Industrial Engineering & Management
ISSN: 2169-0316

Research Article | Open Access

Neural Network Control Chart Architecture for Monitoring Non-Conformities in a Poisson Process

Yousef S Alhammadi[1]* and Michael Adams B[2]

[1]*Assistant Professor, Engineering Systems & Management, Masdar Inisititute, Abu Dhabi, United Arab Emirates*
[2]*Professor of Statistics, Information Systems, Statistics & Management Science, The University of Alabama, Tuscaloosa, USA*

Abstract

The uses of Neural Network (NN) models have recently been recommended as statistical quality control (SQC) tools. The advantages of NNs, particularly the robustness of the nonlinear modeling abilities, are appealing to quality control practitioners for use in process monitoring. Advances in computing power have also made the Neural Network Control Charts (NNCC) an alternative SQC technique.The systematic Design of Experiment (DOE) methodology is employed to find near optimal NN topology for NNCC for Poisson data. A (2k) full factorial design is implemented and supplemented as needed to investigate NN topologies. The effect of the following factors were investigated through a simulation study: the number of the inputs "n", the number of nodes in the hidden layer(s), the training data size, and in-control mean for shift range 0-3σ . The guidelines and steps of constructing the DOE study for the NNCC is given, along with an example.

Keyword: Neural network; System design; Pison process; Neural control charts

Introduction

The control chart was first introduced by Shewhart in the 1920s. The most commonly used attribute control charts available are P-charts, used when the fraction of nonconforming items are monitored in a specific sample of size "n" products. The underlying assumed distribution in this case is the Binomial distribution. To monitor the number of nonconformities or defects, the Poisson distribution is assumed and the C-chart or U-chart is the appropriate chart. Examples include number of scratches in a 1 m^2 table, the number of undesired bumps in one mile of new road, or the number of defaulted loans given by a bank in some specific time period.

This paper aims to investigate and propose the best topology of a neural network to monitor the nonconformities in the quality process, using a Feed-forward multi-layer perceptron (MLP) neural network. The factors to be studied are the in-control mean level, shift size δ, the number of inputs, the number of the neurons in the hidden layer, and the training data size.

The paper is organized in the following fashion: Sections 2 and 3 contain brief descriptions of the C-charts and the MLP neural networks, respectively. Section 4 discusses the design of the proposed neural networks (NN), including the experimental design, noise factors, and their levels. The simulation process is described in detail in Section 5, while Section 6 is reserved to describe the method of finding the appropriate upper control limits (UCL) point that leads to an in-control ARL of 370. Finally, Sections 7 & through 9 provide examples of the results, conclusions, and recommendations.

C-chart and Poisson Distribution

The Poisson distribution is one member of the family of discrete distributions. It is useful in modeling the number of events that accrue over some specific continuous data such as area or time. If X follows the Poisson distribution, the probability of mass function (pmf) of X is as follows

$$f(x)=\frac{e^{-\lambda}\lambda^{x}}{x!} \quad x=0,1,2\ldots \quad \text{and } \lambda>0. \tag{2.2.1}$$

The mean of X, denoted as μ_x, and the standard deviation of X, denoted as σ_x, may be shown to be $\mu_x=\sigma_x^{\,2}=\lambda$.

In quality control, if the number of non-conformities in a Poisson process is monitored, then the C-chart is an appropriate tool. The values C_i, for i=0, 1, 2….m, are plotted over time where C_i is the number of non-conformities that occur in the i^{th} inspection unit.

The 3σ control limits for the C-chart are given by

$$\begin{aligned}&\text{Center line} = \bar{C}\\ &\text{UCL}=\bar{C}+3\sqrt{\bar{C}}\\ &\text{LCL}=\bar{C}-3\sqrt{\bar{C}}\end{aligned} \tag{2.2.2}$$

where $\bar{C}=\sum_{i=1}^{m}\frac{c_i}{m}$ for a historical set of "m" observations, and UCL and LCL denote the Upper Control Limits and the Lower Control Limits, respectively.

The conventional measure of control chart performance is the Average Run Length (ARL) index. Let RL denote the random variable that reports how many inspections are made until the first out-of-control signal occurs. For this application, we have $RL \sim geometric(p)$ where $p=1-P(LCL\le C_i \le UCL)$. It follows that $ARL=\mu_{RL}=\frac{1}{p}$ for an in-control process.

In general, the ARL measures how long, on average, it takes the chart to detect a process shift. A large ARL is desired for in-control cases, while a short ARL is desired for out-of-control cases. More detail concerning C-charts may be found in [1,2].

***Corresponding author:** Yousef S Alhammadi, Assistant Professor, Engineering Systems & Management, Masdar Inisititute, Abu Dhabi, United Arab Emirates, E-mail: yalhammadi@masdar.ac.ae

Received May 02, 2013; **Accepted** May 08, 2013; **Published** May 13, 2013

Citation: Alhammadi YS, Michael Adams B (2013) Neural Network Control Chart Architecture for Monitoring Non-Conformities in a Poisson Process. Ind Eng Manage 2: 115. doi:10.4172/2169-0316.1000115

Neural Networks

Neural networks are distribution-free models that have proven capable of well approximating almost any continuous function. Among the many types of neural networks are the Multi-layer perceptron (MLP), Learning Vector Quantization (LVQ), Radial basis function (RBF), Adaptive Resonance Theory (ART), and Kohonen self-organizing network (SOM). The MLP is perhaps the neural network most widely known and used in Statistical Quality Control (SQC). The MLP consists of at least two layers: inputs which pass the data into the network, and an output layer which communicates the network results to the end-user. In addition to these layers, there is typically at least one hidden layer that connects the inputs and output layer. All calculations of the network are done within the hidden layers. A typical three-layer NN is shown in Figure 1.

The kth value of the output layer is given by:

$$\hat{y}_k = g_2\left(\sum_{j=0}^{M} W_{kj}\, g_1\left(\sum_{i=0}^{N} w_{ji}\, x_i\right)\right) \tag{2.3.1}$$

where

$\hat{y}_k$ is the estimated output value, and

g_1 and g_2 are the activation functions.

W_{kj} is the estimated weight value between the kth output and the jth hidden node,

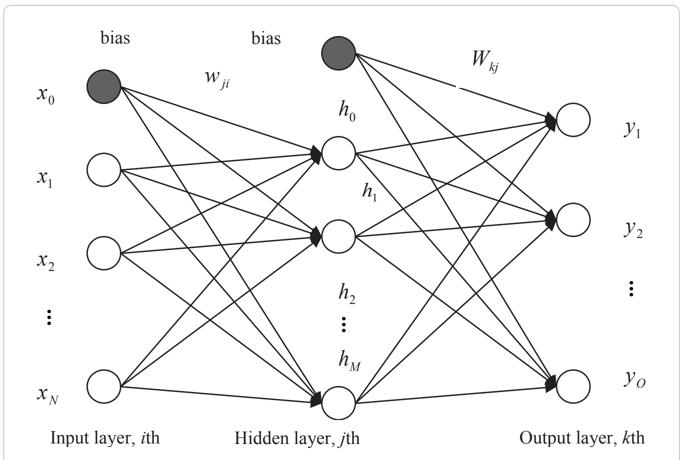

Figure 1: Typical Feed-forward Neural Network.

