Smart Structures and Systems, *Vol. 23, No. 4 (2019) 393-403*
DOI: https://doi.org/10.12989/sss.2019.23.4.393

Faults detection and identification for gas turbine using DNN and LLM

Seyyed Mohammad Emad Oliaee[1a], Mohammad Teshnehlab[*1] and Mahdi Aliyari Shoorehdeli[2b]

[1]*Control Engineering Dept., Electrical Faculty, K.N.Toosi University of Technology, Tehran, Iran*
[2]*Mechatronics Dept., Electrical Faculty, K.N.Toosi University of Technology, Tehran, Iran*

(Received July 21, 2018, Revised March 4, 2019, Accepted March 9, 2019)

Abstract. Applying more features gives us better accuracy in modeling; however, increasing the inputs causes the curse of dimensions. In this paper, a new structure has been proposed for fault detecting and identifying (FDI) of high-dimensional systems. This structure consist of two structure. The first part includes Auto-Encoders (AE) as Deep Neural Networks (DNNs) to produce feature engineering process and summarize the features. The second part consists of the Local Model Networks (LMNs) with LOcally LInear MOdel Tree (LOLIMOT) algorithm to model outputs (multiple models). The fault detection is based on these multiple models. Hence the residuals generated by comparing the system output and multiple models have been used to alarm the faults. To show the effectiveness of the proposed structure, it is tested on single-shaft industrial gas turbine prototype model. Finally, a brief comparison between the simulated results and several related works is presented and the well performance of the proposed structure has been illustrated.

Keywords: curse of dimension; local model network; deep neural network; LOLIMOT; fault detection; gas turbine

1. Introduction

The neuro-fuzzy network is one of the common structures for FDI and modeling systems (Nelles 2006). This structure simultaneously overcomes the weaknesses such as training capability in fuzzy systems and interpretability in neural networks. The LMNs is based on this structure (Nelles 2013). In (Adeniran and El Ferik 2017) a complete review of the LMNs and its training algorithms in identifying and modeling nonlinear systems is presented.

Despite; using the fuzzy systems are widespread, the curse of dimensions is a big trap in illicit models that causing inefficiency in these systems. This inefficiency is a result of both computation costs and the challenge of producing fuzzy rules (Mutlu *et al.* 2016). By increasing the number of input features, the input space becomes much larger and consequently, the search space and the needed samples as training data increase exponentially (Alizadeh *et al.* 2016). Therefore, many researchers have tended to learn algorithms that are able to extract good features from high-dimensional systems.

Mammal's brain is wired in a way that can analyze high-dimensional input very well and offer appropriate answers. So in recent years, scientists were enthusiastic to design the deep neural networks by getting inspired from the human brain structure. But the challenge of the training DNNs led to its inefficiency until 2006 (Hinton and Salakhutdinov 2006). In order to overcome this problem, the idea of greedy unsupervised learning has been proposed for each layer (Bengio *et al*. 2007). So, we utilize AE as DNN with the Stochastic Gradient Descent (SGD) training algorithm to extract wealthy features and eliminate the curse of dimension in LMNs.

On the other hand in the recent years, the FDI has received considerable attention in various fields, such as *in civil engineering:* The (Yi *et al.* 2017) has been proposed a two-phase method for the outlier detection of Global Positioning System (GPS) monitoring data. An innovative sensor fault diagnosis in statistic method has been proposed in (Huang *et al*. 2016) to a benchmark structure developed for bridge health monitoring. Also (Huang *et al*. 2015) has presented a sensor-FDI approach by Principal-component analysis (PCA) with application to bridge health monitoring. Moreover three types of faults has been considered in (Chang *et al.* 2017) including the additive, multiplicative, and slowly drifting faults for health monitoring in the field of civil engineering

In mechanical engineering: (Jung and Koh 2014) has interduced a new method, based on multiscale wavelet scalogram (MWS) features to fault detection and condition monitoring of various damage-level scenarios for a bearing system. In (Shen *et al.* 2014), a new structure consisting of support vector regression machines (SVRMs) has been proposed to recognize bearing fault patterns and track the fault sizes.

Furthermore especially *in industrial power generation* (gas turbine (Pourbabaee *et al*. 2013) and wind turbine(Saleh *et al.* 2016)), and etc.

The paper (Yi *et al.* 2017) provides a comprehensive review in kind of sensor fault and its methodology in detail based on the categorization.

∗Corresponding author, Professor
E-mail: teshnehlab@eetd.kntu.ac.ir
[a] PhD. Student
E-mail: emad.oliaee@ee.kntu.ac.ir
[b] Assistant Professor
E-mail: aliyari@eetd.kntu.ac.ir

ISSN: 1738-1584 (Print), 1738-1991 (Online)

This warm embrace, illustrates the importance of FDI in practical applications.

Therefore, in this paper, a two-part structure of AE as DNN and local linear model (LLM) as LMN is proposed. In this proposed structure, the proper performance of AE in the feature engineering has caused the better performance of LMN and consequently the better results in fault detection and identification of gas turbine as compared to other papers.

Hence, in the next section, AE and its training algorithm are introduced. The input space partitioning algorithms in LMN structure, such as LOLIMOT, are considered in the third part. In part four, the proposed structure for high-dimensional systems is presented. Later, in part 5, the proposed structure is utilized to detect and identify the faults of gas turbine, and its result is compared to the other papers. Finally, the conclusion is offered in part 6.

2. Auto-encoder family

After solving the challenges in training layers of DNNs in 2006, various deep learning methods have been developed in the past years. Convolution Neural Network (CNN), Restricted Boltzmann (RBM), AE and Recurrent Neural Network (RNN) are the most typical deep learning models (Zhang *et al.* 2018). Since AE has a simpler structure and algorithm, so several types of AE have been introduced in recent years (Fig. 1).

Traditional AE (Fig. 2) has two layers (encoder and decoder). It is very similar to conventional neural networks. The Input vector (feature) and output vector (feature) are the same as X. For this reason, its training method is called an unsupervised learning algorithm. Usually, the activation function is the sigmoid (logsig) in the encoder (hidden layer) and is the purline in the decoder (output layer). In AE's structure, the input features are projected to a new space arbitrarily (and often less-dimensional) by the encoder, and then, the decoder attempts to reconstruct the input in the output.

