

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Neural Networks for Gas Turbine Diagnosis

Igor Loboda

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63107>

Abstract

The present chapter addresses the problems of gas turbine gas path diagnostics solved using artificial neural networks. As a very complex and expensive mechanical system, a gas turbine should be effectively monitored and diagnosed. Being universal and powerful approximation and classification techniques, neural networks have become widespread in gas turbine health monitoring over the past few years. Applications of such networks as a multilayer perceptron, radial basis network, probabilistic neural network, and support vector network were reported. However, there is a lack of manuals that summarize neural network applications for gas turbine diagnosis.

A monitoring system comprises many elements, and many factors influence the final diagnostic accuracy. The present chapter generalizes our investigations that are devoted to the enhancement of this system by choosing the best option for each element. In these investigations, a diagnostic process is simulated on the basis of neural networks, and we focus on reaching the highest accuracy by choosing the best network and its optimal tuning to the issue to solve. Thus, helping with enhancement of a whole monitoring system, neural networks themselves are objects of investigation and optimization. As a result of the conducted investigations, the chapter provides the recommendations on choosing and tailoring the network for a particular diagnostic task.

Keywords: gas turbines, gas path diagnosis, fault classification, pattern recognition, artificial neural networks

1. Introduction

As complex and expensive mechanical systems, gas turbine engines benefit a lot from the application of advanced diagnostic technologies, and the use of monitoring systems has

become a standard practice. To perform effective analysis, there are different diagnostic approaches that cover all gas turbine subsystems. The diagnostic algorithms based on measured gas path variables are considered as principal and pretty complex. These variables (air and gas pressures and temperatures, rotation speeds, fuel consumption, etc.) carry valuable information about an engine's health condition and allow to detect and identify different engine abrupt faults and deterioration mechanisms (for instance, foreign object damage, fouling, erosion, tip ribs, and seal wear). Malfunctions of measurement and control systems can be diagnosed as well. Thousands of technical publications devoting to the gas path diagnosis can be found. They can be arranged according to input information and mathematical models applied.

Although advancement of instrumentation and computer science has enabled extensive field data collection, the data with gas turbine faults are still infrequent because real faults rarely appear. Some intensive and practically permanent deterioration mechanisms, for example, compressor fouling, allow their describing on the basis of real data. However, to describe the variety of all possible faults, mathematical models are widely used. These models and the diagnostic methods that use them fall into two main categories: physics-based and data-driven.

A thermodynamic engine model is a representative physics-based model. This nonlinear model is based on thermodynamic relations between gas path variables. It also employs mass, energy, and momentum conservation laws. Such a sophisticated model has been used in gas turbine diagnostics since the work of Saravanamuttoo H.I.H. (see, e.g., [1]). The model allows to simulate the gas path variables for an engine baseline (healthy engine performance) and for different faults embedded into the model through special internal coefficients called fault parameters. Applying system identification methods to the thermodynamic model, an inverse problem is solved: Unknown fault parameters are estimated using measured gas path variables. During the identification, such parameters are found that minimize the difference between the model variables and the measured ones. Besides the better model accuracy, the simplification of the diagnosing process is reached because the fault parameter estimates contain information of current engine health. The diagnostic algorithms based on the model identification constitute one of two main approaches in gas turbine diagnostics (see, for instance, [1–4]).

The second approach uses a pattern recognition theory. Since model inaccuracy and measurement errors impede a correct diagnosis, gas path fault localization can be characterized as a challenging recognition issue. Numerous applications of recognition tools in gas path diagnostics are known, for instance, genetic algorithms [5], correspondence and discrimination analysis [6], k-nearest neighbor [7], and Bayesian approach [8]. However, the most widespread techniques are artificial neural networks (ANNs). The ANNs applications are not limited by the fault recognition, they are also applied or can be applied at other diagnostic stages: feature extraction, fault detection, and fault prediction.

At the feature extraction stage, differences (a.k.a. deviations) between actual gas path measurements and an engine baseline are determined because they are by far better indicators of engine health than the measurements themselves are. To build the necessary baseline model, the multilayer perceptron (MLP), also called a back-propagation network, is usually em-

ployed [9, 10]. To filter noise, an auto-associative configuration of the perceptron is sometimes applied to the measurements [11].

At the fault localization stage, fault classes can be presented by sets of the deviations (patterns) induced by the corresponding faults. Such a pattern-based classification allows to apply the ANNs as recognition techniques, and multiple applications of the MLP (see, e.g., [4, 5]) as well as the radial basis network (RBN) [5], the probabilistic neural network (PNN) [12, 13], and support vector machines (SVM) a.k.a. Support vector network (SVN) [7, 12] were reported. In spite of many publications on gas turbine fault recognition, comparative studies, which allow to choose the best technique [4, 5, 7, 12], are still insufficient. They do not cover all of the used techniques and often provide differing recommendations.

The fault detection stage can also be presented as a pattern recognition problem with two classes to recognize: a class of healthy engines and a class of faulty engines. If the classification for the fault localization stage is available, it does not seem a challenge to use the patterns of this classification for building the fault detection classification. However, the studies applying recognition techniques, in particular the ANNs, for gas turbine fault detection are absent so far. Instead, the detection problem is solved by tolerance monitoring [14, 15].

The fault prediction stage is less investigated than the previous stages, and only few ANNs applications are known. Among them, it is worth to mention book [16] analyzing the ways to predict gas turbine faults and study [17], comparing a recurrent neural network and a nonlinear auto-regressive neural network. We can see that in total for all stages, the perceptron is by far the highest demand network. It is used for filtering the measurements, approximating the engine baseline, and recognizing the faults.

Thus, a brief observation of the neural networks applied for gas turbine diagnosis has revealed that the multiple known cases of their use need better generalization and recommendations to choose the best network. The areas of promising ANNs application were also found. In the present chapter, we generalize our investigations aimed at the optimization of a total diagnostic process through the enhancement of each of its elements. On the one hand, the neural networks help with process realization being its critical elements. On the other hand, the networks themselves are objects of analysis: For known applications, they are compared to choose the best network, and one new application is proposed. During the investigations, the rules of proper network usage have also been established.

The rest of the chapter describes these investigations and is structured as follows: description of the networks used (Section 2), network-based diagnostic approach (Section 3), diagnostic process optimization (Section 4), feature extraction stage optimization (Section 5), fault detection stage optimization (Section 6), and fault localization stage optimization (Section 7).

2. Artificial neural networks

The four networks mentioned in the introduction have been chosen for investigations: MLP, RBN, PNN, and SVN. The PNN is a realization of the Parzen Windows and has the important

property of probabilistic outputs, that is, the gas turbine faults are recognized on the basis of their confidence probabilities. These probabilities are computed through numerical estimates of probability density of fault patterns. For the purpose of comparison, a similar recognition tool, the K-nearest neighbor (K-NN) method has been involved into the investigations. Foundations of the chosen techniques can be found in many books on classification theory, for example, in [18, 19, 20]. The next subsections include only a brief description of techniques required to better understand the present chapter.

2.1. Multilayer perceptron

The perceptron can solve either approximation or classification issues. The scheme shown in **Figure 1** illustrates structure and operation of the MLP [18, 19]. We can see that the perceptron presents a feed-forward neural network in which no feedback is observed, and all signals go only from the input to the output.

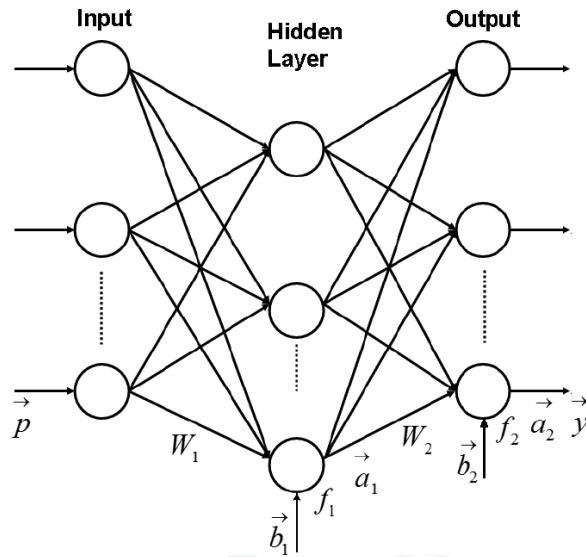


Figure 1. Multilayer perceptron.