Time	x1	x2	x3	x4	x5	$\hat{y}$
1	3	8	7	0	9	0.8477
2	8	7	0	9	3	0.7366
3	7	0	9	3	4	0.5398
4	0	9	3	4	3	0.3057
5	9	3	4	3	4	0.4668
6	3	4	3	4	5	0.3668
7	4	3	4	5	11	0.9284
8	3	4	5	11	11	0.9943
9	4	5	11	11	9	0.9985
10	5	11	11	9	10	0.9996
11	11	11	9	10	6	0.9992
12	11	9	10	6	3	0.982
13	9	10	6	3	11	0.9948
14	10	6	3	11	8	0.9942

Table 1: Example of construction of Neural Network Control Chart.

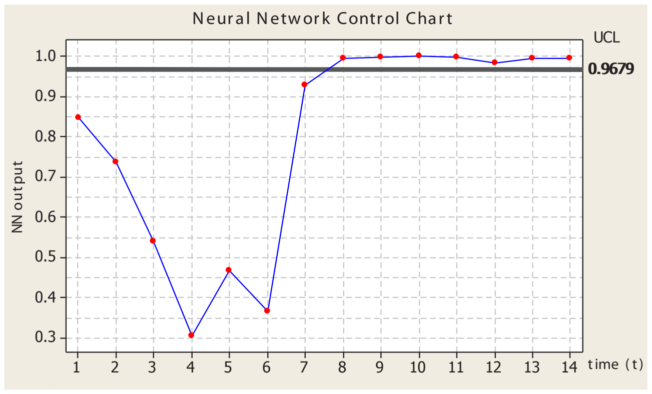

Figure 2: Neural Network Poisson Quality Control Chart

Factor	Low level	High Level
Shift Size (measured in $\hat{y}$)	0	3
Number of inputs	3	5
Number of hidden nodes	3	5
Training Data Size	1,000	10,000
In-control Mean	2	4

Table 2: The experimental factors and their levels.

w_{ji} is the estimated weight value between the jth hidden node and the ith input.

Commonly used activation functions include:

$$g(x) = \frac{1}{1+e^{-x}} \text{ (Logistic or Sigmoid Function)} \tag{2.3.2}$$

$$g(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \text{ (Hyperbolic Function).} \tag{2.3.3}$$

The number of inputs, hidden layers, number of nodes in each hidden layer, and outputs, as well as choice of the activation function, is commonly referred to as the NN topology or architecture [3].

NN Control Chart Construction

To illustrate the construction of a NN Control Chart, suppose a NN has been trained using historical data, after which observations were taken from the process and the time ordered values (3, 8, 7, 0, 9, 3, 4, 3, 4, 5, 11, 11, 9, 10, 6, 3, 11, 8) were obtained. In fact, the first 10 observations were drawn from an in-control Poisson process with mean λ_0 =4, while the last eight observations come from a Poisson process of mean value, λ =7, for illustration purposes. The example considers a NN with 5 inputs (5 most recent data values), one hidden layer and a single output value, $\hat{y}$. Any output exceeding an upper control limit (UCL) of 0.9679 is considered to signal a process shift.

The corresponding x vectors and neural network $\hat{y}$ values are listed in Table 1.

A typical Neural Network Poisson Quality Control Chart for the data in Table 1 is shown in Figure 2. The Neural Network Control Chart (NNCC) signals the shift on the second observation after the shift occurred. Note that the NN input began with the fifth observation in this example. For time period 6, the moving window of span 5 includes observations 2 through 6 as inputs and omits observation 1. Figure 2 contains the time (x-axis) ordered values of the NN outputs (y-axis). A signal occurs at time period 8 as the NN output exceeds the upper control limits. Questions regarding optimal choices for the number of the inputs, the number of the hidden layer nodes, required training data size, determination of the Upper Control Limits (UCL), and others will be answered in the coming sections [4].

The Design

To define optimal topology of the NN suitable for the Poisson data control chart, Design of Experiment (DOE) methodology is used to check for the most important factors that affect the output of the NN. The Full Factorial (2^k) to investigate the effects of the factors is used. The response variable is the ARL, as well as the Median Run Length (MRL). Table 2 and Figure 3 show the experimental factors and their level, while Table 3 contains influential variables held constant within the experiment.

Not included in the project is an investigation of starting weights for the NN. That is, will it help the NN in terms of convergence iteration, time or accurate ARL if we provide initial weights an expert in the field considers the best weights? The comparison between the two ad hoc cases revealed no practical difference as shown in Figures 4 and 5. In both figures, the ARL values (in the Horizontal axis) are plotted against the type of the initial Weights (random weights, R, versus some specific weights, S).

Starting weights are initiated as random values in this experiment. Additionally, the option of having more than 10,000 observations (perhaps 20,000) as a training data set size was investigated. There was no significant difference between training data sizes of 10,000 and 20,000 observations.

The Simulation

The following is the procedure used to carry out the simulation study for the NN for Poisson control chart – the example given here is for in-control λ_0=4 and the code is written using R 2.4.1. (Appendix C).

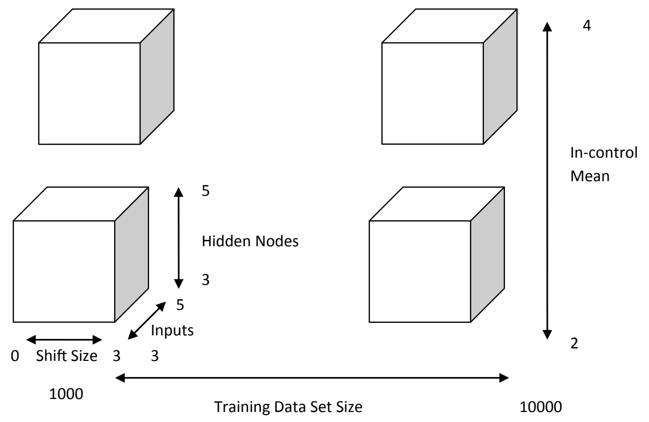

Figure 3: The Design factors and their levels.

Factor	Level
Type of the NN	Feed forward Multilayer Perceptron (MLP)
Type of Connection	Fully connected
Number of hidden layers	One
Number of outputs	One
Activation function	Logistic
Optimization method	Quasi-Newton
Initial Weights	Starts at Random [-0.5,0.5]

Table 3: The noise factors and their fixed level.

