International Journal of Mathematics and Statistics Invention (IJMSI)
E-ISSN: 2321 – 4767 P-ISSN: 2321 - 4759
www.ijmsi.org Volume 4 Issue 10 || December. 2016 || PP-29-34

Artificial Neural Network and Multi-Response Optimization in Reliability Measurement Approximation and Redundancy Allocation Problem

Yuanchen Fang[1], Nasser Fard[1], Huyang Xu[1]
[1](*Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA USA)*

ABSTRACT: *Neural network is an important tool for reliability analysis, including estimation of reliability or utility function which are too complicated to be analytical expressed for large or complex system. It has been demonstrated the neural network has significant improvement in the parameter estimation accuracy over the traditional chi-square test. There are many parameters of a neural network that should be determined while training the dataset, since different setups of algorithm parameters affect the estimation performance in either accuracy or computation efficiency. In this paper, neural network training is used to estimate the utility function for the parallel-series redundancy allocation problem, and weighted principal component based multi-response optimization method is applied to find the optimal setting of neural network parameters so that the simultaneous minimizations of training error and computing time are achieved.*
KEYWORDS: *Design of Experiment, Multi-Response Optimization, Neural Network, Redundancy Allocation Problem, System Utility Estimation*

I. INTRODUCTION

The redundancy allocation problem (RAP) involves finding a suitable allocation for the components of a system possibly with low cost, weight, or other system constraints [1]. Reliability evaluation for RAP constitutes an important computational problem. That is, even a simple RAP in series systems with linear constraints is NP-hard [2]. Focusing on reliability evaluation of RAP, functional relationship between inputs and outputs is usually nonlinear. In this case, artificial neural network (ANN) method, inspired by biological neural networks, are used to estimate the RAP reliability measurement.

When ANN method is used for RAP, it has higher prediction accuracy rate in empirical research. Also, it relaxes the assumption of having samples from specified distribution. However, the accuracy of prediction relies on the parameter settings of neural network as well as the complexities of problems and the neural network architecture; the results of the analysis could be even more significant with the selection of optimal parameters and network architecture [3]. Such details of ANN affect the approximation performance in either accuracy or efficiency. Recent studies point to applying design of experiment (DOE) is a scientific way to improve ANN's performance. Whereas the traditional DOE is designed for optimizing a single measurement, the efficiency of ANN considers both training accuracy and computation time. In order to improve ANN for RAP with both purposes, multi-response optimization based on Taguchi method is proposed. The proposed method considers more than one type of measures (e.g. training error and execution time) to quantify ANN's performance.

On the other hand, many approaches for multi-response optimization such as assigning weight to response variables [4], grey relational analysis [5] and multiple regression model [6] have been proposed in recent years. Among these approaches, multi-response optimization based on principal component analysis (PCA) has gained more attention, since it takes into account the possible correlations between response variables without increasing the computational complexity. Because there are potential relationships among ANN's performance measures, weighted principal component analysis (WPCA) based multi-response optimization approach, which takes all the uncorrelated components into consideration in order to explain all the response variables is used. One problem related to WPCA method is: each eigenvalue obtained from the application of PCA in optimization corresponds to more than one eigenvectors, and different eigenvectors will lead to different results. In order to solve this problem and produce a unique optimal solution, in this paper, the improved WPCA based multi-response optimization which integrates index procedures [7] is used to achieve the neural network parameters optimization.

RAP is a traditional optimization problem with one objective function under some constraints. It should be noted that once the system reliability is evaluated, the reliability of the system is only one of the properties that should be optimized for system operating performance. Based on this perspective, RAP is extended into multiple-objective optimization problem.

II. UTILITY ESTIMATION OF CONTINUOUS-STATE SERIES-PARALLEL SYSTEM BY NEURAL NETWORK

Figure 1 depictsa typical series-parallel system configuration, which consists of M subsystems in series and each subsystem m ($1 \leq m \leq M$) consists of N_m components being placed in parallel. For such system, all subsystems must function for the system to function. For each subsystem, at least one component must be operational for the subsystem to function. One optimization problem for the series-parallel system is to determine the optimal number of parallel components in each subsystem so that the system utility is maximized while the cost, volume, and/or weight are minimized (optimal redundancy allocation problem).

