# Architecture Optimization of a Neural Network Model to Identify Defects of Rotating Machines

**BOUFEDJ Imed**
University of Blaise Pascal - Clermont-Ferrand II, France, boufedjimed@gmail.com
**BOUZAOUIT Azzedine**
LRPCSI Laboratory of research, University of Skikda, Algeria, bouzaouit21@gmail.com
**CHAIB Rachid**
Mentouri University of Constantine, Algeria, r3chaib@yahoo.fr

**Abstract**
The objective of our work is to propose an approach for the development of a surveillance system against vibration faults of rotating machinery. This approach is based on the choice of a neural network architecture optimized, taking into account the activation functions used to know; the log type sigmoid function 'logsig' and the hyperbolic tangent sigmoid function type 'tansig' on one side, and another side, the type of learning algorithm. We considered eight '8' vibratory defects, the indicators chosen as variables of neural network inputs are: the spectral power density 'PSD' and the spectral energy density 'DSE'.

The error function used to compare the output of neural networks optimized architecture compared to the expected output is the mean square error 'MSE'.

Also by adopting a compromise between performance and the execution time. The number of inputs (independent variables) of the optimization process was optimized by Taguchi method.

The adapted neural model and the proposed architecture have increased the probability of network convergence in significant time. The results show the feasibility and efficiency of the neural model used to approximate the cutting process.

**Keywords**
optimization, vibration monitoring, neural networks, correlation coefficients, Levenberg-Marquardt, error gradient

## 1. Introduction

The evolution of the productive apparatus, with machines increasingly complex and automated, allied to the massive arrival of electronic boards on equipment hitherto electromechanical, led to a new perception of the challenges attached to the function maintenance. This function evolved much and continuous to evolve following analyses carried out during these last years and which relate to the trends of the budgets, manpower and the markets of maintenance on the one hand, and with its behavior in the various branches types of industry on the other hand.

These analyses show the major interest of maintenance and conditional maintenance in particular in an industrial context [1].

Indeed, among the tools of conditional maintenance, the vibratory analysis is that which knows today one of the most significant developments because of developments in the technologies in the data processing fields and the signal treatment.

The finality of this monitoring kind is intended to ensure the safety of the installation by avoiding important degradations by the alarms release [2], when the level of the vibrations reaches values considered to be excessive for the good performance or the latter integrity.

There exist various investigations techniques used in the field of the vibratory follow-up of the revolving machines.

Generally, any optimization problem comprises an essential part, it is the mathematical modelling.

It consists of three steps:

- Identification of decision variables: these are the parameters on which the user can act to change the system considered;

- Definition of a cost function (called objective function) to assess the state of system (such as, performance ...);
- Description of the constraints on decision variables.

The optimization problem is then to determine the decision variables leading to the best operating conditions of the system (which is to minimize or maximize the cost function), while respecting the constraints of use as defined, what is my contribution.

## 2. Description of the RSM Method

The determining neural network architecture is an indispensable dedicated to the success of our monitoring system and even its reliability and robustness. For this task and to highlight the investigation of the optimal architecture to our model, the method of the response surface RSM associated with the desirability function approach and the Taguchi method are used. The methodology of response surface is an empirical statistical technique, it is used for multiple regression analysis of quantitative data from statistically designed experiments in solving multivariate equations simultaneously. The graphical representation of these equations is called response surfaces; she used to describe the individual and cumulative effect of test variables on the response and the determination of the mutual interaction between the test variables and their effect on the response. The main objective of the RSM method is to determine the optimal response for system that satisfies the independent variables. The concept of response surface models a dependent variable Y, said response variable, depending on a number of independent variables (factors), $X_1$, $X_2$, ..., $X_k$, for analyzing the influence and interaction of these variables on the response. The model can thus be present for a response (Y) as follows:

$$Y = a_0 + \sum_{i=1}^{3} a_i X_i + \sum_{i=1}^{3} a_{ii} X_i^2 + \sum_{i<j}^{3} a_{ij} X_i X_j \tag{1}$$

where $Y$ is the observed response (cutting forces),

$a_0$ is the constant term,

$a_i$ represents the coefficients of the linear terms,

$a_{ij}$, $a_{ii}$ represent the terms of the interactions between variables and quadratic terms,

$X_i$ represent the independent variables.

