TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 1340 – 1350

doi:10.3906/elk-1110-71

Feature-based fault detection of industrial gas turbines using neural networks

Abbas RASAIENIA,[1,*] **Behzad MOSHIRI,**[2] **Mohammadamin MOEZZI**[3]

[1]Department of Electrical Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2]School of Electrical & Computer Engineering, Center of Excellence for Control and Intelligent Processing, University of Tehran, Tehran, Iran
[3]Department of Instrumentation and Control, Monenco Iran Consulting Engineers Company, Tehran, Iran

Received: 31.10.2011 • **Accepted:** 23.05.2012 • **Published Online:** 12.08.2013 • **Printed:** 06.09.2013

Abstract: Gas turbine (GT) fault detection plays a vital role in the minimization of power plant operation costs associated with power plant overhaul time intervals. In other words, it is helpful in generating pre-alarms and paves the way for corrective actions in due time before incurring major equipment failures. Hence, finding an efficient fault detection technique that is applicable in the online operation of power plants involved with minor computations is an urgent need in the power generation industry. Such a method is studied in this paper for the V94.2 class of GTs. As the most leading stage for developing a feature-based fault detection system and moving from a fixed time-scheduled maintenance to a condition-based one, principal component analysis is used for dimension reduction in the sensor data space and dimensionless key features are employed instead. One healthy condition and 6 faulty conditions are used to provide a realistic data set that is used for feature extraction, training, and testing artificial neural networks. In the proposed method, multilayer perceptron (MLP) and learning vector quantization (LVQ) networks are used for the fault classification. The good performance of the LVQ networks is presented by properly selecting the network architecture and respective initial weight vectors. When comparing the results of the MLP and LVQ networks for the fault classification, the LVQ network shows better classification results.

Key words: Feature extraction, fault detection, gas turbine, principle component analysis, neural network

1. Introduction

Artificial intelligent solutions have been widely adopted for fault detection problems in dynamic systems. The main idea of these approaches is to convert the fault detection problem into an associated optimization problem by introducing a performance index. Intelligent methods such as fuzzy systems and neural networks can learn the plant model from the input-output data or learn from experience instead of being programmed. They can approximate the nonlinear functions in order to construct the analytical model for generating residuals. Moreover, they can be used as classifiers to perform fault detection and analysis [1]. Neural networks are known to approximate any nonlinear function and they are widely used in nonlinear (and robust) fault diagnosis problems [2–7]. They are also able to find the patterns that rule the relationships between the inputs and outputs. Meanwhile, fuzzy logic systems also have the ability to model a nonlinear system and to express it in the form of linguistic rules, making it more transparent, i.e. easier to interpret. They also have inherent abilities to deal with imprecise or noisy data, therefore making them suitable for model-based fault diagnosis [8–13].

*Correspondence: rasaienia.abbas@monenco.com

The development of industrial gas turbine (GT) fault detection configuration has been the aim of many researches in the past years. The most important reasons for employing fault detection in GT engines can be categorized within the following 3 topics:

1. Passage of time and aging reduces the efficiency in the GT package and results in performance reduction.
2. Generally, any undesired change in the performance and efficiency of one subsystem in a GT package will make the other parts operate out of their normal conditions.
3. Because of the large amount of registered alarms and the status in the control system of a power plant, predicting and analyzing the failures that will occur in the future is complicated.

With due consideration of the above mentioned reasons, the predicting and preventing of major failures by analyzing the symptoms of minor faults and the provision of condition-based maintenance schedules is the principal challenge in various maintenance strategies. The literature indicates the level of research carried out in the field of industrial GT fault detection (and feature extraction) as follows:

- Fault detection methods (model-based and data-based methods) and fault-tolerant control systems [14,15];
- Neural networks, soft computing, and principal component analysis (PCA) [16–23];
- Fault detection of GTs [24–32];
- V94.2 GT piping and instrumentation diagram, instrument list, input/output list, and alarm list [33].

In this paper, a feature extraction technique is employed in order to identify the most important symptoms of the V94.2 class of industrial GT engines instead of using all of the sensor data. Since the measuring points in the V94.2 package are known, using static techniques seems to be an appropriate approach for solving this problem. Considering a reasonable amount of instruments installed in the GT engine, the method proposed in this paper attempts to make it possible to generate a limited number of features from the main data set (related to the existing instruments) to use them for classification and detection purposes.

