Neural Networks 76 (2016) 106–121

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines

M. Amozegar, K. Khorasani *

Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada

ARTICLE INFO

Article history:
Received 16 October 2015
Accepted 13 January 2016
Available online 29 January 2016

Keywords:
Ensemble learning
Fault detection and isolation
System identification
Dynamic neural networks
Gas turbine engines

ABSTRACT

In this paper, a new approach for Fault Detection and Isolation (FDI) of gas turbine engines is proposed by developing an ensemble of dynamic neural network identifiers. For health monitoring of the gas turbine engine, its dynamics is first identified by constructing three separate or individual dynamic neural network architectures. Specifically, a dynamic multi-layer perceptron (MLP), a dynamic radial-basis function (RBF) neural network, and a dynamic support vector machine (SVM) are trained to individually identify and represent the gas turbine engine dynamics. Next, three ensemble-based techniques are developed to represent the gas turbine engine dynamics, namely, *two heterogeneous ensemble* models and *one homogeneous ensemble* model. It is first shown that *all* ensemble approaches do significantly improve the overall performance and accuracy of the developed system identification scheme when compared to each of the stand-alone solutions. The best selected stand-alone model (i.e., the dynamic RBF network) and the best selected ensemble architecture (i.e., the heterogeneous ensemble) in terms of their performances in achieving an accurate system identification are then selected for solving the FDI task. The required residual signals are generated by using both a single model-based solution and an ensemble-based solution under various gas turbine engine health conditions. Our extensive simulation studies demonstrate that the fault detection and isolation task achieved by using the residuals that are obtained from the dynamic ensemble scheme results in a significantly more accurate and reliable performance as illustrated through detailed quantitative confusion matrix analysis and comparative studies.

1. Introduction

Fault Detection and Isolation (FDI) of complex systems has captured a wide range of attention in various industries including the aerospace, among others. FDI plays an important role in increasing safety and reducing operational costs of an aircraft. This is applicable to different subsystems of an aircraft, which also includes its engine. Early diagnosis of the gas turbine engine faults reduces both the operational and maintenance costs of an aircraft.

At the highest level, the FDI schemes can be categorized into two main classes, namely: model-based and computational intelligent-based methods. The complications associated with the availability of a highly accurate and reliable mathematical representation of a system is the most significant obstacle in using model-based approaches. Kalman filter is a well-established model-based approach that has been extensively applied to gas turbine engine fault diagnosis (Kobayashi & Simon, 2003; Naderi, Meskin, & Khorasani, 2012).

Computational intelligent-based methods, on the other hand, do not require availability of a mathematical model and they can be trained by using only the available input–output data from the system (Tayarani-Bathaie, Vanini, & Khorasani, 2014; Venkatasubramanian, Rengaswamy, Yin, & Kavuri, 2003). Various computational intelligent-based approaches have been developed for FDI in the literature, among which the neural networks are the most well-known. Neural networks have been widely used in gas turbine engine fault diagnosis. The use of dynamic neural networks for fault diagnosis of gas turbine engines is reported in Tayarani-Bathaie et al. (2014). Feed-forward neural networks are another widely used neural network for the gas turbine engine fault diagnosis (Loboda, Feldshteyn, & Ponomaryov, 2011; Xiao, Eklund, Goebel, & Cheetham, 2007; Yan & Xue, 2008; Zhang, 2005). The use of radial basis function (RBF) neural networks for fault diagnosis of gas turbine engines is also reported in Loboda et al. (2011), Lu, Zhu, and Lv (2012) and Oza, Tumer, Tumer, and Huff (2003).

The main drawback of standard neural network methods is their lack of quantitative *a priori* confidence on their generalization performance, given that their knowledge is distributed over a set of neurons (unlike the model-based approaches where the knowledge is centralized in the mathematical model). To respond

* Corresponding author. Tel.: +1 514 848 2424x3086; fax: +1 514 848 2424.
E-mail address: kash@ece.concordia.ca (K. Khorasani).

http://dx.doi.org/10.1016/j.neunet.2016.01.003
0893-6080/

to this challenge, one can propose a fault detection and isolation scheme that is based on ensemble of neural network architectures. The agreement among the ensemble members reduces the chance of error while increases the overall decision making reliability and confidence (Zhang & Ma, 2012). Ensemble learning has proven to improve the individual learner's generalization capability and performance (Hansen & Salamon, 1990; Sharkey & Sharkey, 1997), and to reduce the chance of selecting a learner with weak performance capability. Ensemble learning has captured a lot of attention in computer science and engineering communities under various names (Polikar, 2012) including: bagging (Breiman, 1996), boosting (Avnimelech & Intrator, 1999; Drucker, Cortes, Jackel, LeCun, & Vapnik, 1994), mixture of experts (Jacobs, Jordan, Nowlan, & Hinton, 1991), and neural networks ensemble (Hansen & Salamon, 1990).

The use of ensemble methods for tackling the FDI problem has been reported in several publications. In Xiao, Eklund, Goebel, and Cheetham (2007), Xiao et al. developed an ensemble classifier for fault diagnosis of an aircraft engine using linear regression (LR), multilayer perceptron (MLP), likelihood ratio test (LRT) and robust ratio thresholding (RRT). Yan and Xue (2008) introduced an ensemble classifier for gas turbine engine fault diagnosis by using support vector machine (SVM), MLP and decision tree (DT). Xiao, Eklund, and Goebel (2007) designed an ensemble classifier for fault diagnosis of gas turbine with generalized regression neural network (GRNN), logistic regression (LoR) and random forest (RF). Varma et al. (2007) used rough sets (RS) and self-organizing maps (SOM) for anomaly detection problem of gas turbines. Donat, Choi, An, Singh, and Pattipati (2007) presented five fault classifiers based on the nearest neighbor classifier (K-NN), SVM, Gaussian mixture models (GMM), probabilistic neural network (PNN) and principle component analysis (PCA). Lu et al. (2012) presented an ensemble system based on the MLP and radial basis function (RBF) neural networks to improve the diagnostic accuracy and reduce the rate of misdiagnosis of an aircraft engine faults. Huang and Wang (2010) proposed a multiple classifier fusion using within-class decision support for fault diagnosis where the base classifiers selected are K-NN and orthogonal quadratic discriminant function (OQDF). Amanda and Sharkey (2002) as well as Sharkey, Chandroth, and Sharkey (2000) used ensemble of MLP networks for fault diagnosis of diesel engines. Lei, Zuo, He, and Zi (2010) presented a multiple classifier system (MCS) for fault detection of a gearbox by combining MLP, RBF and K-NN. Oza et al. (2003) presented an ensemble of MLP and RBF for the aircraft health monitoring problem. Ren, Yan, and Li (2009) combined three classifiers MLP, fuzzy logic (FL) and human–machine interaction (HI) to solve the fault diagnosis problem of an aero-engine.

Based on all the above literature review, *the use of ensemble learning for gas turbine engine fault diagnosis through* system identification has not yet been reported. The work in this paper presents an ensemble of dynamic neural networks for fault detection and isolation of a gas turbine engine through system identification. Several prior research highlighted above have employed ensemble learning ***to evaluate*** the residual signals to detect or isolate a fault but no research has addressed the possible use of ***ensemble learning identifiers for generating*** the residuals. In other words, all previous work has developed *ensemble of classifiers* that receives and operates based on the residual signals; however, no research has addressed the use of *ensemble of identifiers or regressors* to identify the gas turbine engine dynamics and to generate the ***ensemble of dynamically generated residuals*** for accomplishing the FDI problem.

The objectives of this work are therefore to develop an ensemble-based methodology for accomplishing the fault-detection and isolation of gas turbine engines and also to *compare* the results with conventional single neural network-based FDI solutions. It will be shown that by integration of the stand-alone neural networks more accurate ensemble models can be constructed and designed to identify and represent the gas turbine engine dynamics without the need of requiring ad-hoc fine tuning procedures that are necessary in single neural network-based solutions. It should be noted that the main motivation, justification, and argument for developing a more accurate ensemble is to have a large number of ensemble members. In theory, the accuracy of an ensemble model can be improved *arbitrarily* by increasing the number of ensemble members albeit *without requiring to have very accurate individual ensemble members*.

For the purpose of performing gas turbine engine health monitoring, first its dynamics is identified by using three different stand-alone learning algorithms. Specifically, the MLP-NARX, the RBF-NARX, and the SVM-NARX models (NARX denotes Nonlinear AutoRegressive eXogeneous) are trained to individually model the gas turbine engine output measurements. A separate model was trained for each engine output by using individual learning algorithms. The parameters of the individual learning algorithms (e.g., the number of the neural network neurons) are optimized by performing several experimentations. It will be shown that the RBF-NARX model shows a better modeling and representation performance (in terms of the modeling identification accuracy) as compared to the other stand-alone models.

The first ensemble model that attempts to model the gas turbine engine dynamics is a homogeneous ensemble with bagging where several RBF-NARX models are trained by using various subsets of the training data that are generated by the bootstrap sampling to model the gas turbine engine dynamics. The effects of the number of models in an ensemble on its accuracy will also be studied. It is observed that by increasing the number of models in an ensemble the prediction error in general decreases. It is also observed that homogeneous ensemble models based on the MLP, RBF, and SVM structures outperform the stand-alone models in term of the modeling and identification accuracy.

Next, the system identification is accomplished by developing a heterogeneous ensemble where the members are combined by using the weighted averaging where the weights are optimized by using a gradient descent method. For the second heterogeneous ensemble approach the Forward Sequential Selection (FSS) pruning is utilized. We first train a pool of stand-alone models to identify the gas turbine engine output measurements. The ensemble initially utilizes the model yielding the best performance (in terms of modeling and identification accuracy), and then the other models are added to the ensemble based on their contribution in improving the overall ensemble performance.

The selected heterogeneous ensembles with the FSS pruning (given that it will be shown to perform the best among the other ensembles), and the RBF-NARX as a stand-alone neural network model is shown to yield the best performance for solving the fault detection problem. Different fault scenarios are considered based on the fault type, the fault severity, and the gas turbine engine's input profile (the fuel flow rate). Our experiments will demonstrate that the fault detection achieved by using the residuals that are obtained from the ensemble models result in a more accurate fault detection goal and objective.

Finally, fault isolation is solved by evaluating the variations in the ensemble residual signals before and after a fault detection flag is issued by the detection filters. Eight classes corresponding to single faults and ten classes corresponding to multiple concurrent faults are defined (according to the fault type and severity). Our goal is to demonstrate and show that the ensemble-based fault isolation approach will result in a more promising performance as compared to the individual neural network-based classifiers.

It should be pointed out that the improved performance and accuracy that are achieved by using the ensemble methods will be

slightly more computationally costly (in terms of training multiple system identification models as opposed to only one model) as compared to stand-alone neural network methods; however, this additional effort will be compensated by and justified as the ensemble methods do not require time-consuming and labor intensive ad-hoc fine tunings that are necessary for single neural network-based solutions.

Based on the above discussion, the main contributions of this work can therefore be summarized as follows:

(1) A novel approach is proposed for identification of a gas turbine engine dynamics based on an ensemble of dynamic identifiers and learners. This paper does report the first use of dynamic ensemble learning methodology for identifying and representing nonlinear systems.
(2) An extensive comparative study is conducted to verify and validate that the proposed ensemble-based dynamic system identification can reduce the gas turbine engine modeling error for up to 67% as compared to single neural network-based solutions. Consequently, one can expect to guarantee more accurate residual signal generation process that can be utilized for accomplishing the fault detection and isolation (FDI) objectives.
(3) Novel FDI methodologies are proposed by utilizing the ensemble dynamic identifiers of the gas turbine engine by using both *homogeneous* and *heterogeneous* ensemble architectures. Various ensemble schemes are studied to determine the strategy that yields the maximal improvement as compared to single dynamic neural network-based solutions. The constructed residuals are shown to yield ensemble-based gas turbine engine fault detection results that are up to 5% more accurate as compared to the single neural network-based solutions. Moreover, it is shown that one can achieve up to 12% improvement in the fault isolation correct classification rates.