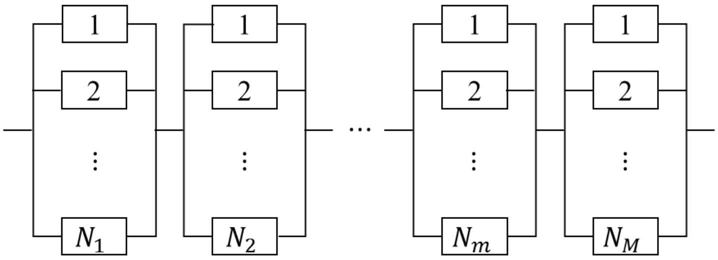

Figure 1 Reliability block diagram for series-parallel system

Liu et al. (2003) modeled the deterioration of each component as a continuous state variable S taking values in the range $[0,1]$, where 1 indicates as good as new and 0 indicates complete failure. By the definition of multi-state series-parallel system, the system state is the state of the worst subsystem while the subsystem state is the state of the best component in this subsystem. Given the utility function of the system when it is in state s, $u(s)$, and the state probability density function of component$i$ in subsystem m, $f_{im}(s)$, the expected utility of the system is [8]

$$U(N_1,N_2,\dots,N_M) = -\int_0^1 u(s)\left(\frac{d}{ds}\left(\prod_{m=1}^{M}\left(1-\prod_{i=1}^{N_m}\left(\int_0^s f_{im}(t)\,dt\right)\right)\right)\right)ds \qquad (1)$$

However, when the number of subsystems M is large or the component state density functions $f_{im}(s)$ is not simple (or expressed in empirical form), Equation (1) is too complicated to be analytical expressed and efficiently solved. Therefore, Liu et al. (2003) proposed a neural network approach to estimate the main and most complex part in Equation (1)

$$g(s,N_1,N_2,\dots,N_M) = \prod_{m=1}^{M}\left(1-\prod_{i=1}^{N_m}\left(\int_0^s f_{im}(t)\,dt\right)\right) \qquad (2)$$

which is the CDF of system state distribution. The detailed neural network training and testing algorithm can be found in [8] and [9]. Replace $g(s,N_1,N_2,\dots,N_M)$ in Equation (1) with its approximation, and the system utility can be solved easily and efficiently.

III. NEURAL NETWORK PARAMETER DETERMINATION BY MULTI-RESPONSE OPTIMIZATION MODEL

When executing the neural network algorithm stated in Section 2, there are many parameters of neural network that should be determined while training the dataset, since different setups of algorithm parameters affect the estimation performance in either accuracy or computation efficiency. For example, it was found that when the number of neurons in the single hidden layer exceeds 15, the accuracy starts to decline [10]. In this paper, the settings of neural network parameters are determined using design of experiment with more than one response variables which simultaneously consider the estimation accuracy and computation time. Multi-response optimization model is applied to determine an optimal combination of parameters for neural network which could efficiently provide the more accurate estimate of the parameters.

3.1 Factors and Levels in Neural Network Parameter Optimization

For estimating $g(s,N_1,N_2,\dots,N_M)$ in Section 2 by neural network, the following factors (parameters) need to be considered: [11]

- Number of neurons in hidden layer (NNHL): Neurons in the hidden layers are used to capture the nonlinear structures in a time series [2]. Too few neurons will lead to lack-of-good fit, and too many neurons will cause any data can be trivially fitted [5]. Here, the factor levels are chosen from $k(n+1)$, where n is the number of input and k = 1, 2, or 3 in our case.

- Training algorithm (TA): It is the algorithm type to calculate the neural network. In this paper, three types of algorithm are considered: backpropagation (backprop), resilient back propagation with weight backtracking (rprop+), and globally convergent algorithm with smallest absolute gradient learning rate (sag).
- Epochs (E): It is a measure of the number of times all of the training vectors are used to update the weights. A higher number of epochs will improve the accuracy of the model but increase the cost as a function of time. In this paper, three factor levels, 10, 50 and 100, will be considered.
- Number of training pairs in training data set (NTP): To produce an efficient neural network architecture, the training dataset must be complete enough to represent the entire range of model with noise included. However, irrationally increasing the size of training datasets may lead to over fitting and waste of computation [12]. This study investigates the number of data samples with three factor levels, which are 500, 1000 and 1500.