In the schematic structure indicated in Figure 1, the information received from a group of sensors installed in the GT package is collected. Next, by extracting a number of features from the collected data (in healthy and in faulty conditions), it will be possible to implement the fault detection procedure using artificial neural networks [multilayer perceptron (MLP) and learning vector quantization (LVQ) classifiers]. The purpose of this paper is to evaluate and compare the abilities of the MLP and LVQ networks for industrial GT fault detection based on the extracted features.

2. V94.2 GT (case study)

V94.2 is a class of heavy-duty GTs designed for reliable, efficient, and flexible operation. As of March 2011, more than 68 turbines of this type were in operation in the Middle East in order to generate more than 10,000 MW of electrical power. The main technical specifications of this class of GTs are as follows [33]:

- 16 stages-adjustable first stator row;
- 4 stage design, optimum as far as aerodynamic efficiency and losses due to cooling air consumption;

- 2 vertical silo-type large-volume combustors;
- Double-walled flame tube with ceramic tiles;
- 2 × 8 burners (same size-different number for different GT sizes).

Furthermore, the main operational data of the V94.2 GT are mentioned in Table 1.

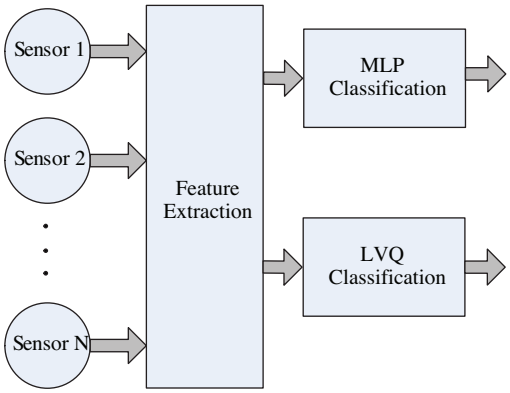

Figure 1. Concept of the feature-based fault detection by neural networks.

Table 1. V94.2 GT general technical data sheet.

GT model	V94.2
Normal output (MW)	159
Frequency (Hz)	50
Efficiency (%)	34.5
Heat rate (BTU/KWh)	9890
Compression ratio	11.1
Exhaust-gas flow (kg/s)	519
Exhaust-gas temperature (°C)	540
NOx emissions at full load (ppm)	
with natural gas	25
with distillate (as defined by US EPA)	42

The data space used for the feature extraction (indicated in Table 2) is obtained by reviewing the list of installed instruments in this class of GT engine in order to remove the redundant data and highlight the independent measurements.

3. Feature extraction by principal component analysis

The goal of PCA is to reduce the dimensionality of the data, while retaining as much of the variation present in the original data set as possible. This method projects the original data onto the orthogonal unit vectors along which the data varies the most.

To illustrate the matter, consider an n-dimensional data space in which the data set consists of $n\times$ *1* $(\underline{\mathbf{x}}_{i,i=1,2,\ldots,k})$ random vectors, where k represents the number of provided samples of $\underline{\mathbf{x}}_i$. For each $\underline{\mathbf{x}}_i$, consider $E(\underline{\mathbf{x}}_i) = \underline{\mathbf{0}}$ and $E(\underline{\mathbf{x}}_i \cdot \underline{\mathbf{x}}_i^T) = \underline{\mathbf{R}}_{\mathbf{x}_i}$, where $E(.)$ denotes the expectation operator and $\underline{\mathbf{R}}_{\mathbf{x}_i}$ is the covariance matrix of $\underline{\mathbf{x}}_i$.

Table 2. Input of the main data set for the feature extraction of V94.2 GT.

Input data of the main data set	Description	Note	Measuring range
a_1	Ambient temperature	1 transmitter	–30 to 70 °C
a_2	Ambient humidity	1 transmitter	0%–100%
a_3	Ambient pressure	1 transmitter	
a_4	Turbine speed (different location-redundant sensors)	6 magnetic field sensors- 2 redundant sets	0.5–25,000 HZ
a_5	Output power	2 measurement	0–160 MW
a_6	Inlet guide vane position	1 measurement	0–90°
a_7	Compressor inlet pressure	1 measurement	0–1300 mbar abs
a_8	Compressor outlet pressure	2 measurement-redundant	0–30 bar
a_9	Compressor inlet temperature	3 RTD	0–400 °C
a_{10}	Compressor outlet temperature	2 RTD	0–350 °C
a_{11}	Turbine outlet temperature (calculation)	Calculated in control system	
a_{12}	Fuel gas volumetric flow	2 measurement	200 – 4000 m^3/h
$a_{13,\ ...,\ a_{20}}$	Bearing temperature	8 triplex thermocouples for turbine/generator and compressors	0–200 °C