The remainder of this paper is organized as follows. Section 2 presents the required background on the ensemble learning and the gas turbine engine model as well as the nature of the considered faults. Section 3 presents our proposed ensemble-based dynamic system identification methodology. Section 4 describes our developed and proposed ensemble-based methodology for the fault detection problem. Section 5 presents our proposed single fault isolation ensemble methodology, and Section 6 presents our proposed multiple fault isolation ensemble approach. The proposed homogeneous and heterogeneous dynamic neural network ensemble identifiers and simulation results corresponding to an engine are presented in Section 7. The FDI simulation results are presented in Section 8. Section 9 concludes the paper.

2. Background information

This section contains three parts. The first part provides a brief overview of the ensemble learning strategy. The second part presents preliminaries on the dynamic neural network learning scheme. Finally, the third part is an overview on single-spool gas turbine engine model and the component faults that are considered and investigated in this paper for the FDI problem.

2.1. Ensemble learning

The error in any learning problem is composed of two components, namely *bias* and *variance* (Bishop, 2006; Hansen & Salamon, 1990; Polikar, 2012), where there is a trade-off relationship. Generally, *bias* would be large if a learning method produces models that are consistently wrong. Assume that a learning problem is solved several times using the same learning algorithm. Then the bias is indicated by the difference between the prediction and the expected values taken over different trained models (Hastie, Tibshirani, & Friedman, 2008). *Variance* would be large if choosing different training sets results in various predictions (assuming that a learning problem is solved several times using the same learning algorithm but different training data) (Hastie et al., 2008). When one compares different learners, in most cases comparisons show that one method has a higher bias and lower variance and another method has a lower bias and higher variance (Bishop, 2006). The decision for selecting one method over another is not simply a matter of selecting the one that has a *small variance* or the one that has a *small bias*. Rather the goal should be to weigh the respective merits of the bias and variance and then properly choose accordingly (Anders & Vedelsby, 1995). Ensemble learning methods have been developed to improve a methodology's accuracy by reducing its variance, while maintaining the bias of the learner low (Polikar, 2012).

Diversity is key to designing ensemble learning systems. Clearly, there is nothing to be gained from combining several identical models. It should be noted that the association and *link* between diversity and *bias–variance trade-off* is that diversity of models boils down to their variances. Therefore, the ensemble members should be different from each other while each must maintain an acceptable level of accuracy.

The specific method for creating diversity plays an important role in training of the ensemble model. At the high level, two different methodologies can be considered for creating diversity among the ensemble members. The first method uses learners having different architectures (e.g. by using different types of neural networks), and the second methodology trains different learners on different sets of training data (Brown, Wyatt, Harris, & Yao, 2005). These two methodologies are referred to as heterogeneous and homogeneous ensembles, respectively.

The source of diversity in heterogeneous ensembles is due to the inherent properties of different learning schemes. On the other hand, the source of diversity in homogeneous ensembles is due to the use of different subsets of the available training data for training the individual models. Homogeneous ensembles have been widely studied in the literature (Brown et al., 2005), whereas the number of work on heterogeneous ensembles is relatively fewer (Mendes-Moreira, Soares, Jorge, & Sousa, 2012), and thus it deserves more attention.

2.2. Neural networks for dynamical system identification

Basic neural network architectures are capable of learning static nonlinear maps between inputs and outputs of a system. In static systems the output at any instant n, that is $y(n)$, depends only on the input $u(n)$ at the same instant through a potentially nonlinear map that is given by $y(n) = f\big(u(n)\big)$. Therefore, static neural networks can be used for modeling such systems. However, in dynamical systems, the output at the present time depends not only on the input at the present time, but also on a certain number of past instances of inputs and outputs. Such systems should be represented by dynamical neural network structures for mapping representation and system identification.

The main characteristic of a dynamic neural network is that it is embedded with memory, which makes it a suitable framework for modeling highly complex nonlinear systems, such as the gas turbine engines. We will show that dynamic neural networks is a promising tool for generating residuals, and hence for fault diagnosis. Dynamic neural networks have recently been employed in achieving FDI of nonlinear systems. In Al-Zyoud and Khorasani (2005), a dynamic neural network was used to detect actuator faults in the attitude control subsystem of a satellite. Valdes, Khorasani, and Ma (2009) used a dynamic neural network for fault detection and isolation of thrusters in satellites. The authors

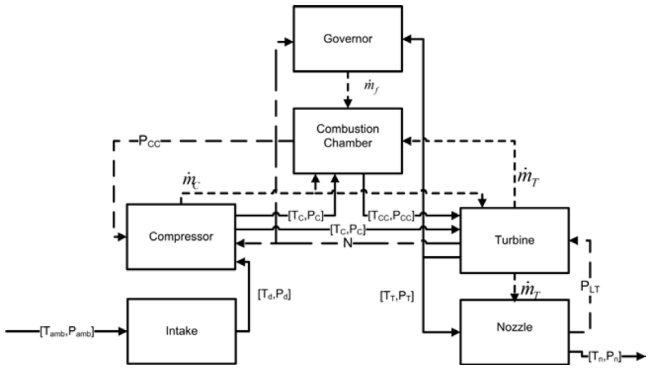

Fig. 1. Gas turbine engine modules and the information flow chart (Naderi et al., 2012).

in Mohammadi, Naderi, Khorasani, and Hashtrudi-Zad (2010), Tayarani-Bathaie et al. (2014), Vanini, Khorasani, and Meskin (2014) have applied dynamic neural networks that was developed in Yazdizadeh and Khorasani (2002) for fault detection of aircraft gas turbine engines.

2.3. Gas turbine engine model

A MATLAB Simulink model of a gas turbine engine is developed in Naderi et al. (2012) based on the available literature (Camporeale, Fortunato, & Mastrovito, 2006; Panov, 2009) and is used in this paper to generate the required data. Alternatively, data from a real engine can be used. However, it should be noted that using a simulation model is of special interest as it easily allows one to generate faulty data, given that for practical reasons it may not always be possible to actually inject faults in a real operational gas turbine engine. The following set of nonlinear equations describes a single-spool gas turbine engine dynamics (Naderi et al., 2012):

$$\dot{T}_{CC} = \frac{1}{c_v m_{CC}}\left[(c_P T_C \dot{m}_C + \eta_{CC} H_u \dot{m}_f - c_P T_{CC} \dot{m}_T) - c_v T_{CC}(\dot{m}_C + \dot{m}_f - \dot{m}_T)\right]$$

$$\dot{N} = \frac{\eta_{mech} \dot{m}_T c_P (T_{CC} - T_T) - \dot{m}_C c_P (T_C - T_d)}{JN\left(\frac{\pi^2}{30}\right)}$$

$$\dot{P}_T = \frac{RT_{M_i}}{V_{M_i}}\left(\dot{m}_T + \frac{\beta}{1+\beta}\dot{m}_C - \dot{m}_n\right)$$

$$\dot{P}_{CC} = \frac{P_{CC}}{T_{CC}}\dot{T}_{CC} + \frac{\gamma R T_{CC}}{V_{CC}}(\dot{m}_C + \dot{m}_f - \dot{m}_T)$$

where T_{CC} denotes the combustion chamber temperature, N denotes the rotational speed, P_T denotes the turbine pressure, P_{CC} denotes the combustion chamber pressure, m_{CC} denotes the mass flow in the combustion chamber, c_v denotes the specific heat at constant volume, c_p denotes the specific heat at constant pressure, J denotes the rotor moment of inertia, R denotes the gas constant, V_{CC} denotes the combustion chamber volume, γ denotes the heat capacity ratio, T_C denotes the compressor temperature, $\dot{m}_C$ denotes the compressor mass flow rate, $\dot{m}_T$ denotes the turbine mass flow rate, η_{CC} denotes the combustion chamber efficiency, H_u denotes the fuel specific heat, $\dot{m}_f$ denotes the fuel mass flow rate, η_{mech} denotes the mechanical efficiency, T_d denotes the diffuser temperature, $\dot{m}_n$ denotes the nozzle mass flow rate, β denotes the bypass ratio, T_{M_i} denotes the mixer temperature, and V_{M_i} denotes the volume mixer.

Fig. 1 depicts the main engine components and their interdependencies. The state variables in a single-spool gas turbine engine are selected as $x = [T_{CC}, N, P_T, P_{CC}]^T$. The output measurements in a single-spool gas turbine engine are selected as $z = [P_C, T_C, N, P_T, T_T]$, where P_C denotes the compressor pressure and T_T denotes the turbine temperature. The control input of the gas turbine engine is the power level angle (PLA) that is adjusted by the pilot and is related to the fuel mass flow rate ($\dot{m}_f$) through a variable gain.

Faults in a gas turbine engine are categorized into the following commonly occurring types, namely component faults, actuator faults, and sensor faults. This paper considers only component faults that are listed in Table 1. The component faults are modeled as decrease in the engine health parameters that are the efficiency and the mass flow rates of the compressor and the turbine sections.

3. Ensemble-based dynamic system identification

3.1. The gas turbine engine model identification: single neural network-based approach

System identification plays an important role in fault detection schemes. One always requires to have a reference healthy model to generate expected system outputs that correspond to the healthy process. The residual signals are then constructed by comparing the output of the actual system that could be subjected to faults with that of the identified healthy reference model output. If the residual is within a certain *a priori* selected bounds, the system under study is said to be healthy, however if the residual signal exceeds these bands then a fault has occurred in the system and the fault flag is declared. In this work, various neural network architectures are trained to identify the dynamics of a healthy gas turbine engine. These models are subsequently integrated together to construct an ensemble model representation of the healthy engine. This section describes the identification of the engine dynamics by using individual neural network learning models, whereas Section 3.2 provides the identification of the engine dynamics by using an ensemble of the learning models.

The gas turbine engine dynamics is identified by utilizing the Nonlinear AutoRegressive eXogenous (NARX) architecture which is commonly used in the system identification domain (Billings, 2013). The NARX approach relates the current output of the to be identified system to its previous inputs and outputs. A general nonlinear system can be represented by an input–output map according to the following nonlinear autoregressive moving average (NARMA) model:

$$y(k) = f(y(k-1), \ldots, y(k-d_y), u(k), \ldots, u(k-d_u)) \quad (1)$$

where $u(k)$ and $y(k)$ denote the input and the output vectors of the system at the discrete-time instant k, respectively and f is an unknown nonlinear function that is to be identified and estimated by the dynamic neural network. The parameters d_y and d_u represent the order of the delays in the output and input channels of the system, respectively. When a system is identified by the NARX model the representation is then expressed by a nonlinear function $\hat{f}(.)$ as follows:

$$\hat{y}(k) = \hat{f}(y(k-1), \ldots, y(k-\hat{d}_y), u(k), \ldots, u(k-\hat{d}_u)) \quad (2)$$

where $\hat{y}(k)$ denotes the estimate of the actual output and the time delays $\hat{d}_y$ and $\hat{d}_u$ should be estimated and selected such that $\hat{d}_y \geq d_y$ and $\hat{d}_u \geq d_u$ (Yazdizadeh & Khorasani, 2002). There are two possible structures that one can utilize for the NARX model. During the training phase, the so-called *series–parallel NARX structure* is used for identification of the system dynamics. In this architecture the actual inputs and outputs of the gas turbine engine are fed to the NARX identification model. During the recall phase, the so-called *parallel NARX structure* is used for verification and validation of the trained NARX representation and model. These two structures are shown in Fig. 2.

Table 1
Gas turbine engine component fault indications.