The methodology for using the RMS method can be summarized in three essential steps [2]:

- The first, during which the number and levels of the parameters to be tested are selected, models will be proposed and discussed their validity.
- A second step, based on the use of graphs of the effects of the factors it will evaluate the effects of different neuronal variables on the performance of the response.
- Finally, in the last step, an optimization approach will be achieved through multi-objective optimization different performance of the neural network are: learning the correlation coefficient, the test correlation coefficient (validation), the overall correlation coefficient and the mean squared error (MSE).

## 3. Type of Neural Network Model and the Monitoring Concept Implemented

The system has two main phases.

The first is the data acquisition phase, the data are acquired as signals form, and they have taken over the monitoring and diagnostic operations.

While the second phase "decision phase" it concerns the treatment by neural networks (classification of defects). We propose in what follows to develop in detail these two phases that make up our surveillance system.

The neural model adopted in our approach of a surveillance system is a multilayer neural network Feed-forward type with the back-propagation algorithm. This type of network is also called "Perceptron type networks" [3, 4], it is considered a nonlinear static neural system, in which each neuron in one layer is connected to all neurons of the previous and follows layers (except for the input and output layers) and there are no connections between neurons of the same layer.

The information spread layer by layer without backtracking is possible. The effectiveness of this model is represented by its ability to predict the nonlinear behaviour of the synthesized values as well as its speed of convergence. Now the most difficult problem is how to obtain an adequate architecture of a neural network, or the method to find the optimal number of hidden layers and the number of neurons in each layer, and the right choice of initial values of the weight of network connections.

## 4. Development of an Optimized Neural Model

The approach implemented for setting developed an optimized neural model is illustrated and presented in Figure 1.

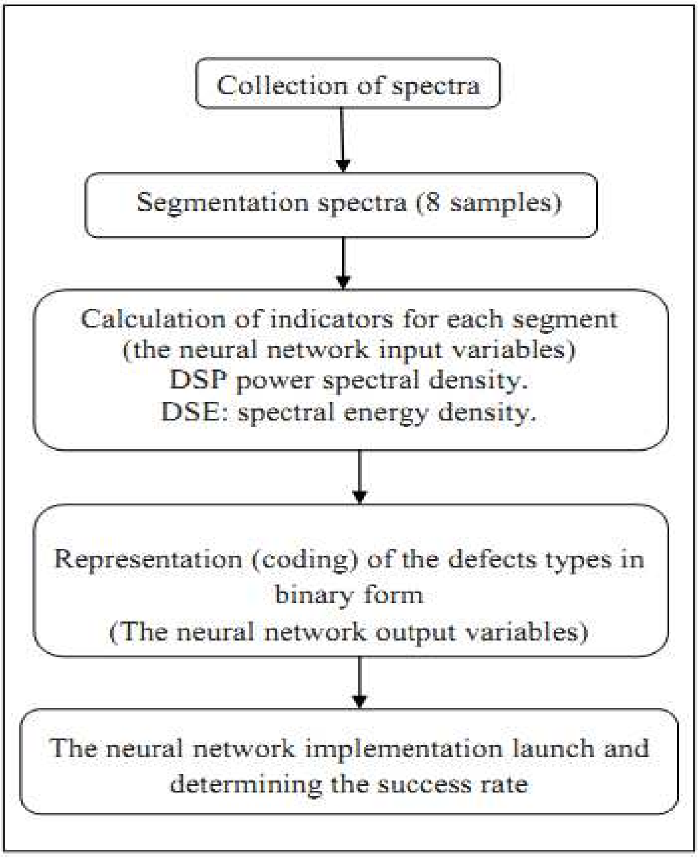



Fig. 1. The approach used in focus a neural model optimizes

In the present work to address the phase of multi-objective optimization, two orthogonal tables of Taguchi: L27 ($3^5$) and L9 ($3^3$) were adopted as scenario plans. The variables to consider and assigning the respective levels are indicated in (Table 1).

Table 1. Independent variables and their levels

| Level | A | B | C | D | E |
|---|---|---|---|---|---|
| | Number of hidden layers | Number of hidden layer neurons | $L_r$ | $M_c$ | Activation function |
| 1 | 1 | 1 | 0.1 | 0.1 | Logsig |
| 2 | 5 | 15 | 0.5 | 0.5 | Tansig |
| 3 | 10 | 30 | 0.9 | 0.9 | Purelin |

The results obtained by the simulation using neural networks are presented and summarized by Table 2, Table 3, Figure 2, and Figure 3.

