to discover mean shifts but also reduces the Average Run Length (ARL) when there is no-shift in the process that is, it increases the number of false alarms. The X-bar control charts perform well for large shifts, While, it is less effective for small shifts. This is due to the memory-less property of the Shewart X-bar control chart. Developed by Page in 1958 , the CUmulative SUM (CUSUM) control chart were introduced to include all previous sample points through summation. Roberts proposed the Exponentially Weighted Moving Average (EWMA) control charts [3]. It also includes a weighted sum of all previous sample points where the importance of any point decline over time by a selected factor. Both charts include a memory of previous points. But while the sum increases the power of discovering small shifts, it dilutes the effect of large shifts in the mean with previous non-shifted points and it also delays large shifts from being discovered. The CUSUM and EWMA detect small shifts earlier that Shewart X-bar control charts, while the X-bar chart outperforms the CUSUM and EWMA charts for large shifts. Lucas developed a V-mask for the CUSUM control chart to improve its abilities[4-5].

1.3. Artificial intelligence based methods:

One main Limitation for the traditional methods is that the basic X-bar and CUSUM charts (no V-mask or sensitizing rules) look only at the last point and not at the pattern. This is traditionally solved by adding sensitizing rules (i.e., 2 of 3 consecutive points outside 2-sigma limit, 4 of 5 consecutive points outside the 1-Sigma limit, 8 consecutive points on one side of the center line, etc). The developed sensitizing rules –if used- have a major

* Corresponding author. e-mail: mabargha@yahoo.com

drawback. In particularly, it increases the number of false alarms.

There is a need for a technique to analyze more than just the last point without increasing the number of false alarms. This raised the reason why many researchers turned to the application of artificial intelligence.

Several researchers have utilized Artificial Intelligence (AI) in control charts. AI methods can handle a specified number of input points at the same time. It categorizes the shift and no-shift cases into two different patterns. Training population is then prepared through computer process simulation or through process sampling. This is then used to train the neural network. That is, to estimate the different parameters (weights and biases) of the neural network. The neural network will then make the decision as shift and no-shift for runs other than those used for the training process. Although, the neural network will give perfect and accurate results for the population used in training, it might not be as successful for new population that has not been tested yet. This is due the infinite number of possible shapes of patterns (being a random process).

Another important perspective of AI technique is that it is mostly a continuous function. The output of the AI is usually a number or a score for each pattern. The usual procedure is to compare this output to a threshold, if the score is greater/less than the threshold, then the pattern is classified as a shift pattern. Threshold is usually set to achieve a low percentage of false alarms.

Early studies compared neural networks and Shewhart control charts for both large and small shifts in the mean and reported that neural networks are comparable to Shewhart X-bar and R-charts for large shifts and outperform them for small shifts [6-7] . Several artificial Intelligence tools were used including: Multi-layer perceptron [6-9], Linear Vector Quantization[10], Probabilistic neural networks[11], Modular neural networks [12], Adaptive Resonance Theory [13], Neuro Fuzzy techniques[14-17], Fuzzy technique[18-21], Support Vector Machine [22], Self-Organized Approaches[23] and Competitive Neural Networks[24].

The basic types of Artificial intelligence techniques, which are referred to “Individual” in this paper have outperformed the traditional techniques, but as will be illustrated shortly in section 5.3, the magnitude of difference was small. This might put some doubt on the abilities of the Artificial Intelligence techniques. However, one reason for this humble performance of individual type control charts is instabilities in performance of the AI techniques[25]. This can be overcome and heavily improved by group type or “ensemble type” Techniques, where multi-stage decision is made as will be illustrated in this paper.

1.4. Instabilities in neural networks and artificial intelligence:

Instability is a feedback control concept. A stable system reaches a target value, while unstable systems oscillate (entails instabilities) and do not reach their required target. Figure (1) shows a sample behavior of ANN’s. The upper chart is the process sample pattern, while the lower part is the decision of the ANN. It can be noticed that, the sample ANN in Figure (1) changes its decision with time. This is a form of instability noted in individual ANN’s. However, this type of behavior can be eliminated in group type or ensemble type ANN systems as will be shown in this work.

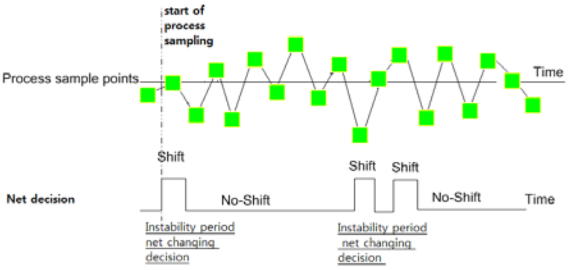

Figure 1: A sample behavior of ANN’s.

Instabilities have direct effect on the number of false alarms. Usually, the threshold for classification is modified to reduce the number of false alarms and thus, reducing -as a consequence- the ability of the neural network to discover small shift.

In this work, the instabilities are handled differently. By tolerating the instabilities initially and keeping the threshold used for classification at 0.5. Thus, the instabilities will occur at the initial stage of the ANN, and then another ANN stage is used for classification. The initial stage ANN, and for the case of large shift size, the neural network will be mostly giving a shift decision, but some time a no-shift decision can occur, as for the small shift, most likely the decision will be a no-shift decision but a shift decision is likely to occur. The second stage ANN needs to see more than one decision from the initial stage to make the final decision.

