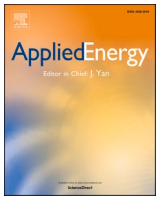
Applied Energy 332 (2023) 120520

Contents lists available at ScienceDirect

Applied Energy

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



A hierarchical structure built on physical and data-based information for intelligent aero-engine gas path diagnostics

Junjie Zhao [a,*], Yi-Guang Li [b], Suresh Sampath [c]

[a] School of Aerospace, Transport and Manufacturing, Cranfield University, Bedfordshire MK43 0AL, UK
[b] Centre for Propulsion Engineering, School of Aerospace, Transport and Manufacturing, Cranfield University, Bedfordshire MK43 0AL, UK
[c] Centre for Propulsion Engineering, School of Aerospace, Transport and Manufacturing, Cranfield University, Bedfordshire MK43 0AL, UK

HIGHLIGHTS

- An Machine Learning method for intelligent aero-engine diagnostics is proposed.
- A hierarchical structure for low complexity and high accuracy is built.
- The proposed method effectively exploits physical and data-based information.
- The error accumulation problem in hierarchical structures is solved.
- NASA ProDiMES software provides a standard benchmark for evaluation.

ARTICLE INFO

Keywords:
Aero engines
Intelligent condition monitoring
Gas path diagnostics
Machine Learning
Hierarchical diagnostic method

ABSTRACT

Future trends in engine health management (EHM) systems are information fusion, advanced analytical methods, and the concept of the Intelligent Engines. Machine Learning (ML)-based aero-engine gas path diagnostic methods are promising under the motivation of these trends. However, previous ML-based diagnostic structures are rarely applied in actual engineering practice because they are purely mathematical and lack physical insight or are limited by the error accumulation problem. Developing an accurate, flexible and interpretable intelligent diagnostic method has always posed a challenge, especially when physical knowledge is also available for more diagnostic information. Instead of modifying and applying existing ML methods for classification or regression, this study proposes a novel hierarchical diagnostic method to get insight into the physical systems, build hierarchies automatically, and recommend the classification structures. The proposed hierarchical diagnostic method is evaluated against a NASA model high-bypass two-spool turbofan engine. NASA's blind test case results show that Kappa Coefficient of the proposed hierarchical diagnostic method is 0.693 and is at least 0.008 higher than the other diagnostic methods in the open literature. It has been proved that the proposed method can quantify the dependence relationships between the fault classes for enhanced diagnostic information, recommend the best diagnostic structure for reduced complexity, and solve the error accumulation problem for improved diagnostic accuracy. The proposed method could support intelligent condition monitoring systems by effectively exploiting physical and data-based information for improved model interpretability, model flexibility, diagnostic visibility, diagnostic accuracy, and diagnostic reliability.

Abbreviations: *AANN*, Auto-Associative Neural Network; *AI*, Artificial Intelligence; *ANN*, Artificial Neural Network; *CCR*, Correct Classification Rate; *C-MAPSS*, Commercial Modular Aero-Propulsion System Simulation; *CNN*, Convolutional Neural Network; *CUSUM*, Cumulative Sum; *DBN*, Deep Belief Network; *DNN*, Dynamic Neural Network; *DLN*, Deep Learning Neural Network; *DOD*, Domestic Object Damage; *DT*, Decision Tree; *EFS*, Engine Fleet Simulator; *EHM*, Engine Health Management; ELM, Extreme Learning Machine; *EMA*, Exponential Moving Average; *FDI*, Fault Detection and Isolation; *FFBP*, Feed-forward Back-propagation; *FOD*, Foreign Object Damage; *FPR*, False Positive Rate; *HAC*, Hierarchical Agglomerative Clustering; *HPC/ LPC*, High/ Low Pressure Compressor; *HPT/ LPT*, High/ Low Pressure Turbine; *KNN*, K-nearest Neighbours; *LCPN*, Local Classifier per Parent Node; *LR*, Logistic Regression; *MCR*, Misclassification Rate; *ML*, Machine Learning; *MLP*, Multilayer Perceptron; *NB*, Naïve Bayes; *NN*, Neural Network; *PLA*, Power Lever Angle; *PNN*, Probabilistic Neural Network; *ProDiMES*, Propulsion Diagnostic Method Evaluation Strategy; *RNN*, Recurrent Neural Network; *SOM*, Self-Organizing Map; *SVM*, Support Vector Machine; *TDNN*, Time Delayed Neural Network; *TPR*, True Positive Rate; *VBV*, Variable Bleed Valve; *VSV*, Variable Stator Vane.