Table 2. Simulation results by neural networks [5]: type of learning algorithm: back-propagation of error gradient associated with the Levenberg-Marquardt algorithm.

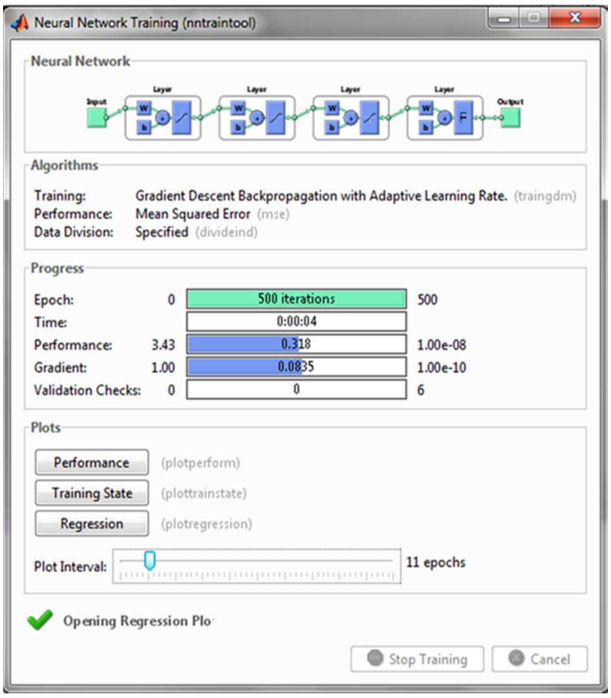
| Coded variables | | | $R_A$ | $R_V$ | $R_G$ | MSE |
|---|---|---|---|---|---|---|
| A | B | C | | | | |
| 1 | 1 | 1 | 0.6737 | 0.5305 | 0.61624 | 0.2581 |
| 1 | 2 | 2 | 0.95322 | 0.93276 | 0.94451 | 0.0621 |
| 1 | 3 | 3 | 0.88811 | 0.84322 | 0.86916 | 0.113868 |
| 2 | 1 | 2 | 0.47257 | 0.27791 | 0.39374 | 0.356837 |
| 2 | 2 | 3 | 0.87497 | 0.86506 | 0.85392 | 0.123867 |
| 2 | 3 | 1 | 0.95322 | 0.93276 | 0.94451 | 0.0721 |
| 3 | 1 | 3 | 0.68835 | 0.54184 | 0.62982 | 0.249523 |
| 3 | 2 | 1 | 0.94927 | 0.92739 | 0.93996 | 0.0821 |
| 3 | 3 | 2 | 0.13031 | 0.09811 | 0.11652 | 3.4195 |

Table 3. Simulation results by neural networks [5], type of learning algorithm: back-propagation of error gradient with "momentum"