3.2 Response Variables in Neural Network Parameter Optimization

The objective of this optimization is to find the best combination of the above factor levels such that the parameter estimation accuracy is maximized while the training time is minimized. When running each setting, the training error (TE), validation error (VE), training steps (TS) and execution time (T, in seconds) are recorded. Therefore, the problem is modeled as a multi-response optimization experimental design with factors and levels as discussed in Section 3.1 and four response variables, training error and execution time. Response variables are the smaller-the-better. For each parameter setting, 5 experiments will be run to minimize bias. The factors and levels under consideration for this neural network algorithm are complex and in a large amount, which leads to a huge number of experiments. Therefore, instead all the combinations are executed, fractional factorial design based on Taguchi method is used, which could significantly increase the optimization efficiency. The 3^{4-2} factorial design problem with 4 response variables and 5 replicates per experiment (run) is given in Table 1.

Table 1 Multi-response optimization experimental design for neural network parameter selection

	Factor Level				Response ($i = 1,2,3,4,5$)			
	NNHL	TA	E	NTP	Training error	Validation error	Training steps	Execution time
1	$n+1$	backprop	10	500	TE_{1i}	VE_{1i}	TS_{1i}	T_{1i}
2	$n+1$	rprop+	50	1000	TE_{2i}	VE_{2i}	TS_{2i}	T_{2i}
3	$n+1$	sag	100	1500	TE_{3i}	VE_{3i}	TS_{3i}	T_{3i}
4	$2(n+1)$	backprop	50	1500	TE_{4i}	VE_{4i}	TS_{4i}	T_{4i}
5	$2(n+1)$	rprop+	100	500	TE_{5i}	VE_{5i}	TS_{5i}	T_{5i}
6	$2(n+1)$	sag	10	1000	TE_{6i}	VE_{6i}	TS_{6i}	T_{6i}
7	$3(n+1)$	backprop	100	1000	TE_{7i}	VE_{7i}	TS_{7i}	T_{7i}
8	$3(n+1)$	rprop+	10	1500	TE_{8i}	VE_{8i}	TS_{8i}	T_{8i}
9	$3(n+1)$	sag	50	500	TE_{9i}	VE_{9i}	TS_{9i}	T_{9i}

3.3 Unique Solution WPCA Multi-Response Optimization

The modified WPCA based multi-response optimization approach proposed by Fard et al. (2016) is applied to the above problem and thus to determine a unique optimal neural network parameter setting. An algorithm based on this proceduredescribed in [7] is developed for calculation of the parameter setting. The procedure is demonstrated as follows:

Step 1: Compute S/N ratio for each response:

Since all of the responses are smaller-the-better, the loss functions for each response are calculated as

$$L(TE_j) = \frac{1}{5}\sum_{i=1}^{5}(TE_{ji})^2, L(VE_j) = \frac{1}{5}\sum_{i=1}^{5}(VE_{ji})^2, L(TS_j) = \frac{1}{5}\sum_{i=1}^{5}(TS_{ji})^2, L(T_j) = \frac{1}{5}\sum_{i=1}^{5}(T_{ji})^2, j = 1, 2, \dots, 9$$

Then the S/N ratio for each response are

$$\eta(TE_j) = -10\log_{10} L(TE_j), \eta(VE_j) = -10\log_{10} L(VE_j), \eta(TS_j) = -10\log_{10} L(TS_j),$$

$$\eta(T_j) = -10\log_{10} L(T_j), j = 1, 2, \dots, 9$$

Step 2: Normalize S/N ratio of each response:

$$\eta^{(N)}(TE_j) = \frac{\eta(TE_j) - \min\{\eta(TE_1), \dots, \eta(TE_9)\}}{\max\{\eta(TE_1), \dots, \eta(TE_9)\} - \min\{\eta(TE_1), \dots, \eta(TE_9)\}},$$

$$\eta^{(N)}(VE_j) = \frac{\eta(VE_j) - \min\{\eta(VE_1), \dots, \eta(VE_9)\}}{\max\{\eta(VE_1), \dots, \eta(VE_9)\} - \min\{\eta(VE_1), \dots, \eta(VE_9)\}},$$
$$\eta^{(N)}(TS_j) = \frac{\eta(TS_j) - \min\{\eta(TS_1), \dots, \eta(TS_9)\}}{\max\{\eta(TS_1), \dots, \eta(TS_9)\} - \min\{\eta(TS_1), \dots, \eta(TS_9)\}},$$
$$\eta^{(N)}(T_j) = \frac{\eta(T_j) - \min\{\eta(T_1), \dots, \eta(T_9)\}}{\max\{\eta(T_1), \dots, \eta(T_9)\} - \min\{\eta(T_1), \dots, \eta(T_9)\}}, j = 1, 2, \dots, 9$$