To determine a hidden layer input vector, the product of a weight matrix W_1 and a network input vector (pattern) \vec{p} is summed with a bias vector \vec{b}_1 . A hidden layer transfer function f_1 transforms this vector in an output vector \vec{a}_1 . A network output \vec{a}_2 is computed similarly considering the vector \vec{a}_1 as an input. In this way, perceptron operation can be expressed by $\vec{y} = \vec{a}_2 = f_2\{W_2 f_1(W_1 \vec{p} + \vec{b}_1) + \vec{b}_2\}$. When we apply the MLP to classify patterns, elements of the vector \vec{a}_2 show how close the pattern \vec{p} is to the corresponding classes. The nearest class is chosen as a class to which the pattern belongs, and such classifying can be considered as deterministic.

To find unknown matrixes W_1 and W_2 and vectors \vec{b}_1 and \vec{b}_2 , a back-propagation learning algorithm distributes a network output error on these unknown quantities. In every learning iteration (epoch), they vary in the direction of error reduction. The iterations continue unless the minimum error has been reached. This algorithm requires differentiable transfer functions, and a sigmoid type is commonly used.

2.2. Radial basis network

Figure 2 illustrates operation of an RBN. It includes two layers: a hidden radial basis layer and an output linear layer. Operation of radial basis neurons is different from the perceptron neurons operation [18, 19, 20]. The neuron's input n is formed as the Euclidean norm $|||$ of a difference between a pattern vector \vec{p} and a weight vector \vec{w} , multiplied by a scalar b (bias). In this way, $n = ||\vec{w} - \vec{p}||b$. Using this input, a radial basis transfer function determines an output $a = \exp(-n^2)$. Where there is no distance between the vectors, the function has the maximum value $a=1$, and the function decreases when the distance increases. The bias b allows changing the neuron sensitivity. The output layer transforms the radial basis output \vec{a}_1 to a network output \vec{a}_2 . Operation of this layer does not differ from the operation of a perceptron layer with a linear transfer function. The radial basis layer usually needs more neurons than a comparable perceptron hidden layer because the radial basis neuron covers a smaller region compared with the sigmoid neuron.

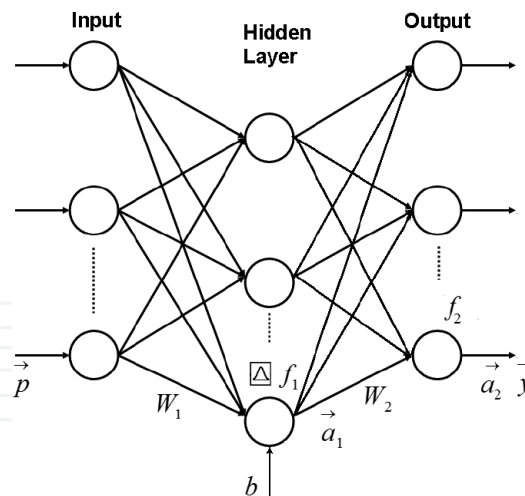


Figure 2. Radial basis network.

2.3. Probabilistic neural networks

The PNN is a specific variation of radial basis network [18]. It is used to solve classification problems. **Figure 3** presents the scheme of this network and helps to understand its operation. Like the RBN, the probabilistic neural network has two layers.

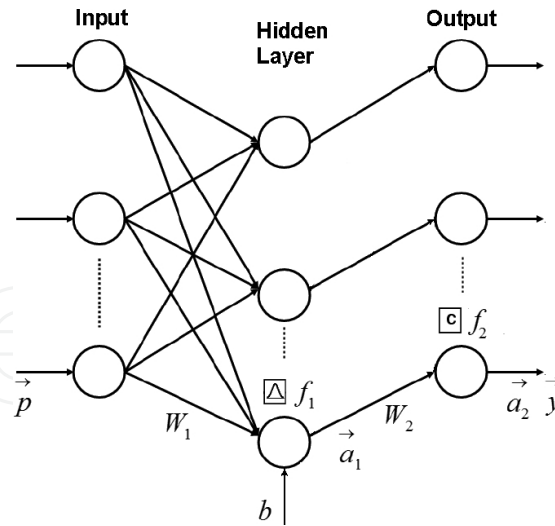


Figure 3. Probabilistic neural network.

The hidden layer is formed and operates just like the same layer of the RBN. It is built from learning patterns united in a matrix W_1 . Elements of an output vector \vec{a}_1 indicate how close the input pattern is to the learning patterns.

The output or classification layer differs from the RBN output layer. Each class has its output neuron that sums the radial basis outputs a_j corresponding to the class patterns. To this end, a weight matrix W_2 formed by 0- and 1-elements is employed. A vector $W_2 \vec{a}_1$ contains probabilities of all classes. A transfer function f_2 finally chooses the class with the largest probability. In this way, the probabilistic network classifies input patterns using a probabilistic measure that is more realistic than the perceptron classifying. The PNN is the most used realization of a Parzen Windows (PW) [18], a nonparametric method that estimates probability density in a given point (pattern) \vec{z} using the nearby learning patterns.

2.4. k-Nearest neighbors

Like the Parzen Windows (PNNs), the k-nearest neighbors is a nonparametric technique [18]. For a given class and point (pattern) \vec{p} , it counts the number k of class patterns in a nearby region of volume V and estimates the necessary probability density in accordance with a simple formula

$$\rho = \frac{k/n}{V} \quad (1)$$

where n stands for a total number of class patterns.

To ensure the convergence of the estimate ρ , we need to satisfy the following requirements

$$\lim_{n \rightarrow \infty} V = 0; \lim_{n \rightarrow \infty} k = \infty; \lim_{n \rightarrow \infty} k/n = 0. \quad (2)$$

To this end, we increase n and can let V be proportional to $1/\sqrt{n}$.

In contrast to the Parzen Window method that fixes the volume V and looks for the number k , the K -nearest neighbor method specifies k and seeks for the sphere of volume V . Since the PW uses constant window size, it may not capture patterns when the actual density is low. The density estimate will be equal to zero, and the classification decision confidence will be underestimated. A solution to this problem is to use the window that depends on learning data. Using this principle, the K -NN increases a spherical window individually for each class until k patterns (nearest neighbors) fall into the window. A sphere radius will change class by class. The greater the radius is, the lower probability density estimate will be according to Eq. (1).

2.5. Support vector network

Any hyperplane can be written in the space R^P as the set of points \vec{p} satisfying:

$$\vec{p}^T \vec{w} + b = 0 \quad (3)$$

where \vec{w} is a vector perpendicular to the hyperplane and b is the bias. Let us present learning data of two classes as pattern vectors $\vec{p}_i \in R^P, i = 1, N$ and their corresponding labels $y_i \in (-1, 1)$, indicating the class to which the pattern \vec{p} belongs.

If the learning data are linearly separable, two parallel hyperplanes without points between them can be built to divide the data. The hyperplanes can be given by $\vec{w}^T \vec{p}_i + b = 1$ and $\vec{w}^T \vec{p}_i + b = -1$. The margin is defined to be the distance between them and is equal to $2/\|\vec{w}\|$ (Figure 4). Intuitively, it measures how good the separation between the two classes is. The points divided in this manner satisfy the following constraint:

$$y_i(\vec{w}^T \vec{p}_i + b) \geq 1 \quad (4)$$

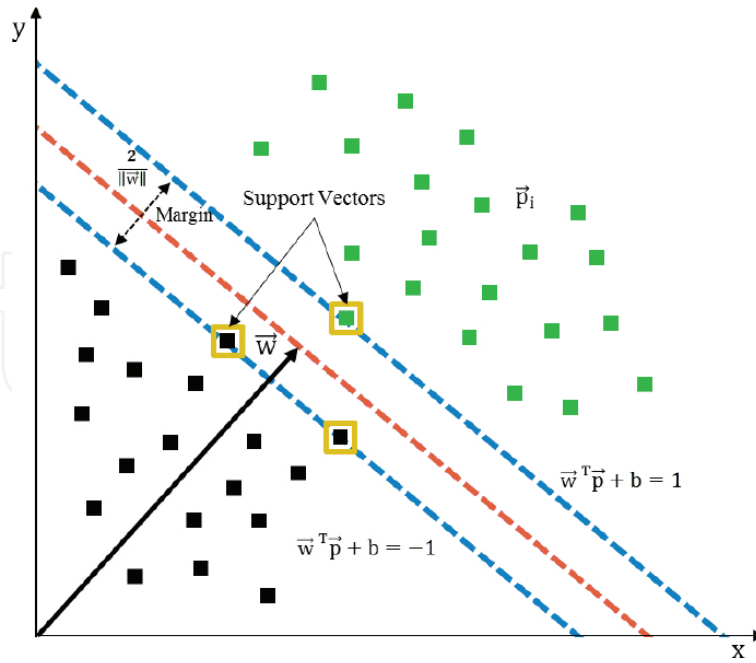


Figure 4. SVN: hyperplanes and separation margin.

