Hellenic Complex Systems Laboratory

Intelligent Quality Control: A Software Tool for Exploring the Application of Convolutional Neural Networks to Quality Control Samples of Very Small Size

Technical Report XXIV

Rallou A. Chatzimichail and Aristides T. Hatjimihail
2022

Intelligent Quality Control: A Software Tool for Exploring the Application of Convolutional Neural Networks to Quality Control Samples of Very Small Size

Rallou A. Chatzimichail, PhD [a], Aristides T. Hatjimihail, MD, PhD [a]

[a] Hellenic Complex Systems Laboratory

Search Terms: quality control, convolutional neural networks, quality control function, probability for rejection, probability for false rejection, quality control measurement, critical error

Introduction

Although there is extensive literature on the application of artificial neural networks (NN) in quality control (QC) to monitor the conformity of a process to quality specifications, at least five QC measurements are required, increasing the related cost. To explore the application of NN to samples of QC measurements of very small size, four one-dimensional (1-D) convolutional NN (CNN) $R_N(\boldsymbol{x}; 0,1, n, a)$ (see Notation), for $a \in \{1,2,3,4,6,8,12,16\}$, were designed, trained, and tested with datasets of n-tuples of simulated standardized normally distributed QC measurements, for $1 \leq n \leq 4$.

The designed CNN are compared to statistical QC functions $R_S(\boldsymbol{x}; 0,1, n, k, l\,)$, for $1 \leq k \leq n$, and $R_M(\boldsymbol{x}; 0,1, n, l\,)$ (see Notation), applied on tuples $\boldsymbol{x}$ of control measurements of the same size n, distributed as $\mathcal{N}(m, s^2)$, with equal probabilities for false rejection (*pfr*).

The program *Intelligent Quality Control* was developed to explore the application of the trained CNN to QC samples of very small size and compare them to alternative QC functions.

The program

The program provides two modules:

QC Plots

This module plots the probabilities for rejection of the application of NN QC functions $R_N(\boldsymbol{x}; 0,1, n, a)$ and statistical QC functions $R_S(\boldsymbol{x}; 0,1, n, k, l\,)$, for $1 \leq k \leq n$, and $R_M(\boldsymbol{x}; 0,1, n, l\,)$ (see Figures 1-2).

QC Calculator

This module calculates the probabilities for rejection of the application of NN $R_N(\boldsymbol{x}; 0,1, n, a)$ and statistical QC functions $R_S(\boldsymbol{x}; 0,1, n, k, l\,)$, for $1 \leq k \leq n$, and $R_M(\boldsymbol{x}; 0,1, n, l\,)$, for user defined critical mean and standard deviation of the measurement process, and their probabilities for false rejection (see Figures 3-4).
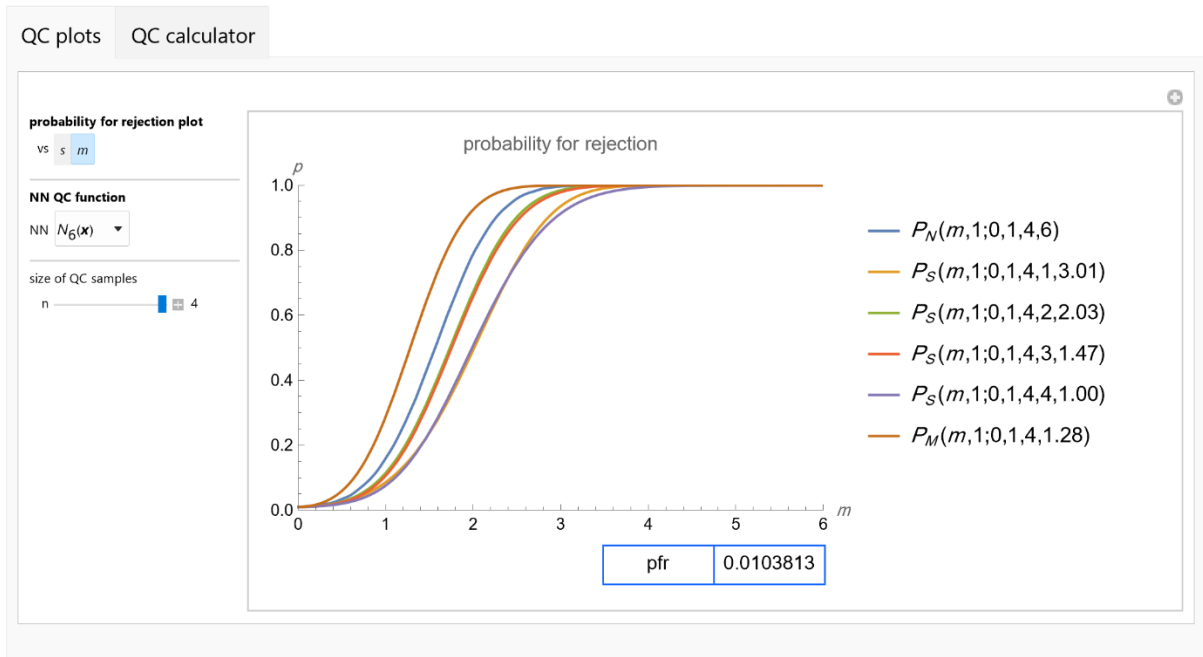
QC plots   QC calculator