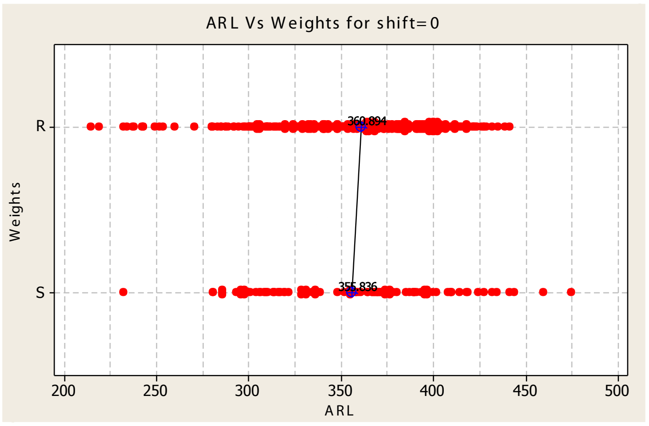

Figure 4: In-control ARL performance of the NNCC for both random weights vs. specific weights for inputs=5, hidden nodes=3 and in-control λ_0=4

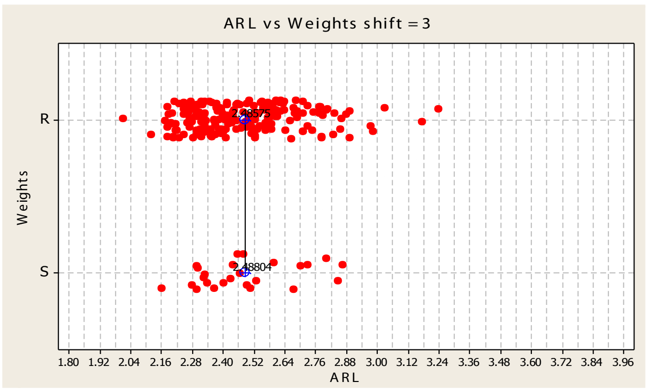

Figure 5: ARL performance of the NNCC for both random weights vs. specific weights for shift δ =3, inputs=5, hidden nodes=3 and in-control λ_0=4.

The simulation consists of five steps as follows:

Generate training data set

The data set contains both in-control ($X \sim Poisson(4)$) and out-of-control data ($X \sim Poisson(7)$) so the neural network will learn to differentiate between the two states. The training data was generated in blocks. For example for the case where the number of inputs=5 and the number of hidden layer nodes=3, a block of data that contains 18 observations is generated using

$$x_i \sim Poisson(\lambda) \quad \text{where}$$

$$\lambda = \begin{cases} 4 & (in-control) & i=1,2,\ldots,9 \\ 7 & (out-of-control) & i=10,11,\ldots,18 \end{cases}.$$

The value λ =4 represents an in-control process, while λ =7 indicates a process shift of size (1.5 σ).

The outputs are:

$$y_i = \begin{cases} 0 & for\, x_i \sim Poisson(4) \\ 1 & for\, x_i \sim Poisson(7) \end{cases}.$$

The inputs and corresponding responses from this block of data will be as follows:

$$
x_i^* = \begin{pmatrix} x_1, x_2, x_3, x_4, x_5 \\ x_2, x_3, x_4, x_5, x_6 \\ x_3, x_4, x_5, x_6, x_7 \\ x_4, x_5, x_6, x_7, x_8 \\ x_5, x_6, x_7, x_8, x_9 \\ x_6, x_7, x_8, x_9, x_{10} \\ x_7, x_8, x_9, x_{10}, x_{11} \\ x_8, x_9, x_{10}, x_{11}, x_{12} \\ x_9, x_{10}, x_{11}, x_{12}, x_{13} \\ x_{10}, x_{11}, x_{12}, x_{13}, x_{14} \\ x_{11}, x_{12}, x_{13}, x_{14}, x_{15} \\ x_{12}, x_{13}, x_{14}, x_{15}, x_{16} \\ x_{13}, x_{14}, x_{15}, x_{16}, x_{17} \\ x_{14}, x_{15}, x_{16}, x_{17}, x_{18} \end{pmatrix} \qquad y_i^* = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}
$$

In total we will have five pure in-control input states, five pure out-of-control input states, and four in-transition out-of-control states, where the status of the state is determined by the most recent data point in the inputs. This block is repeated to create a training data set of a specified size. For inputs n=3 the block will have eight data points, and for inputs n=4 the block will have 12 data points that are evenly distributed between the in- and out-of-control states in the same manner as described above.

Train the NN

Use the subroutine "nnet" in R2.4.1 to get the optimal weights and check for convergence.

Determine upper control limits (UCL) for NN outputs

Generate pure in-control data, that is, obtain the data from the Poisson distribution with the in-control mean (λ_0 =4). Then, apply the trained neural network to this data and get the output in the closed interval [0, 1], where all the values should be close to 0 for in-control data. An appropriate percentile of the in-control output will be used to get an upper control limits (UCL) point, which is used to determine the state of the process in the next step. In-control samples of 100,000 for each Neural Network are used to get accurate UCL values, because most of the percentile points are of 4 significant digits.

Determine ARL for current NN

The Average Run length (ARL) for the neural network is determined using only out-of-control samples. The first set of inputs is from $x_1^* = (x_1, x_2, x_3, x_4, x_5)$ where $x_1, \ldots, x_4$ are randomly generated in-control points and x_5 is out-of-control data point. The trained network is applied to these inputs to obtain $\hat{y}_i$.

Let UCL_{NN} denote the UCL point for the NN signals. If $\hat{y}_i \geq UCL_{NN}$ then a process signal occurs.

If $\hat{y}_1 < UCL_{NN}$, another random sample, $x_2^* = (x_2, x_3, x_4, x_5, x_6)$ where x_6 is an out-of-control data point, is generated as well as its corresponding output $\hat{y}_2$. The process continues until $\hat{y}_i \geq UCL_{NN}$, and then the process is declared as "out-of-control" and the Run-length Index (m_k) is recorded. Then the process is repeated "nRLout" times to produce "nRLout" run length. The ARL= $\sum_{k=1}^{nRLout} \frac{m_k}{k}$ as well as other desired statistics such as Run Length standard deviations or medians are calculated.

Repeat the steps

Steps 1-4 will be repeated several times to produce different realizations of the process and accurate estimations of the ARLs for multiple NNs, Figure 6 displays the steps of the simulation.

Determining the Upper Control Limits (UCL)

For proper comparisons with the other methods examined in this study, equal in-control ARL values for all of competing methods must be considered. To determine nominal 370 in-control ARL for the neural network, the correct value for the UCL_{NN} point must be obtained. This step is required of all experiment combinations of the levels of the inputs and hidden nodes. An attempt to theoretically obtain the UCL point was pursued. Due to the mathematical complexities associated

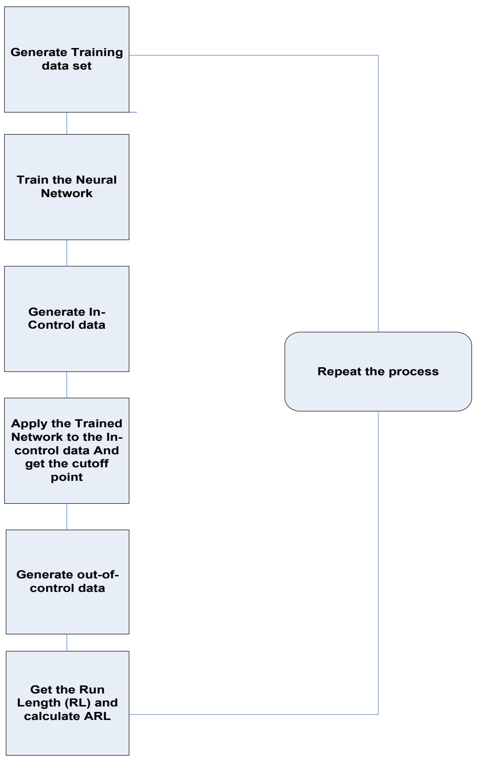
Generate Training data set
Train the Neural Network
Generate In-Control data
Apply the Trained Network to the In-control data And get the cutoff point
Generate out-of-control data
Get the Run Length (RL) and calculate ARL
Repeat the process

Figure 6: Steps of the simulation.