The objective of the SVN is to find the hyperplanes that produce the maximal margin or minimum vector \vec{w} [19, 20]. In this way, SVN needs to solve the following primal optimization problem:

$$\min \frac{1}{2} \vec{w}^T \vec{w} \quad (5)$$

subject to $y_i(\vec{w}^T \vec{p}_i + b) \geq 1$, for $i = 1, \dots, N$

Introducing the Karush-Kuhn-Tucker (KKT) multipliers $\alpha_i \geq 0$, objective function (5) can be transformed to:

$$\min_{w,b} \max_{\alpha} \frac{1}{2} \vec{w}^T \vec{w} - \sum_{i=1}^N \alpha_i (y_i (\vec{w}^T \vec{p}_i + b) - 1) \quad (6)$$

As can be seen, expression (6) is a function of \vec{w} , b , and α . This function can be transformed into the dual form:

$$L = \min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \vec{p}_i^T \vec{p}_j - \sum_{i=1}^N \alpha_i \quad (7)$$

subject to $\alpha_i \geq 0$ and $\sum_{i=1}^N \alpha_i y_i = 0$ for $i = 1, \dots, N$

It can be also expressed as:

$$L = \min_{\alpha} \frac{1}{2} \vec{\alpha}^T Q \vec{\alpha} - 1^T \vec{\alpha} \quad (8)$$

where Q is the matrix of quadratic coefficients. This expression is minimized now only as a function of $\vec{\alpha}$, and the solution is found by Quadratic Programming.

In SVM classification problems, a complete separation is not always possible, and a flexible margin is suggested in reference [21] that allows misclassification errors while tries to maximize the distance between the nearest fully separable points. The other way to split not separable classes is to use nonlinear functions as proposed in reference [22]. Among them, radial basis functions are recommended [23].

SVMs were originally intended for binary models; however, they can now address multi-class problems using the One-Versus-All and One-Versus-One strategies.

A gas turbine diagnostic process using the techniques above described is simulated according to the following approach.

3. Neural networks-based diagnostic approach

The approach described corresponds to the diagnostic stages of feature extraction and fault localization and embraces the steps of fault simulation, feature extraction, fault classification formation, making a recognition decision, and recognition accuracy estimation.

3.1. Fault simulation

Within the scope of this chapter, faults of engine components (compressor, turbine, combustor, etc.) are simulated by means of a nonlinear gas turbine thermodynamic model

$$\vec{Y}(\vec{U}, \vec{\Theta}) \quad (9)$$

The model determines monitored variables \vec{Y} as a function of steady-state operating conditions \vec{U} and engine health parameters $\vec{\Theta} = \vec{\Theta}_0 + \Delta\vec{\Theta}$. Each component is presented in the model by

its performance map. Nominal values $\vec{\theta}_0$ correspond to a healthy engine, whereas fault parameters $\Delta\vec{\theta}$ imitate fault influence by shifting the component maps.

3.2. Feature extraction

Although gas turbine monitored variables are affected by engine deterioration, the influence of the operating conditions is much more significant. To extract diagnostic information from raw measured data, a deviation (fault feature) is computed for each monitored variable as a difference between the actual and baseline values. With the thermodynamic model, the deviations Z_i $i=1,m$ induced by the fault parameters are calculated for all m monitored variables according to the following expression

$$Z_i = \left(\frac{Y_i(\vec{U}, \vec{\theta}_0 + \Delta\vec{\theta}) - Y_{0i}(\vec{U}, \vec{\theta}_0)}{Y_{0i}(\vec{U}, \vec{\theta}_0)} + \varepsilon_i \right) / a_i \quad (10)$$

A random error ε_i makes the deviation more realistic. A parameter a_i normalizes the deviation errors, resulting that they will be localized within the interval $[-1, 1]$ for all monitored variables. Such normalization simplifies fault class description.

Deviations of the monitored variables united in an $(m \times 1)$ deviation vector \vec{Z} (feature vector) form a diagnostic space. Every vector \vec{Z} presents a point in this space and is a pattern to be recognized.

3.3. Fault classification formation

Numerous gas turbine faults are divided into a limited number q of classes D_1, D_2, \dots, D_q . In the present chapter, each class corresponds to varying severity faults of one engine component. The class is described by component's fault parameters $\Delta\theta_j$. Two types of fault classes are considered. The variation of one fault parameter results in a single fault class, while independent variation of two parameters of one gas turbine component allows to form a class of multiple faults.

To form one class, many patterns are computed by expression (10). The required parameters $\Delta\theta_j$ and ε_i are randomly generated using the uniform and Gaussian distributions correspondingly. To ensure high computational precision, each class is typically composed from 1000 patterns. A learning set $\mathbf{Z1}$ uniting patterns of all classes presents a whole pattern-based fault classification. **Figure 5** illustrates such a classification by presenting four single fault classes in the diagnostic space of three deviations.

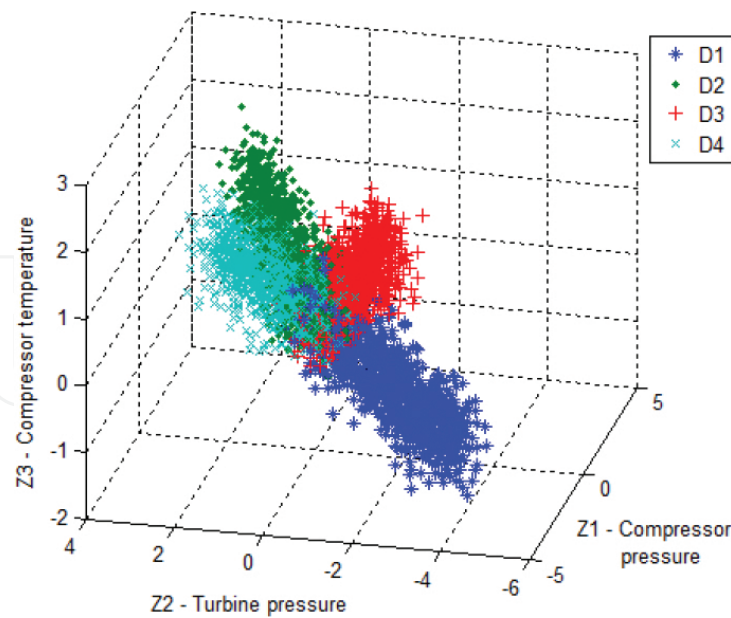


Figure 5. Pattern-based fault classification.

3.4. Making a fault recognition decision

In addition to the given (observed) pattern \vec{Z} and the constructed fault classification $\mathbf{Z1}$, a classification technique (one of the chosen networks) is an integral part of a whole diagnostic process. To apply and test the classification techniques, a validation set $\mathbf{Z2}$ is also created in the same way as set $\mathbf{Z1}$. The difference between the sets consists in other random numbers that are generated within the same distributions.

3.5. Recognition accuracy estimation

It is of practical interest to know recognition accuracy averaged for each fault class and a whole engine. To this end, the classification technique is consequently applied to the patterns of set $\mathbf{Z2}$ producing diagnoses d_i . Since true fault classes D_j are also known, probabilities of correct diagnosis (true positive rates) $P(d_i/D_j)$ can be calculated for all classes resulting in a probability vector \vec{P} . A mean number \bar{P} of these probabilities characterizes accuracy of engine diagnosis by the applied technique. In this chapter, the probability \bar{P} is employed as a criterion to compare the techniques described in Section 2.

4. Optimization of the neural networks-based diagnostic process

The structure and efficiency of a diagnostic algorithm depend on many factors and the options that can be chosen for each factor. The classification of these factors and options is given in

Figure 6 , where the factors are shown in the first line. On the basis of accumulated knowledge and experience, every research center (even a single researcher) chooses an appropriate option for each factor and develops its own diagnostic algorithm. To be optimal, this algorithm should take into account all peculiarities of a given engine, its application, and other diagnostic conditions. Thus, it is not likely that the algorithm be optimal for other engines and applications. As a result, every monitoring system needs an appropriate diagnostic algorithm.