Let $\underline{\mathbf{u}}_{j,(j=1,2,\ldots,n)}$ be $n\times$ *1* orthonormal vectors, which are the unit vectors of the data space while representing the direction of the maximum data variation. The projection of $\underline{\mathbf{x}}_i$ onto $\underline{\mathbf{u}}_j$ is defined by their inner product as Eq. (1):

$$y_j = \underline{\mathbf{x}}_i^T \cdot \underline{\mathbf{u}}_j = \underline{\mathbf{u}}_j \cdot \underline{\mathbf{x}}_i^T. \tag{1}$$

The variance of the projected data along each $\underline{\mathbf{u}}_j$ is represented as Eq. (2):

$$\delta_{yj}^2 = \underline{\mathbf{u}}_j^T \cdot E(\underline{\mathbf{x}} \cdot \underline{\mathbf{x}}^T) \cdot \underline{\mathbf{u}}_j = \underline{\mathbf{u}}_j^T \cdot \underline{\mathbf{R}}_{\mathbf{x}} \cdot \underline{\mathbf{u}}_j, \tag{2}$$

where $\underline{\mathbf{x}}_{n\ \times\ k} = [\ \underline{\mathbf{x}}_1 \underline{\mathbf{x}}_2 \ldots \underline{\mathbf{x}}_k\ ]$.

Considering that PCA is seeking the direction along which the variance of the original data is maximized, an optimization problem should be solved in order to find the desired direction of the maximum variance ($\underline{\mathbf{u}}$). If the Lagrange multiplier optimization method is used, the related cost function J will be defined as Eq. (3):

$$J = \underline{\mathbf{u}}^T \cdot \underline{\mathbf{R}}_{\mathbf{x}} \cdot \underline{\mathbf{u}} + \alpha(1 - \underline{\mathbf{u}}^T \cdot \underline{\mathbf{u}}), \tag{3}$$

where α is named as a Lagrange multiplier.

By solving the above problem, Eq. (4) will be obtained.

$$\underline{\mathbf{R}}_{\mathbf{x}} \underline{\mathbf{u}} = \alpha \underline{\mathbf{u}}. \tag{4}$$

From Eq. (4), it is realized that α is an eigenvalue of the covariance matrix and $\underline{\mathbf{u}}$ is the corresponding eigenvector. From Eqs. (2) and (4), the optimum α will be calculated as in Eq. (5):

$$\delta_{yj}^2 = \alpha. \tag{5}$$

From Eq. (5), it is inferred that the variances of the projected data are equal to the eigenvalues of the covariance matrix of the main data set. Hence, the m greatest variances will lie along the direction of the m related eigenvectors, which are considered as features.

4. MLP and LVQ neural networks

The MLP and LVQ networks belong to the family of feedforward neural networks, which have different structures and different training algorithms.

MLP is formed by 1 input layer, 1 or more hidden layers, and 1 output layer. The most commonly used method of MLP training is the backpropagation algorithm. The simplest implementation of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly, i.e. the negative of the gradient. MLP weights and biases are reinforced as in Eq. (6):

$$\mathbf{V}_{k+1} = \mathbf{V}_{k} - \eta \cdot \boldsymbol{g}_k, \tag{6}$$

where $\mathbf{V} = [\mathbf{w}\ \mathbf{b}]$ is a vector of weights ($\mathbf{w}$) and biases ($\mathbf{b}$), $\boldsymbol{g}_k$ is the current gradient, and η is the learning rate. There are 2 different ways in which a gradient descent algorithm can be implemented: incremental mode and batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode, all of the inputs are applied to the network before the weights are updated [16,18].

LVQ is a competitive learning network that utilizes an unsupervised learning method to solve supervised learning tasks. LVQ includes 2 layers: the competitive layer and the linear layer. The competitive layer is learned to classify the input patterns and the linear layer transforms the output of the competitive layer into the target class [16–18].

In the vector quantization technique, the input space is divided into distinct regions. A reconstruction vector is considered for each region. When an input is fed into the network, a vector quantizer determines the region where the vector lies. Next, a reconstruction vector is provided by the quantizer.

