Proceedings of ASME Turbo Expo 2011
GT2011
June 6-10, 2011, Vancouver, British Columbia, Canada

GT2011-46752

NEURAL NETWORKS FOR GAS TURBINE FAULT IDENTIFICATION: MULTILAYER PERCEPTRON OR RADIAL BASIS NETWORK?

Igor Loboda
National Polytechnic Institute,
School of Mechanical and Electrical Engineering,
Santa Ana Street, 1000, Mexico City,
Federal District,
Post Office 04430, Mexico
Telephone and fax (52-55)5656-2058
E-mail: iloboda@ipn.mx

Yakov Feldshteyn
Compressor Controls Corporation,
4725 121 Street, Des Moines, Iowa,
50323, USA
Telephone (515)251-2782
E-mail: yfeldshteyn@cccglobal.com

Volodymyr Ponomaryov
National Polytechnic Institute,
School of Mechanical and Electrical Engineering,
Santa Ana Street, 1000, Mexico City,
Federal District,
Post Office 04430, Mexico
Telephone and fax (52-55)5656-2058
E-mail: vponomar@ipn.mx

ABSTRACT

Efficiency of gas turbine condition monitoring systems depends on quality of diagnostic analysis at all its stages such as feature extraction (from raw input data), fault detection, fault identification, and prognosis. Fault identification algorithms based on the gas path analysis may be considered as an important and sophisticated component of these systems. These algorithms widely use pattern recognition techniques, mostly different artificial neural networks. In order to choose the best technique, the present paper compares two network types: a multilayer perceptron and a radial basis network. The first network is being commonly applied to recognize gas turbine faults. However, some studies note high recognition capabilities of the second network.

For the purpose of the comparison, both networks were included into a special testing procedure that computes for each network the true positive rate that is the probability of a correct diagnosis. Networks were first tuned and then compared using this criterion. Same procedure input data were fed to both networks during the comparison. However, to draw firm conclusions on the networks' applicability, comparative calculations were repeated with different variations of these data. In particular, two engines that differ in an application and gas path structure were chosen as a test case. By way of summing up comparison results, the conclusion is that the radial basis network is a little more accurate than the perceptron, however the former needs much more available computer memory and computation time.

1. INTRODUCTION

A gas turbine can be considered as a very complex and therefore potentially unreliable system. In order to keep its reliability high and reduce maintenance costs, many advanced condition monitoring systems have been developed in the recent decades. Design and use of these systems were spurred by the progress in instrumentation, communication techniques, and computer technology.

Success of gas turbine monitoring and condition based maintenance is contingent on perfection of monitoring software and, in particular, strongly depends on degree to which engine critical parts are covered, as well as on the accuracy of diagnostic decisions. For example, even a brief description of an advanced power plant monitoring system given in [1] shows that the system is intended to cover all principal gas turbine subsystems. To enhance overall monitoring efficiency, all subsystems should be diagnosed as accurately as possible. For every subsystem, a diagnostic process can be split into four general and relatively independent stages: feature extraction, fault detection, fault identification, and prognosis, and each stage may be presented by specific algorithms. Thus, the monitoring system includes a lot of different algorithms and every algorithm needs to be optimized. The present paper addresses a particular optimization problem: selection of the best technique for gas path fault identification.

In a total monitoring system the gas path fault identification can be considered as an important and sufficiently complex component. It is based on gas path measurements (control variables and monitoring variables), gas turbine models, and pattern recognition techniques.

Since abrupt faults and gradual deterioration mechanisms affect gas turbine component performance, measured gas path variables (pressure, temperature, rotation speed, fuel consumption, etc.) vary accordingly. Thus, gas path measurements can be employed to identify gas path problems.

Many different types of gas path performance degradation are known, such as foreign object damage, fouling, tip rubs, seal wear, and erosion. Their detailed descriptions can be found, for instance, in [1,2]. With the use of gas path measurements, malfunctions of control and measurement subsystems [3] can be localized as well.

In order to distinguish between various degradation types, we need to know their influence on monitored gas path variables. For some intensive and practically permanent deterioration mechanisms, it is possible to estimate such influence through analysis of recorded data. For example, the most common cause of stationary gas turbines' deterioration is compressor fouling; its impact on gas turbine performance is well described, see [2,4,5]. In aircraft engines, especially helicopter engines, compressor airfoils are often affected by erosion because of dust and sand in the sucked air [2].

Nevertheless, the majority of real gas turbine faults occur rarely or they are not so severe as to describe them using data recorded in field conditions. Another option – physical fault simulation in a test cell – seems unfeasible because of very high costs of such experiments. That is why mathematical models are involved. They connect faults of different engine components with the corresponding changes of monitored variables, assisting in this way with fault description. These fault simulation tools are generally nonlinear thermodynamic models based on mass, energy, and momentum conservation laws. Such models require detailed knowledge of the gas turbines under analysis and can be classified as physics-based models. The thermodynamic models are generally complex software. Such sophisticated models have been used in gas turbine diagnostics since the works of Saravanamuttoo H.I.H. (see, for example, [4]). Nowadays, the use of these models is a standard practice.

The fault identification based on gas path measurements presents a complicated pattern recognition problem because of residual model inadequacy, normal measurement noise, and possible sensor malfunctions. A lot of applications of pattern recognition techniques for gas turbine diagnosis can be found in literature including, but not limited to, Artificial Neural Networks (ANNs) [6-12], Genetic Algorithms [6], Support Vector Machines [7], Correspondence and Discrimination Analysis [13], and Bayesian Approach [10,12].

Among the mentioned techniques the ANNs are most widespread. In turn, the multilayer perceptron (MLP), known as a backpropagation network, is by far the most commonly applied for gas turbine fault recognition [6,14]. Following this trend, we also employed the MLP for the fault classification needs [11,12].

On the other hand, some researchers, for example, S. Sampath and R. Singh [6], note a high recognition capability of the Radial Basis Network (RBN). Additionally, our preliminary study [15] indicates that the RBN can be a little bit more efficient than the perceptron. The above explanations encouraged us to conduct a comprehensive comparative study of these two networks, MLP and RBN.

In this paper, the selected networks are thoroughly compared in order to determine what network better performs fault recognition in real gas turbine monitoring systems. To this end, both networks were embedded in a special testing procedure. The procedure embraces a) computing the monitored variables' deviations induced by the faults embedded in a gas turbine thermodynamic model, b) fault classification formation, c) training two selected networks on the classification data used here as a learning set, and d) parallel application of the trained networks (each network makes its particular diagnosis) to new fault data (validation set). Numerous diagnosis cycles with validation set data enable computing probabilities of correct and wrong diagnoses for each compared network. These probabilities are networks' diagnostic performances and criteria to choose the best network.

The described procedure was developed in Matlab. This language of technical computing developed by the MathWorks, Inc. offers convenient tools for experimenting with different neural networks. In particular, it contains a neural networks toolbox that simplifies network creation, training, and use, thus assisting in effective software development. MATLAB also allows choosing between various training functions and calculation options.