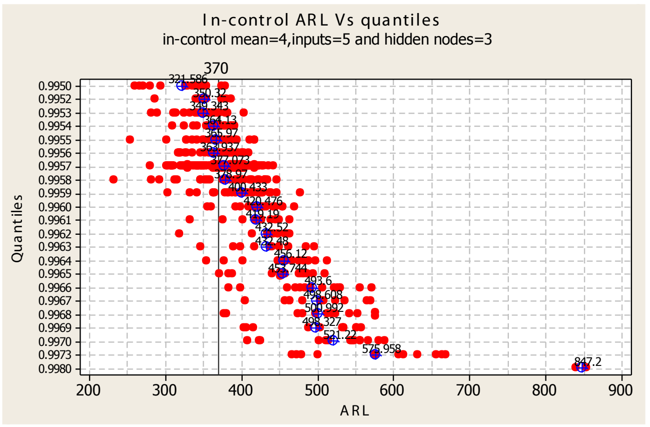

Figure 7: Values of the In-control ARL for different percentile UCLs when inputs=5, hidden layer nodes =3 and in-control λ=4.

with NN models, the solution was intractable. A brief outline of this work is provided in Appendix A.

An intensive search in the percentile range of [.95-.98] was conducted to empirically obtain the right UCL value to produce the nominal in-control ARL of 370. Figure 7 illustrates the searching method for inputs value of 5 and hidden nodes value of 3, where the ARL values are plotted against the percentiles (in ascending order). For example, the smallest in-control ARL for the 99.55th percentile was approximately 280.

The 99.57th percentile is used as the UCL_{NN} for this setting so the in-control ARL=377.07. The following Figure 8 provides a better view of the graph.

Results

Once the UCLs are determined, the experiment is conducted with 50 replicates at each of the experimental units. MINTAB 15 software is used to analyze the data. Table 4 reveals significant main effects for these factors: Inputs (n), hidden layer nodes (h), training data size (x 1,000), and in-control lambda (c). Further, there is a possible high-order interaction especially between inputs (n) and hidden layer nodes (h). Looking to both Table B1 and B2 in Appendix B, all of the combinations of the inputs and hidden nodes factors are giving in-control ARL values close to the nominal 370 ARL. In the high level setting of both factors (that is, five inputs and five hidden layer nodes) with a low training data size of 1,000, the in-control ARLs were in the 264-268 period. With in-control lambda λ_0 = 2 and four nodes in both input and hidden layers, small values of in-control ARL around 288.6348 to 291.373 were obtained for all training data set sizes.

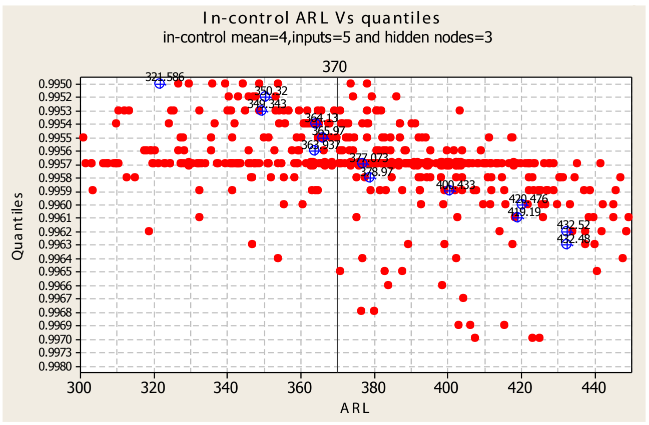

Figure 8: Values of the In-control ARL [300-450] for different percentile upper control limits (UCL)s when inputs=5, hidden layer nodes =3 and in-control λ=4.

Factorial Fit: ARL versus n, h, Shift, training, in-control lambda
Analysis of Variance for Mean (ARL) (coded units)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	5	65260593	65155247	13031049	1067.12	0.000
2-Way Interactions	10	1899501	1939431	193943	15.88	0.000
3-Way Interactions	10	1218185	1215520	121552	9.95	0.000
4-Way Interactions	5	941449	934485	186897	15.31	0.000
5-Way Interactions	1	108601	108574	108574	8.89	0.003
Curvature	1	435	435	435	0.04	0.850
Residual Error	10655	130113274	130113274	12211		
Lack of Fit	177	63004894	63004894	355960	55.58	0.000
Pure Error	10478	67108380	67108380	6405		
Total	10687	199542039				

Table 4: The ANOVA table.

Figures 9-11 display the ARL performance of the NNCC for an in-control λ=2 across all training data sizes (1,000; 5,500 and 10,000) with shifts δ (0.5-3) standard deviation units from the in-control mean for different inputs and hidden layer nodes, compared to the theoretical value of the C-chart (called Poisson in the charts) . For the detailed performance values, see Tables B3-B8 in Appendix B. In Figure 11, the horizontal axis presents shift size with shifts ranging in magnitude from 0.5σ to 3σ. The vertical scale provides the out-of-control ARL. For example, the NNCC (3; 5; 10,000) gives an ARL of approximately 45 for shift δ =0.5. Figure 11 reveals that ARL performance of NNCC (3; 5; 10,000) and NNCC (3; 3; 10,000) are similar.

Notice that the NNCC is performing better than the C-chart for the large training data size (a minimum of 5,500), but poorly for the small training data set (1,000) across all inputs, hidden nodes combinations, and shift sizes. For a training data size of 1,000, the C-chart is outperforming the best NNCC for all shift sizes. For that matter, the best NNCC was the (3, 3) network, and the ARL for shift δ =0.5 is 52.5235, while the C-chart ARL is 32.93. For shift size δ =3, the ARLs are 2.555 and 1.2 for the NNCC (3; 3; 1,000) and C-chart, respectively. The remaining NNCCs are not consistent regarding which one is the best for any specific shift; for example, NNCC(5; 5; 1,000) was the worst for shift δ =0.5, the best for shift δ =1, and the second best for shift δ =3 among the other NNCCs.

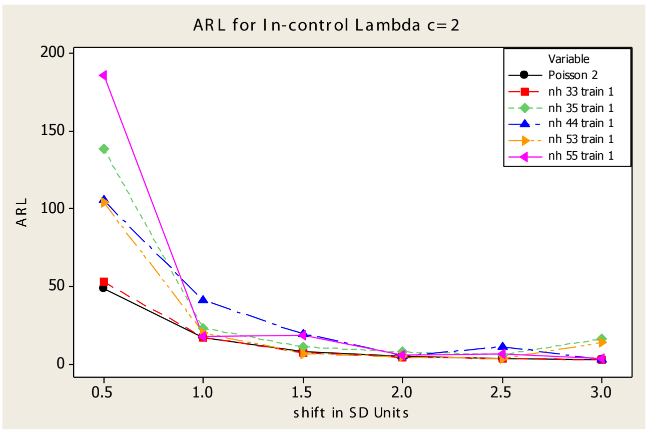

Figure 9: The ARL performance for In-control lambda=2 with training data size of 1,000.