Let $\mathbf{X}_n$ be the data set in an n-dimensional space and k the number of principal components that represent the most data set variation. When the training pattern $\mathbf{u}^k$ from class F_j is loaded to the network, the reconstruction vector ($\mathbf{w}_P$) of the P nodes of the competitive layer is reinforced according to the following supervised rule (Eq. (7)) if the class of winner node P equals the desired class F_j:

$$\mathbf{w}_{P,\ New} = \mathbf{w}_P + \eta(\mathbf{u}^k - \mathbf{w}_P). \tag{7}$$

Otherwise,

$$\mathbf{w}_{P,\ New} = \mathbf{w}_P - \eta(\mathbf{u}^k - \mathbf{w}_P), \tag{8}$$

where η is the learning rate monotonically decreasing with time. The decreasing learning rate allows the network to converge to a point at which the weight vectors are stable. One of the good effects of the LVQ network is to minimize the misclassified points.

5. V94.2 GT feature extraction by PCA and fault detection by neural networks

According to Table 2, the input data space contains 20 measurement variables, including sensor-measured variables and a variable computed in the control system. While these variables are measured and computed

under GT healthy performance conditions, they are called healthy data. Thus, each healthy condition ($\underline{\mathbf{x}}_i$) is defined as Eq. (9), where each $a_{ij,(j\ =\ 1,\ 2,\ ...,\ n)}$ is a healthy datum:

$$\underline{\mathbf{x}}_i = [a_{i1} a_{i2} \quad ... \quad a_{i20}]^T, i\ =\ 1,\ 2,\ ...,\ k\ . \tag{9}$$

The main healthy data set is generated using the V94.2 engine simulator. The variable parameters are the ambient temperature, humidity, and pressure for the different load operations of the GTs (using gas as fuel). In this case, a matrix containing 20 $\times$ 15,000 samples (k = 15,000) is generated for the healthy conditions.

After the healthy data are generated, all of them are normalized between –1 and +1. Since in the PCA method it is important to centralize the data, a correction is made to the data sets by subtracting their mean value from each data value before applying the algorithm. In order to simulate the realistic data, white Gaussian noise with 5% variance is considered to represent the expected sensor noise for all of the parameters (including the calculated or measured parameters).

By performing PCA, the lower dimensional representation of the healthy data will be as in Eq. (10):

$$\underline{\mathbf{x}}_i = c_1 \cdot \underline{\boldsymbol{u}}_1 + c_2 \cdot \underline{\boldsymbol{u}}_2 + \ \ldots + c_m \cdot \underline{\boldsymbol{u}}_m, \tag{10}$$

where m is the number of features that are obtained after extracting the m greater eigenvalues from the covariance matrix computed for the healthy data set. and $\underline{\boldsymbol{u}}_j$s are their corresponding eigenvectors.

According to the diagram depicted in Figure 2, the data variation around the eigenvectors is computed for m = 1, ..., 4. This value is 43.54% for the single feature extraction, 82.11% in the case of the 2-feature extraction, and 99.14% while extracting 4 features. In fact, this diagram confirms the sufficiency of selecting 4 principal components as the 4 features instead of employing the collected sensor data.

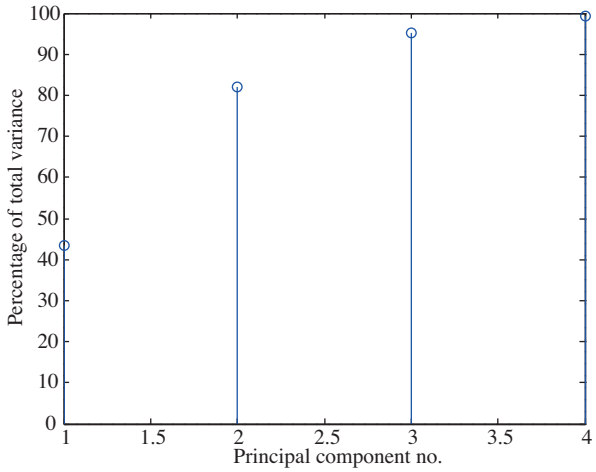

Figure 2. Percentage of variance captured as a number of principal components.