Component fault	Indication	Symbol
Compressor fouling	Decrease in the compressor flow capacity ($\dot{m}_C$)	F_{mc}
Compressor erosion	Decrease in the compressor efficiency (η_C)	F_{ec}
Turbine fouling	Decrease in the turbine flow capacity ($\dot{m}_T$)	F_{mt}
Turbine erosion	Decrease in the turbine efficiency (η_C)	F_{et}

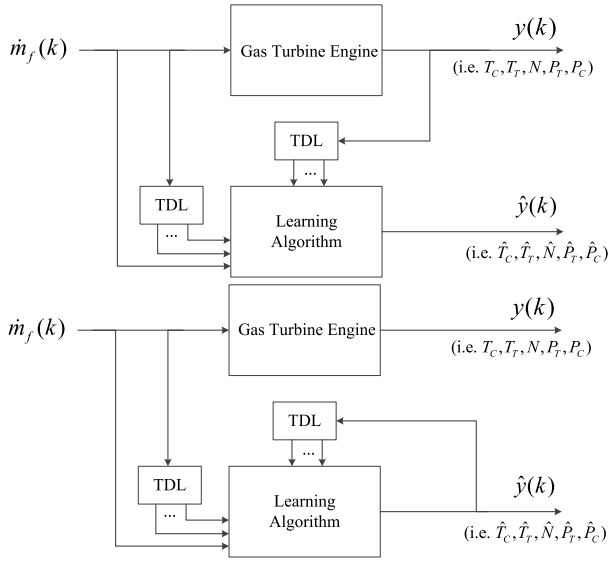

Fig. 2. The gas turbine engine identification process by using the NARX methodology that is used during the training (top plot) and the recall (bottom plot) phases. The block *TDL* represents the Tapped Delay Lines module.

Given that the gas turbine engine is a bounded input bounded output (BIBO) stable system, all the signals that are used for the identification process are bounded. This guarantees that the identified model will also remain BIBO stable. Once it can be ensured that $\hat{y}(k) \approx y(k)$, the series–parallel model will then be replaced by the parallel model. Three separate learning algorithms are now considered for this framework to identify the engine dynamics as described below.

The first learning algorithm uses the MLP neural network within the NARX structure in order to result in what we designate as the MLP-NARX architecture for accomplishing the system identification objective. This structure has previously been reported in the literature (Chen, Billings, & Grant, 1990). The second learning algorithm uses the RBF neural network within the NARX structure to result in what we designate as the RBF-NARX architecture. This structure has also been reported in the literature (Chen, Wang, & Harris, 2008). The third individual learning algorithm is the SVM that is used within the NARX structure to result in what we designate as the SVM-NARX architecture. This structure has also been reported in the literature (Salat, Awtoniuk, & Korpysz, 2013).

3.2. The gas turbine engine model identification: ensemble-based approach

This section develops and presents our proposed methodology for identifying the gas turbine engine dynamics by using ensemble methods. Similar to the single model-based approach that was presented in Section 3.1, a series–parallel architecture is also used for training an ensemble model as shown in Fig. 3. Once the training process is completed where the trained model outputs replicate the outputs of the actual engine, the series–parallel architecture is replaced with a parallel architecture as shown

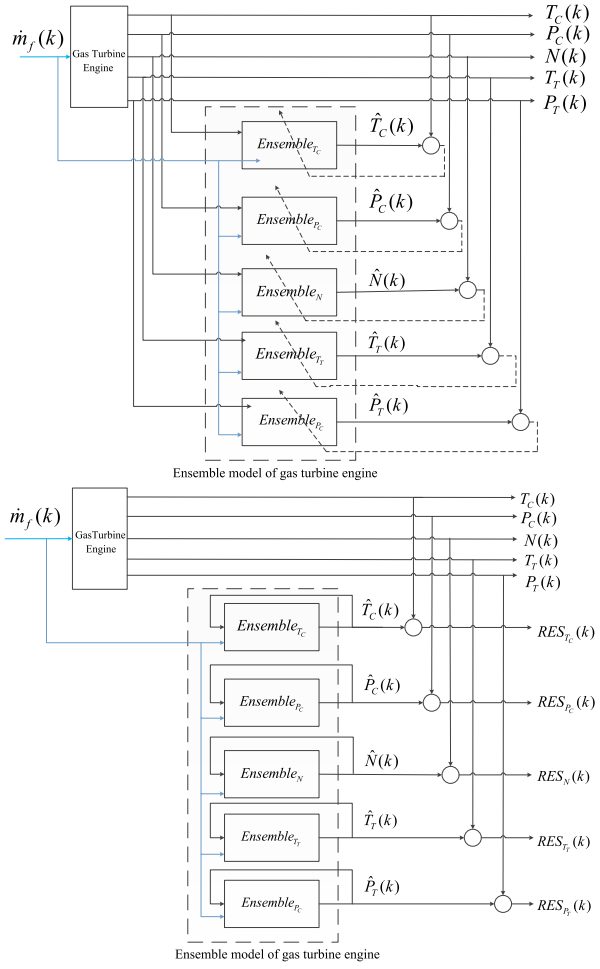

Fig. 3. The architecture of the gas turbine engine ensemble learning model during the learning (top figure) and the recall (bottom plot) phases.

in Fig. 3. Fig. 4 depicts the internal structure of each ensemble learning model. Note that each model has its own specifically optimized parameters, such as the number of neurons and the number of the tapped delay lines (TDLs).

Training of an ensemble learning architecture can be generally accomplished through three steps (Roli, Giacinto, & Vernazza, 2001). The first step is the *ensemble generation*, during which a set of individual models are constructed. The second step is to trim the set of generated models, during what is known as the *ensemble pruning* stage, so that the performance of the ensemble learning model is optimized in terms of the identification accuracy. Finally, the selected models are integrated in the *ensemble integration* step where the final ensemble learning model is constructed. Note that constructing the ensemble model is an iterative procedure as it requires selecting various subsets of the generated models (i.e., in the pruning step) and combining them (i.e., in the integration step) in order to achieve the best possible generalization performance capabilities. These steps are now described in more detail below.

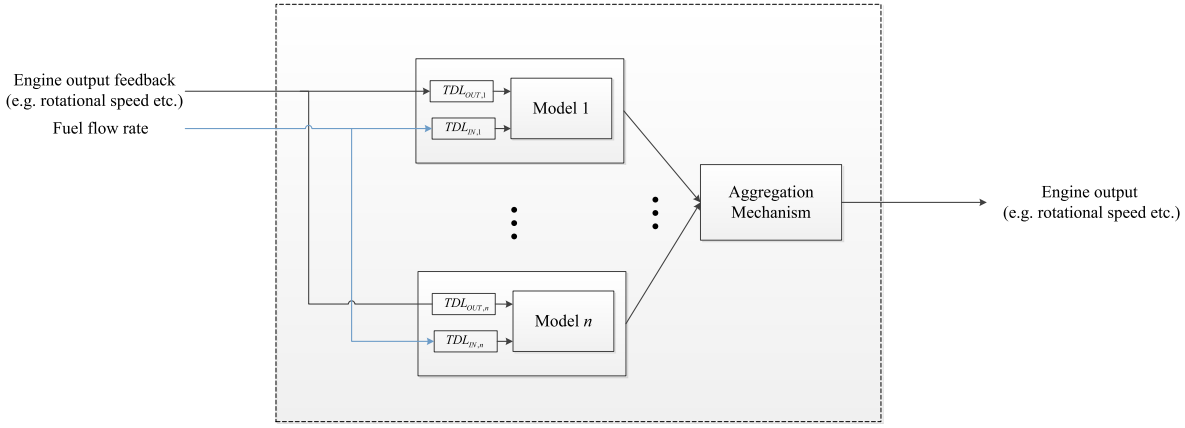

Fig. 4. The internal structure of a given ensemble learning model (refer to Fig. 3).

3.2.1. Ensemble generation

Ensemble generation approaches are divided into (a) *homogeneous* schemes, where the models are generated all by using the same learning algorithm, and (b) *heterogeneous* schemes, where different learning algorithms are used for training each member of the ensemble. As discussed in Section 2, diversity among models is a key requirement and essential for constructing an ensemble of learning models. The source of diversity in the homogeneous ensemble generation is accomplished by using different training data. Whereas, the source of diversity in the heterogeneous ensemble is accomplished through the inherent properties of different learning algorithms. Homogeneous ensemble generation is widely addressed and investigated in the literature (Brown et al., 2005), however in contrast the number of work on heterogeneous ensemble generation is relatively quite limited (Mendes-Moreira et al., 2012). In our work here, one homogeneous and two heterogeneous ensemble learning models for the gas turbine engine are developed by using the MLP-NARX, the RBF-NARX and the SVM-NARX structures as candidate ensemble learning machines.

Homogeneous ensemble generation is the best covered area of the ensemble learning in the literature (Brown et al., 2005). In this approach, ensemble members are generated by using the same learning algorithm (e.g., the same kind of neural networks), and diversity among them are ensured by *altering the training data*. Alternatively, one may diversify homogeneous models by using the same learning method subject to different set of parameters (e.g., neural networks with different number of hidden layers and neurons). Comparative studies between these two approaches have concluded that altering the training data is generally more effective than altering the network parameters (Opitz & Macli, 1999). Several approaches have been suggested in the literature for training ensemble systems that manipulate the training data. Bagging (bootstrap aggregating) has been extensively studied as a homogeneous ensemble method (Mendes-Moreira et al., 2012). In this approach, the original training data is re-sampled by the bootstrap sampling in order to obtain several training sets corresponding to a given training data. The authors in Breiman (1996) and Domingos (1997) have provided valuable insights as to why bagging works.

Boosting is another approach that works by manipulating the training data to ensure diversity. Similar to the bagging, several training data sets are generated by re-sampling the corresponding training data. However, unlike bagging the probability of being selected is not necessarily the same for different samples. In fact, the probability of being selected is initially the same for all samples, however in subsequent iterations the samples that lead to more inaccurate predictions will be given higher probability of being selected. Boosting was originally developed for classification problems. Although, several modifications to it are proposed in the literature for regression problems, none has demonstrated to be as promising as that of the bagging method (Granitto, Verdes & Ceccatto, 2005).

In the *heterogeneous* ensemble, on the other hand, distinct learning models are trained by using the same training data. There are very few works in the literature that use different architectures for ensemble learning systems (Mendes-Moreira et al., 2012). The diversity among the models is ensured through different learning algorithms. This approach is studied much less in the literature; however, some results are reported by using heterogeneous ensembles (Webb & Zheng, 2004). The diversity in this approach is ensured through inherent properties of different learning algorithms. The main challenge is lack of control on the diversity of the ensemble during the generation phase.

3.2.2. Ensemble pruning

Ensemble pruning refers to the procedure for trimming the set of trained models with the goal of improving the generalization error of the resulting ensemble. It is also used to reduce the complexity of the overall ensemble system. Several pruning methods are proposed and compared in the literature, including (a) ranking based on the accuracy, (b) forward selective search (FSS) algorithm, (c) backward selective search (BSS), and (d) BSS with Ranking (BSSwR). It has been reported in Guilherme and Zuben (2006), Roli et al. (2001) that the FSS demonstrates a better performance as compared to the other approaches. In this paper, a heterogeneous ensemble with the FSS as a pruning algorithm will be used to identify the gas turbine engine dynamics. First, several models are trained using each of the MLP-NARX, the RBF-NARX, and the SVM-NARX structures to model, represent, and identify the single input multiple output dynamics of the engine. To limit the complexity of the solution to the problem, a subset of trained models is selected based on their performance (i.e., the 10 best RBF-NARX models, the 10 best MLP-NARX models, and the 10 best SVM-NARX models are selected out of a total of 105 models each). The ensemble learning system members are then selected by using the FSS algorithm. The FSS algorithm is initialized by using the model that yielded the best performance in the set. Each time a new model is added to the ensemble, all the candidates are tested and the model with the maximal improvement is then added as the next model. It should be pointed out that the FSS is an iterative procedure that requires combining various subsets of generated models in order to construct the ensemble with the best achievable generalization performance.

3.2.3. Ensemble integration

Ensemble integration combines identifications that are made by various models into a module for generating and constructing the final ensemble. For system identification and regression problems the integration mechanism combines the models by using a linear combination of identifiers as represented by (Mendes-Moreira et al., 2012):

$$f_{ensemble}(x) = \sum_{i}^{n} \alpha_i f_i(x)$$

where $f_{ensemble}(x)$ represents the output of the ensemble learning model for the input pattern x, $f_i(x)$ denotes the output of the ith model corresponding to the input pattern x, α_i denotes the averaging weight for the ith model, and n denotes the number of selected ensemble members. In other words, the ensemble combination for a system identification or a regression problem can be restated as that of optimizing the averaging weights α_is.