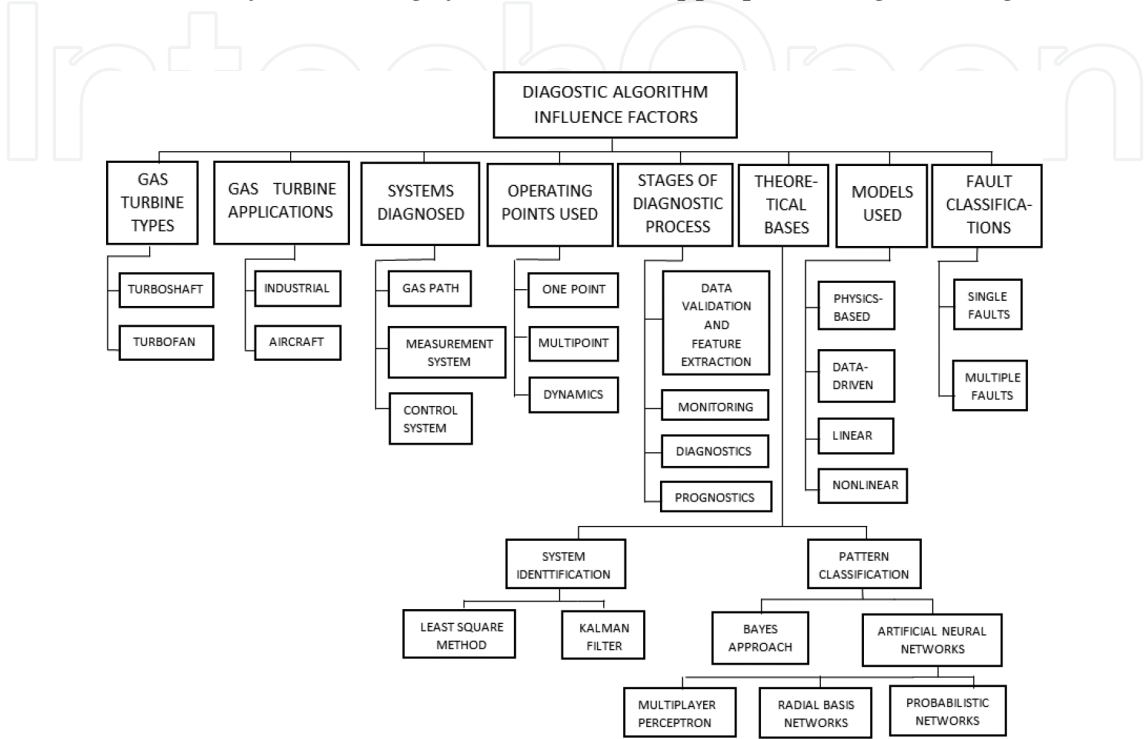


Figure 6. Factors that influence structure and efficiency of gas path diagnostic algorithms.

Thus, comparing complete diagnostic algorithms does not seem to be useful. Instead, comparing options for each above factor and choosing the best option are proposed. When options of one factor are compared, the other factors (comparison conditions) are fixed forming a comparison case. To draw sound conclusions about the best option, the comparison should be repeated for many comparison cases. To form these cases, each comparison condition varies independently according to the theory of the design of experiments. Since every new condition drastically increases the volume of comparative calculations, the most significant conditions are considered first.

To perform the comparative calculations, a test procedure based on the above-described approach has been developed in Matlab (MathWorks, Inc.). For each compared option, the procedure executes numerous cycles of gas turbine fault diagnosis by the chosen technique and finally computes a diagnosis reliability indicator, which is used as a comparison criterion.

Three gas turbine engines (Engine 1, Engine 2, and Engine 3) of different construction and application have been chosen as test cases. Engine 1 and Engine 2 are free turbine power plants. Engine 1 is a natural gas compressor driver; it is presented in the investigations by its thermodynamic model and field data recorded. Engine 2 is intended for electricity production and

is given by field data. Engine 3 is a three-spool turbofan for a transport aircraft; its thermodynamic model is used. The field data called hourly snapshots present filtered and averaged steady-state values recorded every hour during about one year of operation of Engine 1 and Engine 2. Since the data include periods of compressor fouling and points of washing, they are very suitable for testing diagnostic techniques.

Using the network-based approach described in Section 3 and the information about the test case engines, many investigations have been conducted to improve the diagnostic process at the stages of feature extraction, fault detection, and fault localization. The results achieved for the feature extraction stage are described in the next section.

5. Feature extraction stage optimization

As stated in Section 3, the deviations are useful diagnostic features. Although the thermodynamic model can be used as a baseline model for computing the deviations, it is too complex for real monitoring systems and has intrinsic inaccuracy. As mentioned in the introduction, to build a simple and fast data-driven baseline model, only neural networks, in particular the MLP, are applied. On the other hand, in the previous studies we successfully used a polynomial type baseline model. It was therefore decided [24] to verify whether the application of such a powerful approximator as the MLP instead of polynomials yields higher adequacy of the baseline model and better quality of the corresponding deviations.

Given a measured value Y_i^* and data-driven baseline model $\vec{Y}_0(\vec{U})$, the deviation is written as

$$\delta Y_i^* = \frac{Y_i^* - Y_{0i}(\vec{U})}{Y_{0i}(\vec{U})} \quad (11)$$

For one monitored variable, a complete second-order polynomial function of four arguments (operating conditions) is written as

$$\begin{aligned} \vec{Y}_0(\vec{U}) = & a_0 + a_1 u_1 + a_2 u_2 + a_3 u_3 + a_4 u_4 + a_5 u_1 u_2 + a_6 u_1 u_3 + a_7 u_1 u_4 + \\ & a_8 u_2 u_3 + a_9 u_2 u_4 + a_{10} u_3 u_4 + a_{11} u_1^2 + a_{12} u_2^2 + a_{13} u_3^2 + a_{14} u_4^2 \end{aligned} \quad (12)$$

For all m monitored variables and measurements at n operating points, equation (12) is transformed to a linear system $\mathbf{Y}=\mathbf{V}\mathbf{A}$ with matrixes \mathbf{Y} ($n \times m$) and \mathbf{V} ($n \times k$) formed from these data, where $k=15$ is number of coefficients. To enhance coefficient estimates (matrix \mathbf{A}), great volume of input data ($n \gg k$) is involved and the least-squares method is applied.

As to the perceptron, its typical input is formed by four operating conditions, and the output consists of seven monitored variables. Hidden layer size determines a network's capability to

approximate complex functions and varies in calculations. As a result of MLP tuning, we chose 12 nodes at this layer. Thus, the perceptron structure is written as $4 \times 12 \times 7$. Since the MLP has tan-sigmoid transfer functions, and the output varies within the interval $(-1, 1)$, all monitored quantities are normalized.

Many cases of comparison on the simulated and real data of Engines 1 and 2 were analyzed. The MLP was sometimes more accurate at the learning step. At the validation step, the deviations computed with the MLP had a little worse accuracy for Engine 1. For Engine 2, the best MLP validation results are illustrated in **Figure 7**. As can be seen here, both polynomial deviations dT_{tp} and network deviations dT_{tn} reflect the fouling and washing effects equally well. However, in many other cases the polynomials outperformed. Why does the network approximate well a learning set and frequently fail on a validation set? The answer seems to be evident because of an overlearning (overfitting) effect. Due to a greater flexibility, the network begins to follow data peculiarities induced by measurement errors in the learning set and describes worse a gas turbine baseline performance for the validation set.

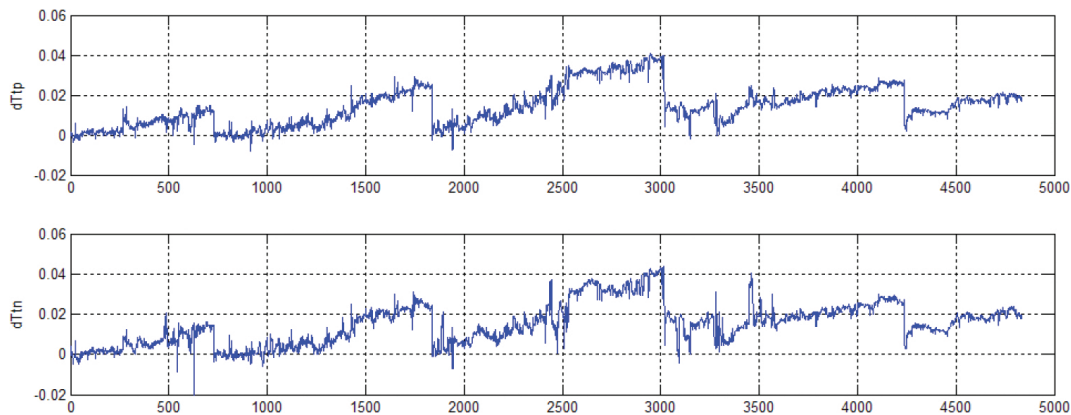


Figure 7. EGT deviations computed on the Engine 2 real data validation set (dT_{tn} —network-based deviation; dT_{tp} —polynomial-based deviation).