The cost function is minimizing reconstruction (representation) error as Eq. (1). where θ is the parameters of network as the weights ($\mathbf{W}^i$) and biases ($\mathbf{b}^i$) in i-th layer. Usually, the error back propagation method based on

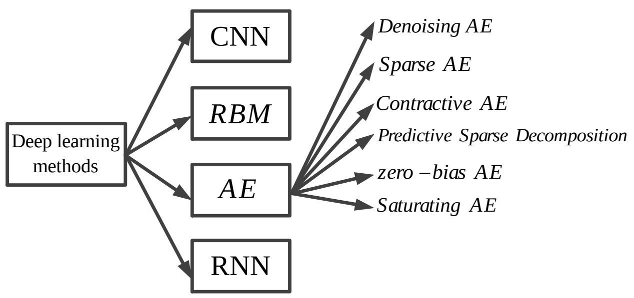


Fig. 1 Categorization of the deep learning methods and AE family

gradient descend is utilized to find θ. Finally, the Stacked Auto-Encoder (SAE) is created by serializing the encoder parts (Fig. 3). In other words, SAE is a deep neural network with number of hidden layers which has trained greedily (Erhan *et al*. 2009).

$$J_{AE}(\theta) = \sum_t \left(x^t - O^2\right)^T \left(x^t - O^2\right) = \sum_t \left(x^t - purline\left(W^2 logsig\left(W^2 x^t + b^1\right)\right)\right)^T \left(x^t - purline\left(W^2 logsig\left(W^2 x^t + b^1\right)\right)\right) \tag{1}$$

Recently, different methods were proposed to train AE parameters with respect to constraints. In (Vincent *et al.* 2008), Denoising AE (DAE) offered to protect the network against noise. It try to reconstruct the clean input in output from the corrupted (noised) input.

Also Sparse AE introduced to make available the volume of the hidden layer. The idea of tied weight was the earliest uses as well as the sparsity regularization (Poultney e*t al.* 2007). Although the L1 penalty (Eq. (2)), the Student-t penalty (Eq. (3)), average output penalty (Eq. (4)) were other sparsity regularization that added terms to the cost function(Eq. (1)), respectively (Bengio *et al.* 2013). Another well-known alternative to sparse regularization was Kullback-Liebler divergence (Eq. (5)) (Ng 2011) that turned off the neurons of the hidden layer whose has action less than the threshold. In this equation $\hat{\rho}_j$ is the average activation of neuron j for a training set (Eq. (6)) and ρ is the sparsity parameter, which is usually small value.

$$+\left\|O_j^1\right\|^1 \tag{2}$$

$$+\log\left(1+O_j^{1^2}\right) \tag{3}$$

$$+\left\|\overline{O^1}\right\|^1 \tag{4}$$

$$+\sum_{i=1}^{n} KL\left(\rho \parallel \hat{\rho}_j\right) = \sum_{i=1}^{n} \rho \log \frac{\rho}{\hat{\rho}_j} + \left(1-\rho\right) \log \frac{1-\rho}{1-\rho_j} \tag{5}$$

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} O_j^1\left(x^i\right) \tag{6}$$

The Contractive Auto-Encoder (CAE), was presented to robust representation in following DAE by (Rifai *et al.* 2011). CAE achieve this robustness by adding Frobenius norm of the encoder's Jacobian to the cost function, while DAE carry out the robustness by injecting noise into the training set.

A practically successful method of sparse coding and AE called Predictive Sparse Decomposition (PSD) (Kavukcuoglu *et al*. 2010), whose uses a fast non-iterative approximator rather than costly and nonlinear encoding step. The PSD has been utilized to object recognition in images and video(Kavukcuoglu *et al.* 2009).

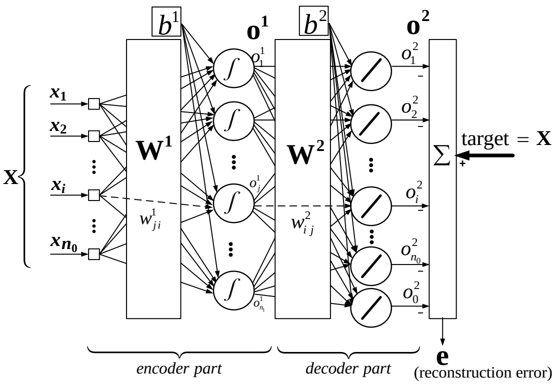


Fig. 2 Structure of Typical Auto-Eencoder

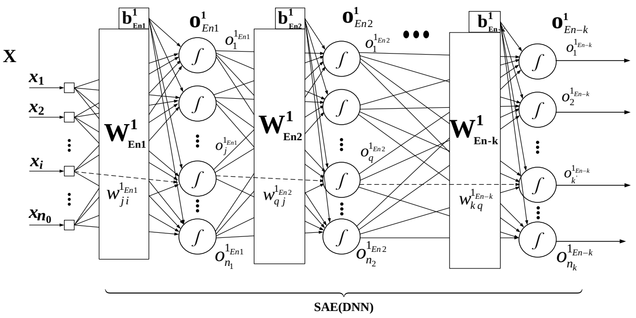


Fig. 3 Structure of Stacked Auto-Encoder (SAE)

The other kind of AE is zero-bias AE (Konda *et al.* 2014). It has been suggested a new activation function *shrinkage* that the AE can be trained without any additional regularization such as sparsity, denoising and contraction.

The Saturating AE (Goroshin and LeCun 2013) introduced another activation functions for hidden layer which contain one zero-gradient region (saturation) at least. This regulator encourages the activation function that acts in the saturation region explicitly. This method limits the reconstruction ability of inputs that are not near the data manifold.

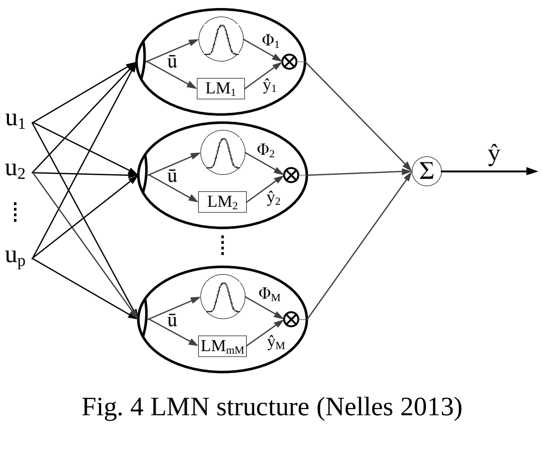


Fig. 4 LMN structure (Nelles 2013)

3. Incremental partitioning in local model networks

In recent years, interpolation of LMs for system identification has attracted a lot of attention. Fig. 4 show the LMN structure.

Adeniran (Adeniran and El Ferik 2017) has illustrated an extensive review of the LMN and its partitioning in modeling and identifying of nonlinear systems. Moreover, (Baghernezhad and Khorasani 2016) is demonstrated that LMN is very proper for FDI mobile robot. Also (Han *et al.* 2017) has compared artificial neural networks and SVM and random forest for fault diagnosis of rotating machinery. In (Aydin and Kisi 2015) an algorithm based on neuro-fuzzy hybrid system has presented to the detection of multiple damages for location and severity predictions of cracks in beam-like structure. Moreover, (Mohammadzadeh1a and Kim 2015) has introduced PANFIS which is integration of three different methodologies: Principal component analysis, Artificial Neural networks, and Fuzzy Inference Systems for modeling nonlinear behavior of civil structures.