Merz and Pazzani (1999) conducted a comparative study to determine the most effective ensemble combination technique. Several ensemble combination techniques were studied (i) the Generalized Ensemble Method (GEM) (Perrone & Cooper, 1994), (ii) the Basic Ensemble Method (BEM) (Perrone & Cooper, 1994), (iii) the Linear Regression (LR), (iv) the Gradient Descent, and (v) the Exponential Gradient Descent (Kivinen & Warmuth, 1997). Among these approaches, the *gradient descent method* was found to demonstrate a better generalization performance as compared to other approaches.

The objective function that is minimized by using the gradient descent approach is the ensemble RMSE as:

$$\min_{\alpha_i} \quad RMSE_{ensemble}(\alpha)$$

$$\textit{Updating rule}: \quad \alpha_{k+1} = \alpha_k - \gamma \nabla RMSE_{ensemble}(\alpha_k)$$

where $RMSE_{ensemble}(\alpha) = \sum_{i=1}^{n} \alpha_i f_i(x_{training}) - f(x_{training})$, $\alpha = [\alpha_1, \ldots, \alpha_n]^T$, $f_i(x_{training})$ denotes the prediction of the ith model corresponding to the training data $x_{training}$, $f(x_{training})$ denotes the target or the actual value corresponding to the training data $x_{training}$, α_k denotes the value of α at the kth iteration, $\nabla RMSE_{ensemble}(\alpha_k)$ denotes the gradient of the $RMSE_{ensemble}$ at the kth iteration, and γ denotes the step size. The step size, as well-known in optimization techniques, should be carefully selected given that a very large step size may lead to divergence of the gradient descent and a very small step size will take a long time for iterations to converge to a fixed point. It should be pointed out that the step size can be either fixed or adaptive (i.e., it is adjusted or changed at each iteration).

4. Fault detection methodology

Our proposed fault detection methodology consists of two phases. The first phase is to identify the dynamics of the engine by using both individual models as well as integrating them to construct ensemble learning models. The identified model under healthy operation of the engine is then used to generate residual signals to be analyzed for detection of faults that maybe present in the engine. The residual signals are evaluated to determine the health status of the engine. Our proposed fault detection methodology is now given below.

4.1. Fault detection logic

As stated above, the first step in the engine fault detection is the identification phase that was developed and described in Section 3. The engine dynamics is identified by using both the ensemble-based and the individual learning models. A separate model (corresponding to both ensemble-based and individual learning models) is developed for **each** of the engine's five measurable outputs, namely the variable z as defined in Section 2.3. The trained models are then utilized for generating the *residual signals RES* by comparing the actual output of the engine with that of the trained model representing the behavior of the healthy engine as shown in Fig. 5.

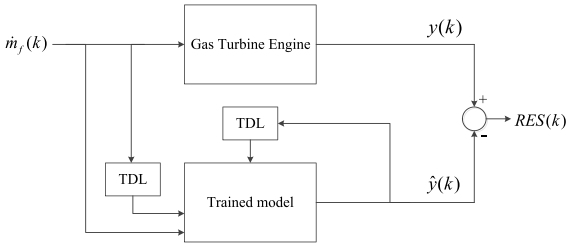

Fig. 5. The schematic for generating the residual signals.

The constructed residual signals are then utilized as indicators of the gas turbine engine health status, given that the residuals change before and after the occurrence of a fault. Consequently, by selecting a proper threshold band the engine faults can be detected by monitoring the variations in the residuals. To generate the threshold bands, the mean (μ) and the standard deviation (σ) of the residuals are obtained when the engine is operating under the healthy condition and under various fuel operating conditions and profiles. The threshold bands are then specified according to $t.h._{upper} = \mu + z\sigma$ and $t.h._{lower} = \mu - z\sigma$, corresponding to the upper and lower bands, respectively. By assuming a normal distribution associated with residuals, a 99% confidence interval can be determined by selecting $z = 2.6$. A fault in the gas turbine engine would then be detected if any of the five residual signals passes its corresponding thresholds that are defined by the band $[t.h._{lower}, t.h._{upper}]$.

5. Fault isolation methodology

This section provides our proposed methodology for evaluating the residuals that were utilized in the previous section for detecting the presence of a fault to now solve the fault isolation problem. This is achieved by using a static ensemble of neural network classifiers. The use of static MLP neural networks for fault isolation of a gas turbine engine is reported in several publications including but not limited to Amanda and Sharkey (2002), Loboda et al. (2011), Sharkey et al. (2000), Xiao, Eklund, Goebel, and Cheetham (2007) and Yan and Xue (2008). Similar methodology is used here for isolating the engine faults, but instead we utilize an ensemble of neural network models that we designed in Section 4 for performing the fault detection task. The inputs to MLPs in this case should be spatial data as opposed to residual signal time-series data that were used for the fault detection task. Therefore, the residuals are pre-processed to make them suitable as inputs to static neural network structures.

We consider only two fault severity ranges for sake of simplicity and without loss of generality. Namely, we consider (a) severities $<3\%$, and (b) severities $>3\%$. It should be emphasized that this is for illustration purposes only as more severity ranges can be easily considered by incorporating additional fault class labels. The class labels that are considered for a single fault scenario are listed in Table 2.

The designed neural network fault classifier receives variations of the residual signals before and after the fault occurrence, and returns a fault label corresponding to the isolated fault. This is the function of the residual evaluation block that is shown in Fig. 6. This block continuously compares the residual signals with their corresponding threshold bands. Once at least one of the residuals

Table 2
The considered single fault classes.

Description	Symbol
Class 1: Decrease in the compressor flow capacity ($\dot{m}_C$) with severity less than or equal to 3%	$F_{mc} \leq 3\%$
Class 2: Decrease in the compressor flow capacity ($\dot{m}_C$) with severity $> 3\%$	$F_{mc} > 3\%$
Class 3: Decrease in the compressor efficiency (η_C) with severity $\leq$ to 3%	$F_{ec} \leq 3\%$
Class 4: Decrease in the compressor efficiency (η_C) with severity $> 3\%$	$F_{ec} > 3\%$
Class 5: Decrease in the turbine flow capacity ($\dot{m}_T$) with severity $\leq$ to 3%	$F_{mt} \leq 3\%$
Class 6: Decrease in the turbine flow capacity ($\dot{m}_T$) with severity $> 3\%$	$F_{mt} > 3\%$
Class 7: Decrease in the turbine efficiency (η_T) with severity $\leq$ to 3%	$F_{et} \leq 3\%$
Class 8: Decrease in the turbine efficiency (η_T) with severity $> 3\%$	$F_{et} > 3\%$

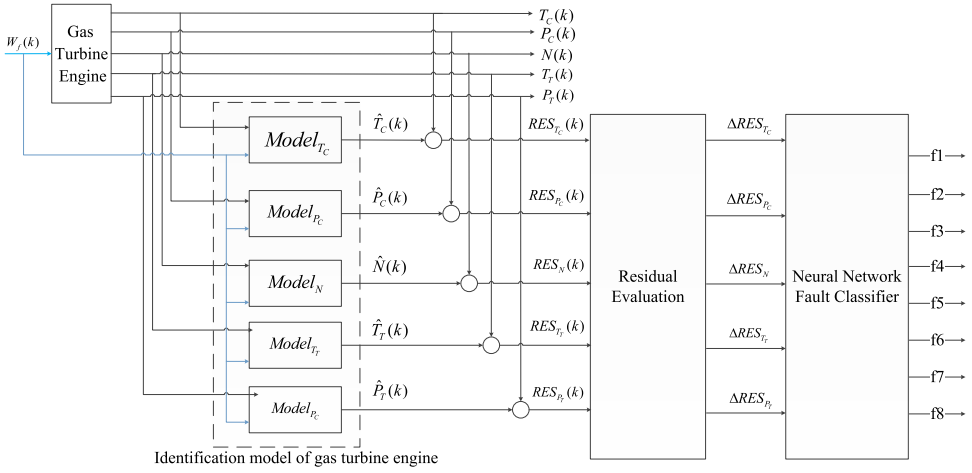

Fig. 6. The schematic of the fault isolation methodology.

has exceeded its threshold band the presence of a fault is declared. The residual evaluation block then computes the variations of the residual signal before and after the fault detection declaration. This is now used as an input to the neural network for performing the fault classification task. In other words, the output of the residual evaluation block can be defined and specified according to:

$$\Delta RES = \left[\Delta RES_{T_C}, \Delta RES_{P_C}, \Delta RES_N, \Delta RES_{T_T}, \Delta RES_{P_T}\right].$$

It should be emphasized that the fault isolation neural network is only utilized when a fault is detected and the ΔRES vector is defined. Note that the value of ΔRES is only defined for $t \geq t_D$, and not defined otherwise, where t_D denotes the fault detection time communicated by the fault detection module.

The fault isolation task will be performed by both using the residuals that are obtained from the single model-based solutions as well as those that are obtained from the ensemble-based solutions.

6. Multiple fault isolation

In the previous section, the fault isolation problem in presence of only a single fault at any given time was considered. In this section, the goal is to isolate simultaneously occurring faults. Isolating multiple faults is a complex problem. Therefore, in order to limit the complexity of the proposed solution, it is assumed that only two concurrent faults may occur at any time. Extensions to more than two concurrent faults are straightforward and not included here. We assume that the first fault occurs at $t_1 = 20$ s and the second fault occurs at $t_2 = 30$ s. The fault scenarios investigated are as follows.

Classes 1–4: These classes are associated with single F_{ec}, F_{et}, F_{mc}, and F_{mt} reductions in severities between 1% and 6%. **Class 5** ($\{F_{ec}, F_{mc}\}$): This class is associated with simultaneous 1% to 6% faults in both compressor efficiency (F_{ec}) and compressor mass flow rate (F_{mc}). **Class 6** ($\{F_{ec}, F_{et}\}$): This class is associated with simultaneous 1% to 6% faults in both compressor efficiency (F_{ec}) and turbine efficiency (F_{et}). **Class 7** ($\{F_{ec}, F_{mt}\}$): This class is associated with simultaneous 1% to 6% faults in both compressor efficiency (F_{ec}) and turbine mass flow rate (F_{mt}). **Class 8** ($\{F_{mc}, F_{et}\}$): This class is associated with simultaneous 1% to 6% faults in both compressor mass flow rate (F_{mc}) and turbine efficiency (F_{et}). **Class 9** ($\{F_{mc}, F_{mt}\}$): This class is associated with simultaneous 1% to 6% faults in both compressor mass flow rate (F_{mc}) and turbine mass flow rate (F_{mt}). **Class 10** ($\{F_{et}, F_{mt}\}$): This class is associated with simultaneous 1% to 6% faults in both turbine efficiency (F_{et}) and turbine mass flow rate (F_{mt}).

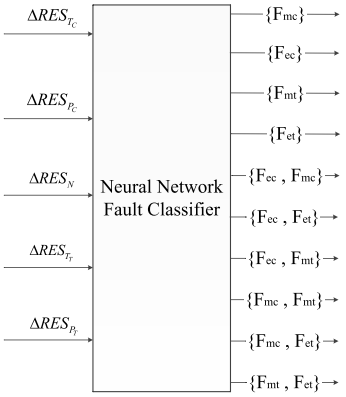

Fig. 7. The schematic of the proposed neural network for achieving multiple fault isolation.

The structure of the ensemble MLP classifier that is used for multiple fault isolation is shown in Fig. 7. The neural network receives the set of variations in residuals before and after the fault detection module declaration and decision.

7. Homogeneous and heterogeneous dynamic neural network ensemble identifiers

7.1. Dynamic neural network system identifiers

The data used for identification of the engine dynamics is obtained from the model that is provided in Section 2.3. It is assumed that the engine is operating for *one hour (3600 s)*. A total of 3601 input–output data samples are collected. The generated data contains the measured five (5) outputs as well as the engine input that is represented by the fuel flow rate.