* Corresponding author.
E-mail address: junjie.zhao@cranfield.ac.uk (J. Zhao).

https://doi.org/10.1016/j.apenergy.2022.120520
Received 23 May 2022; Received in revised form 29 November 2022; Accepted 11 December 2022
Available online 23 December 2022
0306-2619/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Gas turbine performance deterioration may be due to gradual degradations and fault events of an engine [1]. Examples of gradual performance degradations are fouling, erosion, corrosion, etc. [2], and they evolve on a slow timescale. Faults typically refer to sudden damage to the engine, such as Foreign Object Damage (FOD), Domestic Object Damage (DOD) [3], etc. In general, gas turbine gas path faults may be classified as component (for example, fan, compressor, turbine, etc.) faults and sub-system (for example, sensor and actuator) faults [4,5]. Based on the evolution rate, faults may also be classified as abrupt faults that appear instantaneously but do not grow in magnitude over time and rapid faults that initiate and grow in magnitude over a short period of time [4].

Gas turbine diagnostics may be carried out by model-based [6,7] and data-driven methods [8,9]. Typical examples of model-based gas path diagnostic methods are gas path analysis [10,11], kalman filter [12], genetic algorithm [13], and weighted least squares [14], etc. Data-driven methods are based solely on the analysis of measured data. They mainly refer to Artificial Intelligence (AI) methods (also often called soft computing methods [15]) and statistical methods (such as Cumulative Sum (CUSUM) algorithm [16]).

Machine Learning (ML) is a subset of AI [17]. Typical examples of ML-based methods are Artificial Neural Networks (ANNs) [18], K-nearest Neighbours (KNN) [19], Support Vector Machine (SVM) [20], Naïve Bayes (NB) [21], and Ensemble Methods (for example, Random Forest [22]). Some comparisons have been conducted for ML applications in gas turbine diagnostics, such as in the work of Jaw et al. [21] and Loboda et al. [23]. Typical ANNs are Multilayer Perceptrons (MLPs) [24], Probabilistic Neural Networks (PNNs) [25], Auto-Associative Neural Networks (AANNs) [26], Self-Organizing Map (SOM) [27], and some Deep Learning Neural Networks (DLNs) (for example, Deep Belief Networks (DBNs) [28], Autoencoders [29], Convolution Neural Networks (CNNs) [30,31] and Recurrent Neural Networks (RNNs) [32]).

In ML gas turbine fault detection and isolation (FDI), diagnostic structures can be classified into flat and nested structures. A summary of previous research work on ML diagnostic structures is presented in Table 1. It is clear that the majority of the research studies have focused on flat structures, with only a few studies dealing with nested structures. Details of these structures' advantages and disadvantages are described in the following two paragraphs.

Flat structures are commonly used where each instance is classified into one of the possible fault classes, and there is no hierarchical relationship between the classes. Existing ML algorithms for classification and regression could be modified and applied to gas path diagnostics. However, there are some drawbacks to the flat classification structure. Firstly, it lacks understanding of a system as the same classification structure may be applied to different subjects. Secondly, the whole classification system has to be retrained if there are any changes to the physical system due to, for example, maintenance. Thirdly, misclassification may occur if the number of fault classes is large.