The probabilities computed by the testing procedure contain random errors caused by the stochastic nature of the fault simulation and network training. That is why close attention was paid to computational precision. In particular, probability distribution was analyzed and uncertainty intervals were estimated.

To draw firm conclusions on the applicability of the analyzed networks, two different engines were chosen as a test case: an industrial gas turbine and a turbofan. Additionally, for each engine presented by its thermodynamic model, the following factors are varied during the comparison: operating mode, fault classification type, network structure, and some others. All factors are varied independently and therefore produce a great total number of variations of input data for determining the performances of the networks. In this way, the precision analysis and networks comparison under different conditions promise reliable conclusions.

The present paper is structured as follows. A general description of the chosen recognition techniques, MLP and RBN, is given in section 2 followed by the methodology of networks-based fault identification described in section 3. Next, we present the network comparison results for the industrial gas turbine and aircraft engine in sections 4 and 5 correspondingly. Finally, in section 6, these results are discussed and generalized.

NOMENCLATURE

ANN	Artificial neural network
MLP	Multilayer perceptron
GT1, GT2	First and second analyzed gas turbines
RBN	Radial basis network
a_Y	Maximum deviation error
D	Fault class
d	Diagnosis
e	Network error
f_1, f_2	Layer transfer functions
$\overline{P}$	Mean probability of correct diagnoses
S1, S2	Numbers of layer nodes
$\vec{a}_1,\ \vec{a}_2$	Layer output vectors
$\vec{b}_1,\ \vec{b}_2$	Bias vectors

$\vec{p}$	Network input vector
$\vec{U}$	Vector of operating conditions (control variables and ambient conditions)
$\vec{w}$, W_1, W_2	Vector and matrices of weight coefficients
$\vec{Y}$	Vector of monitored gas path variables
$\vec{Z}^*$	Vector of normalized deviations
$\mathbf{Z_L}$, $\mathbf{Z_V}$	Learning and validation sets
ΔG	Flow parameter
$\Delta\eta$	Efficiency parameter
ε	Random error of the normalized deviation
$\vec{\delta Y}$	Vector of relative deviations
$\vec{\Theta}$	Vector of fault parameters

Subscripts and superscripts

*	Measured value
0	Baseline value
i	Index of gas path variable *Y*
j,l	Fault class indices
q	Number of fault classes
av	Average

2. NEURAL NETWORKS COMPARED

Artificial neural networks are fast growing pattern recognition and approximation techniques in many fields of engineering applications, including condition monitoring [1]. Foundations of the multilayer perceptron can be found practically in every book on neural networks or classification, for example, in [16,17]. Information on the radial basis networks is widely available too. Therefore, two subsections below include only a brief description of the networks, which is necessary to gain better understanding of the present paper.

Multilayer perceptron

The ANN and, in particular, the MLP can be classified as a typical black-box model because it is observed only through its input and output without any knowledge of internal operation. During network learning on the known pairs of input and output (target) vectors, unknown network coefficients change in the direction of decreasing a mean squared difference (error) *e* between the target and the network output. To enhance network flexibility, one or more hidden layers can be incorporated in addition to the input and output layers.

The MLP is a feed-forward network in which signals propagate through the network from its input to the output with no feedback. Figure 1 helps to better understand perceptron operation. For each hidden layer neuron, the sum of inputs of a vector $\vec{p}$ multiplied by the weight coefficients of a matrix W_1 is computed and the corresponding bias (element of a vector $\vec{b}_1$) is added. The neuron input obtained in such a way is transformed by a hidden layer transfer function f_1 into a neuron output, element of a vector $\vec{a}_1$. The computation is reiterated for all hidden layer neurons. The same calculations are then performed for the output layer considering $\vec{a}_1$ as an input vector.

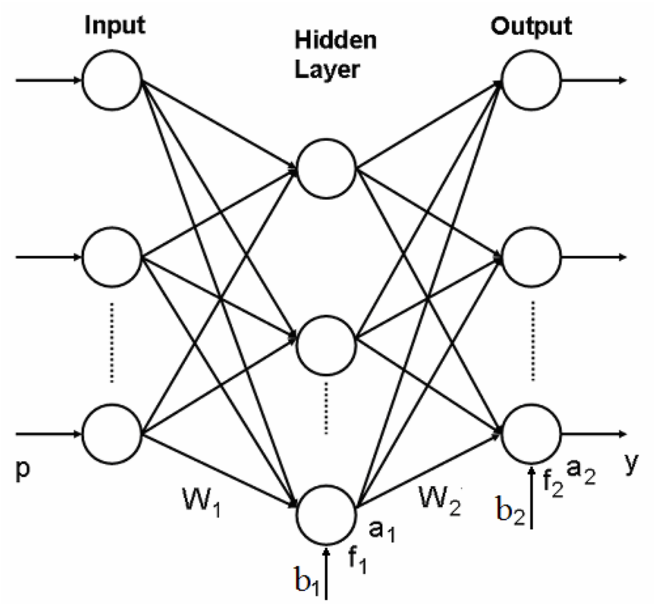


FIG.1. PERCEPTRON DIAGRAM

The MLP has become widespread since a back-propagation training algorithm was proposed. In this algorithm the error between the target and the actual network output is propagated backwards to correct network coefficients. If an incremental training mode is used, all unknown coefficients (weights and biases) are corrected after presentation of every pair of input and target vectors. This elemental corrective calculation (epoch) is repeated successively for all available input and target vectors of the learning set. Since one such series of epochs is not usually sufficient to reach a global minimum of an error function $e(W_1, \vec{b}_1, W_2, \vec{b}_2)$, the series is repeated many times with the same data of the learning set.

Within the bounds of the paper, the MLP training was performed in another mode called a batch mode because the Matlab training functions operate just in this mode. Such training means that a total error between all targets and network outputs of the learning set is computed and used to change the coefficients in every epoch. Generally, this mode is faster because a total number of epochs is drastically smaller.

The back-propagation algorithm requires that all layer transfer functions be differentiable. Usually, they are of a tan-sigmoid, log-sigmoid, or linear type.

Let us consider now the other analyzed network, RBN.

Radial basis network

A radial basis function neuron differs from a perceptron neuron. The input *n* to a radial basis transfer function is a distance between a weight vector $\vec{w}$ and an input vector $\vec{p}$, multiplied by a bias *b* i.e. $n = \left\| \vec{w} - \vec{p} \right\| b$, where $\| \ \|$ denotes the Euclidean norm. The transfer function computes a neuron's output as $a = \exp(-n^2)$. When the distance is 0, this function has a maximum value *a*=1 and the function decreases when the distance increases. The bias *b* allows changing the neuron sensitivity.