In this way, incremental partitioning methods have absorbed greater regard. These methods are in contrast with experimental partitions that require prior knowledge to partitioning. In other words, incremental partitioning requires little or no prior knowledge (Adeniran and El Ferik 2017). So, in each step, two sub-models are added into the model by partitioning the input space. Axis-orthogonal and Axis-oblique methods are the commonly used methods in this area.

3.1 Axis-orthogonal and axis-oblique partitioning

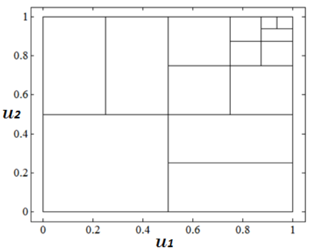
The axis-orthogonal strategy uses orthogonal axis parallel to split the axis of the input space (Fig. 5(a)). The LOLIMOT algorithm (Nelles *et al.* 1996, Nelles 2013) and the J&F algorithm (Johansen and Foss 1995) are pioneers of this type of partitioning. These algorithms have used the interpolation of the local sub-models that describe training data. Although these two algorithms are very similar, they have fundamental differences in the estimation of the local model parameters, and the location of the decomposition regimes. LOLIMOT uses heuristic methods to find structural parameters and works much faster. In the LOLIMOT algorithm, the decomposition of the regime is determined by local error, and the splitting direction is determined by the general error. The parameters of local linear models are easily and conveniently estimated by the Weighted Least Square (WLS) method. That is why LOLIMOT algorithm has gained more popularity.

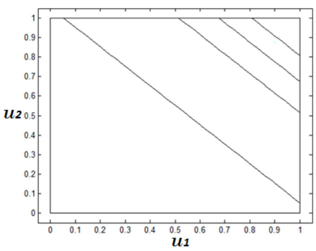
Since the parameters of LM are estimated from the data, hence the polynomial is the best option for LMs. The degree of these Polynomials can be 0 (as fuzzy Mamdani with center average defuzzification), 1 (linear) or higher. Bänfer and Nelles (2009) applied higher-degree polynomials in LMs and named, polynomial model tree (POLYMOT). Two choices exist in each iteration of this algorithm. Choice of either increase complexity of the worst LM or increase number of LMs with splitting of the worst LM. Although, the number of LM decreases for a certain accuracy by increasing polynomials degree; Polynomials of degree 1 (linear) are common and popular choice (Nelles 2006).

In continues, an algorithm similar to POLYMOT algorithm has proposed by Ahmadi and Karrari (2012). Moreover, (Mehran *et al.* 2006) uses PSO to find the best axis-orthogonal partition. Expectation-Maximization (EM) algorithm is executed for identification of LMs by Rezaie (Rezaie *et al.* 2007). To overcome computational effort with fewer LMs, Jakubek and Keuth modified clustering partition and local model statistic which approximates reliability of the obtained model in (Jakubek and Keuth 2006). Sarabi-Jamab and N.Araabi have provided merge and split strategy to decrease the number of LMs by using Piecewise Linear Network (Sarabi-Jamab and Araabi 2011).

Another strategy of splitting is Axis-oblique strategy (Fig. 5(b)), that the splitting is performed at an angle. Breiman (Breiman 1993) was first one who introduced this strategy with hinging hyperplanes. Further, Ernst proposed an algorithm in (Ernst 1998) which based on LOLIMOT construction. Then Nelles set a new stage on LOLIMOT and Ernst algorithm and called it HILOMOT (Nelles 2006, Hartmann *et al.* 2014). HILOMOT uses sigmoid function instead of hinging function as validity function. In continues, Fischer *et al.* (2012) eliminated numerical gradient calculation and applied quasi-Newton optimization to increase nonlinear optimization velocity. Moreover, Hartmann and Nelles (2009) added improvement and considered smoothness in validity functions. Beside that they combined POLYMOT and HILOMOT concept and proposed HILOMOT+ (Hartmann and Nelles 2012). It is true that HILOMOT is much flexible in comparison with LOLIMOT, but its convergence rate to the desired goal is lower than LOLIMOT (Hartmann and Nelles 2009). Hartmann and Nelles (2009) introduced a new strategy based on, Gustafson-Kessel fuzzy clustering to increasing flexibility and called supervised hierarchical clustering (SUHICLUST).

Generally, one of the disadvantages of partitioning algorithms in LMN is the high sensitivity to the size of the input space. In other words, the performance of the algorithm is reduced greatly as the input dimension increases (Nelles and Hartmann 2012).





(a) Axis-orthogonal (b) Axis-oblique

Fig. 5 Two partitioning strategy of input space for 2D input

3.2 LMN properties and LOLIMOT algorithm

The output of the LMN structure ($\hat{y}$), which is presented in Fig. 4, is obtained from the weighted interpolation of M LMs (Eq. (7)).

$$\hat{y}(u_i) = \sum_{j=1}^{M} \hat{y}_j\left(\underline{u_i}\right) \Phi_j(\underline{u_i}) = \sum_{j=1}^{M} (w_{j,0} + w_{j,1}u_{i,1} + w_{j,2}u_{i,2} + \cdots + w_{j,p}u_{i,p})\Phi_j(u_i) \quad (7)$$

where $\underline{u_i}$ is the i-th sample of the input vector and $\Phi_j(.)$ is the validity function of the j-th local model which plays the role of rules in the primary part of the fuzzy structure. It is created from the normalized membership function (Gaussian function) (Eq. (8)).

$$\Phi_j\left(u_i\right) = \frac{\mu_j\left(u_i\right)}{\sum_{j=1}^{M}\mu_j\left(u_i\right)} \quad (8)$$

Hear $\mu_j(u_i)$ is formed as

$$\mu_j(u_i) = -exp\left(-\frac{1}{2}\left(\left(\frac{u_{i,1}-C_{j,1}}{\sigma_{j,1}}\right)+\ldots+\left(\frac{u_{i,p}-C_{j,p}}{\sigma_{j,p}}\right)\right)^2\right) \quad (9)$$

where C and σ are the center coordinate and the individual standard deviation of Gaussian validity functions, respectively. In other words, $\Phi_j(.)$ determines the firing value of the j-th local model. Also, $\hat{y}_j(.)$ is the linear part of the j-th local model, like the consequent part of the j-th rule in the fuzzy structure. In this part, the first-order polynomial is used and $w = \left[w_{j,0}\ w_{j,1} \ldots w_{j,p}\right]^T$ are its coefficients.

Usually, mean Square Error (MSE) is considered as the general cost function (general error) for solving identification problems (Eq. (10)), while the local cost function (local error) is used to decompose models (Eq. (11)) (Nelles 2013).

$$J = \frac{1}{N}\sum_{i=1}^{N} e\left(u_i\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y\left(u_i\right) - \hat{y}\left(u_i\right)\right)^2 \quad (10)$$

$$J_j = \sum_{i=1}^{N} e\left(u_i\right)^2 \Phi_j\left(u_i\right) \quad , \quad j = 1,\ldots,M \quad (11)$$

where i is the number of the input and J_j is the error of j-th local model.