Although the MLP as a powerful approximation technique promised better gas turbine performance description, the results of the comparison have been somewhat surprising. No manifestations of network superiority were detected. When comparing these techniques, it is also necessary to take into consideration that an MLP learning procedure is more complex because it is numerical in contrast to an analytical solution for polynomials. Thus, a polynomial baseline model can be successfully used in real monitoring systems along with neural networks. At least, it seems to be true for simple cycle gas turbines with gradually changed performance, like the turbines considered in this chapter.

6. Fault detection stage optimization

As mentioned in the Introduction, the fault detection is actually based on tolerances (thresholds). However, it seems reasonable to present it as a pattern recognition problem like we do

at the fault recognition stage. Classification D_1, D_2, \dots, D_q created for the purpose of fault localization and presented in **Figure 5** corresponds to a hypothetical fleet of engines with different faults of variable severity. To form the classification for fault detection, we can reasonably accept that the engine fleet and the distributions of faults are the same. Paper [25] explains how to use patterns of the existing classification D_1, D_2, \dots, D_q for two new classes of healthy and faulty engines. The boundary between these classes corresponds to maximal error of the normalized deviations and is determined as a sphere of radius $R = 1$. The patterns, for which a vector of true deviations (without errors) is situated inside the sphere, form the healthy engine class; the others create the faulty engine class. It is clear that the patterns (deviation vectors with noise) of these two classes are partly intersected, resulting in α - and β -errors during the detection. **Figure 8** illustrates the new classification; the intersection is clearly seen. Two variations of the new classification based on single and multiple original classes have been prepared.

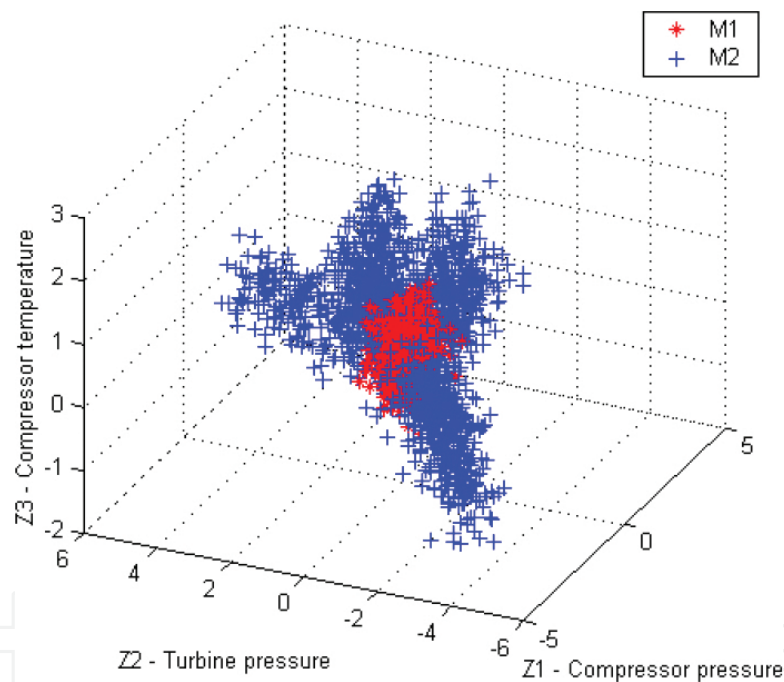


Figure 8. Patterns-based classification for monitoring.

Since new patterns-based classification (learning and testing sets) is ready, we can use any recognition technique to perform fault detection, and the MLP has been selected once more. It conserved sigmoid transfer functions and the hidden layer size of 12. Given that a threshold-based approach, which classifies pattern vectors according to their length, is traditionally used in fault detection, the algorithm with a distance measure (r -criterion) was also developed and compared with the MLP. Since the consequences of α - and β -errors are quite different (α -error is always considered as more dangerous), reduced losses $\bar{c} = P_\beta + \frac{c_\alpha}{c_\beta} P_\alpha$ were introduced to

quantify monitoring effectiveness, where P_α and P_β are probabilities of α - and β -errors, c_α and c_β denote the corresponding losses, and $\frac{c_\alpha}{c_\beta}$ are equal to 10.

Figure 9 shows the plots of the reduced losses versus the radius r . For the MLP the change of r was simulated by the corresponding change of the boundary radius R during pattern separation in the learning set. It can be seen that the introduction of an additional threshold r , which is different from the boundary, reduces monitoring errors for both techniques. The best results correspond to the minimums of the curves. By comparing them, we can conclude that the network (MLP) provides better results for single classes, and the techniques are equal for multiple classes. In general for all comparison cases, the MLP slightly outperforms the r -criterion-based technique. Thus, the perceptron can be successfully applied for real gas turbine fault detection.

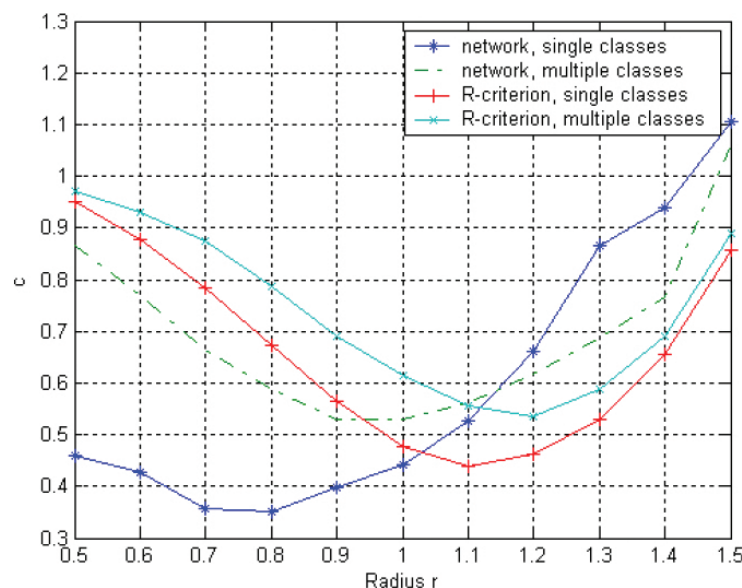


Figure 9. Reduced losses due to monitoring errors versus the threshold radius r .

7. Fault localization stage optimization

To draw sound conclusions about the ANN applicability for gas turbine fault localization, the comparison of the chosen networks was repeated for many comparison cases formed by independent variation of the main influencing factors: engines, operating modes, simulated or real information, and class types. In this way not only the best network is chosen but also the influence of these factors on diagnosis results is determined helping with the optimization of a total diagnostic process. For the purpose of correct comparison, the networks were tailored to a concrete task to solve.

7.1. Neural network tuning

We started to use ANNs applications and their tuning with the MLP [26]. The numbers of monitored variables and fault classes unambiguously determine the size of input and output layers of this network. As to the hidden layer, the number of 12 nodes was estimated as optimal

using the probability \bar{P} as a criterion. To choose a proper back-propagation algorithm, 12 variations were compared by accuracy and execution time. The resilient back-propagation ("rp"-algorithm) provided the best results and has been chosen. It was also found that 200 batch mode training epochs are sufficient for good learning; however, a learning stop by an Early Stopping Option may be useful as well.

Figure 10 illustrates other example of the tuning. Averaged probabilities computed for the PNN are plotted here against spread b , unique PNN tuning parameter. To determine this

probability that has high precision of about ± 0.001 , calculations of \bar{P} were repeated 100 times for each spread value, each time with a different seed (quantity that determines a consequence of random numbers), and an average value was computed. Such computations to find the best value b were repeated for two operating modes of Engine 1 and for two fault class types. It can be seen in the figure that the highest values of probability \bar{P}_{av} does not depend on operating mode. These values are $b=0.35$ for the single fault type and $b=0.40$ for the multiple one.