probability for rejection plot

vs   s   m

NN QC function

NN $N_6(\boldsymbol{x})$ ▼

size of QC samples

n ————————— ⊞ 4

probability for rejection

p

1.0

0.8

0.6

0.4

0.2

0.0

0   1   2   3   4   5   6 m

— $P_N(m,1;0,1,4,6)$
— $P_S(m,1;0,1,4,1,3.01)$
— $P_S(m,1;0,1,4,2,2.03)$
— $P_S(m,1;0,1,4,3,1.47)$
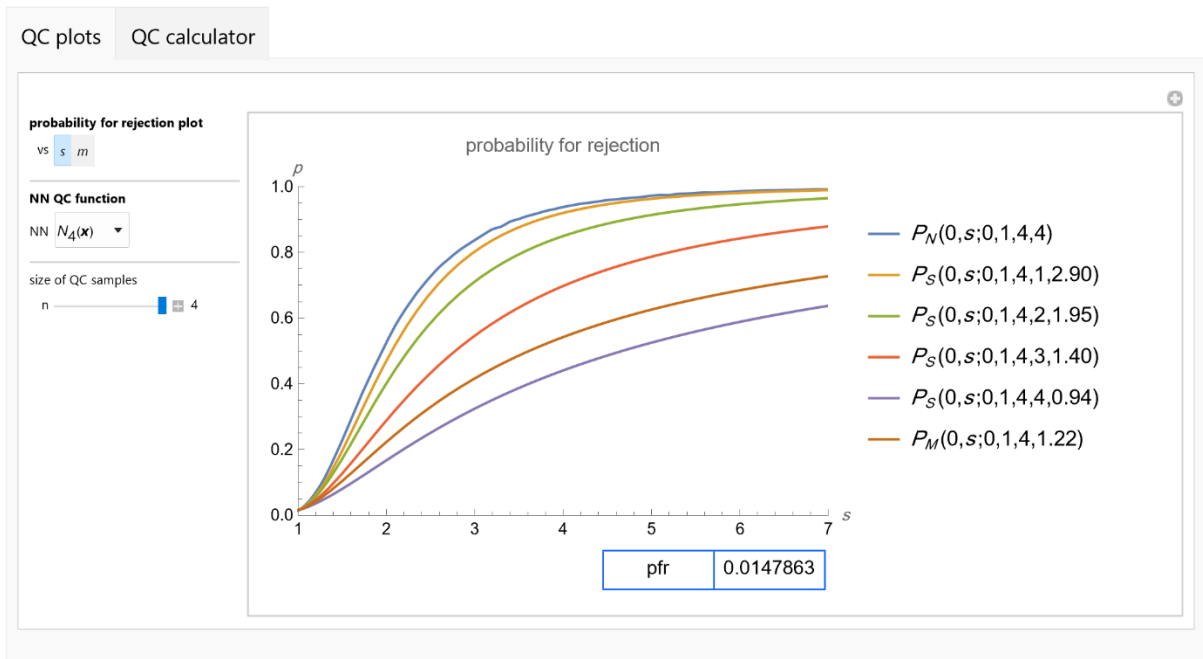— $P_S(m,1;0,1,4,4,1.00)$
— $P_M(m,1;0,1,4,1.28)$

pfr   0.0103813

Figure 1: The plots of the probabilities for rejection of the NN QC function $R_N(\boldsymbol{x}; 0,1, n, a)$ and statistical QC functions $R_S(\boldsymbol{x}; 0,1, n, k, l\,)$, for $1 \leq k \leq n$, and $R_M(\boldsymbol{x}; 0,1, n, l\,)$, with the settings shown at the left.