The LOLIMOT algorithm is as follows:

Step 1: Start with initial model

Step 2: Find the worst LM based on the max local cost function (Eq. (11))

Step 3: Break the worst LM

a. Splitting axis-orthogonal in each input dimension
b. Estimating the parameters of two new LLM by WLS
c. Calculating general cost function (Eq.10) for each splitting

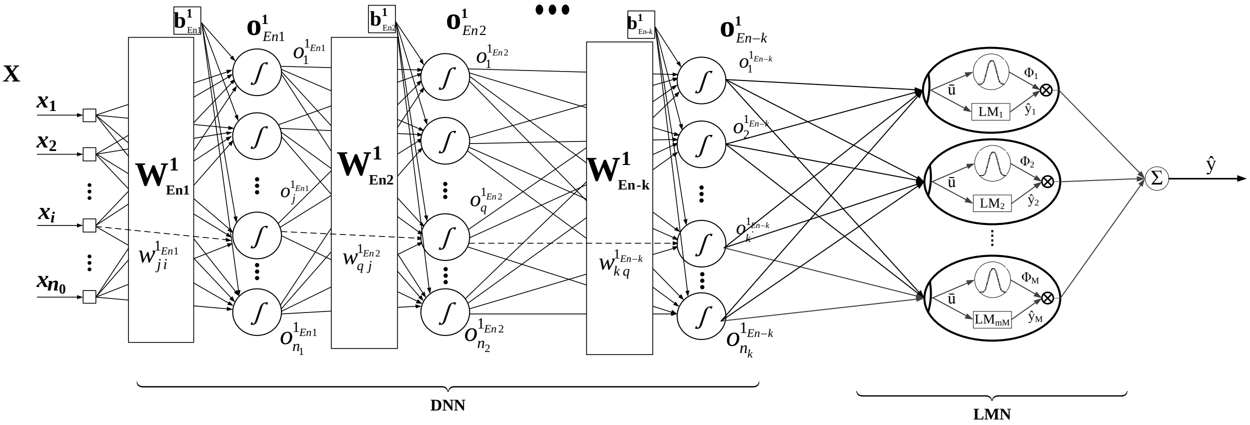


Fig. 6 Proposed combined structure of stacked Auto-Encoder (DNN) and LMN

Step 4: Select the best split based on the lowest general cost function

Step 5: Check the stop condition and go to step 2 if the condition is not met

4. The proposed combined structure of DNN and LMN

One of the advantages of LMN is dividing the system into smaller ones. By doing so, the complex system is divided into several smaller sub-systems with less complexity. But the performance of its algorithms such as LOLIMOT is very sensitive to the size of the input space as previously mentioned (Nelles and Hartmann 2012).

In this paper, Stacked Denoising Auto Encoder (SDAE) is used in the pre-structure of the LMN in order to reduce this sensitivity.

So, at first, the DNN is trained greedily to be encoded data into wealthy information with lower dimensions. Then, the local linear models are used to model outputs. Fig. 6 shows the proposed structure of DNN and LMN.

5. Simulation and results

The case study in this paper is a laboratory model of single-shaft industrial gas turbine Siemens V94.2, which is developed at the ALSTOM-ABB POWER center in the UK. This simulator generates the data set with a sampling rate of 0.08s which have been validated with real measurements in steady-state conditions (Simani and Fantuzzi 2006). The present simulator can apply four fault conditions. The faults are as:

1. Compressor contamination fault
2. Thermocouple sensor fault
3. High-pressure turbine seal damage
4. Fuel actuator friction wear.

These faults are incipient fault (drift) and also incept at the 15-th second.

In this paper, similar to the articles published in this field (Simani *et al*. 2003, Simani 2005, Simani and Fantuzzi 2006, Simani and Patton 2008), the features q_c ,t_3,p_3, p_7 are used as model outputs to FDI of V94.2 gas turbine. In this way, 4 LMNs are considered for modeling outputs, p_3 in normal conditions, q_c in fault1, T_3 in fault2, and P_7 in fault3. Furthermore, 16 sensor measurements with their three dynamics (48 features totally), are considered as inputs. The nomenclature of these feature are described in Table 1**.** The number of the input dynamics is based on trial and error.

In order to build proposed structure, SDAE is trained unsupervised greedily to encode input features by four layers with changing dimensions of 52, 25, 10 and 5 respectively. Fig. 7 shows MSE variations for both train and test sets with respect to epoch for each AEs. Then local linear models are used to model outputs. The proposed structure with four LMNs is presented in Fig. 8.

For a better comparison, the input features are applied to LMN without DNN directly, and once again are applied to the proposed structure. Meanwhile, two other datasets from different operation points are generated to evaluate the generalization of methods. The number of optimal neurons and the MSE of each LMNs are given in Table 2.

Since there are 145 parameters such as center and variance of the validity function and linear model coefficients in each LM with 48 inputs and also there are 16 parameters in each LM with 5 inputs, therefore the number

Table 1 Nomenclature of input features in simulator of gas turbine

a_v	valve angle
ff	fuel flow
q_c	compressor torque
t_i , i=2,3,6,7	ith section (module) temperature
p_i , i=2,3,7	ith section (module) pressure
m_i , i=1,5	ith section (module) mass rate
p_t	turbine power
p_a	ambient pressure
p_c	compressor power
w_t	turbine angular rate