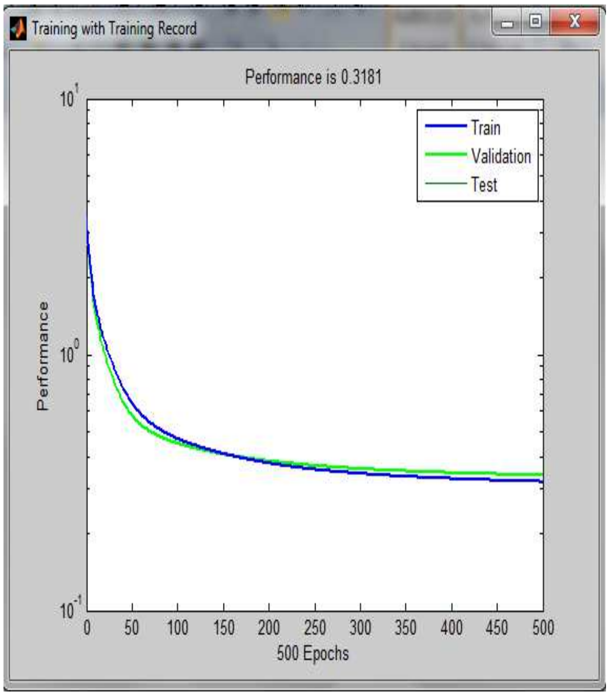
| Coded variables | | | | | $R_A$ | $R_V$ | $R_G$ | MSE |
|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | | | | |
| 1 | 1 | 1 | 1 | 1 | 0.28960 | 0.27251 | 0.283940 | 0.5327 |
| 1 | 1 | 1 | 1 | 2 | 0.30949 | 0.31745 | 0.312150 | 0.6106 |
| 1 | 1 | 1 | 1 | 3 | 0.30256 | 0.36813 | 0.320500 | 0.4831 |
| 1 | 2 | 2 | 2 | 1 | 0.30450 | 0.41910 | 0.337420 | 0.4728 |
| 1 | 2 | 2 | 2 | 2 | 0.41664 | 0.48180 | 0.435840 | 0.3972 |
| 1 | 2 | 2 | 2 | 3 | 0.50957 | 0.43766 | 0.486050 | 0.3457 |
| 1 | 3 | 3 | 3 | 1 | 0.38149 | 0.40570 | 0.388580 | 0.4132 |
| 1 | 3 | 3 | 3 | 2 | 0.46886 | 0.50794 | 0.481730 | 0.3641 |
| 1 | 3 | 3 | 3 | 3 | 0.27169 | 0.37085 | 0.304690 | 17.6309 |
| 2 | 1 | 2 | 3 | 1 | 0.32180 | 0.30053 | 0.314700 | 0.4818 |
| 2 | 1 | 2 | 3 | 2 | 0.19146 | 0.23312 | 0.205230 | 1.3916 |
| 2 | 1 | 2 | 3 | 3 | 0.22974 | 0.22794 | 0.229140 | 0.5382 |
| 2 | 2 | 3 | 1 | 1 | 0.45196 | 0.42124 | 0.441760 | 0.3668 |
| 2 | 2 | 3 | 1 | 2 | 0.53171 | 0.47300 | 0.512240 | 0.3330 |
| 2 | 2 | 3 | 1 | 3 | 0.01310 | 0.01310 | 0.013100 | 0.0131 |
| 2 | 3 | 1 | 2 | 1 | 0.49594 | 0.48886 | 0.493520 | 0.3486 |
| 2 | 3 | 1 | 2 | 2 | 0.56194 | 0.54430 | 0.556030 | 0.3189 |
| 2 | 3 | 1 | 2 | 3 | 0.01410 | 0.01230 | 0.013460 | 0.0131 |
| 3 | 1 | 3 | 2 | 1 | 0.19250 | 0.22784 | 0.204120 | 0.7058 |
| 3 | 1 | 3 | 2 | 2 | 0.20290 | 0.11527 | 0.173680 | 1.5771 |
| 3 | 1 | 3 | 2 | 3 | 0.23111 | 0.16809 | 0.210100 | 0.5822 |
| 3 | 2 | 1 | 3 | 1 | 0.45266 | 0.43320 | 0.446100 | 0.3664 |
| 3 | 2 | 1 | 3 | 2 | 0.45240 | 0.42548 | 0.443410 | 0.3662 |
| 3 | 2 | 1 | 3 | 3 | 0.01310 | 0.01310 | 0.013100 | 0.0131 |
| 3 | 3 | 2 | 1 | 1 | 0.50990 | 0.49232 | 0.504120 | 0.3427 |
| 3 | 3 | 2 | 1 | 2 | 0.53562 | 0.53562 | 0.525485 | 0.3312 |
| 3 | 3 | 2 | 1 | 3 | 0.01310 | 0.01310 | 0.013100 | 0.0131 |

with: $R_A$: Learning correlation coefficient;
$R_V$: Validation correlation coefficient;
$R_G$: Global correlation coefficients;
MSE: Average squared error;
Lr: Learning rate;
Mc: Constant moment**.**