1.5. Ensemble type artificial neural networks:

To reduce the effect of ANN instability and to reach to smooth, robust and stable decision, it is possible to utilize neural networks in a hierarchical manner similar to organizations charts, where a group of “Individual Neural Networks” assess process pattern and make initial decisions (Figure (2)). Later, other networks are used to obtain the final decision. It is possible - at the initial stage- to have multiple decisions where some ANNs signal for a shift and some do not. This is called a neural networks ensemble. Instead of handling the original pattern, the initial neural networks provide their analysis to the following neural networks.

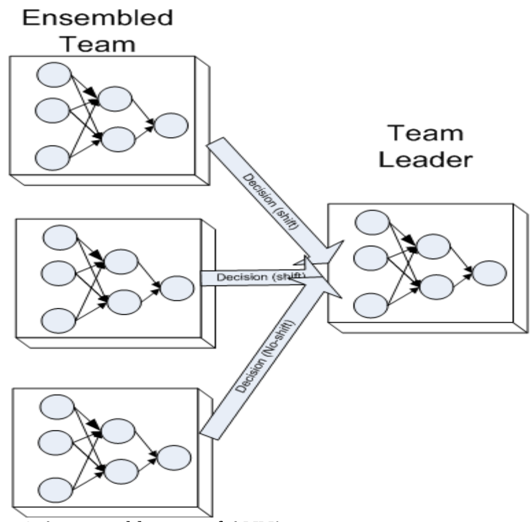

Figure 2:An ensemble team of ANN’s

Ensemble based neural network schemes have been used by several researchers [26-28]. With strong emphasis on ensemble diversity [22]. The collective method can then be: plurality voting, majority voting, winner takes all for classification purposes [24]. The diversity of the neural networks is achieved through varying the topology and initial conditions or algorithm involved. The training for the individual networks can be done by sequential (Ada-Boost) or parallel training (Bagging). The sequential method allows the initial nets to learn the training set. The outputs of the initial nets are the training set for the next group of nets and so on. The Bagging method diversifies neural networks by randomly selecting the input set for each network, then making a collective opinion for these sets [23]. ANNs are extremely affected in their decision by population percentage used for the training [16]. Thus can be used for diversification of the neural networks [15]. Such diversified ANN's are trained using different pattern percentages had shown tendency to have an error in judgment titling towards the pattern available more in the training population [15]. Figure (3) shows an illustrative example of neural network diversity, the process encounter a shift pattern. The first, second and third neural network respond in a different style. The three neural networks encounter instability, in that, they don't have a stable target decision, and instead, they fluctuate. However, the key difference between the three networks is in the number of shift decisions for a specified number of sample points. This is largest for the third neural network and smallest for the first.

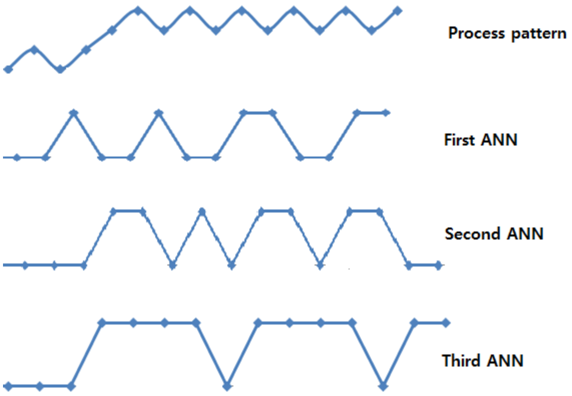

Figure 3: An example of diverse neural network response.

The process variations are the deviations in the process and product parameters. They include both random and assignable causes of product variations. Diversity, on the other hand, is a recent trend in algorithm design accompanied with population based algorithms such as genetic algorithms. A survey of population diversity measures using genetic programming (GP) is done by Burke [29]. Maintaining population diversity in genetic algorithms improves their performance.

Diversity is viewed as an enhancing addition to algorithms and in this work diversity is used to form a team of neural networks. Diversity in this sense is the diversity of the decision makers. Maintaining diversity is important to enhance the decision process. The diversity and its analysis had been proved to be critical in fields such as genetic algorithms; however, the potential of this important concept is not yet fully explored.

1.6. The focus in this paper:

In this work, a diverse team of neural networks is utilized to make an initial decision on shifts in control chart means. A leader neural network then makes the final decision using the initial decisions as inputs. Two types of charts are considered X-bar and CUSUM control charts. The CUSUM of input pattern tends to amplify the shift with the addition of samples. The performance of the ensemble is fine-tuned through a prior scaling factor and threshold based classification. This paper proves that teaming up diverse ANNs and including a leader results in a lower Average Run Length (ARL) and thus a lower Average Time-To-Signal (ATS) than traditional CUSUM control charts.

2. Methodology and Model Construction

2.1. The basic artificial neural network:

A neural network is composed of several interconnected neurons with the following activation function for each neuron [32].

$$output = \varphi\left(\sum_{i=1}^{p} w_i x_i - \theta\right) \quad (1)$$

Where

φ : is the neuron function

wi : is the neuron ith weight

xi : is the ith input to the neuron

θ : is the constant threshold value

p : number of inputs to the neuron.

