Predicting Reliability Via Neural Networks

Marseguerra M. • Department of Nuclear Engineering, Polytechnic of Milan • Milan
Zio E. • Department of Nuclear Engineering, Polytechnic of Milan • Milan
Ammaturo M. • ELASIS S.C.p.A. Vehicle Testing Department • Naples
Fontana V. • ELASIS S.C.p.A. Vehicle Testing Department • Naples

Key Words: Reliability prediction, Neural networks, Automotive components.

SUMMARY & CONCLUSIONS

The objective of this work is to predict the reliability of automotive components and systems from experimental failure data using Artificial Neural Networks. To construct the necessary neural models, the NEural Simulation Tool (NEST), developed by Polytechnic of Milan, has been employed. An operative procedure based on the developed ANN models has been been implemented to predict the trend of the unreliability index $R^{100}(t)$, the number of faults in 100 vehicles at time t (number of months from production time), starting from information on the number of vehicles produced and sold and the predicted number of faults up to the previous time t-1. The procedure has been applied on data from the Fiat Car Group, leading to satisfactory results.

1. INTRODUCTION

The objective of this work is to investigate the feasibility of using neural networks for predicting the reliability of complex systems and components.

Neural networks are information processing systems composed of simple nonlinear processing elements (nodes) linked by weighted connections. The values of the connection weights are determined through a training procedure during which a large number of exemplars (patterns), each one constituted by the inputs and the corresponding outputs, are repeatedly presented to the network. During the training phase, the values of the connection weights are modified so as to minimize a properly defined *error function*, or *Energy function*, which accounts for the deviations of the network outputs from the true values. Through this training procedure, the network is able to build an internal representation of the possibly nonlinear input/output mapping of the problem under investigation. After the training is completed, the final connection weights are kept fixed. New input patterns are presented to the network which is capable of recalling the information stored in the connection weights during training to produce the corresponding output, coherent with the internal representation of the input/output mapping.

The type of neural networks here employed is the classical multi-layered, feed-forward one trained by the error back-propagation method which follows from the general gradient descent method (Refs. 1-3). It consists of an iterative gradient algorithm designed to minimize the mean square error between the actual network output and the true value.

The networks used in the present work have been generated with the software NEST (NEural Simulation Tool) developed at the Department of nuclear engineering of the Polytechnic of Milan (www.cesnef.polimi.it/nest.htm). The training patterns for the ANNs have been constructed from experimental data on automotive components' defects provided by ÉLASIS Pomigliano d'Arco (Naples) Car Research Centre (www.elasis.it).These data are collected and recorded at fixed time intervals of 2-3 months and contain the components' production date, the number of units produced, the number of units sold and the cumulative number of defects recorded, normalized to 100 vehicles, which is taken as an index of the reliability of the component. On the basis of these data, a set of ANNs have been successfully trained to predict with satisfactory accuracy the value of normalized cumulative defects one time step ahead.

In this paper we present the details of the procedure used for building the neural model and discuss in depth the results obtained. Conclusions will be drawn on the feasibility of using the neural approach, particularly in relation to the amount of data available for the neural network training. Future research developments will also be suggested.

2. ARTIFICIAL NEURAL NETWORK (ANN)

Artificial Neural networks are information processing systems composed of simple processing elements (nodes) linked by weighted connections (Refs. 1-3). Here we limit ourselves to describing the most commonly used multi-layered feed-forward neural network which, in its simplest form, consists of three layers of processing elements: the input, the hidden and the output layers, with n_i, n_h and n_o nodes respectively (Fig. 1). The signal is processed forward from the input to the output layer. Each node collects the output values, weighted by the connection weights, from all the nodes of the preceding layer, processes this information through a sigmoidal function $f(x) = (1+e^{-x})^{-1}$ and then delivers the result towards all the nodes of the successive layer. In the present work both input and hidden layers have the additional bias node, which is

0-7803-7717-6/03/$17.00 © 2003 IEEE

often employed as a threshold in the argument of the activation function and whose output always equals unity.