Step 3: Perform indexing PCA to identify eigenvalues and eigenvectors:
(3-1) Calculate the variance of each response variable, and sort the response variables in a descending order of their corresponding variances,
$Var(Y_1) \geq Var(Y_2) \geq Var(Y_3) \geq Var(Y_4), Y_1, Y_2, Y_3, Y_4 = TE, VE, TS, T$
(3-2) Assign indices 1, 2, …, 9 to each response. for example,
$i^{(0)}(y_{1j}) = a, j = 1, 2, \dots, 9, a = 1, 2, \dots, 9$

meansY_{1j} is the ath largest values among $Y_{11}, Y_{12}, \dots, Y_{19}$. Similarly, we have
$i^{(0)}(y_{2j}) = b, i^{(0)}(y_{3j}) = c, i^{(0)}(y_{4j}) = d, j = 1, 2, \dots, 9, b, c, d = 1, 2, \dots, 9$

(3-3) Perform PCA on the normalized data obtained in step 2, which gives 4 eigenvectors, $\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}, \boldsymbol{v_4}$ and their corresponding eigenvalues, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$.
(3-4) For all combinations of eigenvectors with different +/- signs,
$(\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}, \boldsymbol{v_4}), (-\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}, \boldsymbol{v_4}), (\boldsymbol{v_1}, -\boldsymbol{v_2}, \boldsymbol{v_3}, \boldsymbol{v_4}), \dots, (-\boldsymbol{v_1}, -\boldsymbol{v_2}, -\boldsymbol{v_3}, -\boldsymbol{v_4})$
project original data points (normalized S/N ratio) onto the new eigenvector coordinate system and get the rotated data set. Assign indices 1, 2, …, 9 to the rotated data set as (3-2),
$i(PC_{1j}) = a', i(PC_{2j}) = b', i(PC_{3j}) = c', i(PC_{4j}) = d', a', b', c', d' = 1, 2, \dots, 9$
(3-5) Calculate the differences between each pair of rotated data indices with original data indices and find the sum of the absolute differences,

$$d = \sum_{j=1}^{9} \left|i^{(0)}(y_{1j}) - i(PC_{1j})\right| + \sum_{j=1}^{9} \left|i^{(0)}(y_{2j}) - i(PC_{2j})\right| + \sum_{j=1}^{9} \left|i^{(0)}(y_{3j}) - i(PC_{3j})\right| + \sum_{j=1}^{9} \left|i^{(0)}(y_{4j}) - i(PC_{4j})\right|$$

(3-6) Identify the minimum d, and its corresponding eigenvector coordinate system is chosen as the best eigenvector combination.

Step 4: Use the eigenvector and eigenvalue obtained in step 3 to transform normalized S/N ratio into multi-response performance index (MPI),
$\Omega_j = w_1 z_{1j} + w_2 z_{2j} + w_3 z_{3j} + w_4 z_{4j}$
where$w_l = \frac{\lambda_l}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4}, z_{lj} = (y_{1j} \quad y_{2j} \quad y_{3j} \quad y_{4j})\boldsymbol{v_l}, j = 1, 2, \dots, 9, l = 1, 2, 3, 4.$

Step 5: Determine the optimal factor-level combination by identifying the level average of the factors that lead to highest value for MPI. The optimal factor-level corresponds to the best selection of neural network parameters.

IV. EXAMPLE

We use the 4-subsystem parallel-series system solved in [8] as an example to illustrate the proposed method. In this example, the utility function is $u(s) = 10s$, and the state probability density function of components in the four subsystems are

$f_{i1} = 1;$ unit distribution
$f_{i2} = 2s;$ triangular distribution
$f_{i3} = \frac{\Gamma(5.5)}{\Gamma(2)\Gamma(3.5)} s(1-s)^{2.5};$ beta distribution
$f_{i4} = \frac{\Gamma(7)}{\Gamma(5)\Gamma(2)} s^4(1-s);$ beta distribution

The cost of components in the four subsystems are

$c_{i1} = 3200, c_{i2} = 1700, c_{i3} = 830, c_{i4} = 2500$

Besides, we will consider the weight and volume in this paper too. The weight of components in the four subsystems are

$w_{i1} = 30, w_{i2} = 100, w_{i3} = 350, w_{i4} = 50$

and the volume of components in the four subsystems are

$s_{i1} = 20, s_{i2} = 75, s_{i3} = 55, w_{i4} = 40$

The maximum number of components available for each subsystem are

$u_1 = u_2 = u_3 = u_4 = 40$

4.1 Computation of System Utility and Optimization of Neural Network Parameters

Input-output data sets of size 500, 1000, 1500 are randomly generated, and are trained using R language neural network package “neuralnet” [13]. Then WPCA based multi-response optimization method [7] is applied. The optimal combination of factor levels in Table 2,are obtained through the procedure given in section 3.3.