Neural networks have high interconnectivity, where all the neurons are arranged in layers. The outputs of the first layer are the inputs for the second layer as shown in Figure (4). The first layer is the input layer and the final layer is output layer. The layers in between are called hidden layers.

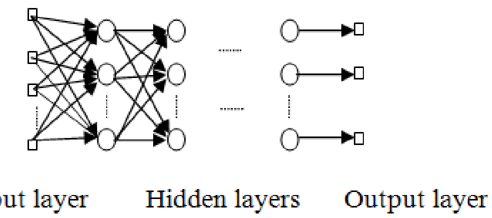

Figure 4: Layers of neural networks.

Training of the neural network is done using back propagation where the error value and the local gradient of the error are used to estimate the new value of the weights and biases.

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \tag{2}$$

Where

Wji	:	is the synaptic weight connecting the output of neuron i to the input of neuron j.
η	:	is the learning rate
δj(n)	:	is the local gradient for the error in output neuron j at iteration n.
yi(n)	:	is the output of the network at neuron i at iteration n.
n	:	iteration number. Several types of neuron activation functions from Mathworks Matlab 7.0 were used including the LogSig sigmoid function, TanSig and PureLin as shown below

$$Logsig(x) = \frac{1}{1+e^{-x}} \tag{3}$$

$$Purelin(x) = b + x \tag{4}$$

$$Tansig(x) = sign\left(\frac{2}{1+e^{-2x}} - 1\right) \tag{5}$$

2.2. Diverse team of nets:

2.2.1. Diversity of the team:

Pattern percentages have been used to achieve ANN’s diversification as shown in Table 1. Column 2 shows the percentages of the patterns for each of the neural networks given in column 1.

Table 1: Population composition and training error for the ANN’s.

NET	Training Population N: Normal (no shift) S: Shift	Training Output N: Normal (no-shift) S: Shift	Average of Square Error
1	(50%, N; 50%, S)	(1, N; 0, S)	$4.01 * 10^{-5}$
2	(75%, N; 25%, S)	(1, N; 0, S)	$3.23 * 10^{-5}$
3	(90%, N; 10%, S)	(1, N; 0, S)	$4.21 * 10^{-6}$
4	(50%, S; 50%, N)	(1, S; 0, N)	$5.32 * 10^{-5}$
5	(75%, S; 25%, N)	(1, S; 0, N)	$4.25 * 10^{-5}$
6	(90%, S; 10%, N)	(1, S; 0, N)	$5.12 * 10^{-5}$

2.2.2. ANN configuration and training:

The number of samples or input size is of little significance. The number of input samples has little impact on the performance of the ANN[16]. Thus, an arbitrary input size of 20 was satisfactory for the purpose of this paper.

The construction of the ANN's was in four layers: input layer, two hidden layers and one output layer. The size of the layers (number of neurons) in not highly significant[16]. Increasing the layer size more the 15 does not improve the performance. However, if we reduce it appreciably below 15 the performance may deteriorate. The selected size for the number of neurons is 15. The layers were composed of LogSig functions for the input layer and the second layer, TanSig for the third layer and PureLin for the fourth layer.

The training average sum of square errors for the different ANNs is shown in Table 1-column 4. The output of the neural networks is shown in column 3. Any deviation from these values will be considered as error and is included in the sum of squares error in column 4.

As is shown in column 4, the ANN's are trained well to a very small sum of square error and can perform extremely well to classify the training population. But the ANN might not be successful in classifying other populations. The average sum of square error is a measure of the training quality for the individual ANN. But it is not a measure of its decision quality.

2.3. Leader network:

The leader network is a neural network that accepts the outputs of the different neural networks and is trained to make the final decision. The leader network is trained in the same manner as the rest of the networks, however, the input to the team leader is the decisions of the other networks. The configuration of the leader ANN's consists of four layers: input layer, two hidden layers and one output layer. The LogSig function was used for the input layer and the second layer, TanSig was used for the third layer and PureLin was used for the fourth layer.

A threshold level classification for the output of the leader net is used. Patterns getting an output of less than a selected threshold value are classified as “no-shift” (normal) and those getting above the threshold are classified as “Shift” patterns. The threshold level classification is the most used classification for the output of the neural nets among the researchers. In this work the threshold of classification was chosen to establish a near-value to the traditional X-bar chart and traditional CUSUM charts for the no-shift case.

2.4. Ensemble construction:

2.4.1. X-bar type ensemble:

As discussed earlier ANN’s suffer from instabilities. The ANN seems to change their decision for the same pattern as more sample points are collected. Thus, in this work, three consecutive decisions for each individual neural network are used. These are then submitted as inputs to the leader pattern (As shown in Figure (5)). The size of the input to the leader pattern is 18 (see Figure (5)). To smooth the output of the Leader-net, a 2-point moving average is used. This procedure is explained through an example later in section 4.0. The 2-point moving average is effective in smoothing the output of the ensemble.