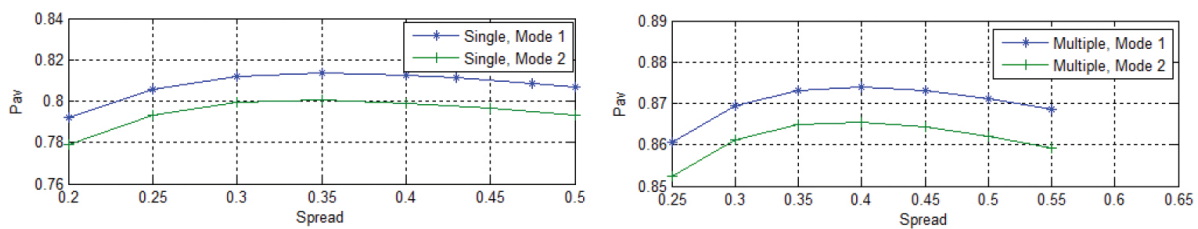


Figure 10. Probabilities versus spread parameter.

For all networks, the value 1000 simulated patterns per fault class has been selected as tradeoff between the required computer resources and the accuracy of the probabilities \bar{P} and \bar{P}_{av} .

It is worth mentioning that the networks tuning is very time consuming. A tuning time can occupy up to 80% of a total investigation time, leaving 20% for the calculations related to final learning and validation of the networks.

7.2. Neural network comparison

The comparison of three tuned networks: MLP, RBN, and PNN, was firstly performed in reference [27], then the SVN was also evaluated. The variations of comparison conditions embraced independent changes of two engines, two operating modes, and two classification variations. The resulting probabilities \bar{P}_{av} are given in **Table 1**. We can see that all networks are practically equal in accuracy for all comparison cases.

Paper [28] provides some additional results extending the comparison on the K-NN technique. The data given in **Table 2** confirm the conclusion about equal performances, now for five different techniques.

Class type	ANN	Engine 1		Engine 3	
		Mode 1	Mode 2	Mode 1	Mode 2
Single	MLP	0.8184	0.8059	0.7338	0.7470
	RBN	0.8186	0.8058	0.7349	0.7485
	PNN	0.8134	0.8004	0.7287	0.7456
	SVN	0.8190	0.8064	-	-
Multiple	MLP	0.8765	0.8686	0.7749	0.7596
	RBN	0.8783	0.8701	0.7787	0.7643
	PNN	0.8739	0.8653	0.7730	0.7617
	SVN	0.8770	0.8698	-	-

Table 1. Results of the network comparison (probabilities computed for Engine 1 and Engine 3).

Technique	Class type	
	Single	Multiple
PNN	0.8134	0.8739
K-NN	0.8154	0.8735
MLP	0.8193	0.8765

Table 2. Additional results of the technique comparison (probabilities for Engine 1).

The PNN and K-NN have probabilistic output, and every pattern recognition decision is accompanied with a confidence probability. This is an important advantage for gas turbine diagnosticians and maintenance staff. It can be taken into account for choosing the best technique when mean diagnosis reliability \bar{P} is equal for all techniques considered. The PNN and K-NN are nonparametric techniques that estimate a probability density for each fault class by counting the patterns that fall into a given volume (window). To accurately estimate the probability density in a multidimensional diagnostic space, the number 1000 of available patterns can be insufficient. To assess possible imprecision of the density and confidence probability estimation by the PNN and K-NN techniques, a more precise analytical density estimation (ADE) technique has been proposed and developed [28]. It analytically determines the density and is employed as a reference to assess imprecision of the PNN and K-NN. To verify the newly developed technique, it was firstly compared with the others by the criterion \bar{P}_{av} . The results were reasonably good: the performances of all the techniques remained very close, but the ADE had the highest probability with the increment of 0.366–0.771 relatively the others.

The results of comparison by the estimated confidence probability are illustrated in **Figure 11**, when the PNN, K-NN, and MLP errors are plotted for 100 patterns. One can see that the bias and scatter for the K-NN estimates are by far greater. As to the MLP outputs, these non-probabilistic quantities look by far more precise than the K-NN probability estimates and seem to have the same precision level as the PW-PNN estimates.

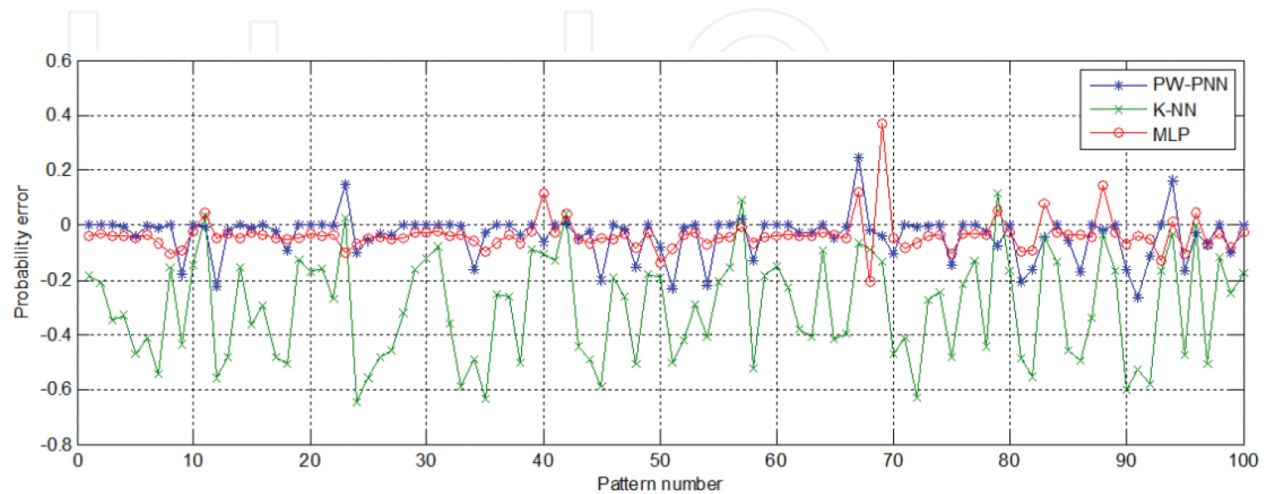


Figure 11. Errors of probability estimation by PW-PNN, K-NN, and MPL techniques (Engine 1, first 100 patterns of the first single fault class).

Table 3 presents the mean estimations errors for the case of the single fault classification. The table data confirm the above conclusion on the compared techniques: The bias and standard deviation of the K-NN errors are by far greater. The table also shows that on average the MLP outputs are even more exact than the PNN probabilities. It is one more argument to apply the perceptron in real gas turbine monitoring systems.

Bias				σ	
PNN	K-NN	MLP	PW-PNN	K-NN	MLP
-0.0444	-0.3293	-0.0419	0.0845	0.2020	0.0791

Table 3. Mean errors of confidence probability estimation (Engine 1, single fault classification).

7.3. Fault classification extension

In the investigations previously described, only two rigid classifications were maintained: one formed by single fault classes and the other constituted from multiple fault classes created by two fault parameters. However, the classification can vary a lot in practice even for the same engine, and it is difficult to predict what classification variation will be finally used in a real monitoring system. To verify and additionally compare the networks for different classification variations, the test procedure was modified for easily creating any new fault classification, more complex and more realistic than the classifications previously analyzed.

Twelve classification variations have been prepared and three networks: MLP, RBN, and PNN, were examined in reference [29]. These classifications have from 4 to 18 gas path and sensor fault classes, 1 to 4 fault parameters to form each class, positive and negative fault parameter changes. All the networks operated successfully for all fault classifications. **Table 4** shows the resulting averaged probabilities of correct diagnosis. Analyzing them, one can state that the differences between the networks within the same classification remain not great (except variation 6), about 0.015 (1.5%), while the difference between the variations can reach the value 0.10. Thus, these results reaffirm once more the conclusion drawn before that many recognition techniques may yield the same gas turbine diagnosis accuracy.

Variation	MLP	RBN	PNN
1	0.8172	0.8169	0.8099
2	0.8732	0.8759	0.8720
3	0.8091	0.8072	0.8037
4	0.8490	0.8524	0.8474
5	0.8033	0.8080	0.8036
6	0.6805	0.7319	0.7316
7	0.7362	0.7616	0.7567
8	0.7828	0.7965	0.7910
9	0.9279	0.9280	0.9260
10	0.7909	0.8017	0.7930
11	0.8075	0.7867	0.7775
12	0.8209	0.8184	0.8076

Table 4. Technique comparison for new classification variations (probabilities \bar{P}_{av} for Engine 1).