The values of the connection weights are determined through a training procedure. In this case we have adopted the usual error back-propagation algorithm which follows from the general gradient descent method (Ref. 2). In short, the back-propagation algorithm performs the steepest descent in the weight space on a surface, whose height at any point is equal to the error function. It consists of an iterative gradient algorithm designed to minimize the mean square error between the actual network output and the true value. Sets of n_p input and associated n_o outputs are repeatedly presented to the network and the values of the connection weights are modified so as to minimize the *average squared output deviation error function*, or *Energy function*, defined as

$$D = \frac{1}{2n_p n_o} \sum_{p=1}^{n_p} \sum_{l=1}^{n_o} (t_{pl} - o_{pl})^2 \qquad (1)$$

where t_{pl} and o_{pl} are the true and the network-computed values of the *l*-th output node, to the *p*-th pattern presented. Through this training procedure, the network is able to build an internal representation of the input/output mapping of the problem under investigation. The success of the training strongly depends on the normalization of the data and on the choice of the training parameters. In this work, each signal has been transformed by an affine mapping in the interval (0.2, 0.8); the learning coefficient and the momentum factor are 0.6 and 0.8, respectively; moreover the connection weights have been initialized randomly within the interval (-0.3, 0.3).

After the training is completed, the final connection weights are kept fixed. New input patterns are presented to the network which is capable of recalling the information stored in the connection weights during training to produce the corresponding output, coherent with the internal representation of the input/output mapping. Notice that the non-linearity of the sigmoidal function of the processing elements allows the neural network to learn arbitrary nonlinear mappings. Moreover, each node acts independently of all the others and its functioning relies only on the local information provided through the adjoining connections. In other words, the functioning of one node does not depend on the states of those other nodes to which it is not connected. This allows for efficient distributed representation and parallel processing, and for an intrinsic fault-tolerance and generalization capability.

These attributes render the artificial neural networks a powerful tool for signal processing, nonlinear mappings and near-optimal solution to combinatorial optimization problems.

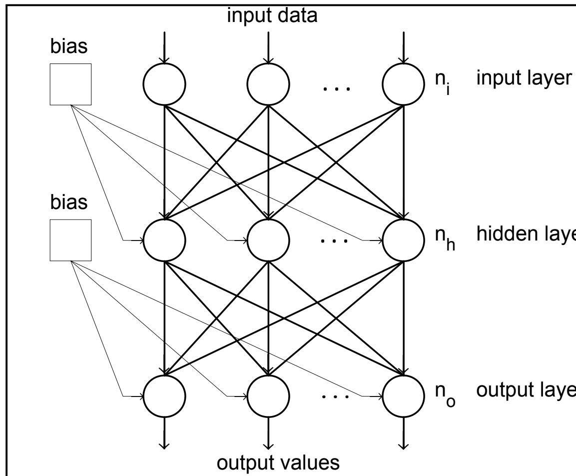


Figure 1: Scheme of a three-layered, feed forward neural network.

3. PROBLEM DEFINITION

For each type of car produced, and with respect to the whole vehicle, its systems and components, Élasis S.C.p.A. has the following data available, from Fiat Auto data banks:

- Number of production lots (max 48);

and for a generic monthly lot *n*:

- Production date, D_n,
- Quantity of produced vehicles, QP_n,
- Cumulative rate of failures, $R_n^{100}(t_i)$, at the time t_i of collection of fault data measured in *months* from the production date, $t_1=1, t_2=3, t_3=6, t_4=9, t_5=12$;
- Quantity of sold vehicles, $QV_n(t_i)$, at the time t_i of collection of fault data measured in *months* from the production date, $t_1=1, t_2=3, t_3=6, t_4=9, t_5=12$ *months*,

To know the value of $R_n^{100}(t_5)=R_n^{100}(12months)$, for a given monthly lot *n*, it is necessary to wait at least 14 months from the date the vehicles are sold (12 months + 2 months of delay in the update of the Data Bank). Correspondingly, the data on all lots which is available at a generic time $\tau = D_{n+2}$, two months after the production date of lot *n*, can be represented in the following matrix $A=(A_{ij})$ with $i=1,\ldots n+2$; $j=1,\ldots 8$, reported in Table 1.