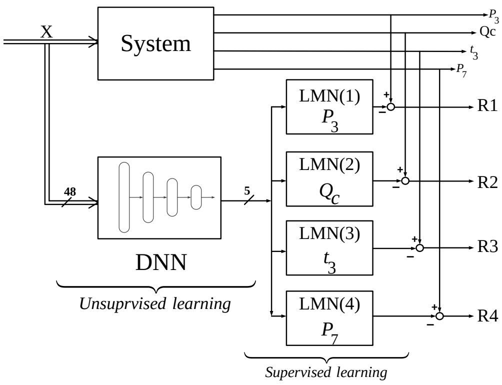


Fig. 8 The proposed combined structure of DNN and four LMNs

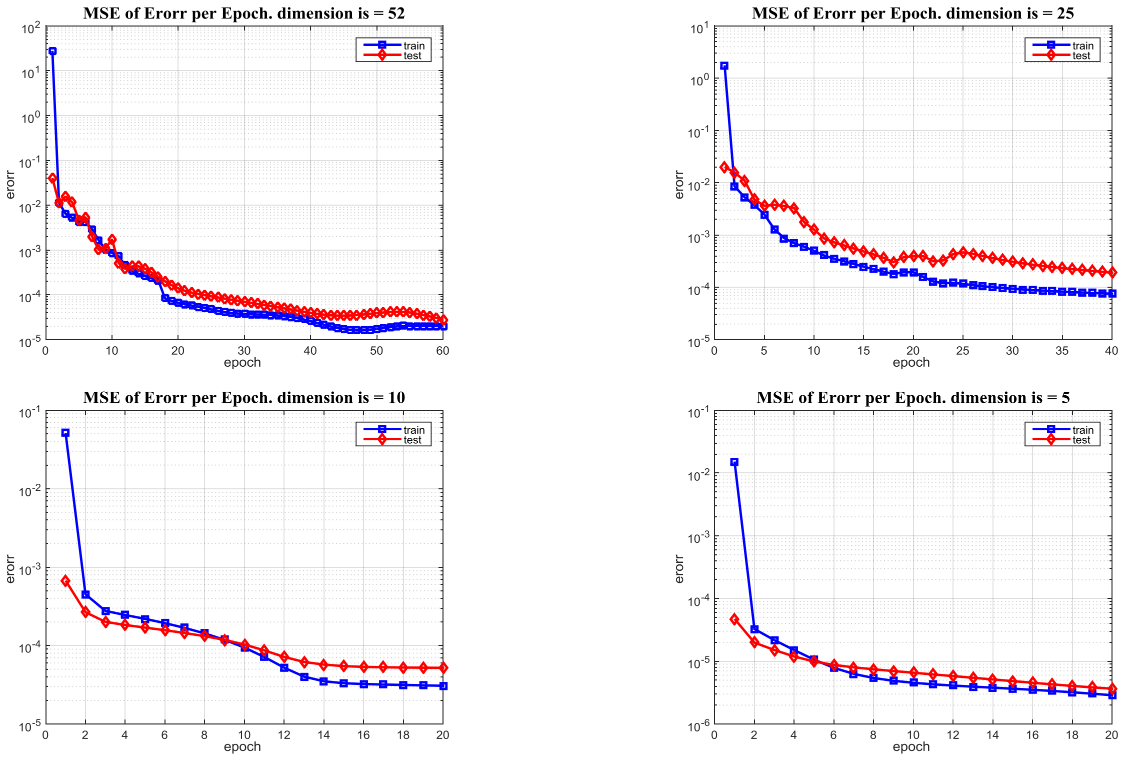


Fig. 7 Variation of the MSE for each Auto-encoders (changing input dimension to 52D,25D, 10D, 5D)

of parameters of each LM without the DNN is about nine times greater than the LM with the DNN approximately. On the other hand, by studying Table 2, it becomes clear that the number of LMs without DNN is more than another one. So, by utilizing the DNN, the required number of parameters is greatly reduced. Moreover, the MSE of the non-DNN model is much greater. This is the same curse of dimension that traps the fuzzy system.

5.1 Fault detection

In order to detect the faults, the residuals (R_1, R_2, R_3, R_4) are obtained from comparing the output of the gas turbine simulator with the output of four LMNs. The residuals with applying different faults are presented in Figs. 9-12. According to these figures, each fault causes different variations and signature in the residuals. This characteristic is utilized to isolate faults in the next sub-section.

The main objective of fault detection is being down the required time to detect faults. On the other hand, un-modelled disturbance and noise cause modeling uncertainty that it jars fault detection with increasing false alarms. Therefore, utilizing constant thresholding and adaptive thresholding are common ideas to reduce this malfunction. The thresholding method creates band around the residuals to make a decision whether a fault occurred or not as follow

Table 2 Learning MSE of LMN with or without DNN in back of the LMN

	Without DNN				With DNN			
Models	Number LM	MSE train	MSE valid1	MSE valid2	Number LM	MSE train	MSE valid1	MSE valid2
$LMN(1) - p_3$	15	3e-3	2e-3	5e-3	10	1.6e-5	7.7e-5	8.3e-5
$LMN(2) - Q_C$	12	2.8e-3	4e—3	3.4e-4	7	4.3e-5	1.7e-4	1.4e-4
$LMN(3) - t_3$	15	1.8e-3	2e-3	2.6e-3	10	1.8e-5	2e-4	2.9e-3
$LMN(4) - p_7$	15	4e-3	1.6e-2	1.4e-2	10	7.9e-5	4e-4	5.3e-4

$$\psi(t) = \begin{cases} 0 \text{ if} & \lambda_{Lower} \leq R(t) \leq \lambda_{Upper} \\ 1 \text{ if} & \lambda_{Lower} \end{cases} \tag{12}$$

where ψ is the fault signature. Since simple thresholding (constant λ) increases the false alarm rate (Eq. (14)), therefore the adaptive thresholding method is used to detect fault occurrences in this paper. Hence λ can be defined as follow

$$\lambda_{Upper/Lower} = m_R \pm \eta S_R \tag{13}$$

where m and s are mean and Standard deviation of residuals. Moreover the False alarm rate is the one of criteria which is used to evaluate the performance of fault detection. The lower this rate, the higher performance. The false alarm rate is given by

$$\text{False Alarm rate} = \frac{NF}{N} \tag{14}$$

where N is the total number of normal pattern data of a class and NF is the number of data samples of same class which detected as faulty patterns incorrectly. Thus, Table 3 shows the fault inception time and the fault detection time and the false alarm rate that were gained in appropriate amounts. The comparison of this table with other works is discussed in sub-Section 5-3.