It is possible to express any of the 4 (m = 4) eigenvectors corresponding to the 4 greatest eigenvalues of the computed matrix (i.e. $\underline{\mathbf{u}}_{j,\ j\ =\ 1,\ 2,\ 3,\ 4}$) as a linear combination of the sensor data. This is indicated in Table 3, where each column contains the calculated constant coefficients (weight factors) by which the effect of each sensor datum (a_i) on the given feature ($\underline{\mathbf{u}}_j$) can be represented. It can be observed that extracted features are severely influenced by ambient temperature, turbine speed, compressor inlet and outlet temperature/pressure, inlet guide vane angle, and fuel gas volumetric flow, while they weakly depend on the bearing temperatures.

Table 3. Relation of each feature with the sensor data of V94.2 GT.

Feature no / Sensor data	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$	$\mathbf{u}_4$
a_1	0.3057	–0.1215	0.8256	–0.0348
a_2	0.2101	–0.0605	–0.0976	–0.1096
a_3	0.1660	–0.0567	–0.0709	–0.0983
a_4	0.3969	–0.1212	–0.1434	–0.2621
a_5	0.3298	–0.0425	–0.0312	–0.0382
a_6	0.3776	–0.1089	–0.1353	–0.2574
a_7	0.2750	0.4601	0.0048	0.3443
a_8	0.2081	0.3381	0.0758	0.5091
a_9	0.2694	–0.5939	–0.1007	0.5901
a_{10}	0.1877	0.2888	–0.4245	0.0197
a_{11}	0.2161	0.1445	–0.0594	0.0893
a_{12}	0.1912	0.3951	0.2220	–0.2486
a_{13}	–0.0434	0.0154	0.0149	0.0411
a_{14}	–0.0018	0.0015	–0.0023	–0.0030
a_{15}	0.0454	–0.0201	–0.0229	–0.0161
a_{16}	–0.0022	0.0035	0.0002	0.0062
a_{17}	–0.0100	0.0038	0.0004	–0.0122
a_{18}	0.2995	–0.0936	–0.1096	–0.1846
a_{19}	0.1250	–0.0319	–0.0407	–0.0764
a_{20}	0.0811	–0.0215	–0.0250	–0.0496

Although the data required for the feature extraction are obtained under healthy operating conditions, in order to investigate the possibility of fault detection by the MLP and LVQ neural networks, it is necessary to determine a group of frequent faults for which timely fault detection is valuable. In the V94.2 industrial GT, this group of faults is considered based on the recommendation of the engine manufacturer (TUGA, under the license of Siemens and Ansaldo), as in Table 4.

Table 4. Fault selection of the V94.2 GT.

Notation	Class	Description
F1	Fault 1	Air filtering in the compressor section
F2	Fault 2	Compressor fouling
F3	Fault 3	Turbine blade aging
F4	Fault 4	Turbine fouling
F5	Fault 5	Combustor fouling
F6	Fault 6	Combustor cracked liner
H	Healthy	Healthy condition (no fault)

According to the diagram depicted in Figure 1, the extracted features ($\underline{\mathbf{u}}_{j,\ j\ =\ 1,\ 2,\ 3,\ 4}$) related to each class of faults (F_j) are used to train the MLP and LVQ networks.

In order to perform the fault detection, a matrix containing 20 × 15,000 samples (which are recorded in the selected power plant's data logger) is collected for all 6 classes of the above-mentioned faults and a new feature-based matrix containing 4 × 15,000 samples is calculated from the first matrix. Next, 4 × 11,900 samples (including 4 × 10,200 faulty data and 4 × 1700 healthy data) are used for the training of the MLP

and LVQ classifiers and 4 × 5600 samples (including 4 × 4800 faulty data and 4 × 800 healthy data) are also used for testing the classifiers and validating the whole configuration.

The $\underline{\mathbf{u}}_{j,\ j\ =\ 1,\ 2,\ 3,\ 4}$ features are the input nodes of the MLP and LVQ networks. The output layer consists of neurons that represent the faulty and healthy classes for both classifiers. The MLP network structure contains 2 hidden layers with 16 and 20 neurons that correspond to the satisfactory results of the training. The LVQ network includes 21 subclasses (the effect of the changes on the number of subclasses is checked and the best training/testing errors are found to be related to 21 subclasses). The training algorithm for the MLP network is backpropagation implemented in the MATLAB toolbox (the error goal value is set to 0.01, which was experimentally found to be satisfactory).