QC plots   QC calculator

probability for rejection plot

vs   s   m

NN QC function

NN $N_4(\boldsymbol{x})$ ▼

size of QC samples

n ————————— ⊞ 4

probability for rejection

p

1.0

0.8

0.6

0.4

0.2

0.0

1   2   3   4   5   6   7 s

— $P_N(0,s;0,1,4,4)$
— $P_S(0,s;0,1,4,1,2.90)$
— $P_S(0,s;0,1,4,2,1.95)$
— $P_S(0,s;0,1,4,3,1.40)$
— $P_S(0,s;0,1,4,4,0.94)$
— $P_M(0,s;0,1,4,1.22)$

pfr   0.0147863

Figure 2: The plots of the probabilities for rejection of the NN QC function $R_N(\boldsymbol{x}; 0,1, n, a)$ and statistical QC functions $R_S(\boldsymbol{x}; 0,1, n, k, l\,)$, for $1 \leq k \leq n$, and $R_M(\boldsymbol{x}; 0,1, n, l\,)$, with the settings shown at the left.

size of QC sample

control measurements ———————————————————— 4

NN QC function

NN $N_8(x)$ ▾

standardized critical errors

systematic error ———————————— 3.

random error ———————————— 4.

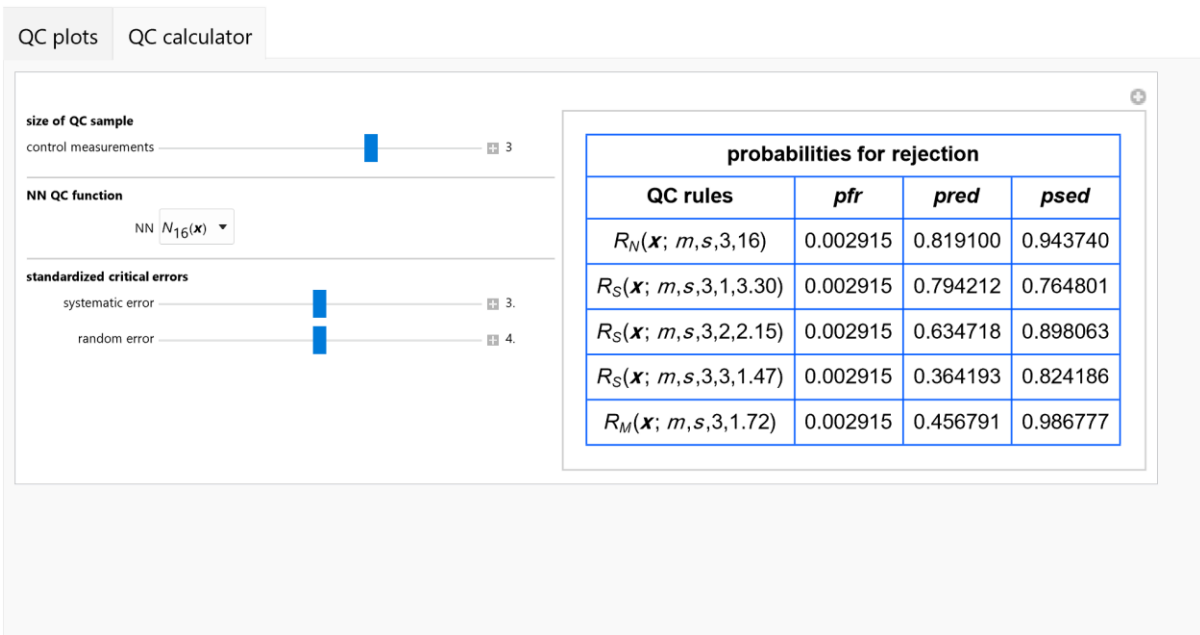| probabilities for rejection			
QC rules	*pfr*	*pred*	*psed*
$R_N(x; m,s,4,8)$	0.005599	0.915460	0.995240
$R_S(x; m,s,4,1,3.19)$	0.005599	0.890356	0.889193
$R_S(x; m,s,4,2,2.15)$	0.005599	0.809246	0.973206
$R_S(x; m,s,4,3,1.57)$	0.005599	0.640726	0.967926
$R_S(x; m,s,4,4,1.09)$	0.005599	0.378357	0.891276
$R_M(x; m,s,4,1.39)$	0.005599	0.488561	0.999380