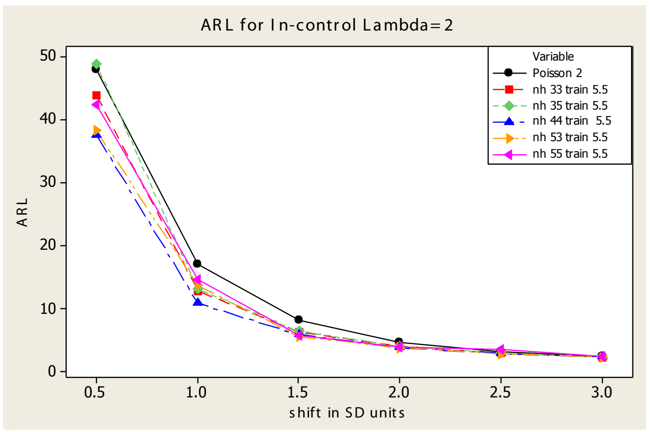

Figure 10: The ARL performance for In-control lambda=2 with training data size of 5,500.

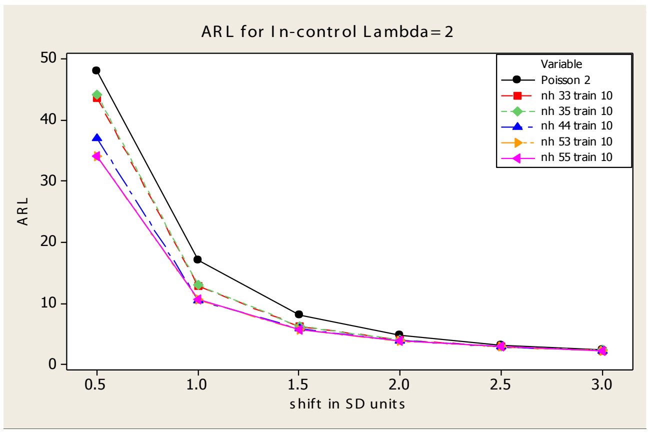

Figure 11: The ARL performance for In-control lambda=2 with training data size of 10,000.

The NNCC exhibits an interesting phenomenon: several of the 50 trained NNCCs are outliers in the sense that NN outputs and ARLs were significantly different from the majority of NNs. Their values affect the overall performance of the NNCC for the given setting. Small training data sets are more susceptible to the problem. This phenomenon is discussed in more detail in Section 2.10.

For a training data size of 5,500, almost all of the NNCC charts outperform the C-chart where both NNCC (4; 4; 5,500) and NNCC (5; 3; 5,500) are the best for small shifts ($\delta < 2$), although NNCC (4; 4; 5,500) has ARL 10.832 and NNCC (5; 3; 5,500) ARL was 13.6778 for shift $\delta=0.5$. However, that is expected since NNCC (4; 4; 5,500) has a smaller in-control ARL 288.6348, compared to 338.504 for NNCC (5; 3; 5,500). For shifts (δ) greater than or equal to 2, all methods are fairly comparable.

For a training data size of 10,000, all NNCC charts outperform the C-chart for shifts $\delta <=2.5$ in standard deviations (SD) units, but perform at the same level as the C-chart once the shift exceeds 2.5 σ. Among the NNCC charts, the NNCC (5; 3; 10,000) and NNCC (5; 5; 10,000) provide optimal ARL performance, especially in the small shift range.

Figures 12-14 show the performance of the NNCC for in-control $\lambda=4$; the ARL details can be found in Tables B9-B14 in Appendix B. With a small training size of 1,000, there is no NNCC better than the C-chart (also called Poisson 4 in the graphs). The best NNCC – especially when the shift size is $\delta<2.5$ – is the NNCC (5; 3; 1,000). The ARL for the C-chart is 73.02, and for NNCC (5; 3; 1,000) the ARL is 82.723 where the biggest ARL is more than the sum of both ARLs, which is 186.254 for NNCC (4; 4; 1,000) for shift $\delta=0.5$. Again, we see the instability of the NNCC when a small size training data set is used, while it is effective for big shift sizes.

Once the training data set size is increased to 5,500, the ARL performances change. The NNCC outperforms the C-chart in almost all shift sizes – except for NNCC (3; 5; 5,500) when the shift $\delta=2.5$; the NNCC ARL is 7.991 and the C-chart ARL is 3.4, which will be explained in later sections. The NNCC (5; 3; 5,500) seems to perform better than the rest of the NNCC, except in the case of the shift size $\delta=0.5$ and 1. The NNCC(5; 3; 5,500) with ARL value of 10.4980 is the second best following NNCC(5; 5; 5,500) with 10.1308 ARL for shift $\delta=1$. In the case of shift size $\delta=0.5$, the best NNCC is NNCC(4; 4; 5,500) with 41.9042 ARL; NNCC(5; 3; 5,500) had 48.6948 ARL, but after removing an outlier value of 656.2, the ARL value dropped to 36.2967, which is the smallest for that shift size.

In the training data set size of 10,000 NNCC outperforms the C-chart across the board. Both NNCC(5; 3; 10,000) and NNCC(5; 5; 10,000) share the close values of the ARL that make them the top

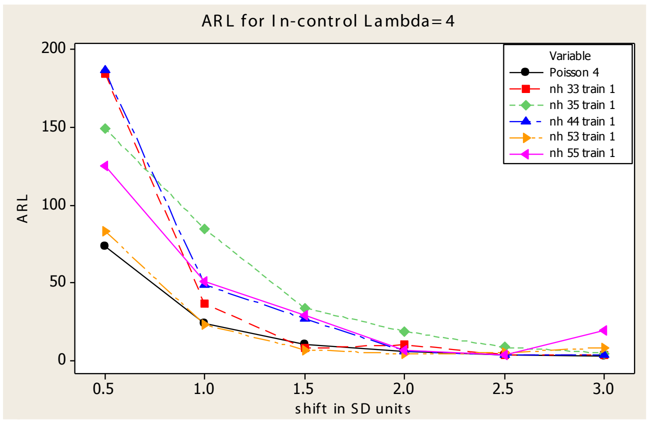

Figure 12: The ARL performance for In-control lambda=4 with training data size of 1,000.

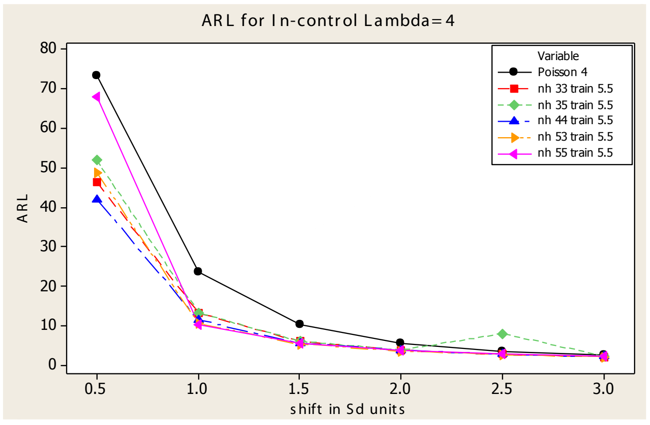

Figure 13: The ARL performance for In-control lambda=4 with training data size of 5,500.