j=1	j=2	j=3	j=4	j=5	j=6	j=7	j=8
Nr. of lots	Production date	Nr. of produced vehicles	Nr. of faults for 100 vehicles after t_1=1 month	Nr. of faults for 100 vehicles after t_2=3 months	Nr. of faults for 100 vehicles after t_3=6 months	Nr. of faults for 100 vehicles after t_4=9 months	Nr. of faults for 100 vehicles after t_5=12 months
1	D_1	QP_1	$R_1(1)$	$R_1(3)$	$R_1(6)$	$R_1(9)$	$R_1(12)$
2	D_2	QP_2	$R_2(1)$	$R_2(3)$	$R_2(6)$	$R_2(9)$	$R_2(12)$
3	D_3	QP_3	$R_3(1)$	$R_3(3)$	$R_3(6)$	$R_3(9)$	$R_3(12)$
4	D_4	QP_4	$R_4(1)$	$R_4(3)$	$R_4(6)$	$R_4(9)$	$R_4(12)$
5	D_5	QP_5	$R_5(1)$	$R_5(3)$	$R_5(6)$	$R_5(9)$	$R_5(12)$
…..	…..	…..	…..	…..	…..	…..	…..
n-11	D_{n-11}	QP_{n-11}	$R_{n-11}(1)$	$R_{n-11}(3)$	$R_{n-11}(6)$	$R_{n-11}(9)$	$R_{n-11}(12)$
n-10	D_{n-10}	QP_{n-10}	$R_{n-10}(1)$	$R_{n-10}(3)$	$R_{n-10}(6)$	$R_{n-10}(9)$	n.a.
n-9	D_{n-9}	QP_{n-9}	$R_{n-9}(1)$	$R_{n-9}(3)$	$R_{n-9}(6)$	$R_{n-9}(9)$	n.a.
n-8	D_{n-8}	QP_{n-8}	$R_{n-8}(1)$	$R_{n-8}(3)$	$R_{n-8}(6)$	$R_{n-8}(9)$	n.a.
n-7	D_{n-7}	QP_{n-7}	$R_{n-7}(1)$	$R_{n-7}(3)$	$R_{n-7}(6)$	n.a.	n.a.
n-6	D_{n-6}	QP_{n-6}	$R_{n-6}(1)$	$R_{n-6}(3)$	$R_{n-6}(6)$	n.a.	n.a.
n-5	D_{n-5}	QP_{n-5}	$R_{n-5}(1)$	$R_{n-5}(3)$	$R_{n-5}(6)$	n.a.	n.a.
n-4	D_{n-4}	QP_{n-4}	$R_{n-4}(1)$	$R_{n-4}(3)$	n.a.	n.a.	n.a.
n-3	D_{n-3}	QP_{n-3}	$R_{n-3}(1)$	$R_{n-3}(3)$	n.a.	n.a.	n.a.
n-2	D_{n-2}	QP_{n-2}	$R_{n-2}(1)$	$R_{n-2}(3)$	n.a.	n.a.	n.a.
n-1	D_{n-1}	QP_{n-1}	$R_{n-1}(1)$	n.a.	n.a.	n.a.	n.a.
n	Dn	QP_n	$R_n(1)$	n.a.	n.a.	n.a.	n.a.
n+1	D_{n+1}	QP_{n+1}	n.a.	n.a.	n.a.	n.a.	n.a.
n+2	D_{n+2}	QP_{n+2}	n.a.	n.a.	n.a.	n.a.	n.a.

(n.a. = data not available at the time D_{n+2}).

Table 1: Matrix A=(Aij) of data available at time Dn+2

For the given production lot *n*, these data do not allow the definition of reliability objectives to be achieved during the preliminary vehicle development stages nor the monitoring, in the early months from the product release, of the likelihood of achieving the reliability objectives at the end of the warranty period of 12 months.

4. RELIABILITY PREDICTION BY NEURAL MODELS

To overcome the above problems of reliability prediction, i.e. to obtain the missing values in the entries of matrix A above, four neural models ANN_i , i= 1..4 have been designed and trained. In the present work, to train the predictive neural models we considered, for greater sensitivity, the cumulative value $R^{100}(t_1)$ and the differences of the successive cumulative values, $R_{t_i t_{i-1}}=[R^{100}(t_i)- R^{100}(t_{i-1})]$, $i=2,3,4,5$. Nevertheless, the predictive results obtained are here presented in terms of the cumulative values at the different times considered.With reference to the training, testing and predictive procedures of the neural approach, in the following we shall use the notation $T_{i,j}$ or $P_{i,j}$ to indicate the elements of the matrix of Table 1 if we are in the training/testing or in the operative (prediction) phase of the procedure, respectively. Then, the training/testing patterns for the four neural networks have the following structure ($T_{input}| T_{output}$):

ANN_1 $T_{i,2}, T_{i,3}, T_{i,4}, T_{i,5}, T_{i,6}, T_{i,7} \mid T_{i,8}$ with i= 1,…n-11;
ANN_2 $T_{i,2}, T_{i,3}, T_{i,4}, T_{i,5}, T_{i,6} \mid T_{i,7}$ with i= 1,…n-8
ANN_3 $T_{i,2}, T_{i,3}, T_{i,4}, T_{i,5} \mid T_{i,6}$ with i= 1,…n-5
ANN_4 $T_{i,2}, T_{i,3}, T_{i,4} \mid T_{i,5}$ with i= 1,…n-2.