The initial vectors of the LVQ network are selected based on the extracted features' probability distribution. The initial learning rate of the LVQ network is set to 0.15. Figure 3 shows the training error of the MLP network (with the backpropagation algorithm) as a function of the epochs number. It is obvious that the training error reaches the error goal after 600 epochs.

In Figure 4, the training error of the LVQ is demonstrated as a function of the epochs number. It is seen that after 57 epochs, the training error reaches 0.

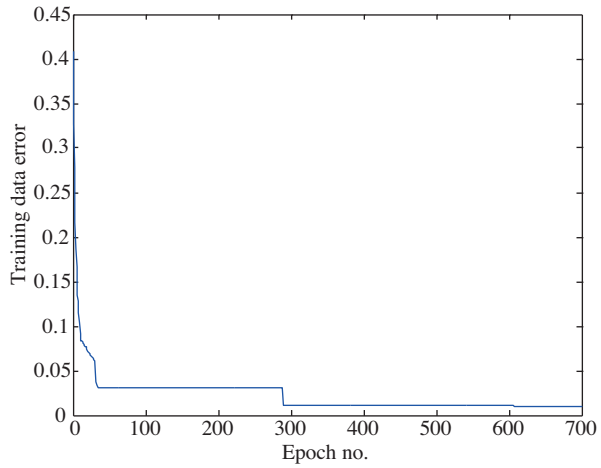

Figure 3. Training error of the MLP network as a function of the Epochs number.

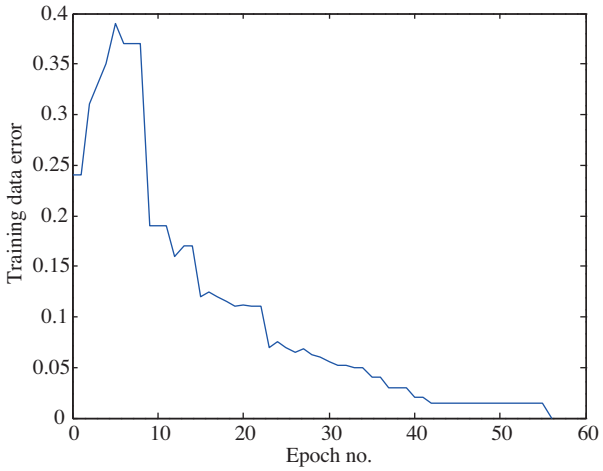

Figure 4. Training error of the LVQ network as a function of the Epochs number.

It is concluded that the MLP algorithm is a slower and less accurate method for reaching the specified error goal in comparison with the LVQ network. The fault classification results of both neural networks are indicated in Tables 5 and 6 using the matrix of the test data.

From Tables 5 and 6, it is clear that the performance of the LVQ classifier is more accurate than that of the MLP network with the backpropagation algorithm in detecting and separating all of the faulty and healthy conditions, while the MLP network shows good results only for the classification of fault 5 (combustor fouling) and fault 6 (combustor cracked liner). Furthermore, it is obvious that the percentage of misclassified conditions (conditions that are not correctly classified) is considerable compared to the results of the LVQ network.

As an important item, it is clearly seen that the classification of faults 1 and 2 (air filtering in the compressor section and compressor fouling) are not basically possible with the MLP network because of the high percentage of unclassified conditions.

Table 5. Performance of the MLP classifier on the GT test data.

Condition classified (%) input fault	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6	H
Fault 1	0.875	0.02125	0	0	0	0	0.10375
Fault 2	0.0325	0.85625	0	0	0	0	0.11125
Fault 3	0	0	0.91375	0.05375	0	0	0.0325
Fault 4	0	0	0.03875	0.95625	0	0	0.005
Fault 5	0	0	0.01375	0	0.9725	0.01125	0.0025
Fault 6	0	0	0	0	0.01	0.99	0
H	0.04625	0	0	0	0	0.00375	0.95

Table 6. Performance of the LVQ classifier on the GT test data.

Condition classified (%) input fault	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6	H
Fault 1	0.995	0	0	0	0	0	0.005
Fault 2	0.0025	0.99375	0	0	0	0	0.00375
Fault 3	0	0	0.94375	0.0375	0	0	0.01875
Fault 4	0	0	0.01	0.9775	0	0	0.0125
Fault 5	0	0	0	0	0.9525	0.03125	0.01625
Fault 6	0	0	0	0	0.00625	0.975	0.01875
H	0	0.00125	0.0025	0	0	0	0.99625

By considering the essence of the defined faults and analyzing their classification condition rates, it is realized that detection of fault 3 (turbine blade aging) and fault 5 (combustor fouling) by the LVQ classifier is less accurate, but is acceptable. Hence, implementation of the extracted features (based on the sensor data) and LVQ fault detection logic is practically possible inside the V94.2 control system software with reasonable accuracy. The overall performance of the LVQ fault classifier can be improved by a fusion structure combining the results of both classifiers (MLP and LVQ), which is now under investigation by the authors.