In applying nested diagnostic structures to gas turbine diagnostics, Zedda and Singh [33] proposed a modular-based diagnostic structure for a twin-spool turbofan gas turbine engine. Ogaji [34] proposed nested NNs for gas turbine component and sensor fault diagnostics [35]. Courdier and Li [36] suggested a sensor fault diagnostic method using nested NNs. Fentaye et al. [37] introduced SVMs to replace NNs in a nested structure for gas path diagnostics. Koskoletos [16] used a PNN to classify components, sensor and actuator faults. These ANN structures can be treated as special types of hierarchical structures where the pre-established class hierarchy is built using a priori knowledge of the systems in concern, i.e., using the facts that we know about the fault classes. These nested structures are purely built on physical knowledge but have little diagnostic performance concerns. Compared with the flat classification structure, these structures provide a better understanding of the relevant systems with lower complexity. However, these structures may lead to lower accuracy due to the accumulation of errors at multiple diagnostic levels. In other words, the errors at upper levels accumulate and propagate downwards to influence the performance of lower levels.

To summarise, ML-based diagnostic structures are either so-called black-box flat structures that lack physical insight or nested structures built on priori knowledge that may lead to lower accuracy due to the accumulation of errors. There is a recognised need to develop an ML diagnostic method that can build an accurate, flexible and interpretable diagnostic structure by exploiting physical and data-based information. Hence, this study proposes a novel hierarchical diagnostic method that can hierarchically decompose a complex fault FDI problem and determine a proper trade-off between complexity and diagnostic accuracy. This study aims to contribute to this growing research area of ML diagnostics, and the main contributions are listed as follows:

a) The proposed hierarchical method is a divide-and-conquer approach that can break a complicated problem into several simple sub-problems, which reduces the computation required by the system, and facilitates fine-tuning and control. This contribution corresponds to the drawback of a flat structure that it is a complicated black-box and misclassification may occur if the number of fault classes is large.
b) The similarity analysis provides physical insight into gas path faults, and the hierarchy provides graphical representations of dependence relationships between the fault classes. They quantify the dependence relationships between the fault classes for enhanced diagnostic information, allowing for efficiencies in both learning and representation. This contribution corresponds to the drawback of a flat structure that it lacks understanding of a system.
c) The proposed diagnostic method can recommend the hierarchical structure automatically by determining a proper trade-off between complexity and diagnostic accuracy. This contribution corresponds to the drawback of a nested structure that it is built on priori knowledge for lower complexity but it may lead to lower accuracy.
d) The error accumulation problem is solved with the proposed distance threshold method, which ensures the hierarchical structure's accuracy. This contribution corresponds to the drawback of a nested structure that it may lead to lower accuracy due to the accumulation of errors at multiple diagnostic levels.

The remaining part of the paper is organised as follows. Section 2 presents the methodology of the proposed method. Section 3 evaluates the proposed method with NASA's Propulsion Diagnostic Method Evaluation Strategy (ProDiMES) software. The results, discussions and conclusions are provided accordingly.

2. Methodology

A generic hierarchical fault diagnostic structure for gas turbine FDI may be organised in a tree-like structure, as shown in Fig. 1, where two fault class levels are included and illustrated. The tree-like structure consists of nodes that include non-leaf or parent nodes and leaf nodes. The first node of the tree structure is called the root and is an example of a non-leaf or parent node. A parent node is connected to child nodes by links called edges. The nodes are labelled to represent the fault classes. Such a hierarchical diagnostic system can break a complicated diagnostic problem into several relatively simple sub-problems, and each sub-problem can be resolved by a relatively simple sub-classifier at different fault class levels.

Fig. 2 shows a setup process of the proposed hierarchical fault diagnostic method for aero gas turbine engines. Firstly, the physical information is quantified, and the similarities between different fault classes are analysed. Secondly, a hierarchy is built based on the similarities. Thirdly, based on the generated hierarchy, a hierarchical diagnostic structure is set up. Finally, local classifiers in the proposed classification structure are set up. The details for each process are described as follows.

Table 1
Overview of previous studies on ML gas turbine gas path diagnostic structures.