We considered a number of lots n=35. The training and testing sets available for the four neural models have been structured as summarized in Table 2:

Neural model	Number of Patterns for Training Set	Number of Patterns for Testing Set	Total Number of Patterns available
ANN_1	20	4	24
ANN_2	20	7	27
ANN_3	25	5	30
ANN_4	25	8	33

Table 2 Number of patterns for training and testing of the four neural models

In Table 3 we report, for each neural model, the values of the most relevant parameters: the input and output nodes are determined by the physics of the problem; the other values have been selected via an automated optimization procedure embedded in the NEST software.

Neural model			Optimized parameters		
	Input Nodes	Output Nodes	Hidden Nodes	Learning Rate	Momentum
ANN_1	6	1	8	0.6	0.8
ANN_2	5	1	2	0.9	0.9
ANN_3	4	1	1	0.9	0.9
ANN_4	3	1	4	0.7	0.4

Table 3 Neural Models parameters

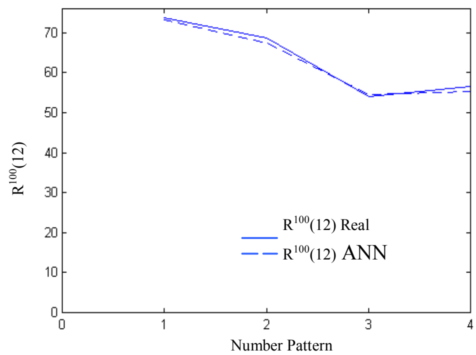


Figure 2: ANN$_1$ predictions

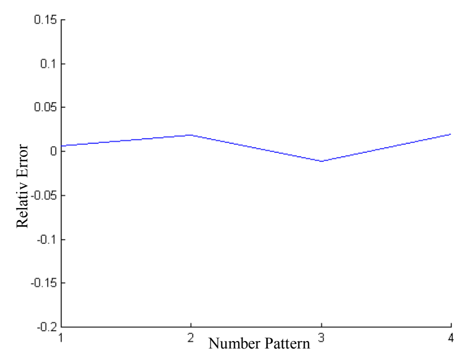


Figure 3: ANN$_1$ relative errors

In Figure 2 we report the comparison between the real values of R^{100} (12) and the corresponding trained-ANN$_1$ network predictions for the 4 lots of the corresponding testing set. Figure 3 reports the corresponding relative error. The performances of the all neural networks are summarized in Table 4.

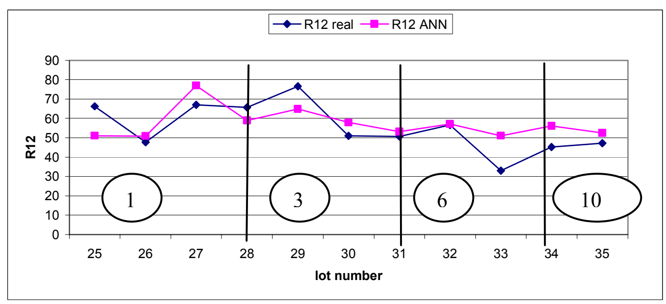
Neural Model	Max. relative error Testing Set	Average absolute relative error Testing Set
ANN$_1$	1.96%	1.39%
ANN$_2$	-9.7%	3.31%
ANN$_3$	15.1%	7.65%
ANN$_4$	-17.9%	7.96%

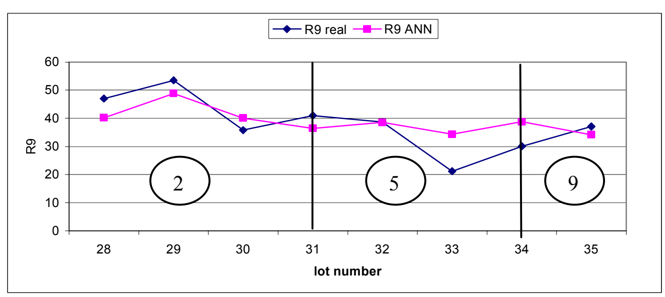
Table 4 Performances of the four neural models