Empirically, it is observed that normalization of data improves the performance of the ensemble learners. Therefore, the following min–max normalization function is applied as a pre-processing step to the data, that is, the normalized data is computed from $X_n = 2 \times \frac{X_{\max} - X}{X_{\max} - X_{\min}}$. During construction of each learning model the generated data is divided into the training, the testing, and the cross-validation subsets.

(1) The Gas Turbine Engine Dynamic Identification using MLP-NARX

Five MLP-NARX structures are trained to model the engine output measurements P_T, T_T, P_C, T_C and N. The networks are denoted by MLP_{P_T}, MLP_{T_T}, MLP_{P_C}, MLP_{T_C}, MLP_N, respectively. Constructing the MLP-NARX structures requires that one determines the network parameters, such as (i) the number of hidden neurons, (ii) the number of hidden layers, (iii) the number of delays, as well as (iv) the size of the training, testing and validation sets. To limit the network complexity, the number of hidden layers is limited to one. The networks are then constructed such that the generalization performance of the trained networks is maximized. The *Root Mean Squared Error (RMSE)* is applied to evaluate the training and the generalization performance of the trained networks. The *Mean of Absolute Error* (μ_{ae}) and *Standard Deviation of Absolute Error* (σ_{ae}) are also evaluated, given that the error is expected to be randomly distributed around zero for an appropriately constructed network.

The construction of MLP_{P_T}, MLP_{T_T}, MLP_{P_C}, MLP_{T_C}, MLP_N networks is repeated several times by using various parameters and different combination of the percentage of the training and the testing data. For example, for the optimal MLP_{P_C} the number of delays is set to 7 and the number of neurons is set to 11, whereas for the optimal MLP_N the number of delays is set to 6 and the number of neurons is set to 10, to name a couple of structures. Consequently, a set of trained models are obtained that will be used later for constructing the ensemble models of the engine. After various experimentations it was determined that 40% of the available training data could be used for the cross-validation stage to construct the most suitable networks.

(2) The Gas Turbine Engine Dynamic Identification using RBF-NARX

Five RBF-NARX structures are trained to model the engine output measurements P_T, T_T, P_C, T_C and N. The networks are denoted by RBF_{P_T}, RBF_{T_T}, RBF_{P_C}, RBF_{T_C}, RBF_N, respectively. Constructing the RBF-NARX structure requires that one determines the network parameters, such as (i) the number of RBF neurons, (ii) the number of delays, and (iii) the size of the training, testing, and validation data. After experimenting with different parameter values the ones that yielded the best generalization performance are selected.

The construction of RBF_{P_T}, RBF_{T_T}, RBF_{P_C}, RBF_{T_C}, RBF_N networks is repeated several times by using various parameters and different combination of the percentage of the training and the testing data. Consequently, a set of trained models are obtained that will be used later for constructing the ensemble models of the gas turbine engine. For example, for the optimal RBF_{P_C} the number of delays is set to 10 and the number of neurons is set to 7, whereas for the optimal RBF_N the number of delays is set to 8 and the number of neurons is set to 11, to name a couple of structures.

(3) The gas turbine engine dynamic identification using SVM-NARX

Five SVM-NARX structures are trained to model the engine output measurements. The construction of SVM_{P_T}, SVM_{T_T}, SVM_{P_C}, SVM_{T_C}, SVM_N networks is repeated several times by applying different network parameters and different percentage combinations of the training and testing data sets. The results produced a set of trained models that will be used subsequently for constructing the ensemble learning models of the gas turbine engine.

7.2. Ensemble I: heterogeneous ensemble with ranked pruning

Heterogeneous ensembles with ranked pruning have been reported in the literature as in Kotsiantis and Pintelas (2005). In this approach, first a pool of individual learners are trained by using different learning algorithms. The most accurate models are then selected for each learning algorithm to be aggregated for generating the final ensemble learning model. The only source of diversity in this approach is the use of heterogeneous ensembles (that is, by using different neural network algorithms).

In this work, the above methodology is first used to identify the engine dynamics. As discussed in the previous subsection, several machine learners are trained from the MLP-NARX, RBF-NARX, and SVM-NARX architectures to identify the engine dynamics. For each learning algorithm (e.g., the MLP-NARX) the identification architecture with the *best performance* is selected from the pool of individual trained learning models. The selected identified models are then combined by using the weighted averaging technique. Two combination methods are used to determine the optimal averaging weights, namely the *generalized ensemble* and the *gradient descent*.

The performance of the *heterogeneous* ensemble with *ranked pruning* and *generalized ensemble method* as the integration methodology was determined. The performance of the *heterogeneous* ensemble with *ranked pruning* and *gradient descent* as the integration method was also determined. Based on the extensive simulations that are conducted it was concluded that the gradient descent approach has a much better performance in terms of the generalization error when compared with the generalized ensemble method. Detailed comparative studies between the ensemble and the individual learners are provided in the following subsections. Table 3 summarizes the main procedure adopted for the heterogeneous ensemble training with ranked pruning.

7.3. Ensemble II: heterogeneous ensemble using the Forward Sequential Selection (FSS)

The use of the heterogeneous ensemble with the Forward Sequential Selection (FSS) as a pruning algorithm has been reported in several publications in the literature, including but not limited to Guilherme and Zuben (2006), Perrone and Cooper (1994), and Roli et al. (2001). Forward selection is initialized with an empty set and iteratively models are added with the aim of decreasing the expected prediction error. Two different versions of the FSS are presented in the literature, namely the Forward Sequential Selection with Ranking (FSSwR) and the Forward Sequential Selection (FSS). The FSSwR ranks all the candidates with respect to their performance on a given training set. It selects the candidate at the top of the list until the performance of the ensemble decreases. In the FSS algorithm, each time a new candidate is added to the ensemble, all the candidates are tested and the one that leads to the maximal improvement in the ensemble performance is selected. In Yates and Partridge (1996), the FSS was modified by adding a diversity measure. In this

Table 3
Summary of the heterogeneous ensemble training with ranked pruning.

1: Several models are trained by using the MLP-NARX, the RBF-NARX, and the SVM-NARX architectures corresponding to the five gas turbine engine outputs.
2: For each architecture, the best trained model is selected for the subsequent integration.
3: The generalized ensemble method and the gradient decent algorithms are used to determine the averaging weights.
4: The *initial condition* of the gradient decent algorithm is selected such that the best learner in the pool has the maximum contribution.

Table 4
Summary of the heterogeneous ensemble training with the Forward Sequential Selection (FSS) pruning algorithm.

1: Several models are trained with MLP-NARX, RBF-NARX, and SVM-NARX architectures to identify the gas turbine engine five output measurements.
2: A subsets of 10 best RBF-NARX models, the 10 best MLP-NARX models, and the 10 best SVM-NARX models are selected from the pool of models trained in Step 1 ***in order to reduce the complexity of the solution***.
3: Corresponding to each gas turbine engine output (i.e., P_C, T_C, N, P_T, T_T), the FSS algorithm is initialized with the best trained model.
4: Each time a new model is added to the ensemble, all the candidates are tested and the model with the largest improvement is then added as the next model.
5: Each time a new model is added to the ensemble, the optimal combining weights are re-calculated by using gradient descent algorithm as discussed in Section 3.2.3.
6: All the evaluations for the FSS algorithm are performed on the training set.

approach, the criterion for inclusion is a diversity measure, and the new model should be diverse from the previously selected models.

In this work, a heterogeneous ensemble with the FSS as a pruning algorithm is utilized for identifying the gas turbine engine dynamics. Several models are trained by employing the MLP-NARX, the RBF-NARX, and the SVM-NARX architectures for modeling the five output measurements of the engine dynamics. As stated earlier, in order to limit the complexity of the resulting solution, here a subset of the trained models is selected based on their achieved performance (i.e., the 10 best RBF-NARX models, the 10 best MLP-NARX models, and the 10 best SVM-NARX models are selected). The members of the ensemble are then selected by invoking the FSS algorithm. The FSS algorithm is initialized by using the model with the best performance in the pool. Each time a new model is added to the ensemble, all the candidates are tested and the model with the maximal improvement is then added as the next model. In each iteration, all the selected models are aggregated by using the gradient descent algorithm. It should be noted that the generalized ensemble method is not employed due to its poor performance as pointed out in the previous subsection. Table 4 summarizes the main procedure adopted for construction of the heterogeneous ensemble system.

7.4. Ensemble III: homogeneous with bagging

Bootstrap sampling or the *bagging* is one of the most extensively used techniques for manipulation of the training data (Mendes-Moreira et al., 2012). Empirical studies have shown that bagging is a simple and effective method in reducing the prediction error in both classification and regression problems (Domingos, 1997). The main idea is to train a learning model by using different subsets of the training data that are generated by the bootstrap sampling procedure. In the bagging method, a training set with the size s is selected where several bootstrap replicates of it are then constructed by taking s samples out of it *with replacement*. Thus, a new training set with the same size is generated where each sample in the original training set may appear once, more than once, or may not appear at all (Domingos, 1997). The learning algorithm then uses this new training set. This procedure will be repeated several times where all the models are then aggregated to construct the final ensemble. Fig. 8 depicts the main components of the bagging procedure.

In this subsection, a homogeneous ensemble is trained by using the bagging for modeling each of the five gas turbine engine outputs. As shown below the RBF-NARX architecture outperforms the MLP-NARX and SVM-NARX models for identifying the dynamics of the engine outputs. Therefore, the RBF-NARX model is used to form the homogeneous ensemble. Corresponding to the network parameters (i.e., the number of neurons, the number of time delays, and the size of training data), those

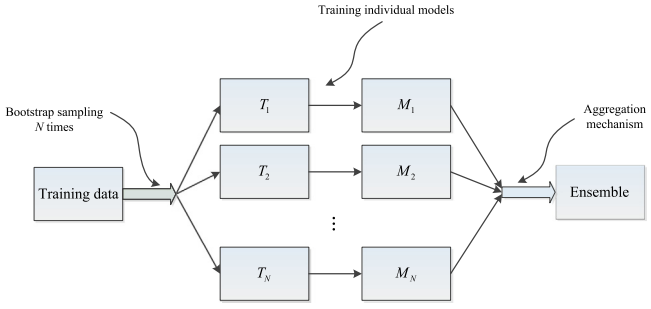

Fig. 8. The schematic of the homogeneous ensemble learning with bagging.

corresponding to models with the best RMSE performance are selected.

Consequently, several RBF-NARX models having exactly the same parameters are trained. The only factor that is different is the training data, where for different homogeneous models the training data is obtained by the bootstrap sampling as indicated above. As previously stated, the number of models in an ensemble plays an important role in its performance (Hansen & Salamon, 1990). Thus, in order to have a fair comparison between the heterogeneous ensembles (i.e., the ensembles I and II) and the homogeneous ensemble (the ensemble III), the same number of models are selected for all of them. A weighted averaging is used to integrate the ensemble models. The weights are optimized by using the gradient descent algorithm with the *objective of minimizing the RMSE on the training data*.

7.5. Comparative results

Table 5 shows comparisons between various methods for modeling only the compressor pressure, for typical illustrative purposes. It can be concluded that the heterogeneous ensemble with the FSS pruning scheme (ensemble II) has a better performance in modeling and identifying the unknown engine dynamics. Moreover, the stand-alone RBF-NARX model has the best performance among the stand-alone trained models.

The gradient descent method (as previously stated in Section 3.2) is used to minimize the RMSE by optimizing the averaging weights of the learning models. The α_is that are obtained for integrating the three individual learners are given in Table 6. A summary of the ensemble system performance for identification of each of the five gas turbine engine outputs is given in Table 7.

A comparative performance study on the heterogeneous ensemble II with each of the individual learners are presented in Tables 8 through 10. To summarize, one can observe that the ensemble model demonstrates a significantly better performance in modeling and identification of the gas turbine engine dynamics.