Figure 2 illustrates operation of a whole network based on the radial basis functions. The RBN presented here includes two layers: a hidden radial basis layer of S1 nodes and an output linear layer (S2 nodes). The input vector $\vec{p}$ and an input weight matrix W_1 produce a distance vector that includes S1 elements. The bias vector $\vec{b}_1$ scales these distances and a radial basis transfer function converts the scaled distances into a hidden layer output vector $\vec{a}_1$. The output layer works similarly to a usual perceptron output layer with a linear transfer function producing an output vector $\vec{a}_2$ of S2 elements. As in the case of the perceptron, unknown coefficients to be determined during the training are elements of the matrices W_1 and W_2 as well as the vectors $\vec{b}_1$ and $\vec{b}_2$.

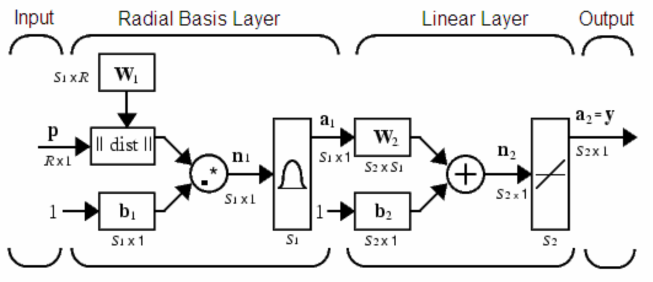


FIG.2. RBN DIAGRAM

In a basic variation of the RBN algorithm, every input vector of the learning set forms a new radial basis neuron. Since the set size equals the radial basis layer size and the latter is limited by available computer time and memory, such a network cannot operate with a large learning set. In contrast to the basic variation, the efficient variation that was used operates in an iterative mode, adding one neuron at a time to the radial basis layer. At every iteration, the input vector that results in the lowest network error is used to create this new neuron. The iterative calculation completes when a network error decreases below an error goal or a maximum neuron number is reached. Although the efficient variation of the RBN can operate with a larger learning set than the basic variation, the maximal volume of this set is usually smaller than the volume of the perceptron's set. This is explained by greater computer memory required for the RBN.

The radial basis network tends to have more neurons than a comparable perceptron. This is explained by the fact that the sigmoid neurons can cover a large region of the input space, while the radial basis neurons only cover a relatively small region. The larger the input region, the more such neurons are required. Although the RBN usually needs more neurons than the perceptron, the former can often be trained within the time necessary for the perceptron. It is noted below in section 3 that in our case the learning set comprises thousands of input and target vectors (patterns). This is favorable for the RBF because this network works better when many training patterns are available [18].

Network validation

The network should follow only general systematic dependencies between simulated variables but not take into account random measurement errors of every input. That is why the errors between the network's outputs and targets decrease but do not completely disappear during the training process.

A neural network capability to generalize data is commonly verified on a new portion of data called a validation set. If the network describes learning data well but loses accuracy on validation data it is an indication of an overlearning (overfitting) effect. The network takes into consideration random peculiarities of the learning set and therefore loses its capability to generalize validation data. In this case we need to stop the training when the overlearning begins. If a number of training epochs is sufficiently large and the overlearning does not appear, the difference between the network performances obtained on the learning and validation sets will be small. The network's tendency to overfit the data depends on the proportion between a volume of the training set and a number of unknown network coefficients. Since excessive hidden layer neurons can provoke the overfitting, the number of these neurons must be optimized.

Let us now consider a total gas turbine diagnosis process that involves two networks described above into recognizing gas path faults.

3. FAULT RECOGNITION PROCESS

A general recognition process can be divided into three principal stages: forming the classification, decision making, and estimating the recognition accuracy. Using monitored gas path variables directly is not convenient to form necessary multidimensional classification space. Although monitored gas path variables reflect the influence of gas turbine degradation, it is difficult to discriminate the degradation effects from the changes that are due to different operating modes when we analyze the variables themselves. That is why to form a fault classification and to recognize faults, a preliminary procedure of computing deviations is fulfilled. These deviations are practically free of the mode influence and therefore they can be good degradation indicators.

Deviations

The deviation, also called a "delta," is defined as a difference between measured and baseline values of a monitored variable. In a relative form the deviation δY^* of the variable *Y* is computed according to an expression

$$\delta Y^* = \frac{Y^* - Y_0(\vec{U})}{Y_0(\vec{U})}, \qquad (1)$$

where Y^* and Y_0 are measured and baseline values, respectively. As an engine baseline depends on operating conditions (atmospheric conditions and engine control variables all together) united in a vector $\vec{U}$, the baseline value is given as a function $Y_0(\vec{U})$, which is usually called a baseline function.

Due to measurement inaccuracy and baseline function uncertainty, the deviations always contain random errors and their level is individual for every monitored variable. If we denote by a_Y a maximum amplitude of the errors in the deviation δY^*, the corresponding normalized deviation will be written down as

$$Z^* = \delta Y^* / a_Y \,. \tag{2}$$

As a result of the normalization, random errors of the deviation Z^* (but not the deviation itself) of every monitored variable are limited by the same value of one. That is why a description of fault classes in a space of these deviations is more uniform and fault diagnosis becomes simpler.

As mentioned in the introduction, the thermodynamic model is widely used to simulate different gas turbine degradation mechanisms. The model relates the gas path variables with the operating conditions and with special fault parameters $\vec{\Theta}$; therefore it can be given by the following expression

$$\vec{Y} = F(\vec{U}, \vec{\Theta}) \,. \tag{3}$$

The fault parameters are used to displace performance maps of engine components that allow simulating the gas path faults. Let us denote a random error of the normalized deviation by a scalar ε, the fault parameters corresponding to a healthy engine by a vector $\vec{\Theta}_0$, and faults embedded in the model by a vector $\Delta\vec{\Theta}$. Given these designations and expression (1), formula (2) is converted to an expression

$$Z^* = \frac{Y(\vec{U}, \vec{\Theta}_0 + \Delta\vec{\Theta}) - Y(\vec{U}, \vec{\Theta}_0)}{Y(\vec{U}, \vec{\Theta}_0) a_Y} + \varepsilon \,. \tag{4}$$

Thus, the normalized deviations are computed by means of the thermodynamic model in accordance with the above formula. The deviations of all *m* monitored variables compose a vector $\vec{Z}^*$ that is a point in the diagnostic space and a pattern to be recognized. Fault classification is formed from these patterns as described below.

Fault classification

Since an existing variety of the gas path faults is too great to distinguish between all of them, the faults are grouped into limited number of classes. The hypothesis is usually accepted in gas turbine diagnostics that an engine state *D* can belong to only one of *q* previously determined fault classes $D_1, D_2, ..., D_q$. Within the scope of the paper, each class is constructed from numerous patterns in a m-dimensional space of deviations $\vec{Z}^*$.

The principle to group the faults into classes is commonly related with gas path components in which the faults occur. It is believed that two fault parameters of one component, usually a flow parameter ΔG and an efficiency parameter $\Delta\eta$, are sufficient to describe all faults of this component. Two types of classes and two corresponding classification variations are dealt with: a single fault type and a multiple fault type. The single fault class has one fault parameter, variation of which allows simulating faults of variable severity. In contrast, the multiple fault class has two independently varied parameters $\Delta\Theta$ of the same component and unites all possible faults of the component. In the present paper one multiple and two single classes are formed for each engine component.