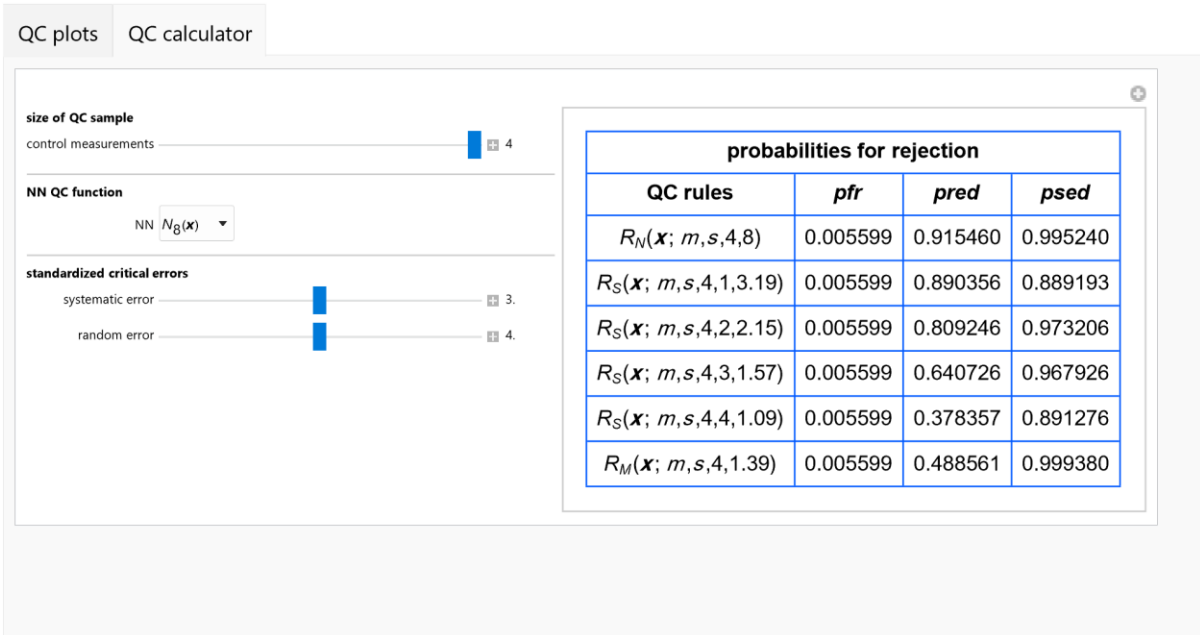
Figure 3: Table presenting the probabilities for rejection of the critical mean and standard deviation of the NN QC function $R_N(x; 0,1,n,a)$ and statistical QC functions $R_S(x; 0,1,n,k,l$), for $1 \leq k \leq n$, and $R_M(x; 0,1,n,l$), with the settings shown at the left.

size of QC sample

control measurements ———————————————— 3

NN QC function

NN $N_{16}(x)$ ▾

standardized critical errors

systematic error ———————————— 3.

random error ———————————— 4.

probabilities for rejection			
QC rules	*pfr*	*pred*	*psed*
$R_N(x; m,s,3,16)$	0.002915	0.819100	0.943740
$R_S(x; m,s,3,1,3.30)$	0.002915	0.794212	0.764801
$R_S(x; m,s,3,2,2.15)$	0.002915	0.634718	0.898063
$R_S(x; m,s,3,3,1.47)$	0.002915	0.364193	0.824186
$R_M(x; m,s,3,1.72)$	0.002915	0.456791	0.986777

Figure 4: Table presenting the probabilities for rejection of the critical mean and standard deviation of the NN QC function $R_N(x; 0,1,n,a)$ and statistical QC functions $R_S(x; 0,1,n,k,l$), for $1 \leq k \leq n$, and $R_M(x; 0,1,n,l$), with the settings shown at the left.

Details

Four 1-D CNN were designed, trained, and tested with datasets of n -tuples of simulated standardized normally distributed QC measurements, for $1 \leq n \leq 4$, as described in [1].