Diagnostic algorithm	ML structures	Reference	Remarks
Decision Tree (DT)	Flat	Batayev et al. (2019) [38]	A DT used for overall fault detection
Logistic Regression (LR)	Flat	Allen et al. (2017) [39]	An LR used for fuel valve binary classification
KNN	Flat	Wang et al. (2014) [40]	A KNN used for multiple fault isolation
SVM	Flat	Zhao et al. (2020) [41]	An SVM used for multiple component fault isolation
NB	Flat	Rabenoro et al. (2014) [42]	An NB used for binary fault detection
Extreme Learning Machine (ELM)	Flat	Lu et al. (2020) [43]	An ELM used for multiple component fault isolation
RF	Flat	Maragoudakis et al. (2008) [22]	An RF used for multiple blading fault isolation
MLP	Flat	Matuck et al. (2009) [44]	An MLP used for multiple component fault isolation
PNN	Flat	Loboda et al. (2014) [45]	A PNN used for multiple component fault isolation
AANN	Flat	Muthuraman et al. (2012) [46]	An AANN used for combustor-related fault detection
Dynamic Neural Network (DNN)	Flat	Sina et al. (2014) [47]	A DNN used for binary fault detection or isolation
Time Delayed Neural Network (TDNN)	Flat	Sina et al. (2015) [48]	A TDNN used for binary fault detection or isolation
DBN	Flat	Yan et al. (2020) [49]	A DNN used for component fault detection
Autoencoder	Flat	Fu et al. (2018) [50]	An autoencoder used for fault detection
CNN	Flat	Zhao et al. (2020) [31]	A CNN used for multiple component fault isolation
RNN	Flat	Cao et al. (2021) [51]	An RNN used for gas path degradation prediction
MLP	Nested	Zedda and Singh (1998) [33]	A nested structure with MLPs for component and sensor fault diagnostics
Optional NN	Nested	Ogaji (2003) [34]	A nested structure with optional local algorithms for component and sensor fault diagnostics
MLP	Nested	Courdier and Li (2016) [36]	A nested structure with MLPs for power setting sensor fault diagnostics
SVM	Nested	Fentaye et al. (2018) [37]	A nested structure with SVMs for component and sensor fault diagnostics
PNN	Nested	Koskoletos (2018) [16]	A nested structure with PNNs for component, sensor and actuator fault diagnostics

2.1. Quantify physical information and analyse similarities

The physical diagnostic information is obtained with a model-based method, a weighted least squares-based diagnostic method. Then, the obtained physical diagnostic information is analysed and quantified with a similarity analysis method. The similarities between fault classes (i.e., class relationships) should reflect the intrinsic characteristics of the classes and provide dependence relationships.

The weighted least squares-based diagnostic method is shown in Fig. 3. More details of the model-based method can be found as described in reference [4]. The fault prediction results of the weighted least squares-based diagnostic method with the test samples labelled with fault classes form the required data set for similarity analysis.

The prediction results from the weighted least squares-based diagnostic method are presented in a confusion matrix in Fig. 4 (a), a specific table that allows visualisation of the classification performance. As shown in Fig. 4 (a), the fault prediction results for the test samples are presented in an $n \times n$ confusion matrix M where the rows represent the true fault classes and the columns represent the predicted fault classes. The elements of M placed at the main diagonal $(M_{ij}, i = j)$ are the number of test samples that the weighted least squares-based method correctly classifies whereas the rest of elements of M at $(M_{ij},\ i \neq j)$ indicate incorrect classifications, i.e., the number of test samples of fault class i are wrongly classified as fault class j. The confusion matrix provides an insight into the similarity between the fault classes in "confusion space". For example, any two classes are similar if the test samples of the two classes are easily misclassified between each other and vice versa.