Variations $\Delta\Theta$ are given by the uniform distribution within the interval [0, 5%], while random errors ε are generated according to the normal distribution. An optimal number of simulated patterns per class was determined in [19]. As a rule, it equals 1000. A whole classification unites patterns of particular classes and all these patterns are employed to train the neural networks chosen. From this point of view a totality of classification's patterns $\mathbf{Z_L}$ can be called a learning set. The trained network, MLP or RBN, is ready to recognize the faults i.e. to make diagnostic decisions.

Diagnostic decision

A nomenclature of possible diagnoses $d_1, d_2, ..., d_q$ corresponds to the accepted classification $D_1, D_2, ..., D_q$. To make a diagnosis *d* for a current pattern $\vec{Z}^*$, each computed network's output y_j is considered as a measure of pattern membership in the class D_j. A decision rule

$$d = d_l \text{, if } \mathrm{y}_l = \max(y_1, y_2, ..., y_q) \tag{5}$$

is then applied.

Since many negative factors affect the diagnosis process, the diagnosis *d* will not always be correct and it should be accompanied by any confidence assessment.

Diagnostic accuracy

Although neither MLP, nor RBN compute accuracy estimations for every diagnosis, it is possible to obtain averaged estimations by means of network testing on numerous input and output vectors. To check the overlearning effect, the network trained on one portion of the random data should be tested on another portion. Consequently, to verify the MLP and RBN, in addition to the learning set, we need one more set. Let us call it a validation set $\mathbf{Z_V}$. This set is formed like the learning set with only exception: other series of random numbers are generated to specify the fault severity and the deviation errors ε. A special number called the seed is used to change a random number series.

During the statistical testing the proper class of every pattern is known a priori. That is why we can compute probabilities of correct and incorrect diagnoses for all fault classes and form the so-called confusion matrix. Each diagonal element of the matrix known as the true positive rate is a probability of correct diagnosis of the corresponding class. A quantity $\overline{P}$ that is computed as a mean number of the diagonal elements can be called as an overall true positive rate. It characterizes a total level of engine diagnosability and will be applied as a criterion to compare the analysed networks, MLP and RBN.

On the basis of the described above approach to gas path fault recognition, a special procedure to test these networks has been developed. By means of this procedure the MLP and RBN were tested and compared on the data of two different engines. The first is a two-shaft free turbine driver for natural gas compressors (referred to as GT1). The second engine, referred to as GT2, is a three-shaft high bypass ratio turbofan.

The networks' comparison conditions and results are given separately for each engine in the two sections below.

4. NETWORKS' COMPARISON: THE CASE OF THE INDUSTRIAL GAS TURBINE (GT1)

Comparison conditions

In all comparative calculations executed for GT1, a structure of monitored gas path variables is constant and includes 6 items. The structure corresponds to a standard GT1 measurement system. Fault simulation during the comparison is based on the variation of nine fault parameters. The monitored variables and fault parameters are described in Table A1 and Table A2 of Appendix I. Further details on the variables and parameters of GT1 can be found in [11]. The structure of these quantities does not change. The other computational conditions (factors) described below are varying.

A. Operating modes. Two gas turbine operating modes, Mode 1 and Mode 2, are analyzed. They are close to engine maximal and idle regimes and are set by different high pressure rotor speeds under standard ambient conditions.

B. Classification variations. Two classification variations are considered. The first variation incorporates 9 single classes with 700 patterns in each class. This number of patterns is less than the recommended value of 1000, because otherwise available computer memory is not sufficient for the RBN. The second variation includes 4 multiple classes corresponding to the principal GT1 components (compressor, combustion chamber, compressor turbine, free turbine). Each multiple class is presented by 1000 patterns.

C. Number of nodes in the hidden layer. According to the structures of monitored variables and fault classes, both networks have 6 nodes on the input layer and 9 nodes (single type classification) or 4 nodes (multiple type classification) on the output layer. All these parameters do not change. As to the hidden layer, two different nodes numbers, basic and enlarged, are considered in order to better understand the influence of a hidden layer size on networks' performances. For the MLP these numbers are 12 and 27, while for the RBN they are 24 and 90.

Comparison results

Each of three described above factors (operating mode, classification variation, and hidden layer node number) has two values. They are varied independently therefore a total number of considered comparison cases is eight. Each case implies a new variant of the compared networks because when any factor changes, the network structure is varied accordingly. In other words, eight variants of the perceptron are compared one by one with the corresponding eight RBN's variants.

For all these cases calculations were first executed with a constant seed (quantity that determines a consequence of random numbers). The results are shown in Table 1. It can be seen for the basic node numbers (cases 1, 3, 5, and 7) that the perceptron is a little better. However, for the enlarged node numbers the RBN demonstrates slightly higher diagnosis accuracy compared with the perceptron: we can see the corresponding increment 0.0032-0.0050 (0.32-0.50%) of the mean probability $\overline{P}$.

During the experimentation with different seeds, we found that random spread of the probability $\overline{P}$ can be larger than the observed difference between the networks. Consequently, the above comparison can only be considered as preliminary. One way to reduce the errors is to perform numerous comparative calculations with different seeds and to average the results. To plan such numeric experiments we need the information on the computational time required.

Table 1. Networks' accuracy comparison on GT1 data (probabilities $\overline{P}$, Seed 1)

Class type	Network type	Mode 1		Mode 2	
		Basic node numbers	Enlarged node numbers	Basic node numbers	Enlarged node numbers
Singular		Case 1	Case 2	Case 3	Case 4
	MLP	0.8129	0.8135	0.8025	0.8027
	RBN	0.8110	0.8167	0.7986	0.8078
Multiple		Case 5	Case 6	Case 7	Case 8
	MLP	0.8755	0.8760	0.8665	0.8650
	RBN	0.8733	0.8805	0.8662	0.8700

In order to make as many calculations as possible and for the sake of a better networks' comparison, computational time was estimated for the same cases as in Table 1. It was found that the time does not depend on an engine operating mode that is why we will analyze below only Mode 1 of GT1 and the corresponding cases 1, 2, 5, and 6. Execution time values are presented in Table 2 separately for these cases and for network's training and diagnosis stages. First of all, a huge time difference between the training and diagnosis is striking. Nevertheless, such difference is easily explainable: one diagnosis cycle implies one network computation, while the training includes millions of such computations. Comparing time values of the analyzed networks for each case presented in Table 2, one can see that the perceptron is always considerably faster. This difference is not perhaps so important at the diagnosis stage because of very short execution time. However, to tailor a neural network to a specific diagnostic application, we may need to repeat the training several thousand times. In this case, longer training time of the RBN can be a serious problem.