For $a \in \{1,2,3,4,6,8,12,16\}$, $n = \#\boldsymbol{x}$ (that is n equal to the cardinality of $\boldsymbol{x}$) and $1 \leq n \leq 4$, each 1-D CNN $R_N(\boldsymbol{x}; m, s, n, a)$ consists of:

1. The input layer, with n nodes.
2. Ten parallel arrays of five convolutional layers each. Every convolutional layer has n output channels, and kernels of size 1x1.
1. A catenating layer, which catenates the ten parallel arrays.
2. A pooling layer, performing 1-D pooling with kernels of size 2x1.
3. Ten convolutional net layers having n output channels, and kernels of size 1x1.
4. A pooling net layer, performing 1-D pooling with kernels of size 2x1.
5. Four linear net layers with output vectors of decreasing size.
6. A softmax net layer, normalizing the exponential of the output vector of the fourth linear layer.
7. The output layer, which is a decoder with two nodes.

Let m and s be the standardized observed mean and the standard deviation of the control measurements. Then the following equations are used to estimate the respective parameters [1, 2, 3, 4]:

Given that $P_S(0,1; 0,1, n, k, l) = P_N(0,1; 0,1; n, \alpha)$, to define l, we solve the following equation:

$$\frac{1}{k!(n-k)!} 2^{-n} \left(erf\left(\frac{l}{\sqrt{2}}\right) - erf\left(\frac{-l}{\sqrt{2}}\right) \right)^{n-k} \left(erf\left(\frac{-l}{\sqrt{2}}\right) - erf\left(\frac{l}{\sqrt{2}}\right) \right)^{k} \Gamma(1+n) \, {}_2F_1\left(1, k-n; 1+k; 1 \right.$$

$$\left. + \frac{2}{erf\left(\frac{-l}{\sqrt{2}}\right) - erf\left(\frac{l}{\sqrt{2}}\right)} \right) = P_N(0,1; 0,1; n, \alpha)$$

Given that $P_M(0,1; 0,1, n, k, l) = P_N(0,1; 0,1; n, \alpha)$, to define l, we solve the following equation:

$$\frac{\left(erf\left(\frac{-l\sqrt{n}}{\sqrt{2}}\right) - erf\left(\frac{l\sqrt{n}}{\sqrt{2}}\right) \right)}{2} = P_N(0,1; 0,1; n, \alpha)$$

The probabilities for the standardized random and systematic error detection of the QC function $R_S(\boldsymbol{x}; 0,1, n, k, l)$ are calculated as follows:

$$P_S(0, s; 0,1, n, k, l)$$

$$= \frac{1}{k!(n-k)!} 2^{-n} \left(erf\left(\frac{l}{s\sqrt{2}}\right) - erf\left(\frac{-l}{s\sqrt{2}}\right) \right)^{n-k} \left(erf\left(\frac{-l}{s\sqrt{2}}\right) - erf\left(\frac{l}{s\sqrt{2}}\right) \right)^{k} \Gamma(1$$

$$+ n) \, {}_2F_1\left(1, k-n; 1+k; 1 + \frac{2}{erf\left(\frac{-l}{s\sqrt{2}}\right) - erf\left(\frac{l}{s\sqrt{2}}\right)} \right)$$

$$P_S(m, 1; 0, 1, n, k, l)$$

$$= \frac{1}{k!\,(n-k)!}\,2^{-n}\left(erf\left(\frac{l+m}{\sqrt{2}}\right) - erf\left(\frac{-l+m}{\sqrt{2}}\right)\right)^{n-k}\left(erf\left(\frac{-l+m}{\sqrt{2}}\right)\right.$$

$$\left. - erf\left(\frac{l+m}{\sqrt{2}}\right)\right)^{k}\Gamma(1+n)\ {}_2F_1\left(1, k-n; 1+k; 1 + \frac{2}{erf\left(\frac{-l+m}{\sqrt{2}}\right) - erf\left(\frac{l+m}{\sqrt{2}}\right)}\right)$$

The probabilities for the standardized random and systematic error detection of the QC function $R_M(x; 0, 1, n, k, l)$ are calculated as follows:

$$P_M(0, s; 0, 1, n, l) = \frac{\left(1 + erf\left(\frac{-l}{s\sqrt{2}}\right) + erfc\left(\frac{l\sqrt{n}}{s\sqrt{2}}\right)\right)}{2}$$

$$P_M(m, 1; 0, 1, n, l) = \frac{\left(1 + erf\left(\frac{(-l+m)\sqrt{n}}{\sqrt{2}}\right) + erfc\left(\frac{(l+m)\sqrt{n}}{\sqrt{2}}\right)\right)}{2}$$