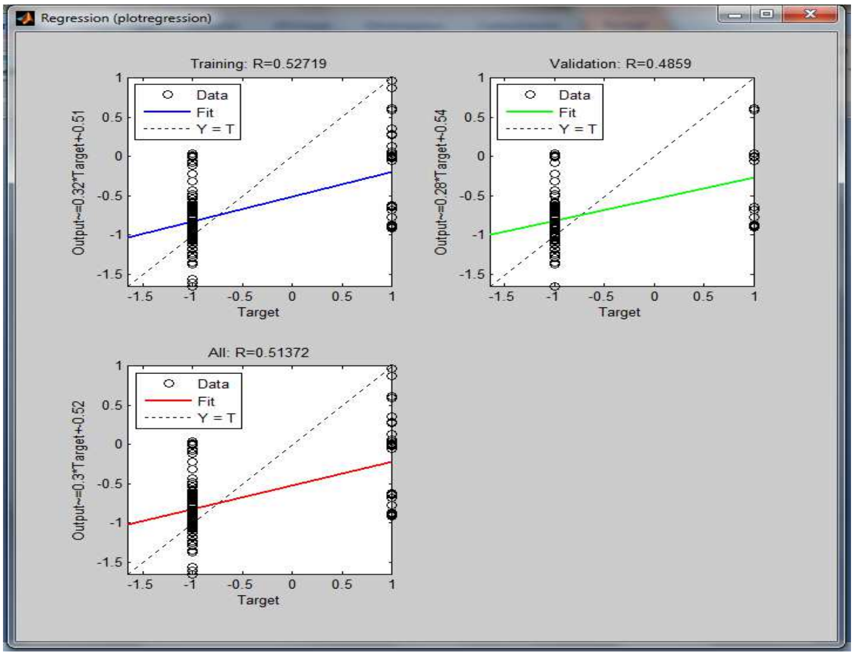
The optimization of a response or the research for a compromise between several answers is to define in the research field adjustment factors, to best meet the requirements in terms of response. In order, to optimize the architecture of the neural model, each parameter in the design (RSM) has been studied at three different levels. The choice of these levels for each variable is required by this design to explore the region near the optimum response surface. The optimization of responses by RSM helps to identify the combination of input variables parameters that optimize a single or set of response. A common optimization must meet the requirements for all the responses. The optimization of multiple responses is a method for a compromise between varieties of responses.




(a)