Table 2. Networks' execution time comparison on GT1 data (the time is given in seconds for Toshiba Tecra: Intel Core 2 Duo CPU, 2.53GHz, and 3.0 GB of RAM)

Class type	Network type	Training stage		Diagnosis stage	
		Basic node numbers	Enlarged node numbers	Basic node numbers	Enlarged node numbers
Singular		Case 1	Case 2	Case 1	Case 2
	MLP	10.3	14.9	7.5×10^{-6}	10.0×10^{-6}
	RBN	72.6	216.2	22.4×10^{-6}	69.4×10^{-6}
Multiple		Case 5	Case 6	Case 5	Case 6
	MLP	5.1	8.4	7.7×10^{-6}	11.8×10^{-6}
	RBN	26.5	79.2	19.5×10^{-6}	66.3×10^{-6}

As mentioned before, averaging the probabilities obtained by calculations with different seeds will improve the precision of probability estimations. It is clear that a number *N* of such calculations should be as large as possible but it is limited by available computational time. As the network training occupies almost all computer time, the training stage time values from Table 2 helped us to choose the number *N*. This number was assigned to 100 for all the cases considered before and each comparative calculation of Table 1 was repeated 100 times, each time with a new seed.

The resulting probabilities $\overline{P}_{av}$ averaged for 100 calculations are given in Table 3, which has the same format as Table 1. It can be seen that the enlargement of the hidden layer results in a probability increment for all considered cases. To estimate a level of this effect, a mean probability increment due to the node number increase was calculated separately for each network. The increment is 0.0027 for the MLP and 0.0049 for the RBN. Thus, we can state that the node number increasing is more favorable for the RBN; however, the observed effect is moderate even for this network.

It becomes clear that a potential advantage of the RBN will be more visible for an enlarged hidden layer. Let us therefore compare MLP and RBN probabilities just in the columns "Enlarged node numbers" of Table 3. We can see that both networks are practically equal for the singular classes, while the RBN is a bit better for the multiple classes. On average for both operating modes and both classification types presented in the table, the RBN gains 0.0009 (0.09%) only. Thus, more precise data of Table 3 allow us to conclude that an advantage of the radial basis network is almost negligible, at least in a diagnostic application to GT1. Nevertheless, the networks' comparison will not be complete without estimating the precision of the Table 3 data.

Table 3. Results of the network comparison on GT1 data (probabilities $\overline{P}_{av}$ averaged for 100 seeds)

Class type	Network type	Mode 1		Mode 2	
		Basic node numbers	Enlarged node numbers	Basic node numbers	Enlarged node numbers
Singular		Case 1	Case 2	Case 3	Case 4
	MLP	0.8157	0.8184	0.8031	0.8059
	RBN	0.8115	0.8186	0.8009	0.8058
Multiple		Case 5	Case 6	Case 7	Case 8
	MLP	0.8738	0.8765	0.8660	0.8686
	RBN	0.8745	0.8783	0.8663	0.8701

Analysis of computational precision

Along with average values $\overline{P}_{av}$ presented in Table 3, standard deviations σ for the probabilities $\overline{P}$ have been also estimated. They are placed in Table 4, which conserves the format of the previous tables. It can be seen from the table that the precision is higher for the single class type than for the multiple one. On the other hand, the probabilities for the radial basis networks are generally more precise than the perceptron's probabilities. However, the differences are not large. That is why to quantify an overall precision level, we will use in the sequel a constant value $\sigma_{\overline{P}}$ = 0.0047 obtained by averaging Table 3 data.

Table 4. Computational precision of the probabilities $\overline{P}$ computed for GT1 (standard deviations σ estimated for 100 seeds)

Class type	Network type	Mode 1		Mode 2	
		Basic node numbers	Enlarged node numbers	Basic node numbers	Enlarged node numbers
Singular		Case 1	Case 2	Case 3	Case 4
	MLP	0.0045	0.0044	0.0049	0.0042
	RBN	0.0045	0.0039	0.0043	0.0041
Multiple		Case 5	Case 6	Case 7	Case 8
	MLP	0.0052	0.0049	0.0054	0.0054
	RBN	0.0051	0.0052	0.0048	0.0050

As a result of the same calculations with 100 seeds, the distributions of the analyzed estimations $\overline{P}$ have been determined for all considered cases. The corresponding diagrams included in Table A5 of the Appendix II help to conclude that the estimations $\overline{P}$ have the normal distribution. If that is the case, an uncertainty interval for $\overline{P}$ is ±2$\sigma_{\overline{P}}$=±0.0094 with 97.7% confidence probability. It becomes clear that the precision of Table 1 data is not indeed sufficient for a final conclusion on the networks.

Let us now estimate precision of the probabilities $\overline{P}_{av}$. Since they are computed by averaging random independent quantities $\overline{P}$, the standard deviation of the probability $\overline{P}_{av}$ can be estimated by a statistical formula $\sigma_{\overline{P}_{av}} = \sigma_{\overline{P}} \big/ \sqrt{N}$, which results in $\sigma_{\overline{P}_{av}}$=0.00047. As is well known, the sum of independent random variables tends to the normal distribution. Therefore, an uncertainty interval for $\overline{P}_{av}$ can be estimated as ±2$\sigma_{\overline{P}_{av}}$=±0.00094 with the same confidence probability of 97.7%. Thus, this interval characterizes computational precision of the data in Table 3.

Since the conclusion on networks' applicability is made on the basis of a difference between the probabilities $\overline{P}^{RBN}$ and $\overline{P}^{MLP}$ of the compared networks, an uncertainty interval for a random quantity $\Delta\overline{P} = \overline{P}^{RBN} - \overline{P}^{MLP}$ is to be determined as well. In this case, it is not obvious that the estimations $\overline{P}^{RBN}$ and $\overline{P}^{MLP}$ are independent. Change of a seed may alter fault classes' distinguishability and, in this way, may influence both estimations in the same direction. Figure A1 of Appendix II illustrates the behavior of these estimations for Case 1 of Table 3. It can be clearly seen that certain correlation between RBN and MLP probabilities takes place. To proceed with the precision analysis, we computed directly a standard deviation of the quantity $\Delta\overline{P}$ and obtained the result $\sigma_{\Delta\overline{P}}$=0.0043. Concerning quantities $\Delta\overline{P}$ obtained for different seeds as random and independent, we come to the following results: a standard deviation for the difference between the averaged probabilities $\overline{P}_{av}^{RBN}$ and $\overline{P}_{av}^{MLP}$ is equal to $\sigma_{\Delta\overline{P}\,av} = \sigma_{\Delta\overline{P}} \big/ \sqrt{N}$ =0.00043 and the corresponding uncertainty interval is ±0.00086. Returning back to Table 3, we can conclude now that the RBN probability increments for Case 6 and Case 8 as well as the mean increment of 0.0009, although being small, are still statistically significant.

In order to have greater confidence in the conclusions on the analyzed networks, the comparative calculations were repeated for GT2, which differs from GT1 in application and gas path structure.