7.4. Real data-based classification

Gas path mathematical models are widely used in building fault classification required for diagnostics because faults rarely occur during field operation. In that case, model errors are transmitted to the model-based classification. Paper [30] looks at the possibility of creating a mixed fault classification that incorporates both model-based and data-driven fault classes. Such a classification will combine a profound common diagnosis with a higher diagnostic accuracy for the data-driven classes. Engine 1 has been chosen as a test case. Its real data with two periods of compressor fouling were used to form a data-driven class of the fouling. **Figure 12** illustrates simulated (without errors) and real data.

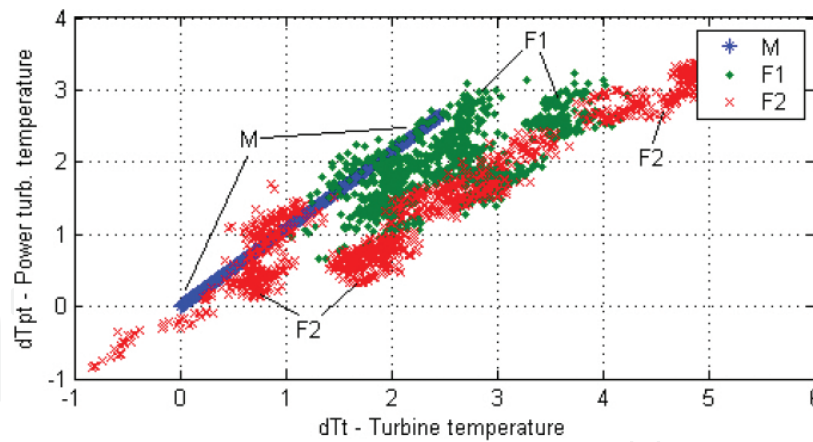


Figure 12. Simulated and real compressor fouling deviations (Engine 1: M—simulated deviations, F1 and F2—real deviations for the first and second fouling periods).

Different variations of the classification were considered and compared using the MLP. In spite of irregular distribution of real patterns, the MLP normally operated at the learning and validation steps. We also found that the perceptron trained on simulated data has 30% recognition errors when applied to real compressor fouling data. However, the use of mixed learning data allows to reduce these errors up to 3%. It was shown as well how to form a representative real fault class, which ensures minimal recognition errors.

Paper [31] presents another way to enhance gas turbine fault classification using real information. Diagnostic algorithms widely use theoretical random number distributions to simulate measurement errors. Such simulation differs from real diagnosis because the diagnostic algorithms work with the deviations, which have other error components that differ from simulated errors by amplitude and distribution. As a result, simulation-based investigations might result in too optimistic conclusions on gas turbine diagnosis reliability. To make error presentation more realistic, it was proposed in reference [31] to extract an error component from real deviations and to integrate it in fault description.

Using simulated and real data of Engine 1, six alternative variations of deviation error were integrated in the fault classification. Diagnosis was performed by the MLP, and the diagnosis reliability was estimated for each variation. Despite irregular real error distribution, the MLP successfully operated for all the variations. Experiments with error representation variations have shown what can happen when the classification formed with accurate simulated deviations is applied to classify less accurate real deviations. In that case, the diagnosis accuracy can fall from $\bar{P} \approx 92\%$ to $\bar{P} \approx 54\%$, but this low diagnostic accuracy can be considerably elevated by including real errors into the description of fault classes.

The fault classifications with integrated real errors were used in reference [32] to compare three networks: MLP, RBN, and PNN, one more time. All networks operated well and they differed in accuracy indicators \bar{P}_{av} by less than 1%, thus confirming again the conclusion about equality of recognition techniques.

7.5. Different operating conditions

Many known studies show that grouping the data collected at different engine operating modes for making a single diagnosis (multipoint diagnosis) yields higher diagnostic accuracy than the accuracy provided by traditional one-point methods. But it is of a practical interest to know how significant the accuracy increment is and how it can be explained. The diagnosis of engines at dynamic modes poses the similar questions. To make one diagnosis, this technique combines data from successive measurement sections of a transient operation mode and in this regard looks like multipoint diagnosis.

Paper [33] analyzes the influence of the operating conditions on the diagnostic accuracy by comparing the one-point, multipoint, and transient options. The MLP is used as a pattern recognition technique. In spite of significant increase of the input dimensionality, the perceptron operated well for all options.

The calculations have revealed that the process of network training has peculiarities for multipoint diagnosis. They are illustrated in **Figure 13**, which shows the plots of the perceptron error versus training epochs for the cases of one-point and multipoint diagnosis. As can be seen, the curves of the error function for the training and validation processes almost coincide for the one-point option, they slow down along with training epochs, and a total epoch number 300 is relatively large. These are indications of no over-training effect. The behavior of the perceptron applied for the multipoint diagnosis is quite different. We can see that the validation curve falls behind the training curve after the 30th epoch, this gap rapidly increases, and the training process stops earlier (108 epochs) because of the over-training phenomenon. We can conclude that the Early Stopping Option is more required here. The differences indicated above can be explained by the ratio of input data volume to the unknown perceptron parameter number. For both cases, the volume of the training set is equal to 7000 patterns, but the numbers of unknown quantities significantly differ: 144 for the first case and 1540 for the second. Consequently, in the case of multipoint diagnosis, the trained network is much more flexible and the over-training becomes possible. An increase of the reference set volume can improve the training process; however, this increase is presently limited by the computation time.

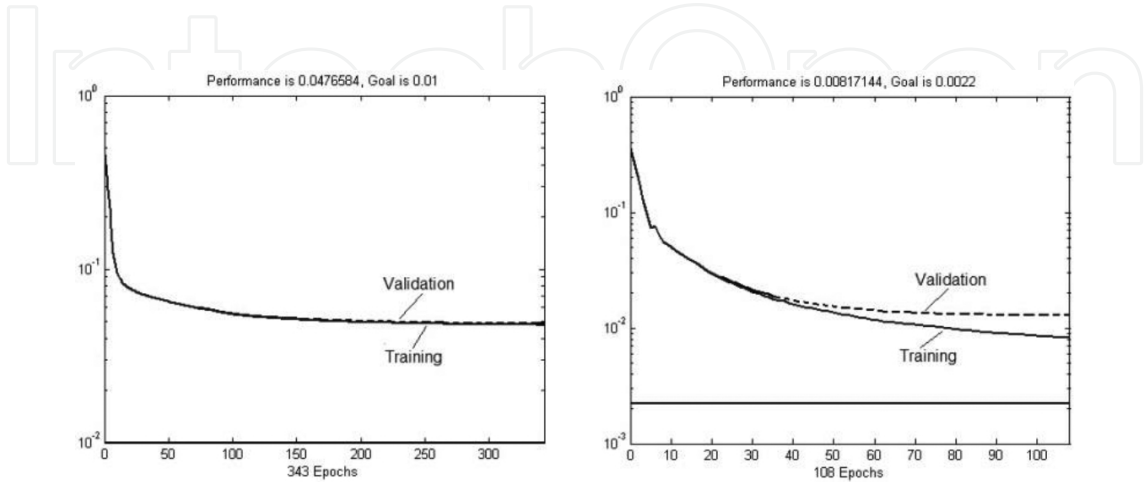


Figure 13. Training process (Engine 1, left plot—one-point diagnosis, right plot—multipoint diagnosis).

The results of the option comparison (probabilities \bar{P}) are grouped in **Table 5**. One can see that a total growth of diagnosis accuracy due to switching to the multipoint diagnosis and data joining from different steady states is significant: The diagnosis errors decrease by two to five times. The diagnosis at transients causes further accuracy growth, but it is not great. It has been found that this positive effect of the data joining is mainly explained by averaging the input data and smoothing the random measurement errors.

Option	Single fault classification	Multiple fault classification
One-point	0.7316	0.7351
Multipoint	0.8915	0.9444
Transient	0.9032	0.9561

Table 5. Comparison of the one-point, multipoint, and transient options (Engine 1).