The probability for rejection for the random error s and the systematic error m of the QC function $R_S(x; m, s, n, k, l)$ is calculated as follows:

$$P_S(m, s; 0, 1, n, k, l)$$

$$= \frac{1}{k!\,(n-k)!}\,2^{-n}\left(-1 + erf\left(\frac{l+m}{s\sqrt{2}}\right) + erfc\left(\frac{-l+m}{s\sqrt{2}}\right)\right)^{n-k}\left(-1 + erf\left(\frac{-l+m}{s\sqrt{2}}\right)\right.$$

$$\left. + erfc\left(\frac{l+m}{s\sqrt{2}}\right)\right)^{k}\Gamma(1+n)\ {}_2F_1\left(1, k-n; 1+k; 1 + \frac{2}{erf\left(\frac{-l+m}{s\sqrt{2}}\right) - erf\left(\frac{l+m}{s\sqrt{2}}\right)}\right)$$

The probability for rejection for the random error s and the systematic error m of the QC function $P_M(x; m, s, n, l)$ is calculated as follows:

$$P_M(m, s; 0, 1, n, l) = \frac{\left(1 + erf\left(\frac{(-l+m)\sqrt{n}}{s\sqrt{2}}\right) + erfc\left(\frac{(l+m)\sqrt{n}}{s\sqrt{2}}\right)\right)}{2}$$

The program *Intelligent Quality Control* compares each designed CNN $R_N(x; 0, 1, n, a)$ to statistical QC functions $R_S(x; 0, 1, n, k, l)$, for $1 \leq k \leq n$, and $R_M(x; 0, 1, n, l)$, applied on tuples x of the same size n, of standardized control measurements, distributed as $\mathcal{N}(m, s^2)$, with equal probabilities for false rejection (*pfr*).

Conclusion

The program *Intelligent Quality Control* can be used as an educational or laboratory tool for the evaluation and comparison of the designed CNN and statistical QC functions.

References

1. Chatzimichail RA, Hatjimihail AT. Quality Control Using Convolutional Neural Networks Applied to Samples of Very Small Size. Technical Report XXI. Drama: Hellenic Complex Systems Laboratory, 2022. Available at: https://www.hcsl.com/Documents/hcsltr21.pdf

Abbreviations

QC: quality control
NN: neural network(s)
CNN: convolutional neural networks

Notation

$\boldsymbol{x}$: n-tuple $(x_1, x_2, \ldots, x_n)$ of QC measurements

$\#\boldsymbol{x}$: the cardinality of $\boldsymbol{x}$

$\mathcal{N}(\mu, \sigma^2)$: normal distribution with mean μ and standard deviation σ.

$R_N(\boldsymbol{x}; m_0, s_0, n, a)$: NN QC function applied on a n-tuple $\boldsymbol{x}$ of control measurements, for $a \in \{1,2,3,4,6,8,12,16\}$. The index a is related to the ratio of the simulated QC measurements of the training set, distributed as $\mathcal{N}(m_0, s_0{}^2)$ and $\mathcal{N}(m, s^2)$, for $0 < |m - m_0| < 6s_0$ or $s_0 < s < 7s_0$.

$N_a(\boldsymbol{x}) = R_N(\boldsymbol{x}; 0, 1, n, a)$

$R_S(\boldsymbol{x}; m_0, s_0, n, k, l)$: statistical QC function with decision limits $m_0 \pm ls_0$, applied on a n-tuple $\boldsymbol{x}$ of control measurements. The function rejects an analytical run if for k of the n $x_i \in \boldsymbol{x}: x_i < m_0 - ls_0 \vee x_i > m_0 + ls_0$.

$R_M(\boldsymbol{x}; m_0, s_0, n, l)$: statistical QC function with decision limits $m_0 \pm ls_0$, applied on a n-tuple $\boldsymbol{x}$ of control measurements. The function rejects an analytical run if for the mean of $\boldsymbol{x}: \bar{x} < m_0 - ls_0 \vee x_i \vee \bar{x} > m_0 + ls_0$.