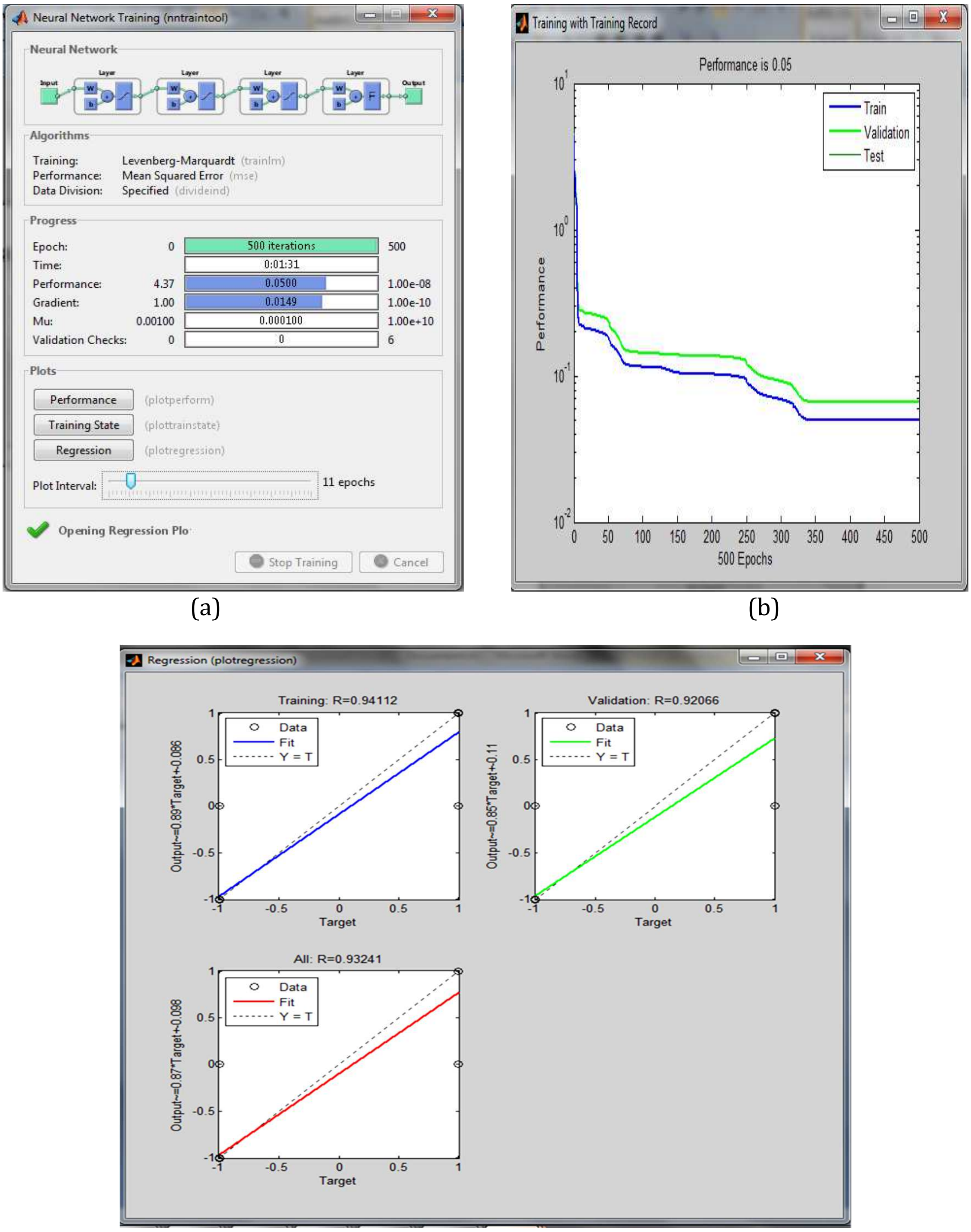



(b)




(c)

Fig. 2. Result of simulation with the optimal neuronal architecture 16-25-25-16 with a learning type algorithm: back-propagation error gradient with "momentum" (Lr = 0.5, Mc = 0.3) and 500 iterations




(a) (b)

(c)

Fig. 3. Result of simulation with the optimal neuronal architecture 16-12-12-12-16 with a learning algorithm type: error back-propagation gradient associated with the Levenberg-Marquardt and 500 iterations

Table 4 present the results of the optimization by RSM architecture of the neural network model. The optimal architecture obtained of the learning algorithm in neural network model type: Back propagation of error gradient with "momentum" [5] is:

(16-25-25-16) two hidden layers, with: $L_r = 0.5$; $M_c = 0.3$ and the Number of neurons in each hidden layer is 25.

Table 4. Learning algorithm type of back-propagation error gradient with "momentum"

| Reply | Objective | Global solution | | | | | Lower value | Target | Higher value | Predicted response | Individual desirability |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | A | B | C | D | E | | | | | |
| $R_A$ | Maximum | 2 | 25 | 0.5 | 0.3 | Tansig | 0.4 | 0.58 | 1 | 0.5071 | 1 |
| $R_V$ | Maximum | 2 | 25 | 0.5 | 0.3 | Tansig | 0.4 | 0.58 | 1 | 0.5274 | 1 |
| $R_G$ | Maximum | 2 | 25 | 0.5 | 0.3 | Tansig | 0.4 | 0.58 | 1 | 0.5101 | 1 |
| MSE | Maximum | 2 | 25 | 0.5 | 0.3 | Tansig | 0.1 | 0.2 | 0.8 | 0.2581 | 1 |

The combination of the activations functions is '*tansig*' / '*tansig*' / '*tansig*' / '*Purelin*' with $R_A$ = 0.5071; $R_V$ = 0.5274; $R_G$ = 0.5101 and MSE = 0.2581.

The optimal architecture obtained in the case of a neural network type of learning algorithm "Backpropagation gradient of error" associated with the Levenberg-Marquardt algorithm is given by the Table 5, it is:(16-12-12-12-16) three hidden layers, the number of neurons in each hidden layer is 12, and the combination of activations functions is: '*tansig*' / '*tansig*' / '*tansig*' / '*tansig*'/ '*Purelin*', $R_A$ = 0.9756; $R_V$ = 0.9679; $R_G$ = 0.9682 and MSE = 0.0296.