8. Conclusions

A monitoring system comprises many elements, and many factors influence the final diagnostic accuracy. The present chapter has generalized our investigations aimed to enhance this system by choosing the best option for each element. In every investigation, a diagnostic process was simulated mainly on the basis of neural networks, and we focused on reaching the highest accuracy by choosing the best network and its optimal tuning to the issue to solve. As can be seen, all the examined techniques (MLP, RBN, PNN, SVN, and K-NN) use a pattern-based classification. Such a classification can be formed from complex classes in which faults are simulated by the nonlinear thermodynamic model. Moreover, this classification allows its description by real fault displays that completely exclude a negative effect of model inaccuracy. Thus, being objects of investigation and optimization, neural networks help with enhancement of a whole monitoring system. As a result of the conducted investigations, some methods to elevate diagnostic accuracy were proposed and proven. The chapter also provides the recommendations on choosing and tailoring the networks for different diagnostic tasks. For solving many tasks, the utility of the multilayer perceptron has been proven on simulated and real data.

Acknowledgements

The work has been carried out with the support of the National Polytechnic Institute of Mexico (research project 20150961).

Author details

Igor Loboda

Address all correspondence to: igloboda@gmail.com

National Polytechnic Institute, Mexico

References

- [1] Saravanamuttoo, H.I.H., MacIsaac, B.D., 1983, Thermodynamic models for pipeline gas turbine diagnostics, *ASME Journal of Engineering for Power*, Vol. 105, pp. 875–884.
- [2] Doel, D.L., 2003, Interpretation of weighted-least-squares gas path analysis results, *Journal of Engineering for Gas Turbines and Power*, Vol. 125, Issue 3, pp. 624–633.
- [3] Aretakis, N., Mathioudakis, K., Stamatis, A., 2003, Nonlinear engine component fault diagnosis from a limited number of measurements using a combinatorial approach, *Journal of Engineering for Gas Turbines and Power*, Vol. 125, Issue 3, pp. 642–650.
- [4] Volponi, A.J., DePold, H., Ganguli, R., 2003, The use of Kalman filter and neural network methodologies in gas turbine performance diagnostics: a comparative study, *Journal of Engineering for Gas Turbines and Power*, Vol. 125, Issue 4, pp. 917–924.
- [5] Sampath, S., 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.
- [6] Pipe K., 1987, Application of advanced pattern recognition techniques in machinery failure prognosis for turbomachinery, *Condition Monitoring 1987 International Conference*, British Hydraulic Research Association, UK, pp. 73–89.
- [7] Lokesh Kumar S., et al., 2007, Comparison of a few fault diagnosis methods on sparse variable length time series sequences, *IGTI/ASME Turbo Expo 2007*, Montreal, Canada, 8 p., ASME Paper GT2007-27843.
- [8] Romessis, C., Mathioudakis, K., 2006, Bayesian network approach for gas path fault diagnosis, *ASME Journal of Engineering for Gas Turbines and Power*, Vol. 128, Issue 1, pp. 64–72.
- [9] Fast, M., Assadi, M., De, S., 2008, Condition based maintenance of gas turbines using simulation data and artificial neural network: a demonstration of feasibility, *IGTI/ASME Turbo Expo 2008*, Berlin, Germany, 9 p., ASME Paper GT2008-50768.

- [10] Palme, T., Fast, M., Assadi, M., Pike, A., Breuhaus, P., 2009, Different condition monitoring models for gas turbines by means of artificial neural networks, *IGTI/ASME Turbo Expo 2009*, Orlando, Florida, USA, 11 p., ASME Paper GT2009-59364.
- [11] Palme, T., Breuhaus, P., Assadi, M., Klein, A., Kim, M., 2011, Early warning of gas turbine failure by nonlinear feature extraction using an auto-associative neural network approach, *IGTI/ASME Turbo Expo 2011*, Vancouver, British Columbia, Canada, 12 p., ASME Paper GT2011-45991.
- [12] Butler, S.W., Pattipati, K.R., Volponi, A., et al., 2006, An assessment methodology for data-driven and model based techniques for engine health monitoring, ASME Paper No. GT2006-91096.
- [13] Romessis, C., Mathioudakis, K., 2003, Setting up of a probabilistic neural network for sensor fault detection including operation with component fault, *Journal of Engineering for Gas Turbines and Power*, Vol. 125, pp. 634–641.
- [14] Jaw, L. C., Wang, W., 2006, Mathematical formulation of model-based methods for diagnostics and prognostics, *IGTI/ASME Turbo Expo 2006*, Barcelona, Spain, 7 p., ASME Paper GT2006-90655.
- [15] Borguet, S., Leonard, O., Dewallet, P., 2015, Regression-based modelling of a fleet of gas turbine engines for performance trading, *IGTI/ASME Turbo Expo 2015*, Montreal, Canada, 12 p., ASME Paper GT2015-42330.
- [16] Vachtsevanos, G., Lewis, F.L., Roemer, M., Hess, A., Wu, B., 2006, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, John Wiley & Sons, Inc., New Jersey, 434 p.
- [17] Vatani, A., Korasani, K., Meskin, N., 2015, Degradation prognostics in gas turbine engines using neural networks, *IGTI/ASME Turbo Expo 2015*, Montreal, Canada, 13 p., ASME Paper GT2015-44101.
- [18] Duda, R.O., 2001, *Pattern Classification*, Wiley-Interscience, New York, 654 p.
- [19] Haykin, S., 1994, *Neural Networks*, Macmillan College Publishing Company, New York.
- [20] Bishop, C.M., 2006, *Pattern Recognition and Machine Learning*, Springer Science, New York.
- [21] Cortes, C., Vapnik, V., 1995, Support-vector networks, *Machine Learning*, Vol. 20, pp. 273–297.
- [22] Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992, A training algorithm for optimal margin classifiers, Fifth Annual Workshop on Computational Learning Theory—COLT '92, ACM Press, New York, USA, pp. 144–152.
- [23] Hsu, C.W., Chang, C.C., Lin, C.J., 2010, A practical guide to support vector classification, National Taiwan University. <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

- [24] Loboda, I., Feldshteyn, Y., 2011, Polynomials and neural networks for gas turbine monitoring: a comparative study, *International Journal of Turbo & Jet Engines*, Vol. 28, Issue 3, pp. 227–236 (also see ASME paper GT2010-23749).
- [25] Loboda, I., Yepifanov, S., Feldshteyn, Y., 2009, An integrated approach to gas turbine monitoring and diagnostics, *International Journal of Turbo & Jet Engines*, Vol. 26, Issue 2, pp. 111–126 (also see ASME paper GT2008-51449).
- [26] Loboda, I., Yepifanov, S., Feldshteyn, Y., 2007, A generalized fault classification for gas turbine diagnostics on steady states and transients, *Journal of Engineering for Gas Turbines and Power*, Vol. 129, Issue 4, pp. 977–985.
- [27] Loboda, I., Yepifanov, S., 2013, On the selection of an optimal pattern recognition technique for gas turbine diagnosis, *IGTI/ASME Turbo Expo 2013*, San Antonio, Texas, USA, 11 p., ASME Paper GT2013-95198.
- [28] Loboda, I., 2014, Gas turbine fault recognition using probability density estimation, *ASME Turbo Expo 2014*, Dusseldorf, Germany, 13 p., ASME Paper GT2014-27265.
- [29] Perez Ruiz, J.L., Loboda, I., 2014, A flexible fault classification for gas turbine diagnosis, *Aerospace Techniques and Technology*, Vol. 113, Issue 6, pp. 94–102.
- [30] Loboda, I., Yepifanov, S., 2010, A mixed data-driven and model based fault classification for gas turbine diagnosis, *International Journal of Turbo & Jet Engines*, Vol. 27, Issue 3–4, pp. 251–264 (also see ASME Paper GT2010-23075).
- [31] Loboda, I., Yepifanov, S., Feldshteyn, Y., 2013, A more realistic scheme of deviation error representation for gas turbine diagnostics, *International Journal of Turbo & Jet Engines*, Vol. 30, Issue 2, pp. 179–189 (also see ASME Paper GT2012-69368).
- [32] Loboda, I., Olivares Robles, M.A., 2015, Gas turbine fault diagnosis using probabilistic neural networks, *International Journal of Turbo & Jet Engines*, Vol. 32, Issue 2, pp. 175–192.
- [33] Loboda, I., Feldshteyn, Y., Yepifanov, S., 2007, Gas turbine diagnostics under variable operating conditions, *International Journal of Turbo & Jet Engines*, Vol. 24, Issues 3–4, pp. 231–244 (also see ASME Paper GT2007-28085).