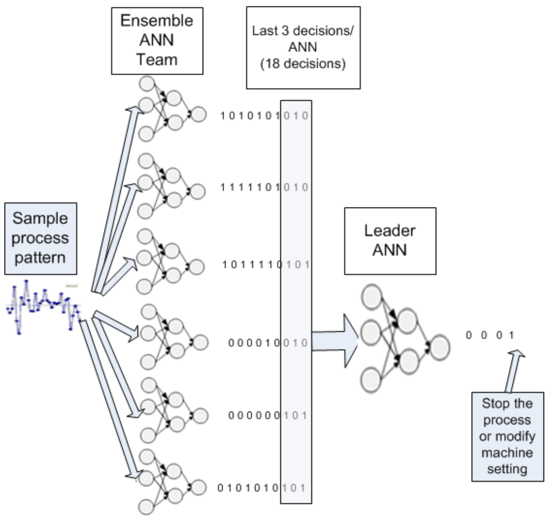

Figure 5: The framework of the ANN ensemble for mean shift detection in X-bar control charts.

2.4.2. CUSUM based ensemble:

Figure (6) shows the framework for the mean shift detection. The data from the process are standardized and summed according to the following equations:

CUSUM (S_0 = 0)

$$S_i = \overline{X}_i + S_{i-1} \tag{6}$$

$$\text{Normalized CUSUM} = \frac{S_i - \mu}{\sigma_{\bar{x}}} \tag{7}$$

Where

i : is the sample number

The Normalized CUSUM is divided by a factor "A" before passing them to the ensemble of nets. A is a tuning parameter for the performance of the ensemble.

Q=Normalized CUSUM/A (8)

For the initial networks, the input size is 20 process sample points for each decision. Three consecutive decisions / net are assembled for the input to the leader net. Altogether 18 inputs are used for the leader net. To smooth the leader net output a "two-point moving average" is used. A threshold based classification is then used. If the smoothed output of the leader net is below the threshold process, the process is considered normal (i.e., 0), otherwise, a shift (i.e., 1) decision is made.

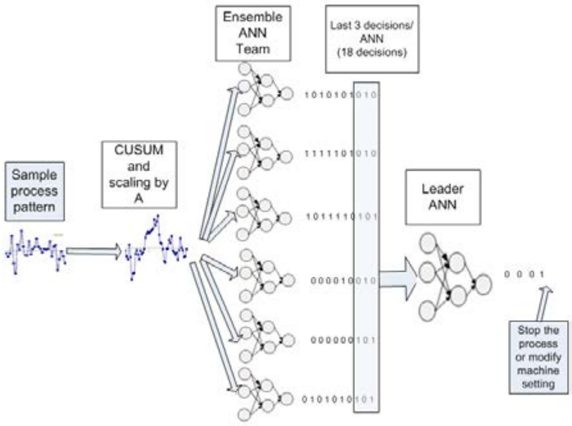

Figure 6: The framework for Ensemble ANN mean shift detection in CUSUM control charts.

3. Pattern Generation

Equation (3) is used to generate the shift and no-shift patterns:

$$\textit{Generated pattern } y(i) = \mu + n(t) + d(t) \tag{9}$$

where

- μ : is the process mean
- n(t) : is the normal cause of variations which is N(0, σ).
- d(t) : is the special cause of variations

For the case of a shift:

d(t) = u * s

Where:

- u : is the position of the shift (it can be either 0 or 1)
- s : is the value of the shift in the mean.

The following standardization was chosen for the patterns:

$$Y(i) = \frac{y(i) - \mu}{\sigma_{\bar{x}}} \tag{10}$$

The final variable Y(i) is a standardized variable with mean zero and a standard deviation of one for the no shift case. The mean shift values (s) can be expressed in a standardized form: as 4, -3, -2,-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1, 2, 3 and 4 or they can be expressed as a multiple of the standard deviation. The "s" variable is set to zero for the no-shift case.

4. An Illustration of the Suggested Model for the X-bar Chart

In this section, the basic mechanics of the suggested model (shown in Figure (5)) will be illustrated. I will use a numerical example with the following assumptions: Those were selected to eliminate the need for the standardization step. I will consider two cases: no-shift case d(t)= 0, and shift case with d(t)=0.25. The function in MATLAB generates normally distributed data with a mean of zero, and a standard deviation of one. The chosen input size for the neural network ensemble is 20, thus 20 sample points need to be generated.

Iteration 1 samples 1-20:

Using y = randn(20,1); to obtain 20 sample points:

y(1),…,y(20) = {-0.1172 ,0.9572 ,0.3863 ,0.5589,0.9682 , -0.1059,-1.6905 ,-1.2510 ,-0.8900 0.3454, 0.7927,0.2004, -0.2301, -0.5378, 1.8070, -0.8698, 0.0232, -0.1273, 0.8690, -0.2178}.

This data acts as the input for the individual ANNs. Table 2 shows the decisions of the individual nets. These values are obtained using the function "sim (network_name, data_vector)". I cannot still submit these inputs to the leader net because two more consecutive decisions / net are required. So, I move to the next two iterations.

Table 2: Iteration 1 results.