Table 5. Learning algorithm type of back-propagation error gradient associated with the Levenberg-Marquardt algorithm

| Reply | Objective | Global solution | | | Lower value | Target | Higher value | Predicted response | Individual desirability |
|---|---|---|---|---|---|---|---|---|---|
| | | A | B | E | | | | | |
| $R_A$ | Maximum | 3 | 12 | Tansig | 0.9 | 0.95 | 0.99 | 0.9756 | 1 |
| $R_V$ | Maximum | 3 | 12 | Tansig | 0.9 | 0.95 | 0.99 | 0.9679 | 1 |
| $R_G$ | Maximum | 3 | 12 | Tansig | 0.9 | 0.95 | 0.99 | 0.9682 | 1 |
| MSE | Maximum | 3 | 12 | Tansig | 0.06 | 0.07 | 0.08 | 0.0296 | 1 |

The results presented in Table 4 and Table 5 show that the optimal architecture of the neural network model that should be used for the classification process defects of rotating machines is: 16-12-12-12-16, the functions of activation are '*tansig*' / '*tansig*' / '*tansig*' / '*tansig*' / '*Purelin*'. The learning algorithm type is: Back-propagation of error gradient associated with the Levenberg-Marquardt algorithm.

The proposed model is considered more efficient ($R_A$ = 0.9756, $R_V$ = 0.9679, $R_G$ = 0.9682, MSE = 0.0296, the execution time t = 1.52 min), so it can be used for classification of rotating machines defects.

## 5. Classification of the Rotating Machines Defects

### 5.1. The neural network reliability criterion

To quantify the quality of predictions, three indicators are used in this case:

- The designed success rate of network.
- The mean square error MSE (the network performance).
- The overall correlation coefficient.

The success rate of neural network is the ratio between the number of detected and total defects.

In this context, we opted for an absolute uncertainty of ± 20%, so the success rate ∈ [80%; 120%].

To know: for the values '1' of defects coding chain, if this value equal $1^{\pm 0.2}$ the test is positive, and the neural model predicts defects successfully, the same for the codification of values '0', if this value becomes $0^{\pm 0.2}$, the test is also positive and the neural model predicts defects successfully.

### 5.2. Analysis and discussion of results

The results of the classification of defects according to the optimal architecture model (learning and test) are illustrated in Table 6 and Table 7.

Table 7 gives the testing results of the defaults classification from the optimal neural model.

Table 6. The learning (training) results of the classification from the optimal neural model

| | Defects codes considered | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| The outputs in binary code | **1.000** | **0.999** | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | **1.000** | **1.000** | 0.000 | 0.000 | -0.001 | -0.001 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 |
| | **1.000** | **0.999** | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 |
| | **1.000** | **1.000** | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | **1.000** | **1.001** | 0.000 | 0.000 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 |
| | **1.000** | **1.000** | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | **0.999** | **1.001** | 0.001 | -0.001 | 0.003 | 0.000 | -0.001 | 0.002 | 0.001 | 0.004 | 0.002 | -0.002 | 0.000 | 0.002 | 0.001 | -0.001 |
| | **1.001** | **1.000** | 0.000 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | -0.001 | 0.000 |
| | **0.999** | **1.000** | 0.000 | 0.001 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | **1.000** | **0.999** | -0.001 | 0.000 | -0.001 | 0.000 | 0.000 | -0.001 | -0.001 | -0.001 | -0.001 | 0.000 | 0.000 | -0.002 | 0.000 | 0.000 |
| | -0.001 | -0.001 | **1.000** | **1.000** | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | **1.000** | **1.000** | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | **0.999** | **0.999** | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | **0.867** | **0.867** | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | **1.000** | **1.000** | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | **1.000** | **1.000** | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | **0.933** | **0.933** | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.005 | 0.005 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | **0.967** | **0.967** |
| | 0.001 | -0.001 | -0.001 | 0.001 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | -0.001 | 0.001 | 0.000 | -0.001 | **0.999** | **1.001** |
| | -0.001 | 0.001 | 0.001 | -0.001 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.001 | -0.001 | 0.000 | 0.001 | **1.000** | **0.999** |