5. NETWORKS' COMPARISON: CASE OF THE AIRCRAFT ENGINE (GT2)

Five variables are used to monitor GT2 health condition and 14 fault parameters are involved in fault simulation. These variables and parameters are described in Table A3 and Table A4 of Appendix I. Single and multiple fault classifications consist from 14 and 7 items, respectively. More information about the monitored variables and the classification variations for the given engine can be found in [20].

After numerous preliminary experiments with classification and network parameters, the following optimal values have been found:

- 500 patterns per a single fault class and 1000 patterns per multiple one,
- 36 perceptron hidden layer nodes,
- 77 RBN hidden nodes for a single fault classification and 105 nodes for multiple one.

We consider the chosen node numbers as enlarged. The comparison will be performed only for these numbers because they allow to better reveal possible RBN's advantages.

Given the chosen parameters, calculations with 100 different seeds were performed and the same probabilities $\overline{P}_{av}$ were obtained as in the case of GT1. The results for two operating modes and two class types (four computational cases in total) are shown in Table 5. The case numeration for the enlarged node numbers is conserved here.

First, we compared Table 3 and Table 5 to conclude on diagnosability of the analyzed engines. Averaging the data in these tables separately for single and multiple class types, we can state that GT2 has lower overall diagnosability than GT1: the true positive rate decreases from 0.8082 (GT1, enlarged node numbers) to 0.7410 (GT2) for the single class type and respectively from 0.8734 to 0.7694 for the multiple class type. This result has a natural explanation: as described above, GT2 has less monitored variables and more fault classes.

The given above probabilities also allows estimating the effect of a classification type change from single to multiple. For both gas turbines the true positive rates rises: by 0.0654 for GT1 and by 0.0284 for GT2. The increase of diagnostic accuracy can be interpreted as a result of two opposite tendencies. On the one hand, a multiple type classes occupy greater areas than single ones resulting in higher class' intersection and lower diagnostic accuracy. On the other hand, the change to the multiple type classification means considerable reduction of a total number of fault classes and the corresponding increase in accuracy. As can be seen, the second tendency prevails for the analyzed engines and their classifications.

Table 5. Results of the network comparison on GT2 data (probabilities $\overline{P}_{av}$ averaged for 100 seeds)

Class type	Network type	Mode 1	Mode 2
Single		Case 2	Case 4
	MLP	0.7338	0.7470
	RBN	0.7349	0.7485
Multiple		Case 6	Case 8
	MLP	0.7749	0.7596
	RBN	0.7787	0.7643

Returning to the main purpose of this paper of selecting a diagnostic technique, let us compare the performances of the networks under analysis. We can see from Table 5 that the RBN is slightly better in all four considered cases and the difference is greater for the multiple fault classes. The corresponding average probability increment is 0.0028 (0.28%). In this way, an advantage of the radial basis network in the application to the analyzed turbofan seems to be a little more notable than in the case of the industrial gas turbine.

As regards the computational precision, it was analyzed for GT2 like it was done in the previous section for GT1. Standard deviations σ of the probabilities $\overline{P}$ are included in Table 6. We can see that the presented data do not differ considerably from the corresponding GT1 data given in Table 4. A new standard deviation mean value $\sigma_{\overline{P}}$ = 0.0051 is close to the previous one $\sigma_{\overline{P}}$ = 0.0047 as well. As explained before, the uncertainty interval for averaged probabilities of Table 5 will be $\pm 2\sigma_{\overline{P}} \big/ \sqrt{N}$ =±0.00102. Consequently, one can conclude that the RBN probability increments observable in this table are statistically significant.

Table 6. Computational precision of the probabilities $\overline{P}$ computed for GT2 (standard deviations σ obtained for 100 seeds)

Class type	Network type	Mode 1	Mode 2
Single		Case 2	Case 4
	MLP	0.0053	0.0047
	RBN	0.0054	0.0046
Multiple		Case 6	Case 8
	MLP	0.0050	0.0050
	RBN	0.0051	0.0052

6. DISCUSSION

The presented network comparison has shown that the change from the multilayer perceptron to the radial basis network can enhance gas turbine diagnosis accuracy. This positive effect is statistically significant but very small: the corresponding increase of correct diagnosis probability is **0.09%** for the first analyzed gas turbine and **0.28%** for the second one. On the other hand, the RBN is much more demanding in terms of necessary computation time and physical computer memory. In general, the advantage of this neural network seems to be disputable. We can conclude that there is no reason to change from the MLP to the RBN in gas turbine diagnostic applications.

The fact that the fault recognition performances of two different neural networks are very close is worthy of discussion. Furthermore, comparing different gas turbine diagnostic techniques in our previous study [12], we arrived to a similar conclusion that the perceptron and the Bayesian approach provide equal levels of diagnostic accuracy. So, three different recognition techniques have the same accuracy. What explanation can be provided for this? We believe that all three techniques are sophisticated enough and are well suited for solving this specific problem – gas turbine fault recognition. No one of these techniques can further enhance diagnostic accuracy because the accuracy level achieved is near a theoretical accuracy level inherent to the analyzed engine with its fault classification. Following this idea, we suppose that within the approach used and the classification accepted no other recognition technique will be capable to considerably enhance diagnostic accuracy.

Instead, all efforts should be made to reduce fault class intersections, for example, by reducing measurement inaccuracy, installing more sensors in the gas path, and decreasing deviation errors. The options of multipoint diagnosis and diagnosis at transient operation will also result in a higher diagnostic accuracy. They are now considered in gas turbine diagnostics and it would be interesting to compare the networks in the application to these options. However, as shown in [20], both options imply a drastic increase of the number of network's unknown coefficients. At the same time, the radial basis network is already demanding of computer memory. For these reasons, it will be too difficult to perform calculations with an elevated number of the coefficients.

As can be seen, in the present paper we tried to perform the network comparison in maximally varying conditions of engine health, operation, and diagnosis. Nevertheless, we could not embrace all possible conditions within the limits of one paper. For example, classes that suppose simultaneous faults in two gas turbine modules were not considered. It was not possible to realize them in the presented study because the number of considered cases is already large and a new classification type will require a lot of additional calculations. However, we are going to extend the network comparison on the case of multiple module faults in our further studies.

Although the idea to compare different fault recognition techniques is not new, we think that this paper makes a practical contribution to gas turbine diagnostics. The point is that this discipline has a history of approximately five decades and embraces a lot of approaches, particular methods, and their variations. However, a designer of a real monitoring system would lack practical recommendations to develop the system in an optimal manner. To help him, more comparative investigations should be conducted beforehand. Such investigations focused on elaborating practical advices will allow optimal design of all monitoring system's elements in a short time. In this way, the present paper contributes to a better selection and tuning of a fault classification technique.

CONCLUSIONS

Thus, the comparison of two recognition techniques, multilayer perceptron (MLP) and radial basis network (RBN), has been performed in the present paper. The necessary fault classification was formed with the use of the thermodynamic model. The classification and the recognition techniques were embedded into a special testing procedure that computes for each technique the probabilities of a correct diagnosis.