Table 5
Comparison of various identification methods for the compressor pressure.

	$RMSE_{total}$	$RMSE_{train}$	$RMSE_{test}$	μ_{ae}	σ_{ae}
MLP-NARX	0.0531	0.060215	0.040119	0.033786	0.040972
RBF-NARX	0.026612	0.027382	0.026087	0.021582	0.015573
SVM-NARX	0.041185	0.037001	0.046766	0.028081	0.03013
Ensemble I	0.024672	0.024926	0.024285	0.02894	0.02103
Ensemble II	0.023135	0.023115	0.023166	0.023113	0.0010121
Ensemble III	0.031684	0.027609	0.042234	0.030476	0.031543

Table 6
The gradient descent coefficients for integration of ensemble learning models.

	α_{MLP}	α_{RBF}	α_{SVM}
P_C	9.5212×10^{-5}	1.0001	9.4425×10^{-5}
T_C	0.0646	0.8164	0.1190
N	0.0685	1.7562	−0.8261
P_T	0.0048	0.8974	0.109
T_T	−0.0995	1.086	0.0131

Table 7
The performance of the heterogeneous ensemble with the FSS pruning.

	$RMSE$	μ_{ae}	σ_{ae}
P_C	0.023135	0.023113	0.00101
T_C	0.5049	0.3883	0.1045
T_T	10.139	7.848	3.820
P_T	0.016865	0.016841	0.000896
N	6.8672	4.563	3.654

Table 8
Comparison of single learning models and the ensemble method for identification of the compressor pressure.

	$RMSE_{total}$	$RMSE_{train}$	$RMSE_{test}$	μ_{ae}	σ_{ae}
MLP	0.053	0.060215	0.040119	0.03378	0.04097
RBF	0.02661	0.027382	0.026087	0.02158	0.01557
SVM	0.04118	0.037001	0.046766	0.02808	0.0301
Ensemble II	0.02313	0.023115	0.023166	0.02311	0.001012

Table 9
Comparison of single learning models and the ensemble method for identification of the rotational speed.

	$RMSE_{total}$	$RMSE_{train}$	$RMSE_{test}$	μ_{ae}	σ_{ae}
MLP	22.7076	24.7451	21.2417	19.597	11.4732
RBF	17.8349	19.7104	15.7359	14.2826	10.6831
SVM	24.6728	26.5164	22.6791	21.236	12.562
Ensemble II	6.8672	6.7947	6.9391	4.5636	3.6543

Table 10
Comparison of single learning models and the ensemble method for identification of the turbine temperature.

	$RMSE_{total}$	$RMSE_{train}$	$RMSE_{test}$	μ_{ae}	σ_{ae}
MLP	41.9615	40.4995	43.3752	37.1441	19.5246
RBF	13.4734	15.2044	12.1843	9.8925	9.1487
SVM	104.3983	107.0395	102.6002	90.6818	51.7326
Ensemble II	10.1397	9.8846	10.3887	7.8483	3.821

Also, it can be concluded that the RBF-NARX model has the best performance among the individual learning models.

Consequently, in the remainder of this work, the trained heterogeneous ensemble model II and the RBF-NARX model will be used for generating the residuals for performing the FDI task. A comparative study will be conducted between the performance of the ensemble-based and single-model based FDI methodologies.

8. Fault detection and isolation case studies

As described earlier, the engine FDI problem and task consists of two phases. In the first phase, the engine dynamics is identified by using both single model-based and ensemble-based learning approaches, and corresponding to each methodology residuals are generated. In the second phase, the obtained residuals from both methodologies are evaluated to detect and isolate the presence of a fault. A comparative study is conducted between the two approaches to evaluate and demonstrate the advantages and capabilities of the ensemble-based methodology.

8.1. Fault detection results

Four engine component faults are considered in this work as described in Table 1. Multiple fault scenarios are considered that vary in terms of (a) the fault type, (b) the fault severity or magnitude, and (c) the fuel flow rate. The simulations are conducted where the fuel flow rate varies between $\dot{m}_f = 0.7, 0.75, 0.8, 0.85$ of its maximum rate. The residual signals are generated by using (a) the heterogeneous ensemble model II, and (b) the individual RBF-NARX model. The residuals are evaluated against the computed thresholds as described in Section 4.1. A fault is detected if any of the residuals exceeds its corresponding determined threshold.

8.2. Scenario I: faults in the compressor efficiency

Consider a decrease in the efficiency of the compressor faults with severities of 1%, 2%, 4%, 6%, and 8% in the gas turbine engine. The fuel flow rate is allowed to vary within the range of 70%–85% of its maximum rate. The instant where the faults are injected to the engine is $t = 20$ s. The residuals corresponding to both the ensemble-based II and the single-model (RBF-NARX) based methodologies are obtained. Figs. 9 and 10 show comparisons between the fault detection results of the ensemble approach (Fig. 10) and the one corresponding to the single-model approach (Fig. 9). These figures indicate an improvement in the fault detection accuracy that is achieved by using the ensemble learning scheme. Note that in Fig. 9 only one residual has exceeded its threshold bands and remained consistently outside the bands, whereas in Fig. 10 two residuals have achieved this property, leading to a more reliable decision in detection of the fault. Moreover, the other residuals in Fig. 10 practically remain at all times within their bands, whereas the residuals in Fig. 9 oscillate consistently in and out of the threshold line generating numerous undesirable false negative flags.

8.3. Scenario II: faults in the compressor mass flow rate

Consider a decrease in the effectiveness of the compressor mass flow rate faults with severities of 1%, 2%, 4%, 6%, and 8% in the gas turbine engine. The fuel flow rate is allowed to vary between 70% and 85% of its maximum rate. The instant where the faults are injected to the engine is $t = 20$ s. The residuals corresponding to both the ensemble-based II and the single-model (RBF-NARX) based methodologies are obtained. The results (graphs are not shown) confirm that the ensemble approach yields more a accurate detection performance.

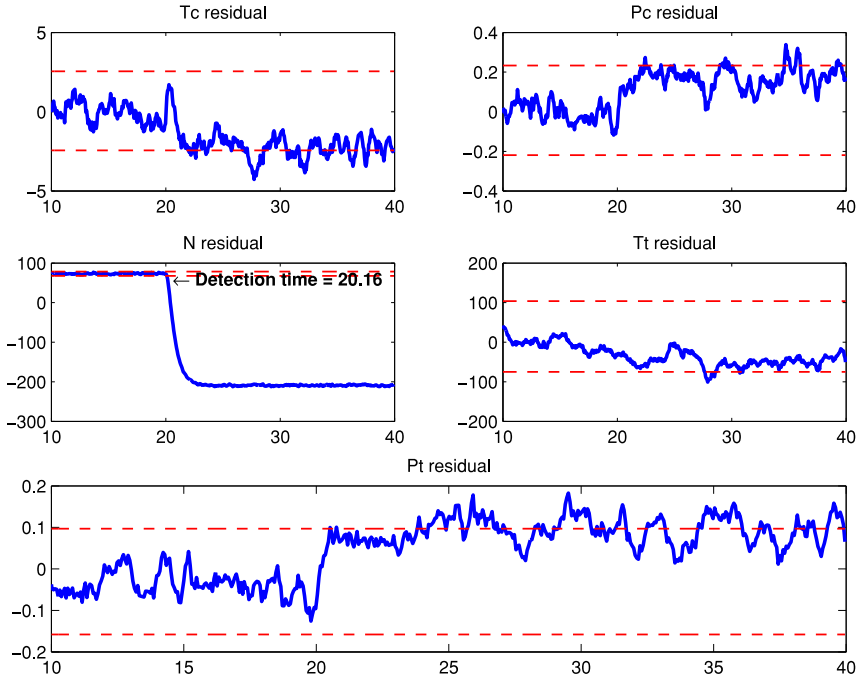

Fig. 9. The residuals that are generated by using the RBF-NARX model subject to a 2% decrease in the compressor efficiency injected at $t = 20$ s.

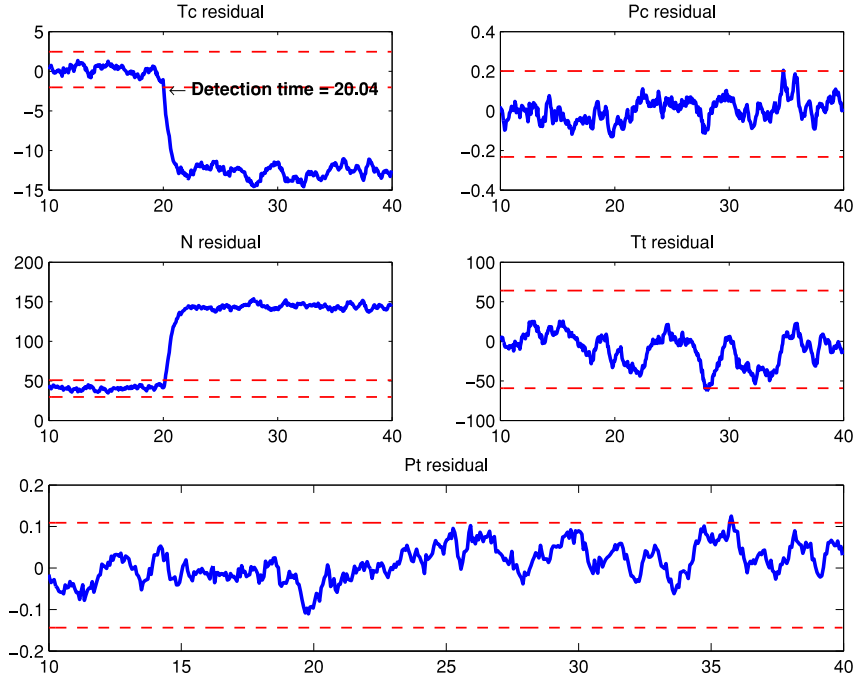

Fig. 10. The residuals that are generated by using the ensemble II model subject to a 2% decrease in the compressor efficiency injected at $t = 20$ s.

8.4. Scenario III: faults in the turbine efficiency

Consider a decrease in the turbine efficiency fault with severities of 1%, 2%, 4%, 6%, and 8% in the gas turbine engine. The fuel flow rate is allowed to vary within the range between 70% and 85% of its maximum rate. The instant where the faults are injected to the engine is $t = 20$ s. The residuals corresponding to both the ensemble-based II and the single-model (RBF-NARX) based methodologies are obtained. The results (graphs are not shown) confirm that the ensemble approach yields a more accurate detection performance.

8.5. Scenario IV: faults in the turbine mass flow

Consider a decrease in the effectiveness of the turbine mass flow rate faults with severities of 1%, 2%, 4%, 6%, and 8% in the gas turbine engine. The fuel flow rate is allowed to vary between 70% and 85% of its maximum rate. The instant where the faults are injected to the engine is $t = 20$ s. The residual signals corresponding to both the ensemble-based II and the single-model (RBF-NARX) based methodologies are obtained. Figs. 11 and 12 show comparisons between the fault detection results of the ensemble approach (Fig. 12) and the one corresponding to the single-model approach

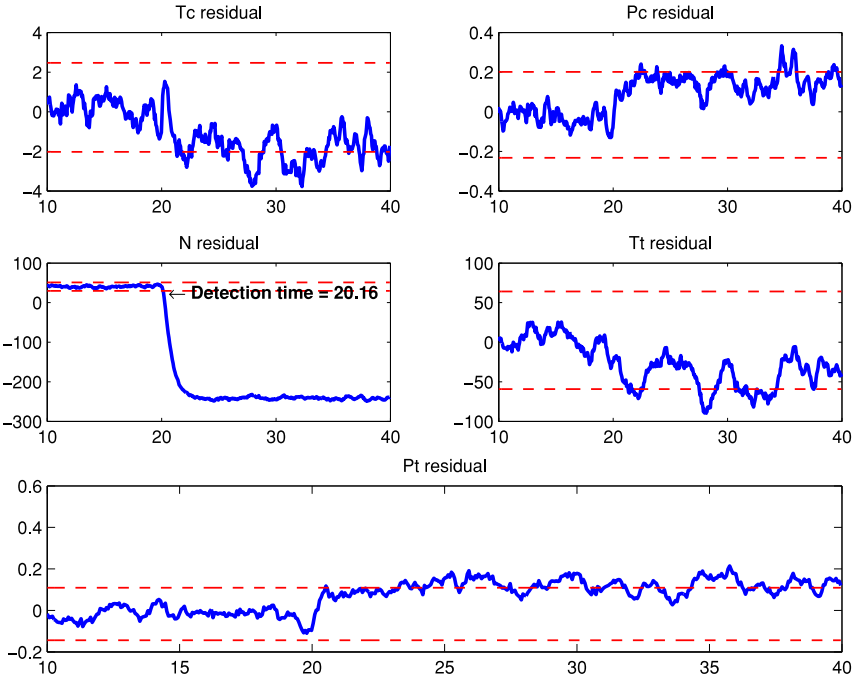

Fig. 11. The residuals that are generated by using the RBF-NARX model subject to a 2% decrease in the turbine mass flow injected at $t = 20$ s.