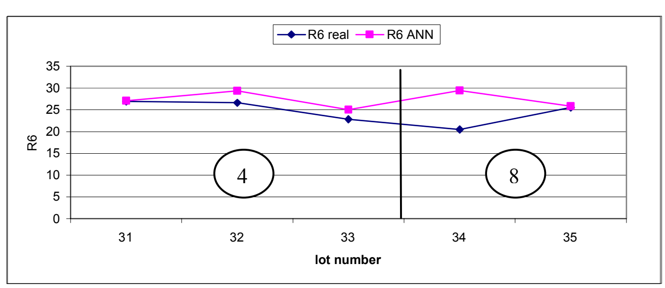
Given the relatively satisfactory performance of the four neural networks, the successive operative (predictive) phase for the prediction of $R^{100}(t_i)$ was tackled according to a procedure which entails the recycling in input of values estimated at previous time steps:

1. Prediction by ANN$_1$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, $P_{i,5}$, $P_{i,6}$, $P_{i,7}$ | $P^*_{i,8}$ with i=n-10,…n-8
2. Prediction by ANN$_2$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, $P_{i,5}$, $P_{i,6}$, | $P^*_{i,7}$ with i=n-7,….n-5
3. Prediction by ANN$_1$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, $P_{i,5}$, $P_{i,6}$, $P^*_{i,7}$ | $P^{**}_{i,8}$ with i=n-7,….n-5
4. Prediction by ANN$_3$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, $P_{i,5}$, | $P^*_{i,6}$, with i=n-4,….n-2
5. Prediction by ANN$_2$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, $P_{i,5}$, $P^*_{i,6}$, | $P^{**}_{i,7}$ with i=n-4,….n-2
6. Prediction by ANN$_1$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, $P_{i,5}$, $P^*_{i,6}$, $P^{**}_{i,7}$ | $P^{***}_{i,8}$ with i=n-4,….n-2
7. Prediction by ANN$_4$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, | $P^*_{i,5}$, with i=n-1,….n
8. Prediction by ANN$_3$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, $P^*_{i,5}$, | $P^{**}_{i,6}$, with i=n-1,….n
9. Prediction by ANN$_2$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, $P^*_{i,5}$, $P^{**}_{i,6}$, | $P^{***}_{i,7}$ with i=n-1,….n
10. Prediction by ANN$_1$ $P_{i,2}$, $P_{i,3}$, $P_{i,4}$, $P^*_{i,5}$, $P^{**}_{i,6}$, $P_{i,7***}$ | $P_{i,8****}$ with i=n-1,….n.

where the symbol * indicates that the value is an estimated value; the symbol ** indicates that the value is an estimated value obtained using as input one previous estimated value; the symbol *** indicates that the value is an estimated value obtained using as input two previous estimated values; the symbol **** indicates that the value is an estimated value obtained using as input three previous estimated values.
The procedure yields the predicted values of the unreliability index $R^{100}(t)$ at the various times of interests. The results of this phase are reported in Figure 4 and summarized in Table 5 for each of the 10 sequential steps of the procedure.







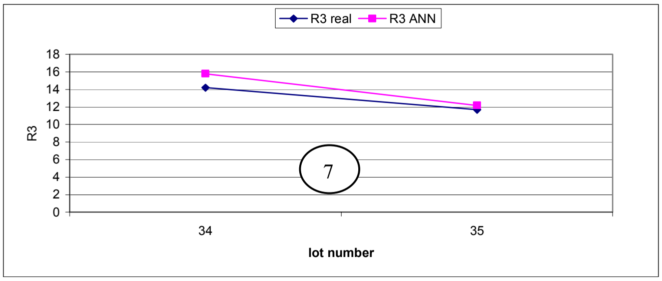


Figure 4 ANN prediction results relative to the various lots. Comparison with the real values.

5. CONCLUSION AND FUTURE DEVELOPMENTS

The satisfactory accuracy achieved on the test case considered in this work seems to demonstrate the feasibility of using ANN models for the estimation of the complete trend of evolution of the unreliability index $R^{100}(t)$, t=3,6,9,12 months. Using four ANN models, the estimates are obtained through a step by step procedure which makes use of the estimated values at previous times for estimating values in the future. As expected, the output predictions of those networks which receive in input a large number of previously estimated values are affected by greater errors than those of networks receiving in input exclusively, or almost exclusively, collected experimental data.