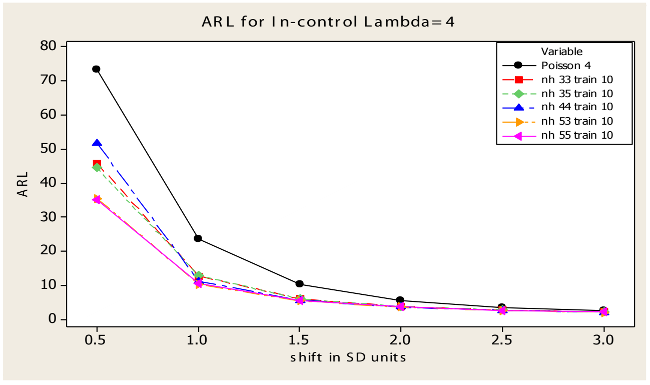

Figure 14: The ARL performance for In-control lambda=4 with training data size of 10,000.

choices, but NNCC(5; 3; 10,000) has a closer in-control ARL to the nominal value of 370, which 366.774 versus 356.666 in-control ARL for NNCC(5; 5; 10,000). It performs better across all shifts except for the slight difference between 35.421 and 34.919 (ARL of the NNCC(5; 3; 1,000) and NNCC(5; 5; 1,000), respectively) for shift δ=0.5.

From the previous analysis, the NNCC with inputs 5 and hidden layer nodes 3 seems a natural choice for the best candidate of the different NNCCs to be used in monitoring non-conformities in the Poisson processes. It always has the smallest ARL or is in-line with one that has the smallest shift for all type of shifts and training data set sizes.

Determination of the Critical Size for Training Data Sets

The next question that must be answered is: what is the best training set size that will perform at least as well as the C-chart?

For in-control λ=2, Figure 15 displays the out-of-control ARL performance for NN with 5 inputs and 3 hidden layer nodes. On the x-axis, the shift magnitude (0.5σ -3σ) is graphed against the ARL (y-axis) performance. For example, the NNCC (5; 3; 5,500) has ARL value of about 40 for shift δ =0.5. Figure 15 shows that the training data set size of 1,000 is not performing as well. It is always has a larger out-of-control ARL than the C-chart. It shows that the NNCC with inputs = 5, hidden nodes = 3, and training set data size of 5,500 will always outperform the C-chart in terms of the ARL values across all shifts δ=0.5-3 standard deviation units from the nominal mean, as seen in Table 5.

It is also true for the performance in the Median Run Length as shown in Figure 16.

For the in-control case with λ=4, the NNCC (5; 3; 1,000) also exhibited poor performance and instability when compared to the C-chart. However, the NNCC with 5 inputs, 3 hidden layer nodes, and

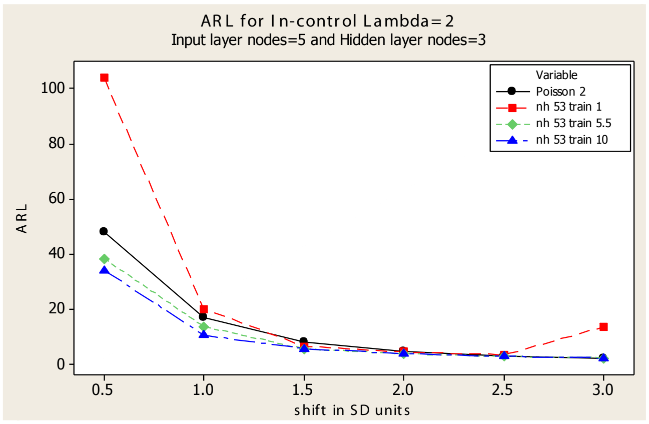

Figure 15: The ARL performance for In-control lambda=2 with inputs=5 and hidden nodes=3.

Shift δ	C-chart	NNCC(5; 3; 5,500)
0.5	48.01	38.31
1	16.98	13.68
1.5	8.08	5.57
2	4.69	3.77
2.5	3.13	2.84

Table 5: ARL performance for both C-chart and Neural Control Chart with inputs n=5, hidden nodes=3, and training data set size=5,500 for in-control lambda=2.

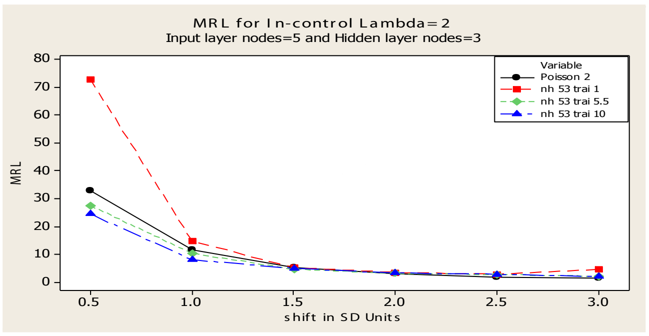

Figure 16: The MRL performance for In-control lambda=2 with inputs=5 and hidden nodes=3.

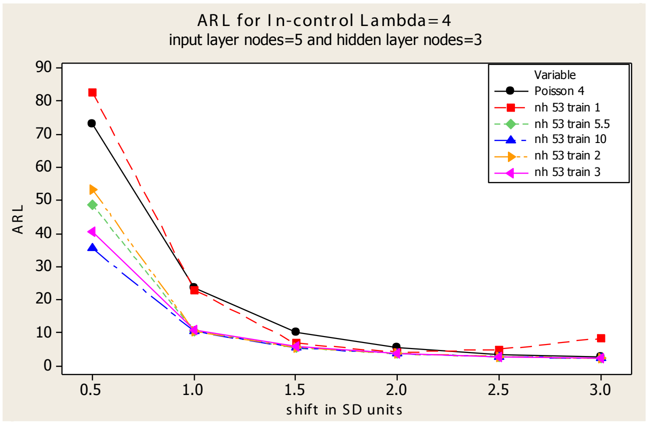

Figure 17: The ARL performance for In-control lambda=4 with inputs=5 and hidden nodes=3.`

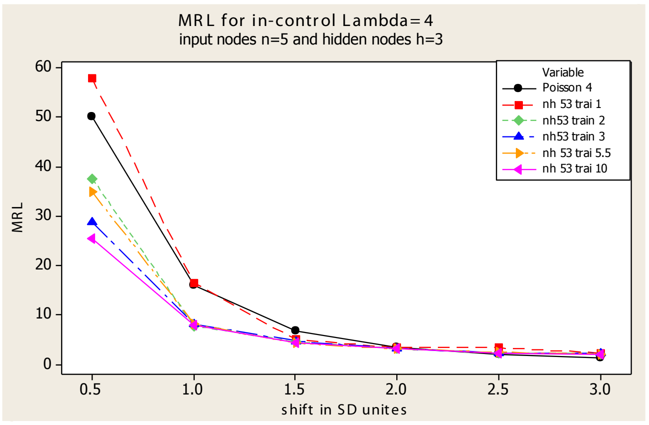

Figure 18: The MRL performance for In-control lambda=4 with inputs=5 and hidden nodes=3.

training data set size 2,000 outperforms the C-chart in monitoring non-conformities in the Poisson process in both ARL and MRL, as we can conclude from Figures 17 and 18 and Table 6.