In order to reduce random computational errors, every comparative calculation was repeated many times by means of the mentioned procedure under the same conditions and the resulting probabilities were averaged. In addition, to draw firm conclusions about the compared techniques, the comparative calculations were executed under varying conditions: different engines, their operating modes, network parameters, and diagnostic conditions.

Summing up the comparison results for all considered cases, it can be stated that the application of the RBN can bring in only slight enhancement of gas turbine diagnosis reliability. Moreover, greater memory and computation time required for the RBN should be also taken into account when a recognition technique is chosen for a real gas turbine monitoring system. In general, it seems to be unnecessary to change from the multilayer perceptron traditionally used in gas turbine diagnostics to the radial basis networks.

ACKNOWLEDGMENTS

The work has been carried out with the support of the National Polytechnic Institute of Mexico (research project 20101199).

REFERENCES

[1] Rao, B. K. N., 1996, *Handbook of Condition Monitoring*, Elsevier Advanced Technology, Oxford.

[2] Meher-Homji, C. B., Chaker, M, A., and Motivwala, H. M., 2001, "Gas Turbine Performance Deterioration," *Proc. of the 30th Turbomachinery Symposium*, Texas A&M University, Houston, Texas, pp. 139-175.

[3] Tsalavoutas, A., Stamatis, A., Mathioudakis, et al., 2000, "Identifying Faults in the Variable Geometry System of a Gas Turbine Compressor," ASME Paper No. 2000-GT-0033.

[4] Saravanamuttoo, H. I. H., and MacIsaac, B. D., 1983, "Thermodynamic Models for Pipeline Gas Turbine Diagnostics," *ASME Journal of Engineering for Power*, Vol. 105, pp. 875-884.

[5] Tarabrin, A. P., Schurovsky V. A., Bodrov A. I., et al., 1998, "An Analysis of Axial Compressor Fouling and a Blade Cleaning Method," *ASME Journal of Turbomachinery*, Vol. 120, pp. 256-261.

[6] Sampath, S., and Singh, R., 2006, "An Integrated Fault Diagnostics Model Using Genetic Algorithm and Neural Networks," *ASME Journal of Engineering for Gas Turbines and Power*, Vol. 128, Issue 1, pp. 49-56.

[7] Butler, S. W., Pattipati, K. R., Volponi, et al., 2006, "An Assessment Methodology for Data-Driven and Model Based Techniques for Engine Health Monitoring," ASME Paper No. GT2006-91096.

[8] Roemer, M. J., and Kacprzynski, G. J., 2000, “Advanced Diagnostics and Prognostics for Gas Turbine Engine Risk Assessment,” ASME Paper No. 2000-GT-30.

[9] Greitzer, F. L., Kangas, L. J., Terrones, K. M. et al., 1999, “Gas Turbine Engine Health Monitoring and Prognostics,” *International Society of Logistics (SOLE) Symposium*, Las Vegas, Nevada, 7p.

[10] Romessis, C., and Mathioudakis, K., 2006, “Bayesian Network Approach for Gas Path Fault Diagnosis,” *ASME Journal of Engineering for Gas Turbines and Power*, Volume 128, Issue 1, pp. 64-72.

[11] Loboda, I., Yepifanov, S., and Feldshteyn, Y., 2007, “A Generalized Fault Classification for Gas Turbine Diagnostics on Steady States and Transients” *ASME Journal of Engineering for Gas Turbines and Power*, Vol. 129, Issue 4, pp. 977-985.

[12] Loboda, I., and Yepifanov, S., 2006, “Gas Turbine Fault Recognition Trustworthiness,” *Cientifica*, Instituto Politecnico Nacional, Mexico, Vol. 10, Issue 2, pp. 65-74.

[13] Pipe K., 1987, “Application of Advanced Pattern Recognition Techniques in Machinery Failure Prognosis for Turbomachinery,” *Proc. Condition Monitoring 1987 International Conference,* British Hydraulic Research Association, UK, pp. 73-89.

[14] Volponi, A. J., DePold, H., and Ganguli, R., 2003, “The Use of Kalman Filter and Neural Network Methodologies in Gas Turbine Performance Diagnostics: a Comparative Study,” *ASME Journal of Engineering for Gas Turbines and Power*, Vol. 125, Issue 4, pp. 917-924.

[15] Loboda, I., Miro Zarate, L. A., and Leal Bolanos, A. E., 2010, “Radial Basis Functions for Gas Turbine Fault Recognition,” *Aerospace Technics and Technology*, National Aerospace University, Kharkov, Ukraine, Issue 77, pp. 182-186.

[16] Haykin, S., 1994, *Neural Networks*, Macmillan College Publishing Company, New York.

[17] Duda, R. O., Hart, P. E., and Stork D. G., 2001, *Pattern Classification*, Wiley-Interscience, New York.

[18] MATLAB manual.

[19] Loboda, I., Nakano Miyatake, M., Goryachiy A. et al., 2005, “Gas Turbine Fault Recognition by Artificial Neural Networks”, *The Fourth International Congress of Electromechanical Engineering and Systems*, National Polytechnic Institute, Mexico City, Mexico, 5p.

[20] Loboda, I., Feldshteyn, Y., and Yepifanov, S., 2007, “Gas Turbine Diagnostics under Variable Operating Conditions,” ASME Paper No. GT2007-28085.

APPENDIX I. MONITORED VARIABLES AND FAULT PARAMETERS

Table A1. Monitored variables of GT1.

№	Variables
1	Compressor total pressure
2	Gas generator turbine total pressure
3	Compressor total temperature
4	Gas generator turbine total temperature
5	Power turbine total temperature
6	Fuel consumption

Table A2. Simulated fault parameters of GT1.

№	Parameters
1	Compressor flow parameter
2	Compressor efficiency parameter
3	High pressure turbine flow parameter
4	High pressure turbine efficiency parameter
5	Power turbine flow parameter
6	Power turbine efficiency parameter
7	Combustion chamber total pressure recovery parameter
8	Combustion efficiency parameter
9	Inlet device total pressure recovery factor

Table A3. Monitored variables of GT2.