Table 2 Optimal settings of neural network parameters

Parameter	Setting level
NNHL	12
TA	rprop+
E	100
NTP	1000

Following the results in Table 4.1, we set the number of neurons in the hidden lay as 12, use resilient back propagation with weight backtracking as training algorithm, and set the number of epochs as 100. 1000 pairs of training data are randomly generated and put into the neural network to be trained again. We obtain the weights between neural network layers, and the estimation of system utility is

$$\begin{aligned}\widehat{U}(N_1, N_2, \dots, N_M) = 10\Bigg\{0.2158\Bigg(1 - \frac{1}{1 + e^{-(3.9704 - 0.7055 - 3.5444 n_1 + 2.0949 n_2 + 0.8166 n_3 - 2.4428 n_4)}} \\ + \frac{1}{-0.7055}\ln\frac{1 + e^{-(3.9704 - 0.7055 - 3.5444 n_1 + 2.0949 n_2 + 0.8166 n_3 - 2.4428 n_4)}}{1 + e^{-(3.9704 - 3.5444 n_1 + 2.0949 n_2 + 0.8166 n_3 - 2.4428 n_4)}}\Bigg) + \cdots \\ + 1.1089\Bigg(1 - \frac{1}{1 + e^{-(5.3948 + 1.4417 - 0.8414 n_1 - 0.4393 n_2 + 4.19 n_3 - 0.821 n_4)}} \\ + \frac{1}{1.4417}\ln\frac{1 + e^{-(5.3948 + 1.4417 - 0.8414 n_1 - 0.4393 n_2 + 4.19 n_3 - 0.821 n_4)}}{1 + e^{-(5.3948 - 0.8414 n_1 - 0.4393 n_2 + 4.19 n_3 - 0.821 n_4)}}\Bigg)\Bigg\} \qquad (3)\end{aligned}$$

4.2 Optimization of System Redundancy Allocation

In this paper, the objective for series-parallel system redundancy allocation problem is to find the optimal number of components designed in each subsystem which maximizes the system utility, and minimizes the design cost, system weight, and system volume. System utility U, is estimated by Equation (3). The factors being considered here are the redundancy numbers of subsystems, and the multiple responses to be simultaneously optimized are utility, cost, weight, and volume.

Unique solution WPCA based multi-response optimization is designed to determine the best number of redundancies for this four-subsystem example. Table 3 shows the initial experimental array.

Table 3 Initial experimental array

	Factor Levels				Response			
	N_1	N_2	$\cdots$	N_M	U	C	W	S
1	$\left\lceil\frac{u_1}{4}\right\rceil$	$\left\lceil\frac{u_2}{4}\right\rceil$	$\cdots$	$\left\lceil\frac{u_M}{4}\right\rceil$	${}^{(1)}u_1$	${}^{(1)}c_1$	${}^{(1)}w_1$	${}^{(1)}s_1$
2	$\left\lceil\frac{u_1}{4}\right\rceil$	$\left\lceil\frac{u_2}{2}\right\rceil$	$\cdots$	$\left\lceil\frac{u_M}{2}\right\rceil$	${}^{(1)}u_2$	${}^{(1)}c_2$	${}^{(1)}w_2$	${}^{(1)}s_2$
3	$\left\lceil\frac{u_1}{4}\right\rceil$	$\left\lceil\frac{3u_2}{4}\right\rceil$	$\cdots$	$\left\lceil\frac{3u_M}{4}\right\rceil$	${}^{(1)}u_3$	${}^{(1)}c_3$	${}^{(1)}w_3$	${}^{(1)}s_3$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
3^M	$\left\lceil\frac{3u_1}{4}\right\rceil$	$\left\lceil\frac{3u_2}{4}\right\rceil$	$\cdots$	$\left\lceil\frac{3u_M}{4}\right\rceil$	${}^{(1)}u_{3^M}$	${}^{(1)}c_{3^M}$	${}^{(1)}w_{3^M}$	${}^{(1)}s_{3^M}$

The multi-response optimization experiments are designed stage by stage, where unique solution WPCA based method is applied continuously. As experiments go on, the factors will tend to their optimal levels. When the

change of factor levels does not significantly affect the response measurements, the iteration stops, and we get the final result as shown in Table 3.
Therefore, for this four-subsystem configuration design, the optimal number of components allocated for subsystem 1 is 10, for subsystem 2 is 20, for subsystem 3 is 36, and for subsystem 4 is 21.