Table 7. Testing results of the classification from the optimal neural model

| | Defects codes considered | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| The outputs in binary code | **1.000** | **1.000** | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | **0.999** | **1.001** | 0.001 | -0.001 | 0.003 | 0.000 | -0.001 | 0.002 | 0.001 | 0.004 | 0.002 | -0.002 | 0.000 | 0.002 | 0.001 | -0.001 |
| | **1.001** | **1.000** | 0.000 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | 0.000 | 0.001 | 0.000 | 0.000 | -0.001 | 0.000 |
| | **0.999** | **1.000** | 0.000 | 0.001 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | **1.000** | **0.999** | -0.001 | 0.000 | -0.001 | 0.000 | 0.000 | -0.001 | -0.001 | -0.001 | -0.001 | 0.000 | 0.000 | -0.002 | 0.000 | 0.000 |
| | -0.001 | -0.001 | **1.000** | **1.000** | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.001 | 0.001 | **1.000** | **1.000** | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | **0.867** | **0.867** | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | **0.500** | **0.500** | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | **0.933** | **0.933** | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.005 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | **0.999** | **1.000** | 0.000 | 0.000 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.005 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | **0.933** | **0.933** |
| | 0.001 | -0.001 | -0.001 | 0.001 | -0.001 | 0.000 | 0.000 | 0.000 | 0.000 | -0.001 | -0.001 | 0.001 | 0.000 | -0.000 | **0.999** | **1.001** |
| | -0.001 | 0.001 | 0.001 | -0.001 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.001 | -0.001 | 0.000 | 0.001 | **1.000** | **0.999** |

It is clearly shows that the neural model used for the classification of defects assumed could not detect a single case (case in state in the test phase). Furthermore, we see the emergence of two values 0.5 instead of ‘0’. This value is above the maximum threshold limit values ($0^{\pm0.2}$) in code ‘0’, therefore, the test is normally negative.

However, the value 0.5 is small compared to the minimum threshold limit values in code 1 ($1^{\pm0.2}$), it does not affect the reliability of the classification, thus it is neglected.

This implies that our neural model has an overall success rate equal to 97.14% with:
- The success rate of learning is 100%.
- The test success rate is 93%.

## 6. Conclusion

The use of neuronal method poses certain difficulties. The main one was the optimization of the learning phase. Choosing the right architecture sometimes has a major drawback of neural models, since a wrong choice can lead to poor performance of the corresponding network.

The first attempts to resolve the architecture determination problem consisted in testing several networks with different architectures to achieve the desired performance.

For this reason, the model presented is not the only model. To this, the use of response surface method (RSM) associated with the desirability function (global and elementary) and the Taguchi method helped to effectively solve our multi-objective optimization problem, it is considered a highly complex mathematical problem.

The proposed optimization approach enables us to achieve the following results:

The most used is the architecture of the perceptron type of neural network multilayer (MLP).

The multi-objective optimization methodology based on RSM and the desirability function, appears to be a very consistent and robust tool, to allow the development of an optimal configuration for the MLP network.

It should be noted that the method was developed taking into account the performance of each configuration. Also by adopting a compromise between performance and the execution time. The number of inputs (independent variables) of the optimization process was optimized by Taguchi method.

The optimal restraint neural pattern is as follows:

- The architecture proposed is: 16-12-12-12-16 three hidden layers with 12 neurons.
- The combination of activations functions used is: '*tansig*' / '*tansig*' / '*tansig*' / '*tansig*' / '*Purelin*'.

This choice is motivated by the binary variables, we use to locate the fault (rapidly approaches 1 if there is default, and to 0 if there is none).

The learning algorithm is of type: Back-propagation of error gradient associated with the Levenberg-Marquardt algorithm. This type of algorithm interpolates the Gauss-Newton algorithm and gradient descent method. It is more stable than that of Gauss-Newton; he accomplished the same optimal solution if started far from a minimum.

The learning parameters are: Maximum number of iterations (Epochs) = 500; Mean square error MSE = $10^{-8}$; Minimum gradient = $10^{-10}$.

The neural network allows, with the same data available to make a more accurate approximation that multiple linear regression.

## References

1. Boulenger, A., Pachaud, C. (2013): *Analyse vibratoire en maintenance*. Dunod, 3rd ed., EAN13: 9782100596959
2. Boufedj, I., Bouzaouit, A., Bennis, O. (2013): *Classification of the revolving machines defects by a neuronal model*. IEEE, (Int. Conf. on) Control Decision and Information Technologies (CoDIT), p. 899-903, ISBN 978-1-4673-5549-0, DOI: 10.1109/CoDIT.2013.6689662, Hammamet, Tunisia
3. Matlab (2007): *Math Works, Inc, Matlab R2007a*
4. Chong, E.K.P, Sak, S.H. (2001): *An introduction to optimisation*. Wiley Inter-Science, ISBN 0 471 39126 3
5. Parizeau, M. (2006): *Réseaux de neurones*. 'Gif-21140 et Gif-64326'. Université Laval

Received in October 2016