$P_N(\mathrm{m}, \mathrm{s}; m_0, s_0, n, a)$: probability for rejection of the QC function $R_N(\boldsymbol{x}; m_0, s_0, n, a)$, applied to n-tuples of control measurements, distributed as $\mathcal{N}(m, s^2)$, for $a \in \{1,2,3,4,6,8,12,16\}$, and for $0 < |m - m_0| < 6s_0$ or $s_0 < s < 7s_0$.

$P_S(\mathrm{m}, \mathrm{s}; m_0, s_0, n, k, l)$: probability for rejection of the QC function $R_S(\boldsymbol{x}; m_0, s_0, n, k, l)$ applied on n-tuples of control measurements, distributed as $\mathcal{N}(m, s^2)$

$P_M(\mathrm{m}, \mathrm{s}; m_0, s_0, n, l)$: probability for rejection of the QC function $R_M(\boldsymbol{x}; m_0, s_0, n, l)$ applied on n-tuples of control measurements, distributed as $\mathcal{N}(m, s^2)$

erf(z): error function

erfc(z): complementary error function

$\Gamma(z)$: gamma function

${}_2F_1(a, b; c; z)$: hypergeometric function

The program interface

1. QC Plots

1.1. Input.

1.1.1.Plot options

1.1.1.1.x-axis

1.1.1.1.1. The standardized standard deviation (σ).

1.1.1.1.2.   The standardized mean (μ).

1.1.2. QC parameters

The following are defined:

1.1.2.1.   The NN function $N_a(\pmb{x}) = R_N(\pmb{x}; 0,1, n, a)$, $a \in \{1,2,3,4,6,8,12,16\}$.

1.1.2.2.   The size n of the QC samples ($1 \leq n \leq 4$).

1.2. Output

1.2.1. Plots of:

1.2.1.1   The probabilities for rejection $P_N(m, s; 0,1, n, a)$, where $a \in \{1,2,3,4,6,8,12,16\}$, $P_S(m, s; 0,1, n, k, l\,)$, for $1 \leq k \leq n$, and $P_M(m, s; 0,1, n, n, l\,)$.

1.2.2. Table of QC parameters

1.2.2.1.   The probability *pfr* for false rejection of the QC functions.

2. QC Calculator

2.1. Input.

2.1.1. Parameters of of the measurement process

2.1.1.1. The standardized critical systematic error (m_c).

2.1.1.2. The standardized critical random error (s_c).

2.1.2. QC parameters

The following are defined:

2.1.2.1. The NN functiom $N_a(\pmb{x}) = R_N(\pmb{x}; 0,1, n, a)$, $a \in \{1,2,3,4,6,8,12,16\}$.

2.1.2.2. The size n of the QC samples ($1 \leq n \leq 4$).

2.2. Output

2.2.1. Table of QC parameters

2.2.2.1.   The QC functions $R_N(\pmb{x}; 0,1, n, n, a)$, where $a \in \{1,2,3,4,6,8,12,16\}$, and $R_S(\pmb{x}; 0,1, n, k, l\,)$, for $1 \leq k \leq n$, and $R_M(\pmb{x}; 0,1, n, n, l\,)$.

2.2.2.2.   Their probability *pfr* for false rejection.

2.2.2.3.   Their probabilities for critical random error detection.

2.2.2.4.   The probabilities for critical systematic error detection.

Source Code

Programming language: Wolfram Language

Availability: The updated source code is available at:

https://www.hcsl.com/Tools/IntelligentQualityControl/ IntelligentQualityControl.nb

Supplement

The trained CNN models, are available as Wolfram Language Objects at:

https://www.hcsl.com/Supplements/hcsltr24s.zip

Software Requirements

Operating systems: Microsoft Windows, Linux, Apple iOS

Other software requirements: Wolfram Player®, freely available at: https://www.wolfram.com/player/ or Wolfram Mathematica®.

System Requirements

Processor: Intel Core i7® or equivalent CPU

System memory (RAM): 16GB+ recommended.

Permanent Citation:

Chatzimichail RA, Hatjimihail AT. *Intelligent Quality Control*: A Software Tool for Exploring the Application of Convolutional Neural Networks to Quality Control Samples of Very Small Size. Drama: Hellenic Complex Systems Laboratory, 2022. Available at: https://www.hcsl.com/Tools/IntelligentQualityControl/

License

First Published: September 14 2022

Revised: October 1 2022