NET	Output	Classification >0.5 <0.5 1 shift 0 no-shift	Decision after classification
1	0.4821	Shift	0
2	0.7995	No-shift	1
3	0.9970	No-shift	1
4	-0.0038	No-shift	0
5	-0.2756	No-shift	0
6	0.9063	Shift	1

Iteration 2 and Iteration 3:

For iteration 2 and 3, two new sample points are generated using randn():
y(21) and y(22) = 1.1071, -0.0009.
With each new point, the newly generated point is added at the end of the already generated pattern in iteration 1 and remove the first point. The number of inputs to the individual neural network should remain 20. The input to the networks in iteration 2 becomes: {0.9572 , 0.3863, 0.5589, 0.9682 , -0.1059,-1.6905 , -1.2510 , -0.8900, 0.3454, 0.7927, 0.2004, -0.2301, -0.5378, 1.8070, -0.8698, 0.0232, -0.1273, 0.8690 , -0.2178, 1.1071}.
The modifications to iteration 2 are shown in bold font. For the third iteration it becomes: {0.3863, 0.5589, 0.9682, -0.1059, -1.6905, -1.2510, -0.8900, 0.3454, 0.7927, 0.2004, -0.2301, -0.5378, 1.8070, -0.8698, 0.0232,-0.1273, 0.8690, -0.2178, 1.1071, -0.0009}.
Table 3 shows the individual neural network decisions for iterations 2 and 3.

Table 3: Iteration 2 and 3 results.

	Iteration 2		Iteration 3	
NET	Output	Value for the decision	Output	Value for the decision
1	0.1813	0	-0.1959	0
2	1.0486	1	1.1628	1
3	0.9974	1	0.9998	1
4	0.0026	0	0.0011	0
5	-0.0486	0	-0.1628	0
6	0.8187	1	1.1959	1

At the end of iteration 3, enough data will be available to start activating the leader network. The input to the leader net is formed as row-by-row combination of the decision columns combining Tables 3 and 4, that is: {0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,1,1,1}.

Iteration 4 (final iteration for the no-shift case)

We then use the MATLAB function sim(leader_net,data_vector) to obtain the output of the leader net. MATLAB result is given as 0.0142. This value is below the threshold value and is classified as no-shift pattern.

For the shift pattern case, I have randomly selected 0.25σ or 0.25 standard shift starting at the 20th sample.

Iteration 2 (following iteration 1):

The variables u, s, d and y in equation (3) are given as follows, note that the changes to iteration 1 are shown in bold:
u= [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1] ,
s =0.25,
d = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.25 0.25 0.25],
y =[-0.1172, 0.9572, 0.3863, 0.5589, 0.9682, -0.1059, -1.6905, -1.2510, -0.8900, 0.3454, 0.7927, 0.2004, -0.2301, -0.5378, 1.8070, -0.8698, 0.0232, -0.1273, 0.8690, 0.0322, 0.3571, 0.2491].

The same procedure shown above for the no-shift case is then used, by selecting sample points 1-20, to obtain the individual networks decision, then the sample points 2-21, and sample points 3-22. The inputs for the leader net are obtained in the same manner from the networks individual decision. Then the leader network can be used to make the final decision.

5. Results and Discussion

The results part of this work is composed of three sections: the first section analyses the output of the individual networks and the leader net. The second section analyses and optimizes the threshold and the third part compares the optimized results obtained to those of the traditional X-bar chart and individual neural networks.

5.1. Analyzing the output of the neural networks:

The intention of this section is to analyze the output of the individual networks having a threshold level equal to 0.5 for the individual nets and a level of 0.85 for the leader net. A standardized shift of 1 is chosen for the analysis because it has a relatively short ARL. The mean shift occurs at the sample number 21.

Figure (7) shows a sample run results for the suggested model. Figure (7)-a shows the sample points generated. It clearly shows that a mean shift has occurred around sample number 21. Figure (7- b, c and d) shows the response of individual Networks 1-3. These are classifying the pattern: as "1" for no-shift case and "0" for the shift case (see Table 2). Some instability is observed in their decisions. It is clearly noted in Net 2 which had changed its decision 6 times (i.e., 1 → 0 and 0 → 1). This instability reduces the benefit of individual networks for pattern classification and definitely justifies the need for an ensemble model. Figure (7- e,f and g) shows the decision of the Nets 4,5 and 6. Instabilities are noted also in the behavior of the Nets 4,5 and 6. Instabilities are noted also in the behavior of Net 5. Figure (7-h) shows the output of the leader net. The output of the leader net is certainly stable and consistent and no instabilities are noted in its behavior. The value of the output exceeds the threshold level at sample 27 and a sample run length of 6 is recorded for this case (the run started at sample 21 and is detected at sample 27).

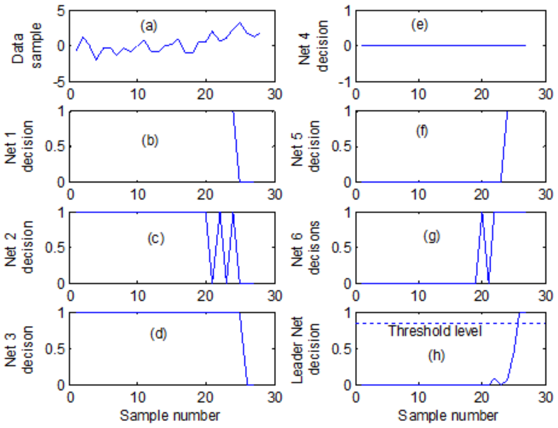

Figure 7: Output for the individual and the leader nets for a sample pattern for a standardized mean shift size of 1.

High variability is noticed at some sample points. Sample 21 -for example- has one of the highest decision variability where the Nets 1,3 and 6 clearly state it is a shift pattern, while the other Nets considers it as no-shift pattern. Sample 27 on the other hand, has the lowest variability. This conflicting decision behavior also advocates the need for an ensemble based technique rather than individual neural network.