After deciding that the NNCC with inputs = 5 and hidden layer nodes = 3 is the most suitable for use in monitoring the number of the non-conformities in the Poisson processes, we investigated its performance in terms of ARL and MRL for the in-control case of λ=6. The results are shown in Figures 19-20 and Table B15 in Appendix B.

Shift δ	C-chart	NNCC(5; 3; 2,000)
0.5	73.02	53.25
1	23.46	10.25
1.5	10.15	5.51
2	5.43	3.58
2.5	3.4	2.63

Table 6: ARL performance for both C-chart and Neural Control Chart with inputs n=5, hidden nodes=3, and training data set size=2,000 for in-control lambda=4.

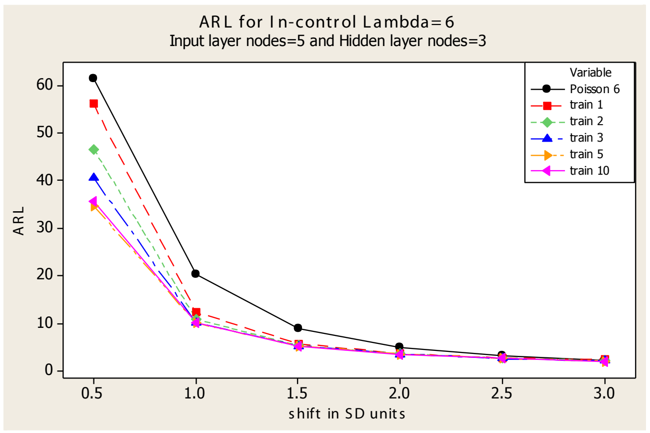

Figure 19: The ARL performance for In-control lambda=6 with inputs=5 and hidden nodes=3.

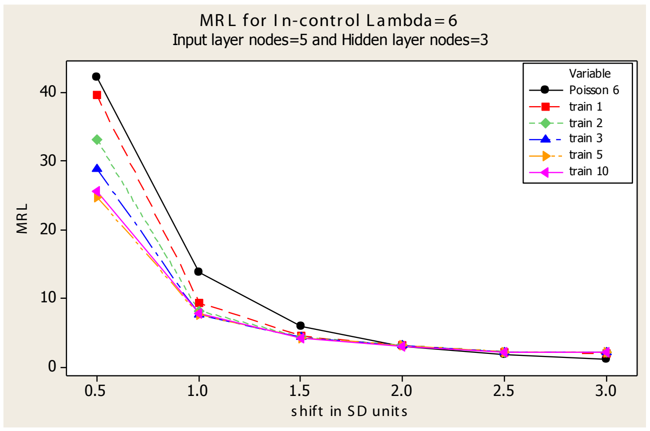

Figure 20: The MRL performance for In-control lambda=6 with inputs=5 and hidden nodes=3.

When comparing the C-chart (Poisson 6 in the graphs) to the NNCCs, we see that for small shifts, that is $\delta \leq 1.5$, all NNCCs outperform the C-chart in terms of the both ARL and MRL. For the remaining shift values, the C-chart is performing better than all of the NNCCs in terms of MRL; meanwhile the NNCC continues to perform better than the C-chart for all types of shifts, except shift δ=3 and training data size=1,000.

As we can see, the training data set size is not crucial in this case, because even with a small training size of 1,000, the previously noticed phenomenon is still valid. The NNCC (5; 3; 1,000) is outperforming the C-chart for in-control λ= 6 in almost all shifts δ<3, for example, for shift δ= 0.5, the NNCC (5; 3; 1,000) has an ARL value of 56.14, but the C-chart ARL is 61.275. For shift δ=3 the C-chart is outperforming the NNCC (5; 3; 1,000) since the ARLs are 2.2778 and 2.14958 for NNCC (5; 3; 1,000) and C-chart, respectively (Table 7).

Alternatively, the picture is somewhat different in the MRL performance. For shift δ=0.5, the NNCC (5; 3; 1,000) has 39.6 MRL and the C-chart has 42.125, but for shifts δ>1.5, the C-chart outperforms the NNCC; for example, when shift δ=2 the values of the MRL were 3.04 and 2.9494 for NNCC (5; 3; 1,000) and C-chart, respectively. It is worth noting that the NNCC (5; 3; 1,000) has a closer in-control ARL value than the C-chart to the nominal 370, as the in-control values were 306.1214 and 275.6 for NNCC (5; 3; 1,000) and C-chart, respectively.

Choosing the NNCC with 5 inputs and 3 hidden layer nodes and a training data set size of 1,000 is a justifiable choice for monitoring the number of the non-conformities in the Poisson processes.

Unusual Outliers

As mentioned earlier, the neural network performance suffers from the instability of the outputs. That is, there will be few outlier networks, which mean that among all the 50 networks that were trained, few networks were noticeably different from the rest of the networks. The question must be considered: do the different networks just have some outlier Run Length(s) that are driving the ARL to be large, or is it the case that the whole Run Lengths for those networks were relatively large?

Figure 21 is a dotplot for the 50 individual in-control ARLs where the ARL of each NNCC is plotted on the vertical scale. As can be seen in Figure 21, for NNCC (5; 3; 1,000), shift δ=1.5 and in-control λ_0=2 among the 50 networks that were trained, it seems there is at least one network with a significantly large ARL reported relative to the remaining networks. Some of the 50 networks ARLs – including three

Shift δ	C-chart	NNCC(5; 3; 1,000)
0.5	61.27	56.14
1	20.23	12.22
1.5	8.85	5.69
2	4.77	3.54
2.5	3.02	2.63
3	2.15	2.28

Table 7: ARL performance for both C-chart and Neural Network Control Chart with inputs n=5, hidden nodes=3 and training data set size=1,000 for in-control lambda=6.

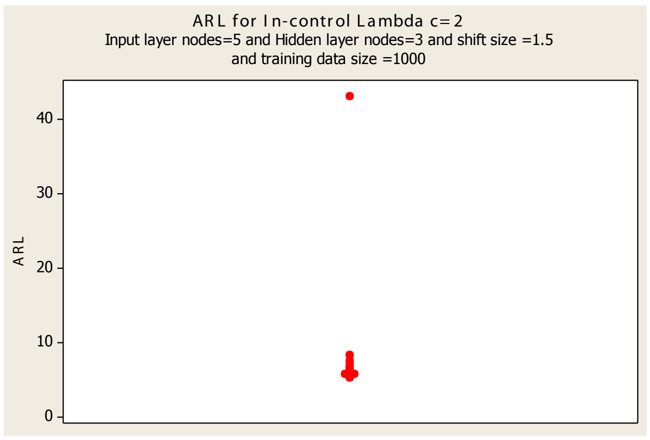

Figure 21: The ARL performance for In-control lambda=2 with inputs=5, hidden nodes=3, shift size δ =1.5, and training data size=1,000.

suspect outliers – were investigated through the mean of the Boxplot. Figure 22 gives the ARL boxplot of seven NNs. Each of the boxplots represents the Run Length distribution for a specific NN. Noticeably, the second network is in fact different than the majority of the networks in both median (and mean) and the spread of the data (range). This network has an ARL value of 42.91 and MRL value of 31; we can get values as extreme as 377 run length for that specific shift δ =1.5. Also, networks 3 and 4 differ from the rest of the networks and have bigger ARLs – which are 8.342 and 8.217 for network 3 and network 4, respectively. The ARL for all 50 networks was 6.53824, and after removing these three outlier networks it dropped to 5.69028 with SE (Standard Error) = 0.066, meaning it is statistically smaller.