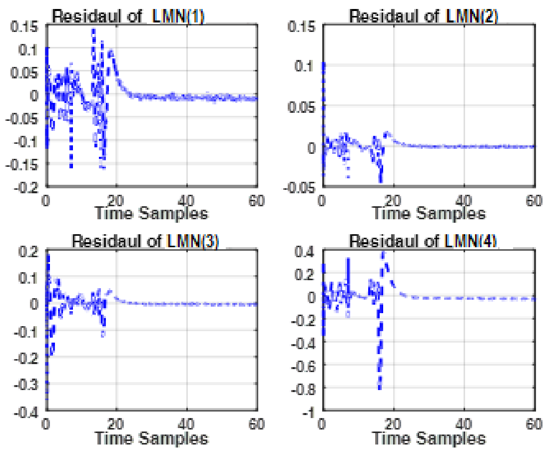


Fig. 9 L Residuals of all four LMNs by applying the fault1

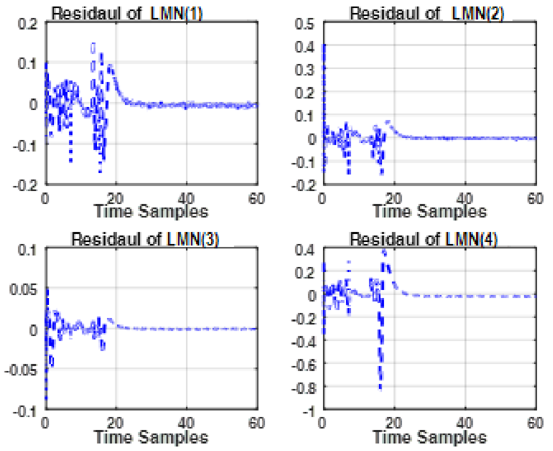


Fig. 10 Residuals of all four LMNs by applying the fault1

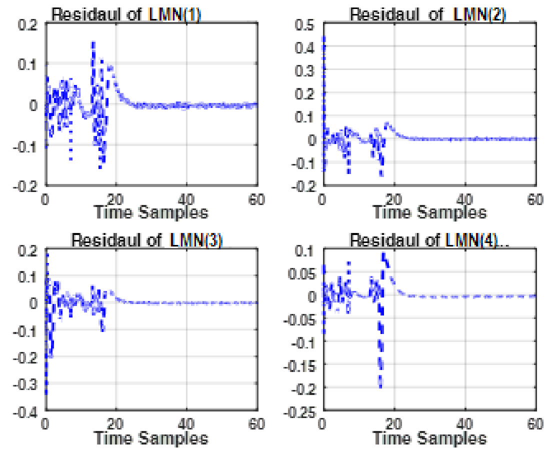


Fig. 11 Residuals of all four LMNs by applying the fault3

Table 3 Fault detection results based on the proposed structure

	Fault inception time(s)	Detection time(s)	False alarm rate(%)
$Fault1$	15	15.66	8.44
$Fault2$	15	15.66	10.45
$Fault3$	15	15.66	8.1
$Fault4$	15	15.66	6.4

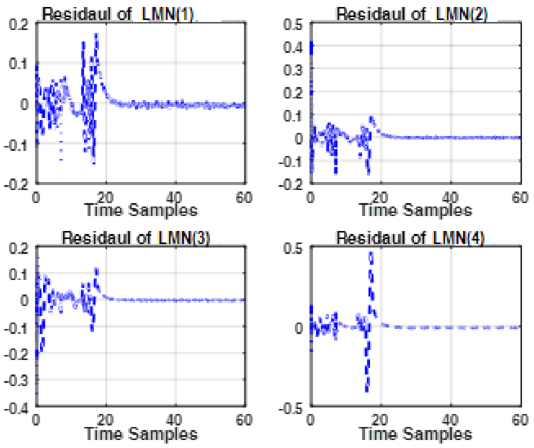


Fig. 12 Residuals of all four LMNs by applying the fault4

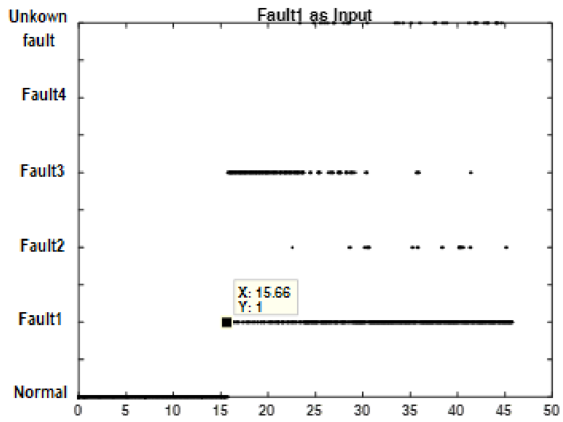


Fig. 13 The output of FDI system in Presence of Fault1

5.2 Fault isolation

Different variations and signature in the residuals in Figs. 9-12 are useful to identify and isolate faults. Table 4 shows these signatures. The signs ↑ and ↓ mean that the residual is non-zero and oriented towards the upper or lower threshold. In other words, the sign – indicates that the residual is close to zero. This table shows that any fault will make unique signature in the residuals. So FDI was done with taking these sign into consideration. Figs. 13-16 show the isolation of faults. In these figures, the value 0 indicate normal condition, and the values 1 up to 4 represent the fault 1 up to 4 respectively, and the value 5 represents a class of faults whose fault's type is not detected. Moreover, the fault isolation percentile is shown in a confusion matrix in Table 5. This table illustrates that almost all classes of conditions are separated with acceptable accuracy. Also, the percentage of non-class is located in the last column. Misdiagnosis or non-recognition of a class may be due to closeness or subscription of residual signs.

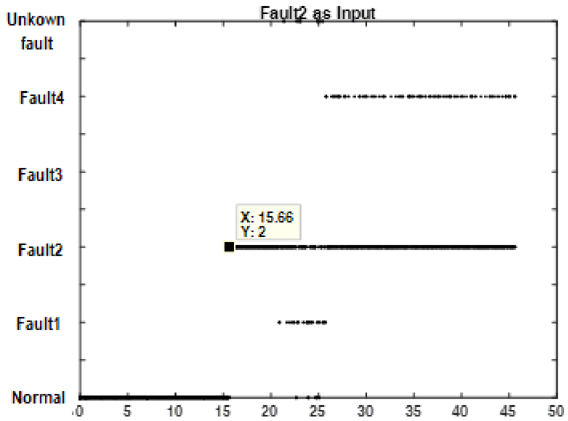


Fig. 14 The output of FDI system in Presence of Fault2

Table 4 Fault signature table

	$Fault1$	$Fault1$	$Fault1$	$Fault1$
$R1$	↓	–	↑	↓
$R2$	–	↑	↑	↑
$R3$	↓	–	↑	–
$R4$	↓	↓	–	↓

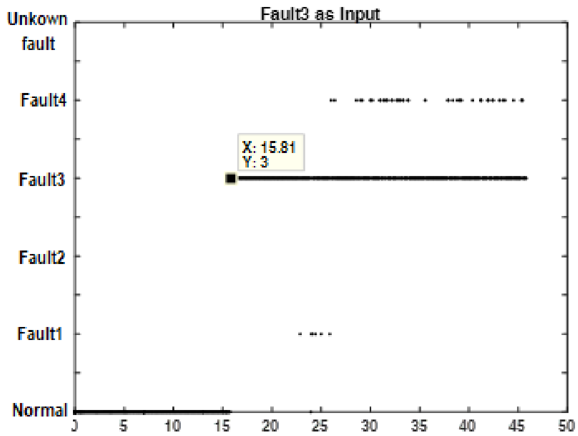


Fig. 15 The output of FDI system in Presence of Fault3

Table 5 Confusion matrix for fault isolation (%)

		Predicted				
		$Fault1$	$Fault1$	$Fault1$	$Fault1$	No class
Actual	$Fault1$	79/31	1/46	16/98	0	2/25
	$Fault2$	4/02	82/26	0	12/43	1/28
	$Fault3$	0/31	0	98/9	0/79	0
	$Fault4$	0	0/05	1/3	98/5	0/05

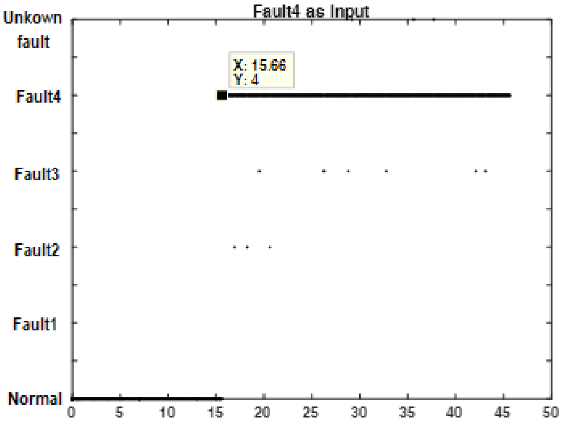


Fig. 16 The output of FDI system in Presence of Fault4

5.3 Compare with other work

The references (Simani *et al.* 1998, Palade *et al*. 2002, Simani 2005, Simani and Fantuzzi 2006, Nozari *et al.* 2012) are considered to compare with our proposed structure on this gas turbine benchmark.