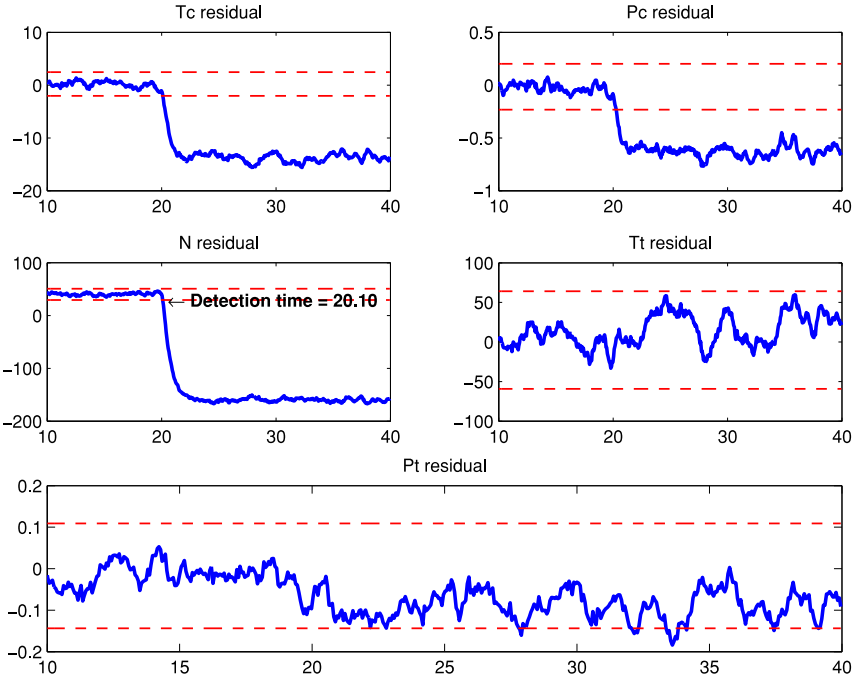

Fig. 12. The residuals that are generated by using the ensemble II model subject to a 2% decrease in the turbine mass flow injected at $t = 20$ s.

(Fig. 11), which indicate an improvement in the fault detection accuracy that is achieved by using the ensemble learning scheme. Note that in Fig. 11 only one residual has exceeded its threshold bands and remained consistently outside the bands, whereas in Fig. 12 three residuals have achieved this property, leading to a more reliable decision in detection of the fault. Moreover, the other residuals in Fig. 12 practically remain at all times within their bands, whereas the residuals in Fig. 11 oscillate consistently in and out of the threshold line generating numerous undesirable false negative flags.

8.6. Fault detection performance analysis with confusion matrix

The following metrics are now used to quantitatively evaluate the performance of our proposed fault detection methodologies, namely *Correct Classification Ratio* (*CCR*) $= \frac{t.p.+t.n}{t.p.+t.n.+f.p.+f.n.}$, *Precision* $= \frac{t.n}{t.n.+f.n.}$, *True Positive Rate* (*TPR*) $= \frac{t.p}{t.p.+f.n.}$, *False Positive Rate* (*FPR*) $= \frac{f.p}{f.p.+t.n.}$, *True Negative Rate* (*TNR*) $= \frac{t.n}{t.n.+f.p.}$, and *False Negative Rate* (*FNR*) $= \frac{f.n}{t.p.+f.n.}$, where t.p. (true positive) denotes the number of cases that are classified as faulty

Table 11
The fault detection accuracy of the single model-based RBF-NARX solution (fault severity = 1%).

	F_{mc}	F_{ec}	F_{mt}	F_{et}
CCR	60%	90%	100%	90%
Precision	57.14%	83%	100%	83%
TPR	40%	80%	100%	80%
FPR	20%	0%	0%	0%
TNR	80%	100%	100%	100%
FNR	60%	20%	0%	20%

Table 12
The fault detection accuracy of the ensemble-based II solution (fault severity = 1%).

	F_{mc}	F_{ec}	F_{mt}	F_{et}
CCR	70%	100%	100%	100%
Precision	100%	100%	100%	100%
TPR	100%	100%	100%	100%
FPR	60%	0%	0%	0%
TNR	40%	100%	100%	100%
FNR	0%	0%	0%	0%

Table 13
The fault detection accuracy of the single model-based RBF-NARX solution (fault severity = 2%).

	F_{mc}	F_{ec}	F_{mt}	F_{et}
CCR	70%	100%	100%	100%
Precision	60%	100%	100%	100%
TPR	67%	100%	100%	100%
FPR	25%	0%	0%	0%
TNR	75%	100%	100%	100%
FNR	33%	0%	0%	0%

Table 14
The fault detection accuracy of the ensemble-based II solution (fault severity = 2%).

	F_{mc}	F_{ec}	F_{mt}	F_{et}
CCR	80%	100%	100%	100%
Precision	80%	100%	100%	100%
TPR	80%	100%	100%	100%
FPR	20%	0%	0%	0%
TNR	80%	100%	100%	100%
FNR	20%	0%	0%	0%

Table 15
The fault detection accuracy of the single model-based RBF-NARX solution (fault severity = 8%).

	F_{mc}	F_{ec}	F_{mt}	F_{et}
CCR	90%	100%	100%	100%
Precision	100%	100%	100%	100%
TPR	100%	100%	100%	100%
FPR	20%	0%	0%	0%
TNR	80%	100%	100%	100%
FNR	0%	0%	0%	0%

and the engine is also actually faulty; f.p. (false positive) denotes the number of cases that are classified as faulty, but the engine is healthy; t.n. (true negative) denotes the number of cases that are classified as healthy and the engine is also healthy; and f.n. (false negative) denotes the number of cases that are classified as healthy but the engine is actually faulty. Tables 11 through 16 provide a complete comparative analysis between the ensemble-based II fault detection methodology and that of the single-model based RBF-NARX fault detection scheme.

8.7. Single model-based fault isolation results

The residuals that are generated corresponding to the single-model approach RBF-NARX are utilized for achieving the fault isolation task. The fault scenarios are described in the previous subsections and the data samples associated with these scenarios are collected (a total of 100 scenarios are considered for sake of illustration). Among the collected data we randomly select 50 samples (after experimenting with different training data sizes, namely, 30, 40, 50, and 60 samples). Starting from a small structure we construct a neural network classifier to perform the fault isolation task. We observe that an acceptable performance is achieved with a single layer MLP having 15 hidden neurons. Table 17 provides the corresponding confusion matrix for the testing results of the available data. The obtained performance results in terms of the CCR metric are as follows: $CCR_{training} = 88\%$, $CCR_{testing} = 80\%$ and $CCR_{total} = 84\%$.

Table 16
The fault detection accuracy of the ensemble-based II solution (fault severity = 8%).

	F_{mc}	F_{ec}	F_{mt}	F_{et}
CCR	100%	100%	100%	100%
Precision	100%	100%	100%	100%
TPR	100%	100%	100%	100%
FPR	0%	0%	0%	0%
TNR	100%	100%	100%	100%
FNR	0%	0%	0%	0%

8.8. Ensemble-based fault isolation results

The residuals that are generated corresponding to the ensemble-based II approach are utilized for achieving the fault isolation task. The fault scenarios having a sample size of 50 data as well as the MLP classifier structure are the same as in the previous subsection. Table 18 provides the corresponding confusion matrix for testing results of the available data. The performance results in terms of the CCR metric are as follows: $CCR_{training} = 98\%$, $CCR_{testing} = 100\%$ and $CCR_{total} = 98.75\%$.

8.9. Single model-based multiple faults isolation results

The residuals that are generated corresponding to the single-model RBF-NARX approach are utilized for achieving fault isolation of multiple concurrent faults. The fault scenarios having a sample size of 50 data as well as the MLP classifier structure are the same as in the previous subsection. Table 19 provides the confusion matrix for testing of the available data. The performance results in terms of the CCR metric are as follows: $CCR_{training} = 98.33\%$, $CCR_{testing} = 87.5\%$ and $CCR_{total} = 94\%$.

8.10. Ensemble-based multiple faults isolation results

The residuals that are generated corresponding to the ensemble-based II approach are utilized for achieving fault isolation of multiple concurrent faults. The fault scenarios having a sample size of 50 data as well as the MLP classifier structure are the same as in the previous subsections. Table 20 shows the confusion matrix for testing of all the available data. The performance results in terms of the CCR metric are as follows: $CCR_{training} = 99.17\%$, $CCR_{testing} = 93.75\%$ and $CCR_{total} = 97\%$.

9. Conclusion

In this paper, a new methodology and framework for fault detection and isolation (FDI) of a gas turbine engine is proposed by using ensemble of neural networks. Proposed ensemble methods integrate various identification models to ensure reduction in the modeling error and increase in the prediction accuracy. By combining individual neural network models, enhanced robustness

Table 17
Testing data confusion matrix by using single model-based (RBF-NARX) fault isolation.

		Actual fault class							
		$F_{ec} > 3\%$	$F_{ec} < 3\%$	$F_{mc} > 3\%$	$F_{mc} < 3\%$	$F_{mt} > 3\%$	$F_{mt} < 3\%$	$F_{et} > 3\%$	$F_{et} < 3\%$
Predicted fault class	$F_{ec} > 3\%$	7	0	0	0	1	0	1	0
	$F_{ec} < 3\%$	0	3	0	0	0	0	0	0
	$F_{mc} > 3\%$	0	0	6	0	0	0	0	0
	$F_{mc} < 3\%$	0	0	0	5	0	0	0	0
	$F_{mt} > 3\%$	1	0	0	0	5	1	1	2
	$F_{mt} < 3\%$	0	0	0	0	0	4	0	0
	$F_{et} > 3\%$	0	0	0	0	0	0	5	0
	$F_{et} < 3\%$	0	0	0	0	0	0	3	5

Table 18
Confusion matrix for testing data using ensemble-based II fault isolation.

		Actual fault class							
		$F_{ec} > 3\%$	$F_{ec} < 3\%$	$F_{mc} > 3\%$	$F_{mc} < 3\%$	$F_{mt} > 3\%$	$F_{mt} < 3\%$	$F_{et} > 3\%$	$F_{et} < 3\%$
Predicted fault class	$F_{ec} > 3\%$	9	0	0	0	0	0	0	0
	$F_{ec} < 3\%$	0	6	0	0	0	0	0	0
	$F_{mc} > 3\%$	0	0	4	0	0	0	0	0
	$F_{mc} < 3\%$	0	0	0	5	0	0	0	0
	$F_{mt} > 3\%$	0	0	0	0	5	0	0	0
	$F_{mt} < 3\%$	0	0	0	0	0	4	0	0
	$F_{et} > 3\%$	0	0	0	0	0	0	4	0
	$F_{et} < 3\%$	0	0	0	0	0	0	0	3

Table 19
Testing data confusion matrix by using the single model-based (RBF-NARX) fault isolation.