It can be noticed that Net 2 and Net 6 have discovered the change in the process mean rather early. Their first signal for a change in the process mean comes in sample 21. The ensemble on the other hand detected the shift at sample 27. However, these two individual nets suffer from extreme instabilities.

5.2. Output analysis for suggested ensemble and threshold selection:

Table 4 shows the ARL results for several shift patterns. Column 1 is the standardized value for the shift and columns 2 to 5 are selected results at different threshold levels.

Table 4: ARL at different threshold level.

Shift	Threshold level			
	0.85	0.94	0.95	0.96
4.00	3	3.04	3	3.04
3.00	3.44	3.56	3.44	3.4
2.00	4.2	4.24	5.04	4.52
1.00	10.56	10.28	11.64	9.72
0.75	14.68	13.56	14.88	14.52
0.50	27.6	34.12	33.48	32.76
0.25	100.6	114.28	134.56	84.64
0 *(no shift case)*	235.52	237.32	294.28	416.28
-0.25	47.36	102.64	81.64	75.16
-0.50	22.64	20.68	20.12	37.52
-0.75	9.16	11.52	14.44	14.48
-1.00	9	9.72	9.36	8.76
-2.00	3.68	4.16	4.36	3.96
-3.00	2.44	3.2	2.88	3
-4.00	2.28	2.6	2.76	2.56

The results show that as the threshold increases, the ARL for the no shift case increases. However, there is slight change for the shift case.

5.3. Comparing the ensemble model to X-bar and individual ANN's:

Table (5) compares the ARL for traditional X-bar chart and individual ANN to the suggested ensemble model. The average run length for the best arrangement in Table 4 (column 5) is selected for comparison and is given in Table 5- column 4. It is compared to traditional techniques X-bar and individual neural network results. Column 2 and 3 are obtained for benchmarking purpose from Yi et al (2001).

Table 5: Comparison of the results of the X-bar chart, NN and the suggested ensemble.

Shift	X-bar chart 3-sigma limit	NN[30]	**Selected ensemble results**
4.00	1.19	1.19	3.04
3.00	1.98	1.97	3.4
2.00	6.28	6.17	4.52
1.00	44.9	43.48	9.72
0.75	83.27	79.48	14.52
0.50	157.1	151.09	32.76
0.25	280.16	272.85	84.64
0 *(no shift case)*	372.28	371.96	416.28
-0.25	282.48	292.57	75.16
-0.50	153.81	163.16	37.52
-0.75	43.94	45.87	14.48
-1.00	14.99	15.68	8.76
-2.00	6.31	6.51	3.96
-3.00	2.02	2.05	3
-4.00	1.18	1.19	2.56

 © 2011 Jordan Journal of Mechanical and Industrial Engineering. All rights reserved - Volume 5, Number 4 (ISSN 1995-6665)

Table (5)-column 4 shows that the suggested model is detecting small process mean shifts far earlier than both individual ANN and X-bar chart. Let's take for example the 0.25 standardized shift, here, the ARL for the suggested ensemble is 97.72 as compared to 272.85 for the individual NN and 280.16 for the X-bar 3 sigma limit. This represents a large difference and shows that the ensemble model is superior to the two other methods. It is also noticed that the ensemble model detects very large shifts such as 3-sigma or 4 sigma nearly one sample point afterward as compared to the other methods. This is due to using more than one decision point and using the moving average technique to smooth the output results.

5.4. Results for the CUSUM control chart:

Table 6 shows the various results that can be obtained from the CUSUM through varying the upper and lower control limits (CL factor).

Table 6: Comparison of tradition CUSUM and new the framework.

	Traditional CUSUM		Selected ensemble results	
Standard mean shift size	CL =4	CL=5	Threshold=0.09 A=2.8	Threshold=0.09 A=2.7
4	1.71	2.01	4.36	5.2
3	2.19	2.57	5.4	6
2	3.34	4.01	6.24	7.52
1	8.38	10.4	10.32	13.64
0.75	13.3	17	12.24	18.72
0.5	26.6	30	15.08	23.44
0.25	74.2	140	39.68	51.76
0	336	930	324.04	906.84
-0.25	74.2	140	33.2	38.12
-0.5	26.6	30	18.64	27.48
-0.75	13.3	17	15.2	15.72
-1	8.38	10.4	10.36	11.8
-2	3.34	4.01	7.24	8.64
-3	2.19	2.57	4.44	6.52
-4	1.71	2.01	4.6	5.36

Table (6) shows the results obtained for a traditional CUSUM for different control limits (columns 2, 3) and the suggested ensemble (columns 4, 5). It is noted that for low shift values the suggested model is highly superior to that of the traditional X-bar technique. Let us for example take the 0.25 standardized shift case and compare columns 3 and 5. The traditional CUSUM has a high ARL value of 140, while the suggested model has an ARL of 51.76. A large difference in value is noticed which shows the superiority of the suggested model.

5.5. A sample ARL distribution:

Figure (8) shows a sample ARL histogram for the CUSUM ensemble for a shift size = 0.5*sigma. ARL distribution is generally characterized with wide spread. The current ensemble ARL is not different. 500 ARL data points are used in the construction of the histogram in Figure (8). The mean ARL is nearly 23. However, an ARL more 4 time this amount is found (nearly 106). This is a general characteristic of ARL distributions and is noticed in traditional X-bar charts and in CUSUM control charts as well.