In these references the number of input feature does not exceed 15 features. Furthermore, in most of these references, there has been no attempt to propose a nonlinear FDI method. While, non-linear identification is more appropriate due to the presence of noise and uncertainty. However (Simani and Fantuzzi 2006) was used the classic observer to detect faults in a gas turbine, but this approach is applicable for linear systems not as the same as gas turbine. (Simani *et al.* 1998) was considered the dynamic observer and neural networks to detect sensor fault which faults were modelled as step functions, while in our work the faults are incipient fault (drift) which are more difficult to diagnose. (Palade *et al.* 2002) was used a neuro-fuzzy method to detect and isolate only two faults (F1 and F4) but we are studied on four faults. In (Simani 2005), four faults (as drift fault) were considered, and the linear dynamical identification has been used for FDI based on the observer. By comparing results of this reference, its lower performance and higher detection time than our work are clear to see. In (Nozari *et al.* 2012), a combination of MLP and LMN structures were presented to robust FDI for the gas turbine in a steady state. It is also, every four faults were considered as the incipient fault. By comparing the fault isolation results, it is clearly seen that almost the same accuracy performance in the fault isolation percentile was achieved but our performance in detection time is better. One of the reasons for the improvement of our performance is the use of more information (more features) from the system in input.

6. Conclusions

In this paper, a new multiple model-based FDI method has been proposed for a laboratory single-shaft gas turbine. The gas turbine is a system that has high complexity and high dimensions. However other papers have utilized only a few of its features, thus that they did not have a dimensional problem. Unlike other articles, in this paper, all measurements of this gas turbine have been considered to achieve better performance. Since the input dimension is a serious constraint in modeling with fuzzy systems, therefore the two-part structure including DNN and LMN has been proffer. The combination of these artificial intelligent methods has led to obtaining a highly sensitive FDI method for the industrial gas turbine. In this paper, it has been shown that using DNN, not only reduces the number of parameter of each local model but also greatly improves the training accuracy. Finally, by comparing the simulation results with other works, it can be said that FDI with the proposed structure is more sensitive to the faults. Since this method is offered in the presence of noise, it can be used in industrial gas turbine software applications, especially when measurements are unreliable due to noise and uncertainties.

References

Adeniran, A.A. and El Ferik, S. (2017). “Modeling and identification of nonlinear systems: A review of the multimodel approach—Part 1”, *IEEE T. Syst, Man, Cy. : Syst*., **47**(7), 1149-1159.

Ahmadi, S. and Karrari, M. (2012), “An iterative approach to determine the complexity of local models for robust identification of nonlinear systems”, *Int. J. Control, Autom. Syst.*, **10**(1), 1-10.

Alizadeh, S., Kalhor, A., Jamalabadi, H., Araabi, B.N. and Ahmadabadi, M.N. (2016), “Online local input selection through evolving heterogeneous fuzzy inference system”, *IEEE T. Fuzzy Syst*., **24**(6), 1364-1377.

Aydin, K. and Kisi, O. (2015), “Damage detection in structural beam elements using hybrid neuro fuzzy systems”, *Smart Struct. Syst*., **16**(6), 1107-1132.

Baghernezhad, F. and Khorasani, K. (2016), “Computationally intelligent strategies for robust fault detection, isolation, and identification of mobile robots”, *Neurocomput*., **171**, 335-346.

Bänfer, O. and Nelles, O. (2009), “Polynomial model tree (POLYMOT)—A new training algorithm for local model networks with higher degree polynomials”, *Proceedings of the 2009 IEEE International Conference on Control and Automation*.

Bengio, Y., Courville, A. and Vincent, P. (2013), “Representation earning: A review and new perspectives”, *IEEE T. Pattern Anal. Machine Intel*., **35**(8), 1798-1828.

Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H. (2007), “Greedy layer-wise training of deep networks”, *Adv. Neural Inform. Process. Syst*., 153-160

Breiman, L. (1993), “Hinging hyperplanes for regression, classification, and function approximation”, *IEEE T. Inform. Theory* **39**(3), 999-1013.

Chang, C.M., Chou, J.Y., Tan, P. and Wang, L. (2017), “A sensor fault detection strategy for structural health monitoring systems”, *Smart Struct. Syts*., **20**(1), 43-52.

Erhan, D., Manzagol, P.A., Bengio, Y., Bengio, S. and Vincent, P. (2009), “The difficulty of training deep architectures and the effect of unsupervised pre-training”, *Artif. Intell. Stat.*, 153-160

Ernst, S. (1998), “Hinging hyperplane trees for approximation and identification”, *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, **2**, 1266-1271.

Fischer, T., Hartmann, B. and Nelles, O. (2012), “Increasing the Performance of a Training Algorithm for Local Model

Networks", *Proceedings of the World Congress of Engineering and Computer Science (WCECS)*, San Francisco, USA.

Goroshin, R. and LeCun, Y. (2013), "Saturating auto-encoders", *arXiv preprint arXiv*, 1301.3577.

Han, T., Jiang, D., Zhao, Q., Wang, L. and Yin, K. (2018), "Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery", *T. Inst. Measurement Control* **40**(8), 2681-2693.

Hartmann, B., Ebert, T., Fischer, T., Belz, J., Kampmann, G. and Nelles, O. (2014), "LMNTOOL–Toolbox zum automatischen Trainieren lokaler Modellnetze", *Proceedings of the 22. Workshop Computational Intelligence (Hoffmann, F.; Hüllermeier, E., Hg.)*,

Hartmann, B. and Nelles, O. (2009), "Advantages of hierarchical versus flat model structures for high–dimensional mappings", *Workshop Computational Intelligence. Bommerholz.*

Hartmann, B. and Nelles, O. (2009), "On the smoothness in local model networks", *Proceedings of the American Control Conference (ACC), St. Louis, USA (June 2009)*.

Hartmann, B. and Nelles, O. (2012), "Structure trade-off strategy for local model networks", *Proceedings of the Control Applications (CCA), 2012 IEEE International Conference on*.

Hartmann, B., Nelles, O., Skrjanc, I. and Sodja, A. (2009), "Supervised hierarchical clustering (SUHICLUST) for nonlinear system identification", *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Control and Automation.*

Hinton, G.E. and Salakhutdinov, R.R. (2006), "Reducing the dimensionality of data with neural networks", *Science*, **313**(5786), 504-507.