Actual	Prediction									
	F_{ec}	F_{mc}	F_{mt}	F_{et}	F_{mc}, F_{ec}	F_{mc}, F_{mt}	F_{mc}, F_{et}	F_{ec}, F_{et}	F_{ec}, F_{mt}	F_{mt}, F_{et}
F_{ec}	4	0	0	0	0	0	0	0	0	0
F_{mc}	0	11	0	0	0	2	0	0	0	0
F_{mt}	0	0	8	0	0	0	0	0	0	0
F_{et}	0	0	0	8	0	0	0	0	0	0
F_{mc}, F_{ec}	0	0	0	0	6	0	0	0	0	0
F_{mc}, F_{mt}	0	0	0	0	0	5	0	0	0	0
F_{mc}, F_{et}	0	0	0	0	0	0	8	0	0	0
F_{ec}, F_{et}	8	0	0	0	0	0	0	7	0	0
F_{ec}, F_{mt}	0	0	0	0	0	0	0	0	5	0
F_{mt}, F_{et}	0	0	0	0	0	0	0	0	0	8

Table 20
Testing data confusion matrix by using the ensemble-based II multiple fault isolation.

Actual	Prediction									
	F_{ec}	F_{mc}	F_{mt}	F_{et}	F_{mc}, F_{ec}	F_{mc}, F_{mt}	F_{mc}, F_{et}	F_{ec}, F_{et}	F_{ec}, F_{mt}	F_{mt}, F_{et}
F_{ec}	9	0	0	0	0	0	0	0	0	0
F_{mc}	0	11	0	0	0	2	0	0	0	0
F_{mt}	0	0	8	0	0	0	0	0	0	0
F_{et}	0	0	0	8	0	0	0	0	0	0
F_{mc}, F_{ec}	0	0	0	0	6	0	0	0	0	0
F_{mc}, F_{mt}	0	0	0	0	0	5	0	0	0	0
F_{mc}, F_{et}	0	0	0	0	0	0	8	0	0	0
F_{ec}, F_{et}	3	0	0	0	0	0	0	7	0	0
F_{ec}, F_{mt}	0	0	0	0	0	0	0	0	5	0
F_{mt}, F_{et}	0	0	0	0	0	0	0	0	0	8

and accurate system representations are almost always achievable without the need of ad-hoc, labor intensive and time consuming fine tunings that are required for single neural network solutions. To accomplish engine health monitoring, its dynamics is first identified and represented by using three different stand-alone neural network learning algorithms. Specifically, a dynamic MLP, a dynamic RBF neural network, and a dynamic SVM are trained to individually identify the engine dynamics. Three ensemble-based schemes are then proposed and developed for representing the gas turbine engine dynamics. Namely, two heterogeneous ensemble models and one homogeneous ensemble model. It is first concluded that all the heterogeneous ensemble models improve the system identification modeling accuracy when compared to the stand-alone solutions. The best selected stand-alone model (i.e., the dynamic RBF neural network) and the best selected ensemble model (i.e., a heterogeneous ensemble) in term of the engine modeling accuracy are then selected to perform the FDI task and objective. The residuals are obtained by using both single and ensemble-based methodologies under various engine health conditions to detect component faults. Our simulation results demonstrate that residuals that are obtained from the ensemble approach result in more accurate fault detection capability and performance. The fault isolation task is then performed by evaluating variations in the ensemble residual signals (before and after a

fault detection flag is issued) by using an MLP neural network classifier. As in the fault detection results, it is concluded through extensive simulation studies that the ensemble-based fault isolation methodology results in a more promising, accurate, and reliable performance.

Acknowledgment

This publication was made possible by NPRP Grant No. 4 - 195 - 2 - 065 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Al-Zyoud, I. A.-D., & Khorasani, K. (2005). Detection of actuator faults using a dynamic neural network for the attitude control subsystem of a satellite. In *Proc. of int. joint conf. on neural networks*.

Amanda, J., & Sharkey, C. (2002). Types of multinet system. In *Multiple classifier systems and lecture notes in computer science* (pp. 108–117).

Anders, K., & Vedelsby, J. (1995). Neural network ensembles cross validation and active learning. In *Advances in neural information processing systems* (pp. 231–238).

Avnimelech, R., & Intrator, N. (1999). Boosting regression estimators. *Neural Computation*, *11*, 491–513.

Billings, S. A. (2013). *Nonlinear system identification: NARMAX methods in the time and frequency and and spatio-temporal domains*. Wiley.

Bishop, M. (2006). *Pattern recognition and machine learning*. Springer.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, *6*, 5–20.

Camporeale, S. M., Fortunato, B., & Mastrovito, M. (2006). A modular code for real time dynamic simulation of gas turbines in simulink. *Journal of Engineering for Gas Turbines and Power*, *128*(3), 506–517.

Chen, S., Billings, S. A., & Grant, P. M. (1990). Non-linear system identification using neural networks. *International Journal of Control*, *51*, 1191–1214.

Chen, S., Wang, X. X., & Harris, C. J. (2008). Narx-based nonlinear system identification using orthogonal least squares basis hunting. *IEEE Transactions on Control Systems Technology*, *16*, 78–84.

Domingos, P. (1997). Why does bagging work? A Bayesian account and its implications. In *International conference on knowledge discovery and data Mining* (pp. 155–158).

Donat, W., Choi, K., An, W., Singh, S., & Pattipati, K. (2007). Data visualization data reduction and classifier fusion for intelligent fault detection and diagnosis in gas turbine engines. In *Proceedings of the ASME turbo expo* (pp. 883–892).

Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., & Vapnik, V. (1994). Boosting and other ensemble methods. *Neural Computation*, *6*, 1289–1301.

Granitto, P. M., Verdes, P. F., & Ceccatto, H. A. (2005). Neural network ensembles: evaluation of aggregation algorithms. *Artificial Intelligence*, *163*, 139–162.

Guilherme, P., & Zuben, F. J. V. (2006). The influence of the pool of candidates on the performance of selection and combination techniques in ensembles. In *International joint conference on neural networks* (pp. 10588–10595).

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(10), 993–1001.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The elements of statistical learning*. Springer.

Huang, J., & Wang, M. (2010). Multiple classifiers combination model for fault diagnosis using within-class decision support. In *Proceedings of information science and management engineering. ISME* (pp. 226–229).

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, *3*, 79–87.

Kivinen, J., & Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, *132*, 1–63.

Kobayashi, T., & Simon, D. L. (2003). Application of a bank of kalman filters for aircraft engine fault diagnostics. In *Proceedings ASME turboexpo* (pp. 461–470).

Kotsiantis, S. B., & Pintelas, P. E. (2005). Selective averaging of regression models. *Annals of Mathematics and Computing & Teleinformatics*, *1*, 65–74.

Lei, Y., Zuo, M. J., He, Z., & Zi, Y. (2010). A multidimensional hybrid intelligent method for gear fault diagnosis. *Expert Systems with Applications*, *37*, 1419–1430.

Loboda, I., Feldshteyn, Y., & Ponomaryov, V. (2011). Neural networks for gas turbine fault identification: Multilayer perceptron or radial basis network? In *Proceedings of the ASME turbo expo* (pp. 465–475).

Lu, F., Zhu, T. B., & Lv, Y. Q. (2012). Data-driven based gas path fault diagnosis for turbo-shaft engine. *Applied Mechanics and Materials*, *249*, 400–404.

Mendes-Moreira, J., Soares, C., Jorge, A. M., & Sousa, J. F. D. (2012). Ensemble approaches for regression: A survey. *ACM Computing Surveys (CSUR)*, *45*, 101–114.

Merz, C. J., & Pazzani, M. J. (1999). A principal components approach to combining regression estimates. *Machine Learning*, *36*, 9–32.

Mohammadi, R., Naderi, E., Khorasani, K., & Hashtrudi-Zad, S. (2010). Fault diagnosis of gas turbine engines by using dynamic neural networks. In *Proc. ASME turbo expo*.

Naderi, E., Meskin, N., & Khorasani, K. (2012). Nonlinear fault diagnosis of jet engines by using a multiple model-based approach. *Transactions of the ASME Engineering for Gas Turbines and Power*, *134*, 319–329.

Opitz, D., & Macli, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, *11*, 169–198.

Oza, N. C., Tumer, K., Tumer, I. Y., & Huff, E. M. (2003). Classification of aircraft maneuvers for fault detection. In *Multiple classifier systems and lecture notes in computer science* (pp. 375–384).

Panov, V. (2009). Gasturbolib: Simulink library for gas turbine engine modelling. In *Proceedings of ASME turbo expo 2009. Vol. 1*.

Perrone, M. P., & Cooper, L. N. (1994). When networks disagree: Ensemble methods for hybrid neural networks. *Neural networks for speech and image processing*, 126–142.

Polikar, R. (2012). Ensemble learning. In *Ensemble machine learning* (pp. 1–34).

Ren, C., Yan, J. F., & Li, Z. H. (2009). Improved ensemble learning in fault diagnosis system. In *IEEE international conference on machine learning and cybernetics, Vol. 1* (pp. 54–60).

Roli, F., Giacinto, G., & Vernazza, G. (2001). Methods for designing multiple classifier systems. *Multiple Classifier Systems*, *2096*, 78–87.

Salat, R., Awtoniuk, M., & Korpysz, K. (2013). Black-box system identification by means of support vector regression and imperialist competitive algorithm. *Przeglad Elektrotechniczny*, *9*, 223–226.

Sharkey, A. J. C., Chandroth, G. O., & Sharkey, N. E. (2000). A multi-net system for the fault diagnosis of a diesel engine. *Neural Computing and Applications*, *9*, 152–160.

Sharkey, A. J. C., & Sharkey, N. E. (1997). Combining diverse neural nets. *Knowledge Engineering Review*, *12*, 1–17.

Tayarani-Bathaie, S. S., Vanini, Z. S., & Khorasani, K. (2014). Dynamic neural network-based fault diagnosis of gas turbine engines. *Neurocomputing*, *125*, 153–165.

Valdes, A., Khorasani, K., & Ma, L. (2009). Dynamic neural network-based fault detection and isolation for thrusters in formation flying of satellites. In *Advances in neural networks—ISNN 2009: 6th international symposium on neural networks*.

Vanini, Z. N. S., Khorasani, K., & Meskin, N. (2014). Fault detection and isolation of a dual spool gas turbine engine using dynamic neural networks and multiple model approach. *Information Sciences*, *259*, 234–251.

Varma, A., Bonissone, P., Yan, W., Eklund, N., Goebel, K., Iyer, N., & Bonissone, S. (2007). Anomaly detection using non-parametric information. In *Proceedings of the ASME turbo expo* (pp. 813–821).

Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, *27*(3), 293–311.

Webb, G. I., & Zheng, Z. (2004). Multistrategy ensemble learning: reducing error by combining ensemble learning techniques. *IEEE Transactions on Knowledge and Data Engineering*, *16*, 980–991.

Xiao, H., Eklund, N., & Goebel, K. (2007). A data fusion approach for aircraft engine fault diagnostics. In *Proceedings of the ASME turbo expo* (pp. 767–775).

Xiao, H., Eklund, N., Goebel, K., & Cheetham, W. (2007). Hybrid change detection for aircraft engine fault diagnostics. In *Proceedings of IEEE aerospace conference* (pp. 1–10).

Yan, W., & Xue, F. (2008). Jet engine gas path fault diagnosis using dynamic fusion of multiple classifiers. In *Proceedings of IEEE world congress on computational intelligence* (pp. 1585–1591).

Yates, W., & Partridge, D. (1996). Use of methodological diversity to improve neural network generalization. *Neural Computing and Applications*, *4*, 114–128.

Yazdizadeh, A., & Khorasani, K. (2002). Adaptive time delay neural network structures for nonlinear system identification. *Neurocomputing*, *47*(4), 207–240.

Zhang, J. (2005). Improved on-line process fault diagnosis through information fusion in multiple neural networks. *Computers & Chemical Engineering*, *30*, 558–571.

Zhang, C., & Ma, Y. (2012). *Ensemble machine learning: methods and applications*. Springer.