The phenomenon is more noticeable with, but not limited to, the small size of the training data sets. For example, as we can see in Figure 23 where in-control λ=4 with inputs = 3, hidden nodes = 5, big shift size δ=2.5, and a moderate training data size = 5,500, we still found an outlier network that suffers from the instability and causes the overall ARL to increase dramatically. The ARL was 7.99064 when that single outlier network was included, but decreased to 2.69453 with SE = 0.0116 when that network is removed from the calculations.

Thus, caution must be exercised when using the Neural Network Control Chart for Poisson process. Getting a variety of trained neural networks will help prevent this instability and getting unrealistic values for the neural network parameters (the weights).

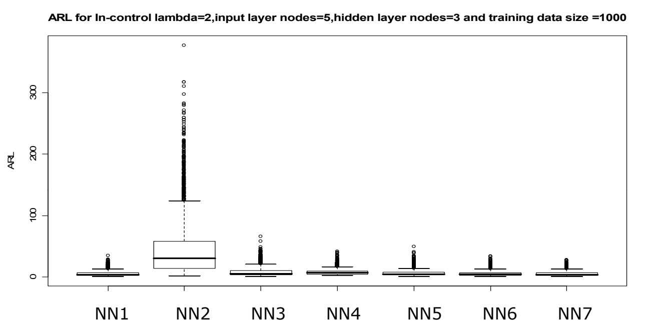

Figure 22: The Boxplot of some of the ARL performance for In-control lambda=2 with inputs=5, hidden nodes=3, shift size δ =1.5 and training data size=1,000.

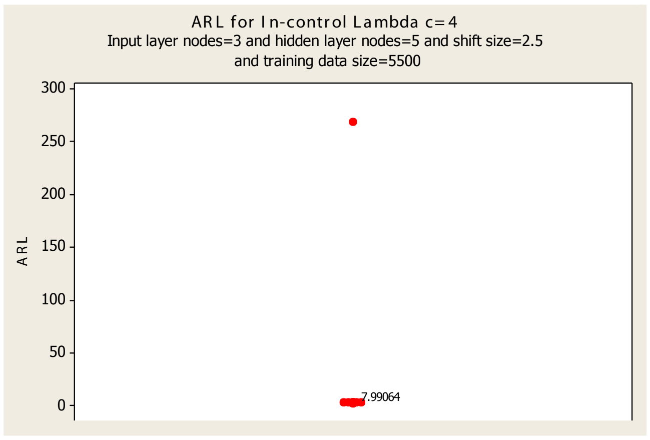

Figure 23: The ARL performance for In-control lambda=4 with inputs=3, hidden nodes=5, shift size δ =2.5 and training data size=5,500.

Shift	In-control Mean=2	In-control Mean=4	In-control Mean=6
0.5	NNCC(5,5,10000)=34.03 NNCC(5,3,10000)=34.03	NNCC(5,5,10000)=34.92 NNCC(5,3,10000)=35.41	NNCC(5,3,4000)=34.61
1	NNCC(4,4,10000)=10.52 NNCC(5,3,10000)=10.54	NNCC(5,5,5500)=10.13 NNCC(5,3,2000)=10.25	NNCC(5,3,3000)=10.00
1.5	NNCC(5,3,5500)=5.57	NNCC(5,3,5500)=5.30	NNCC(5,3,5000)=5.17
2	NNCC(5,3,5500)=3.77	NNCC(5,3,5500)=3.57	NNCC(5,3,5000)=3.44
2.5	NNCC(5,3,5500)=2.84	NNCC(5,3,10000)=2.61	NNCC(5,3,3000)=2.52
3	NNCC(5,3,10000)=2.25	NNCC(5,3,10000)=2.06	NNCC(5,3,5000)=1.97

Table 8: The Best Control Charts in ARL.

Shift	In-control Mean=2	In-control Mean=4	In-control Mean=6
0.5	NNCC(5,5,10000)=24.42 NNCC(5,3,10000)=24.41	NNCC(5,5,10000)=24.99 NNCC(5,3,10000)=25.46	NNCC(5,3,5000)=24.76
1	NNCC(5,3,10000)=7.96	NNCC(5,3,2000)=7.74	NNCC(5,3,5000)=7.56
1.5	NNCC(4,4,10000)=4.18	NNCC(4,4,10000)=4.02	NNCC(5,3,10000)=4.08
2	C-chart=2.89	NNCC(3,3,5500)= NNCC(3,3,10000)=3 NNCC(3,5,10000)= NNCC(4,4,10000)=3	C-chart=2.95
2.5	C-chart=1.8	C-chart=1.99	C-chart=1.72
3	C-chart=1.22	C-chart=1.28	C-chart=1.11

Table 9: The Best Control Charts in MRL.

Conclusion

Using the Design of Experiment (DOE) methodology, we found that number of the inputs, hidden layer nodes, training data size and the value of the in-control mean are all significant factors in having the optimal Neural Network Control Chart for the number of non-conformities in Poisson processes.

The NNCC with inputs "n" = 5 and hidden layer nodes "h" = 3 seems to perform well across all in-control lambda values and outperforms the C-chart especially for small and moderate shifts. The smaller the in-control lambda, the larger the training data needed – training = 5,500; 2,000 and 1,000 were sufficient enough for the NNCC (5, 3) to outperform the C-chart for in-control λ=2, 4 and 6, respectively.

Table 8 and 9 shows the best control chart preferred for use in terms of the ARL and MRL for a specific combination of in-control λ and shift size that we seek to detect. For example, if we wish to detect a shift δ =2 and the in-control λ is 4, then the NNCC(5; 3; 5,500) – which means the NNCC with 5 inputs, 3 hidden nodes and 5,500 training data size – has the smallest ARL (3.573) among all the NNCC and C-charts.

We observe that the NNCC (5, 3) is almost always either the optimal or the nearest optimal NNCC for detecting any shift for any given λ values. There is always a NNCC that outperforms the C-chart in ARL for all types of shifts and the in-control mean values. The C-chart outperforms the NNCC for large shifts (δ ≥ 2) in terms of MRL, except for shift δ=2 and λ=4.

References

1. Montgomery DC (2005) Introduction to Statistical Quality Control. New York: John Wiley & Sons.
2. Woodall WH (1997) Control Charting Based on Attribute Data: Bibliography and Review. Journal of Quality Technology 29: 172-183.
3. Saithanu K (2006) Neural Networks and Multivariate Quality Control. The University of Alabama, Information Systems, Statistics and Management Science.
4. Zorriassatine F, Tannock JDT (1998) A Review of Neural Networks for Statistical Process Control. J Intell Manuf 9: 209-224.