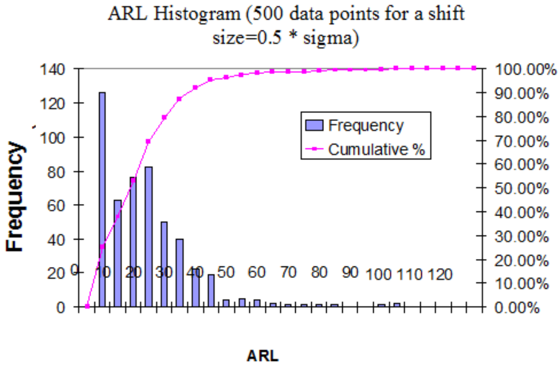

Figure 8: A sample ARL distribution for the CUSUM based ensemble with shift size=0.5*sigma.

6. Case study: Material Variations in plastics sheet

6.1. Introduction:

In this section, the developed model will be applied to plastics sheet extrusion. Plastics manufacturing is a batch type process and the batch-to-batch variations is a common disturbance in polymer manufacturing processes. If the material is highly variable then defective parts may be produced. Figure (9) shows the sheet extrusion process, The extrusion process melts the plastics pellets to a viscosity level V and the viscous form is pumped using a rotating screw extruder at head speed HS towards a heated die to temperature DT, the sheet shaped extruded material is pulled in between two rollers rotating at roller speed RS.

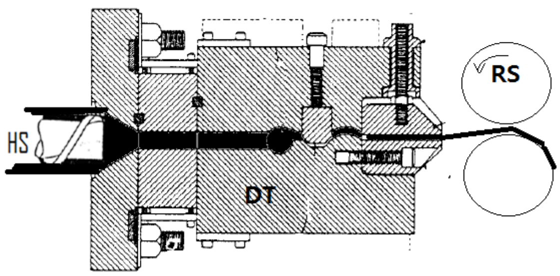

Figure 9: The sheet metal extrusion process schematic.

6.2. SIMULATOR software:

The following section uses simulation for the plastic sheet extrusion generated by the simulation package called SIMULATOR. Simulator performs Monte-Carlo simulation of products and processes for the purpose of training in designed experiments, robust design and tolerance analysis[31]. The Simulator software generates random values of sheet thickness simulating actual process values. Although, the focus of the software is for robust process design, it can be used for control charts design.

6.3. Data Generation:

The sheet thickness is an important customer quality objective. A model relating thickness to the input parameters is found by the following expression[31].

$$TH = 64.93479 - 0.10766RS - 0.001058RS^2 + 0.48434DT - 0.001058DT^2 + 0.001084RS*DT + 5.0V + 0.0075HS + ERR \quad (5)$$

Where, RS: The Roller speed (rpm), DT: The Die temperature oF, V: The Viscosity (coef), HS: Head speed (rpm). The data is generated using the Simulator software randomly. The mean and the standard deviation of the data is estimated for normalizing purpose. The CUSUM framework is then used to analyze the data.

6.4. Viscosity variation detection:

The batch-to-batch variation is an important variation cause in sheet extrusion. It affects the product thickness and material strength. In this subsection, I have used the CUSUM based enseble to detect the viscosity variations. Figure (10) shows the ARL for the CUSUM framework. The X-axis being the viscosity variation and the Y-axis represents the ARL required to detect the change. The threshold and scaling factor A are set to achieve a low false alarm rate.

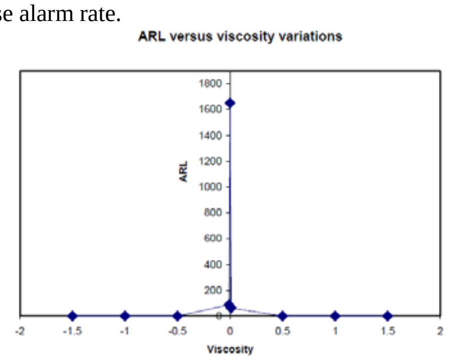

Figure 10: ARL versus viscosity variations for the sheet extrusion process.

Table 7: The tabulated results of ARL versus viscosity.

Viscosity variation	ARL
-1.5	1.5
-1.2	1
-0.5	2.9
-0.01	87
0	1650
0.01	63
0.5	3
1.2	4
1.5	1.3

From Figure (10) and Table (7), the suggested framework can be used to detect the variations as much as 0.5 in an ARL less than 3. Smaller variations as low as 0.01 can be detected in less than 87 ARL. However, if there was no viscosity variations false alarms will appear once every 1650 points which is a relatively low rate of false alarms.

7. Conclusion

It was found that the new suggested model can be tuned through added parameters such as threshold or a scaling factor. If the new model is tuned such that it gives close to the same amount of false alarms as that of the X-bar or CUSUM control charts. Then this model will be able to detect small shifts (0.25sigma) far earlier than the traditional X-bar or CUSUM control charts. However, it will detect larger shifts slightly later than the traditional control charts.

The ensemble was successfully tuned to detect batch-to-batch variation in sheet extrusion, with a false alarm ARL of 1650 and it can discover a small shift of 0.01 in viscosity with an average ARL = 87.