6. Conclusion

The current research claims to introduce a novel feature-based fault detection configuration for V94.2 GTs. The study involved the feature extraction of GTs using PCA. Four principal components were extracted by the PCA method and represented as linear combinations of the sensor data. The MLP and LVQ neural networks were evaluated for V94.2 GT fault detection, which is regarded to be one of most difficult detection problems, using 20 instruments and 6 typical faults.

The LVQ architecture provides the advantage of a simple topology that is faster and simpler than the MLP with backpropagation algorithm. The performance of the LVQ classifier shows reasonable error (compared to the MLP classifier), which is acceptable in power plant fault detection systems. Therefore, this configuration allows the operator to improve the performance of the V94.2 GT by setting minor and major inspection times for the V94.2 components based on the detected conditions.

Notations

a_i	Sensor data
a_{ij}	Healthy data
α	Eigenvalue of the covariance matrix $\underline{\mathbf{R}}_{\mathbf{x}}$
$\mathbf{b}$	Bias vector
c_j	Constant coefficient
η	Learning rate
$\boldsymbol{g}$	Gradient in MLP
J	Cost function
m	Number of greatest eigenvalues of covariance matrix $\underline{\mathbf{R}}_{\mathbf{x}}$ selected for PCA
n	Dimension of data space
$\underline{\mathbf{R}}_{\mathbf{x}_i}$	Covariance matrix of $\underline{\mathbf{x}}_i$
δ^2_{yj}	Variance of the projected data along each $\underline{\mathbf{u}}_j$
$\underline{\mathbf{u}}$	Data space unit vector
$\mathbf{w}$	Weights vector
$\underline{\mathbf{x}}_i$	$n\times$ 1 random vector
y_j	Projection of $\underline{\mathbf{x}}_i$ onto $\underline{\mathbf{u}}_j$

References

[1] R.F. Stengel, "Toward intelligent flight control", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, pp. 1699–1717, 1993.

[2] T. Sorsa, H.N. Koivo, H. Koivisto, "Neural networks in process fault diagnosis", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, pp. 815–825, 1991.

[3] Y. Maki, K.A. Loparo, "A neural network approach to fault detection and diagnosis in industrial processes", IEEE Transactions on Control Systems Technology, Vol. 5, pp. 529–541, 1997.

[4] M. Ayoubi, R. Isermann, "Neuro-fuzzy systems for diagnosis", Fuzzy Sets and Systems, Vol. 89, pp. 289–307, 1997.

[5] M.M. Polycarpou, A. Vemuri, "Learning methodology for failure detection and accommodation", IEEE Control Systems Magazine, Vol. 15, pp. 16–24, 1995.

[6] A. Vemuri, M.M. Polycarpou, "Robust nonlinear fault diagnosis in input-output systems", International Journal of Control, Vol. 68, pp. 343–360, 1997.

[7] M. Demetriou, M.M. Polycarpou, "Incipient fault diagnosis of dynamical systems using online approximators", IEEE Transactions on Automatic Control, Vol. 43, pp. 1612–1617, 1998.

[8] K.M. Passino, P.J. Antsaklis, "Fault detection and identification in an intelligent restructurable controller", Journal of Intelligent and Robotic Systems, Vol. 1, pp. 145–161, 1988.

[9] E.G. Laukonen, K.M. Passino, V. Krishnaswami, G.C. Luh, G. Rizzoni, "Fault detection and isolation for an experimental internal combustion engine via fuzzy identification", IEEE Transactions on Control Systems Technology, Vol. 3, pp. 347–355, 1995.

[10] H. Schneider, P. Frank, "Observer based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation", IEEE Transactions on Control Systems Technology, Vol. 4, pp. 274–282, 1996.

[11] P.M. Frank, B. Koppen-Seliger, "Fuzzy logic and neural network application to fault diagnosis", International Journal of Approximate Reasoning, Vol. 16, pp. 67–88, 1997.