The procedure is somewhat cumbersome and may result in input errors if applied without any automatic support. Hence, the need of developing a procedure to be integrated in the software NEST, as support to the end user.

Further future developments regard the following aspects:

- the use, as input to successive training phases, of the estimated data appropriately weighed so as to control the approximation in the information due to the inaccuracies in the input estimated data;
- the use of noisy training data so as to expose the ANNs to the approximations inherent in the estimated data;
- the adaptive calibration of the neural models as new failure data become available (adaptative training).

Finally, this preliminary success of the approach makes us hope for good results in its application to the prediction of reliability during the design of new vehicles, or parts, on the basis of fault data relative to similar vehicles of a previous design generation. For this kind of application the knowledge base to be provided to the empirical neural models must be extended to include qualitative information in terms of linguistic judgments about the new products, as formulated by technicians and designers. In this context the most suitable methodology seems to be an integration between Neural Networks and Fuzzy Logic (Ref.5), a very efficient method for handling quantitative and qualitative data (Ref. 4).

REFERENCES

1. Jain L.C. e Martin N.M. eds., *Fusion of Neural Networks, Fuzzy Sets and Genetic Algorithms*, CRC Press 1999.
2. Muller B. e Reinhardt J., *Neural Networks – An introduction*, Springer- Verlag, New York, 1991.
3. Uhrig R.U. e Tsoukalas L.H., *Soft Computing Technologies in Nuclear Engineering Applications*, *Progress in Nuclear Energy*, vol. 34, No 1, pp. 13 – 75, 1999
4. Li-Xin Wang e M. Mendel, Generating Fuzzy Rules by Learning from Examples, *IEEE Trans. On Sys., Man and Cyber.*, **22**, No. 6, pp. 1414-1427, 1992..
5. Zadeh L., Fuzzy Sets, *Information and Control*, **8**, pp. 338-353, 1965.

BIOGRAPHIES

Marseguerra M., *Professor*
Dipartimento di Ingegneria Nucleare
Politecnico di Milano
Via Ponzio 34/3, 20133 Milano ITALIA

E-Mail: marzio.marseguerra@polimi.it

Full Professor, currently head of the laboratory of signal analysis and reactor physics. His current interests include the development of innovative techniques such as neural computing, genetic algorithms, biased Monte Carlo simulation, for applications in safeguard analysis, contaminant transport in porous media, reliability and safety analysis, and medical imaging.

Zio E., *Professor*
Dipartimento di Ingegneria Nucleare
Politecnico di Milano
Via Ponzio 34/3, 20133 Milano ITALIA

E-Mail: enrico.zio@polimi.it

Professor. His current interests include the development of innovative techniques such as neural computing, genetic algorithms, biased Monte Carlo simulation, for applications in safeguard analysis, contaminant transport in porous media, reliability and safety analysis.

Step	Employed model	Lot Number (i)	$R_i(3)$ estimate	$R_i(3)$ true	Relative error	$R_i(6)$ estimate	$R_i(6)$ true	Relative error	$R_i(9)$ estimate	$R_i(9)$ true	Relative error	$R_i(12)$ estimate	$R_i(12)$ True	Relative error
1	ANN_1	25										51.10	66.3	0.23
		26										50.90	47.7	-0.07
		27										77.00	67.0	-0.15
2	ANN_2	28							40.18	47.0	0.145			
		29							48.85	53.5	0.087			
		30							40.05	35.8	-0.119			
3	ANN_1	28										58.85	65.7	0.10
		29										64.82	76.6	0.15
		30										57.90	51.0	-0.14
4	ANN_3	31				27.12	26.9	-0.01						
		32				29.38	26.6	-0.10						
		33				25.02	22.8	-0.10						
5	ANN_2	31							36.47	41.0	0.110			
		32							38.52	38.7	0.005			
		33							34.31	21.2	-0.618			
6	ANN_1	31										53.18	50.7	-0.05
		32										57.19	56.6	-0.01
		33										51.12	33.1	-0.54
7	ANN_4	34	15.79	14.2	-0.112									
		35	12.19	11.7	-0.042									
8	ANN_3	34				29.41	20.5	-0.43						
		35				25.81	25.5	-0.01						
9	ANN_2	34							38.76	30.1	-0.288			
		35							34.19	37.1	0.078			
10	ANN_1	34										56.18	45.2	-0.24
		35										52.47	47.2	-0.11

Table 5 Application of procedure: operative phase results