Table 3 Optimal factor levels for redundancy allocation

Subsystem	Number of Components in Subsystem
N_1	10
N_2	20
N_3	36
N_4	21

V. CONCLUSION

For large andcomplex parallel-series system redundancy allocation problem, the utility function is usually too complicated to be explicitly solved. Feed-forward neural network provides an efficient way to approximate the utility function. In the neural network training, there are several parameters need to be determined in order to make the neural network efficient and accurate.
In this paper, ANN algorithm is presented to estimate the expected system utility, while the optimal parameter settings of the ANN are obtained by means of design of experiments (DOE). Since there are more than one criteria to measure the performance of ANN,multi-response optimization based on Taguchi method is used to investigate the response variables at the same time. Meanwhile, some of these criteria have potential relationship, so that the possible correlations between response variables need to be considered without increasing the computational complexity.
Therefore,WPCA multi-response optimization based on Taguchi method is applied to determine the best choice of four main neural network parameters - number of neurons in hidden layer, training algorithm, epochs, and number of training pairs in training data set, which simultaneously optimize four neural network performance measurements - the training error, validation error, training steps and execution time.
After estimating the utility function, multi-response experimental design is applied to determine the optimal number of redundancies for the parallel-series system, while simultaneously maximizing the system utility, and minimizing the total cost, system weight and system size.

REFERENCES

[1]. A. A. Najafi, H. Karimi, A. Chambari and F. Azimi, Two metaheuristics for solving the reliability redundancy allocation problem to maximize mean time to failure of a series–parallel system,*ScientiaIranica, 20(3),*2013, 832-838.

[2]. M. S. Chern,On the computational complexity of reliability redundancy allocation in a series system,*Operations research letters, 11(5),*1992, 309-315.

[3]. M. Y. Chen, M. H. Fan, Y. L. Chen and H. M. Wei, Design of experiments on neural network's parameters optimization for time series forecasting in stock markets,*Neural Network World, 23(4),*2013, 369-393.

[4]. C. Hung, *A cost-effective multi-purpose off-line quality control for semiconductor manufacturing,*doctoral diss., National Chiao Tung University, Taiwan, 1990.

[5]. J. Lin and C. Lin, The use of the orthogonal array with grey relational analysis to optimize the electrical discharge machining process with multiple performance characteristics,*International Journal of Machine Tools and Manufacture, 42(2),* 2002, 237-244.

[6]. S. Gauri and S. Pal, Multi-response optimization using multiple regression-based weighted signal-to-noise ratio (MRWSN),*Quality Engineering, 22(4),* 2010, 336-350.

[7]. N. Fard, H. Xu and Y. Fang, A unique solution for principal component analysis-based multi-response optimization problems,*The International Journal of Advanced Manufacturing Technology, 82(1),* 2016, 697-709.

[8]. P. Liu, M. Zuo and M. Meng, Using neural network function approximation for optimal design of continuous-state parallel-series systems,*Computers & Operations Research, 30(3),* 2003, 339-352.

[9]. M. Zuo, Z. Tian and H. Huang, Neural networks for reliability-based optimal design," in Computational Intelligence in Reliability Engineering, vol. 40, Springer Berlin Heidelberg, 2007, pp. 175-196.

[10]. M. Liu, W. Kuo and T. Sastri, "An exploratory study of a neural network approach for reliability data analysis, *Quality and Reliability Engineering International, 11(2),*1995, 107-112.

[11]. P. Balestrassi, E. Popova, A. Paiva and J. Marangon Lima, Design of experiments on neural network's training for nonlinear time series forecasting,*Neurocomputing, 72(4-6),* 2009, 1160-1178.

[12]. D. Korsmeyer, T. Rajkumar and J. Bardina, Training data requirement for a neural network to predict aerodynamic coefficients,*Proc. SPIE 5102, Independent Component Analyses, Wavelets, and Neural Networks*, 2003, 92-103.

[13]. S. Fritsch, F. Guenther and M. Suling, Package 'neuralnet', 20 February 2015. [Online]. Available: *https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf*.