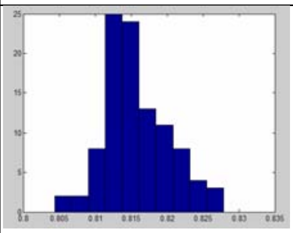
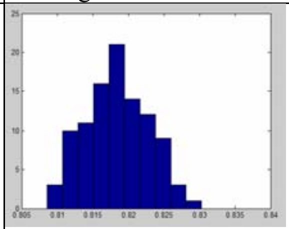
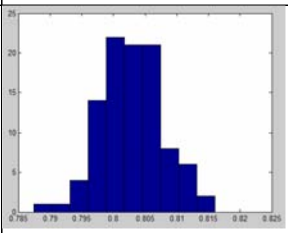
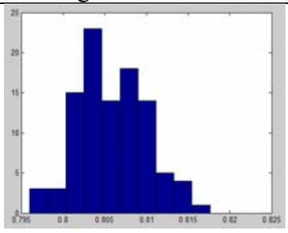
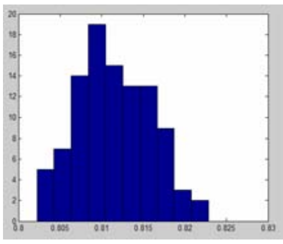
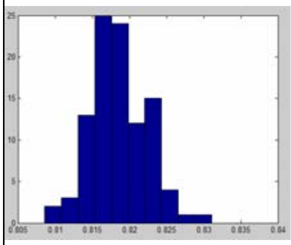
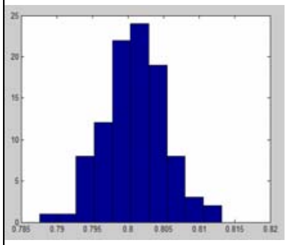
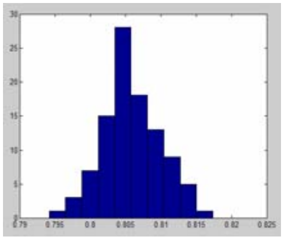
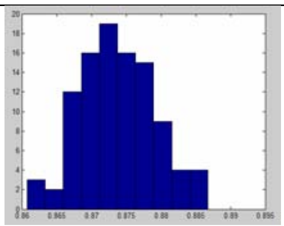
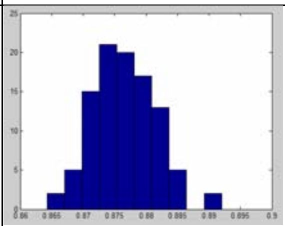
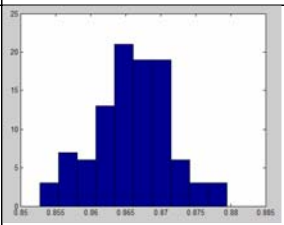
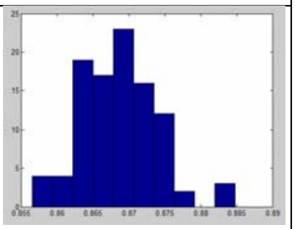
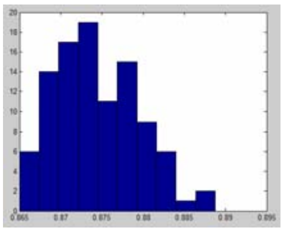
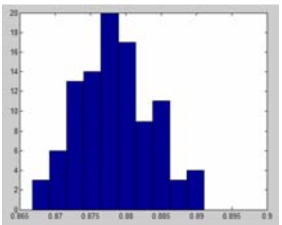
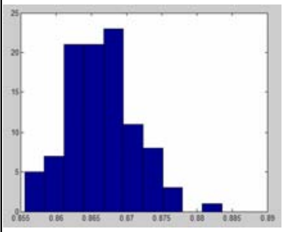
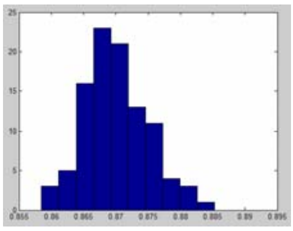
№	Variables
1	Intermediate pressure turbine temperature
2	Low pressure rotor speed
3	Intermediate pressure rotor speed
4	High pressure rotor speed
5	Total pressure ratio

Table A4. Simulated fault parameters of GT2.

№	Parameters
1	Low pressure compressor flow parameter
2	Low pressure compressor efficiency parameter
3	Intermediate pressure compressor flow parameter
4	Intermediate pressure compressor efficiency parameter
5	High pressure compressor flow parameter
6	High pressure compressor efficiency parameter
7	Low pressure turbine flow parameter
8	Low pressure turbine efficiency parameter
9	Intermediate pressure turbine flow parameter
10	Intermediate pressure turbine efficiency parameter
11	High pressure turbine flow parameter
12	High pressure turbine efficiency parameter
13	Combustion chamber total pressure recovery parameter
14	Combustion efficiency parameter

APPENDIX II. BEHAVIOR OF PROBABILITY ESTIMATIONS

Table A5. Distributions of the probabilities $\overline{P}$ (GT1)

Class type	Network type	Mode 1		Mode 2	
		Basic node numbers	Enlarged node numbers	Basic node numbers	Enlarged node numbers
Singular	MLP				
	RBN				
Multiple	MLP				
	RBN				

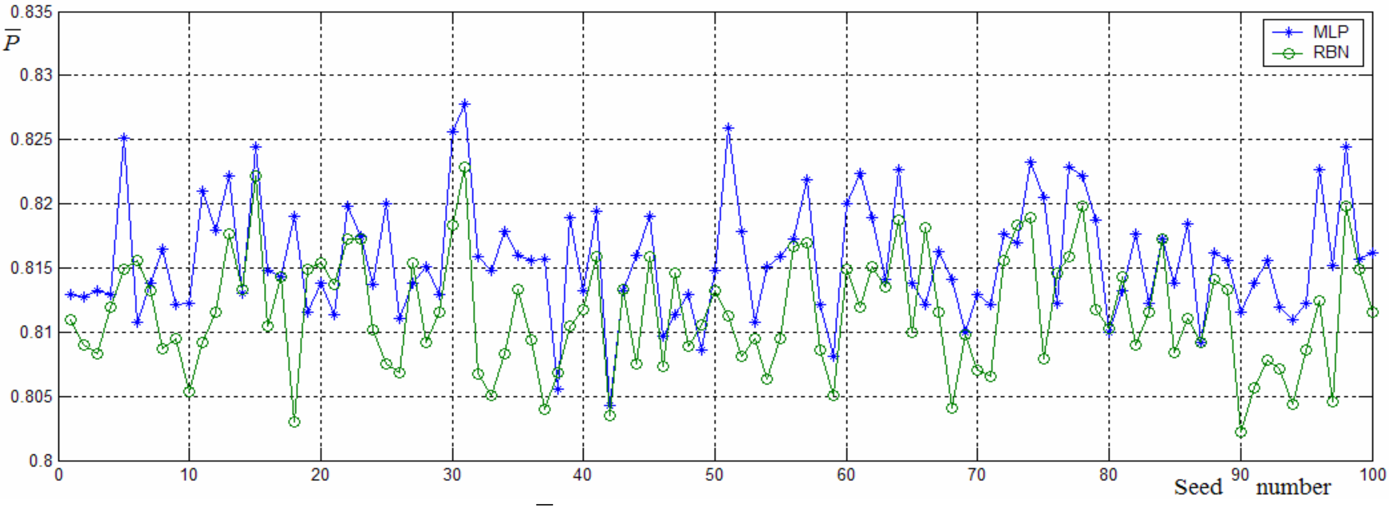


Fig. A1. Correlation of the networks' probabilities $\overline{P}$ (GT1, Case 1)