References

[1] Western Electric Company. Statistical quality control handbook. Indianapolis: Western Electric; 1958.

[2] L.S. Nelson, "The Shewart control chart-tests for special causes". Journal of quality technology, Vol. 16, 1984, 237-239.

[3] S.W. Roberts, "Control chart tests based on geometric moving averages". Technometrics, Vol. 1, 1959, 239-250.

[4] J.M. Lucas, "Combined Shewhart-CUSUM quality control schemes". J. Qual. Technol., Vol. 14, 1982, 51-59.

[5] J.M. Lucas, "A modified V mask control scheme". Technometrics, Vol. 15, 1973, 833-847.

[6] A.E. Smith, "X-bar and R control charts interpretation using neural computing". International Journal of Production Research, Vol. 32, 1994, 309-320.

[7] G.A. Pugh, "A comparison of neural networks to SPC charts". Computers in industrial engineering, Vol. 21, 1991, 253-255.

[8] D.T. Pham, E. Oztemel, "Control chart pattern recognition using neural networks". Journal of Systems Engineering, Vol. 2, No. 4, 1992, 256-62.

[9] C.S. Cheng, "A multi-layer neural network model for detecting changes in the process mean". Computers and Industrial Engineering, Vol. 28, No.1, 1995, 55- 61.

[10] D.T. Pham, E. Oztemel, "Control chart pattern recognition using learning vector quantization networks" .International Journal of Production research, Vol. 32, No. 3, 1994, 721-9.

[11] J. Plummer, "Tighter process control with neural networks". AI Expert, Vol. 8, 1993, 49-55.

[12] C S. Cheng, "A neural network approach for the analysis of control charts". International Journal of Production Research, Vol. 35, No. 3, 1997, 667-97.

[13] HB. Hwrang, WC. Chong, "Detecting process non-randomness through a fast and cumulative learning ART-based pattern recognizer". International Journal of Production Research, Vol. 33, 1995,1817-1833.

[14] Chang, CA. Aw, "A neural fuzzy control chart for detecting and classifying process mean shifts". International Journal of production research, Vol. 34, 1996, 2265- 2278.

[15] N. Santarisi , M A Barghash, "Neuro-fuzzy based model for pattern recognition in control charts". Proceeding of the 37th international conference on computers and Industrial Engineering 2007, 1881-1897.

[16] M. Barghash, N. Santarisi, "Analyzing the Effect of Training Parameters on the Performance of the ANN Using Fractional

Factorial Experimental Design". Journal of intelligent manufacturing Vol. 15, 2004, 635-644.

[17] DT Pham, MA Wani, "Feature based control charts pattern recognition". International Journal of production research,Vol. 35 No.7, 1997, 1875-1890.

[18] M. Gülbaya, C. Kahramana, "Development of fuzzy process control charts and fuzzy unnatural pattern analyses". Computational Statistics & Data Analysis,Vol. 51, 2006, 434- 451.

[19] S.K. Guuri, S. Chakraborty, "Recognition of control charts using improved selection of features". Computers in Industrial Engineering, 2008,

[20] S.K. Guuri, S. Chakraborty, "Feature-based recognition of control chart patterns". Computers in Industrial Engineering, Vol. 51, 2006, 726–742.

[21] K. Assaleh, Y. Al-Assaf, "Features Extraction and Analysis for Classifying Causable Patterns in Control Charts". Computers and Industrial Engineering, Vol. 49, 2005, 168-181.

[22] C.S. Cheng, H.P. Cheng, "Identifying the Source of Variance Shifts in the Multivariate Process Using Neural Networks and Support Vector Machines". Expert Systems with Applications, Vol. 35, 2008, 198-206.

[23] R.S. Guh, Y.R. Shiue, "On-Line Identification of Control Chart Patterns Using Self-Organizing Approaches". International Journal of Production Research, Vol. 43, 2005, 1225-1254.

[24] 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, 2007, 227-244.

[25] M. Barghash, N. Santarisi , " A study on the response and stability of Artificial Neural Networks for pattern recognition in control charts". 12th International Conference on Machine Design and Production, Kusadasi, Turkey 2006.

[26] S. Geman, E. Bienenstock, R. Doursat, "Neural networks and the bias/variance dilemma". Neural computations, Vol. 4, 1992, 1-58.

[27] J. Yu, L. Xi, "A neural network ensemble-based model for on-line monitoring and dignosis of out of control signal in multivariate manufacturing processes". Expert systems with applications, Vol. 36, 2009, 909-921.

[28] Z H. Zhou, JX. Wu, W. Tang, "Ensembling neural networks: many could be better than all". Artificial intelligence, Vol. 137, 2002, 239-263

[29] Edmund Burke, Steven Gustafson, Graham Kendall, "A survey and analysis of diversity measures in genetic programming". In GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, pages 716-723. Morgan Kaufmann Publishers, 2002.

[30] J. Yi, V R. Prybutok , HR. Clayton, "ARL Comparisons Between Neural Network Models and $\bar{x}$-Control Charts for Quality Characteristics that are Nonnormally Distributed". Economic Quality Control, Vol. 16 No.1, 2001, 5- 15.

[31] SIMULATORsoftware,2009,http://www.variation.com/sim/

[32] S. Haykin. Neural Networks a Comprehensive Foundation. New Jersey: Macmilllan; 1994.