Huang, H.B., Yi, T.H. and Li, H.N. (2015), "Sensor fault diagnosis for structural health monitoring based on statistical hypothesis test and missing variable approach", *J. Aerosp. Eng.*, **30**(2), B4015003.

Huang, H.B., Yi, T.H. and Li, H.N. (2016), "Canonical correlation analysis based fault diagnosis method for structural monitoring sensor networks", *Smart Struct. Syst***.**, **7**(6), 1031-1053.

Jakubek, S. and Keuth, N. (2006), "A local neuro-fuzzy network for high-dimensional models and optimization", *Eng. Appl. Artif. Intel.*, **19**(6), 705-717.

Johansen, T.A. and Foss, B.A. (1995), "Identification of non-linear system structure and parameters using regime decomposition", *Automatica*, **31**(2), 321-326.

Jung, U. and Koh, B.H. (2014), "Bearing fault detection through multiscale wavelet scalogram-based SPC", *Smart Struct. Syst.*, **14**(3), 377-395.

Kavukcuoglu, K., Fergus, R. and LeCun, Y. (2009), "Learning invariant features through topographic filter maps", *Proceedings of the Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 1605-1612.

Kavukcuoglu, K., Ranzato, M.A. and LeCun, Y. (2010), "Fast inference in sparse coding algorithms with applications to object recognition", *arXiv preprint arXiv*,1010.3467.

Konda, K., Memisevic, R. and Krueger, D. (2014). "Zero-bias autoencoders and the benefits of co-adapting features." *arXiv preprint arXiv:*1402.3337.

Mehran, R., Fatehi, A., Lucas, C. and Araabi, B.N. (2006), "Particle swarm extension to LOLIMOT", *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications*.

Mohammadzadeh1a, S. and Kim, Y. (2015), "PCA-based neuro-fuzzy model for system identification of smart structures", *Smart Struct. Syst.*, **15**(4), 1139-1158.

Mutlu, B., Sezer, E.A. and Nefeslioglu, H.A. (2016), "A defuzzification-free hierarchical fuzzy system (DF-HFS): Rock mass rating prediction", *Fuzzy Set. Syst.*, **307**, 50-66.

Nelles, O. (2006), "Axes-oblique partitioning strategies for local model networks", *Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control.*

Nelles, O. (2013), "Nonlinear system identification: from classical approaches to neural networks and fuzzy models", *Springer Science & Business Media*.

Nelles, O. and Hartmann, B. (2012), "Structure trade-off strategy for local model networks", *Proceedings of the IEEE International Conference on Control Applications (CCA), Dubrovnik, Croatia, Part of 2012 IEEE Multi-Conference on Systems and Control*.

Nelles, O., Sinsel, S. and Isermann, R. (1996), "Local basis function networks for identification of a turbocharger", *Proceedings of the Control'96, UKACC International Conference on (Conf. Publ. No.* **427**), *IET* 7-12.

Ng, A. (2011), "Sparse Autoencoder", vol. **72** of. *CS294A Lecture Notes.*

Nozari, H.A., Shoorehdeli, M.A., Simani, S. and Banadaki, H.D. (2012), "Model-based robust fault detection and isolation of an industrial gas turbine prototype using soft computing techniques", *Neurocomput*., **91**, 29-47.

Palade, V., Patton, R.J., Uppal, F.J., Quevedo, J. and Daley, S. (2002), "Fault diagnosis of an industrial gas turbine using neuro-fuzzy methods", *IFAC Proceedings Volumes*, **35**(1), 471-476.

Poultney, C., Chopra, S. and Cun, Y. L. (2007), "Efficient learning of sparse representations with an energy-based model", *Adv. neural information processing syst.*, 1137-1144.

Pourbabaee, B., Meskin, N. and Khorasani, K. (2013), "Multiple-model based sensor fault diagnosis using hybrid kalman filter approach for nonlinear gas turbine engines", *Proceedings of the 2013 American Control Conference, IEEE* 4717-4723.

Rezaie, J., Moshiri, B., Rafati, A. and Araabi, B.N. (2007), "Modified LOLIMOT algorithm for nonlinear centralized Kalman filtering fusion", *Proceedings of the Information Fusion, 2007 10th International Conference on, IEEE* 1-8.

Rifai, S., Vincent, P., Muller, X., Glorot, X. and Bengio, Y. (2011), "Contractive auto-encoders: Explicit invariance during feature extraction", *Proceedings of the 28th International Conference on International Conference on Machine Learning, Omnipress* 833-840.

Saleh, A.E., Moustafa, M.S., Abo-Al-Ez, K.M. and Abdullah, A. A. (2016), "A hybrid neuro-fuzzy power prediction system for wind energy generation", *Int. J. Elec. Power Energy Syst*., **74**, 384-395.

Sarabi-Jamab, A. and Araabi, B.N. (2011), "PiLiMoT: A modified combination of LoLiMoT and PLN learning algorithms for local linear neurofuzzy modeling", *J, Control Sci. Eng*., 2011.

Shen, C., Wang, D., Liu, Y., Kong, F. and Tse, P.W. (2014), "Recognition of rolling bearing fault patterns and sizes based on two-layer support vector regression machines", *Smart Struct. Syst*., **13**(3), 453-471.

Simani, S. (2005), "Identification and fault diagnosis of a simulated model of an industrial gas turbine", *IEEE T. Ind. Inform.*, **1**(3), 202-216.

Simani, S. and Fantuzzi, C. (2006), "Dynamic system identification and model-based fault diagnosis of an industrial gas turbine prototype", *Mechatronics*, **16**(6), 341-363.

Simani, S., Fantuzzi, C. and Patton, R.J. (2003), "Model-based fault diagnosis in dynamic systems using identification techniques", *Springer***,** 19-60.

Simani, S., Fantuzzi, C. and Spina, R. (1998), "Application of a neural network in gas turbine control sensor fault detection", *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on, IEEE*, **1**, 182-186.

Simani, S. and Patton, R.J. (2008), "Fault diagnosis of an industrial gas turbine prototype using a system identification approach", *Control Eng. Pract*., **16**(7), 769-786.

Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.A. (2008), "Extracting and composing robust features with denoising autoencoders", *Proceedings of the 25th international conference on Machine learning*, ACM 1096-1103.

Yi, T.H., Huang, H.B. and Li, H.N. (2017), "Development of sensor validation methodologies for structural health monitoring: A comprehensive review", *Measurement*, **109**, 200-214.

Yi, T.H., Ye, X., Li, H.N. and Guo, Q. (2017), "Outlier detection of GPS monitoring data using relational analysis and negative selection algorithm", *Smart Struct. Syst*., **20**(2), 219-229.

Zhang, Q., Yang, L.T., Chen, Z. and Li, P. (2018), "A survey on deep learning for big data", *Inform. Fusion*, **42**, 146-157.

HJ