[12] R. Isermann, "On fuzzy logic applications for automatic control, supervision and fault diagnosis", IEEE Transactions on Systems, Man, and Cybernetics, Part A, Vol. 28, pp. 221–235, 1998.

[13] W.A. Kwong, K.M. Passino, E.G. Laukonen, S. Yurkovich, "Expert supervision of fuzzy learning systems for fault-tolerant aircraft control", Proceedings of the IEEE, Special Issue on Fuzzy Logic in Engineering Applications, Vol. 83, pp. 466–483, 1995.

[14] Y. Zhang, J. Jiang, "Bibliographical review on reconfigurable fault - tolerant control systems", Annual Reviews in Control, Vol. 32, pp. 229–252, 2008.

[15] M. Basseville, I. Nikiforov, "Detecting changes in signals and systems a survey", Automatica, Vol. 24, pp. 309–326, 1993.

[16] S. Haykin, Neural Networks: A Comprehensive Foundation, New York, Macmillan Publishers, 1994.

[17] T. Kohonen, Self-Organization and Associative Memory, 2nd ed., Berlin, Springer-Verlag, 1987.

[18] H. Demuth, M. Beale, Neural Network Toolbox (for use with MATLAB), MathWorks, Inc. User's Guide Version 3, 2000.

[19] K. Diamantaras, S. Kung, Principal Component Analysis Neural Networks: Theory and Applications, New York, Wiley, 1996.

[20] A.M. Martinez, A.C. Kak, "PCA versus LDA", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, pp. 228–233, 2001.

[21] J. Yang, Z. Jind, J. Yang, D. Zhang, A.F. Frangi, "Essence of kernel Fisher discriminant: KPCA plus LDA", Journal of Pattern Recognition, Vol. 37, pp. 2097–2100, 2004.

[22] Y. Li, M.J. Pont, N.N. Jones, J.A. Twiddle, "Using MLP and RBF classifiers in embedded condition monitoring and fault diagnosis applications", Transactions of the Institute of Measurement and Control, Vol. 23, pp. 313–339, 2001.

[23] S. Simani, C. Fantuzzi, P.R Spina, "Application of a neural network in gas turbine control sensor fault detection", Proceedings of the 1998 IEEE International Conference on Control Applications, Trieste.

[24] C. Li, J. Jeng, "Multiple sensor fault diagnosis for dynamic processes", ISA Transactions, Vol. 49, pp. 415–432, 2010.

[25] M. Zedda, R. Singh, "Neural-network-based sensor validation for gas turbine test bed analysis", Proceedings of the IMechE, Part I: Journal of Systems and Control Engineering, Vol. 215, pp. 47–56, 2001.

[26] S.O.T Ogaji, R. Singh, "Advanced engine diagnostic using artificial neural networks", Applied Soft Computing, Vol. 3, pp. 259–271, 2003.

[27] R. Verma, N. Roy, R. Ganguli, "Gas turbine diagnostics using a soft computing approach", Applied Mathematics and Computation, Vol. 172, pp. 1342–1363, 2005.

[28] W.Z. Yan, J.C. Li, K.F. Goebel, "On improving performance of aircraft engine gas path fault diagnosis", Transactions of Institute of Measurement and Control, Vol. 31, pp. 275–291, 2009.

[29] N.H. Afgan, M.G. Cavalho, P.A. Pilavachi, A. Tourlidakis, G.G. Olkhonski, N. Martins, "An expert system concept for diagnosis and monitoring of gas turbine combustion chambers", Applied Thermal Engineering, Vol. 26, pp. 766–771, 2006.

[30] M. Fast, M. Assadi, S. De, "Development and multi-utility of an ANN model for an industrial gas turbine", Applied Energy, Vol. 86, pp. 9–17, 2009.

[31] Y.G. Li, P. Nilkitsaranont, "Gas turbine performance prognostic for condition-based maintenance", Applied Energy, Vol. 86, pp. 2152–2161, 2009.

[32] J. Arriagada, M. Genrup, A. Loberg, M. Assadi, "Fault diagnosis system for an industrial gas turbine by means of neural networks", Proceedings of the International Gas Turbine Congress, pp. 2–7, 2003.

[33] Tuga Co. Publication Operation and Maintenance Manuals for V94.2 Gas Turbines, Damavand Combined Cycle Power Plant Archive, Revision D, 2008.