The similarities between fault classes are measured by their Euclidean Distance. In the confusion matrix M, the n true fault classes can be treated as n dimensions in n-dimensional Euclidean space, and each line segment of the predicted results can be treated as a vector. The Euclidean Distance between two line segments, i.e. between two fault classes, is used to measure the similarity between the two fault classes. The Euclidean Distance for any two fault classes can be obtained by applying the following steps:

a) The normalisation of the Confusion Matrix is carried out by using Eq. (1).

$$\overline{M_{ij}} = \frac{M_{ij}}{\sum_{j=1}^{n} M_{ij}} \tag{1}$$

Where, $\sum_{j=1}^{n} M_{ij}$ is the sum of the elements in row i of the confusion matrix M, which is the total number of the test samples for the fault class i. The normalised confusion matrix $\overline{M}$ can be obtained by dividing each element by the sum of its corresponding row of the confusion matrix M, as shown in Fig. 4 (b).

b) The similarity between any two fault classes represented by the Euclidean Distance between the two fault classes p and q is calculated by Eq. (2).

$$d_C(p,q) = \sqrt{\sum_{j=1}^{n} \left(\overline{M_{pj}} - \overline{M_{qj}}\right)^2} \tag{2}$$

Where p and q are the row counts for the fault classes p and q of the normalised confusion matrix $\overline{M}$. After applying d_C, a symmetric matrix $\overline{M_C}$ called similarity matrix is obtained, as shown in Fig. 4 (c). In the similarity matrix, the bigger the Euclidean Distance between the two fault classes, the smaller the similarity between the two fault classes will be. For example, $d_C(p,q) = 0$ indicates that the fault classes p and q are completely overlapping.

2.2. Build hierarchy

Once the similarity matrix is obtained, the class hierarchy can be built by applying a Hierarchical Agglomerative Clustering (HAC) algorithm [52]. A schematic of the hierarchy building up process is shown in Fig. 4 (d), where a Cluster u is assumed to be a fault class or a group of fault classes, and a Forest w is formed with all the fault clusters. The HAC algorithm works in a bottom-up manner, assigning each fault class to a cluster and then iteratively merging pairs of clusters until only one cluster is left. Clusters are merged based on the distance between them, referred to as the linkage distance. Because there may be more than one fault class in a cluster, a linkage criterion, which is a function of the pairwise distances between two clusters, is applied to determine the linkage distance between the two clusters. There are some commonly used linkage criteria available [52], and the one used in this study is the Complete Distance (i.e. the maximum distance between the elements of clusters). The hierarchical clustering may be graphically represented as a binary tree called a dendrogram, as shown in Fig. 5. It shows how the clusters are grouped based on the distance by which the grouping has taken place. Different hierarchies could be obtained depending on the

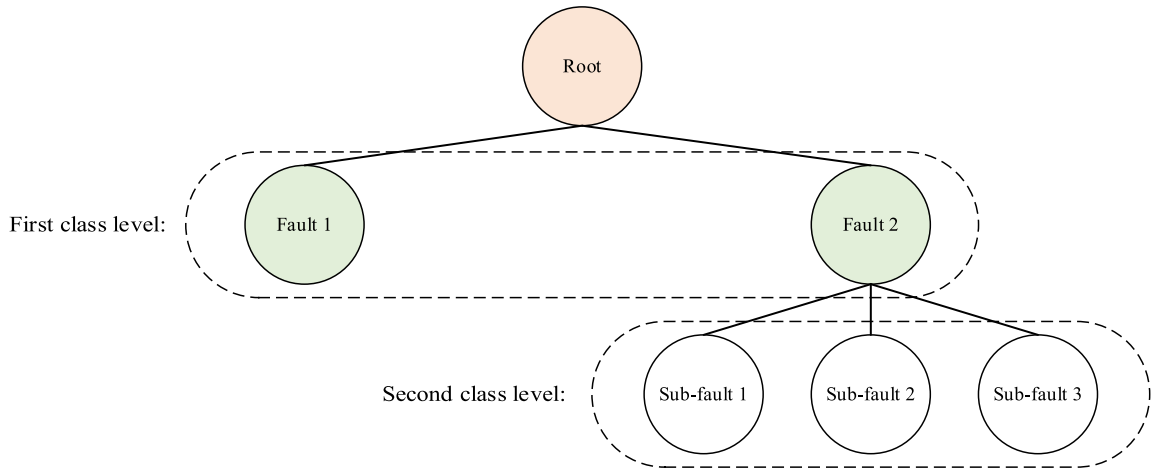


Fig. 1. An example of a tree-based hierarchical diagnostic structure.

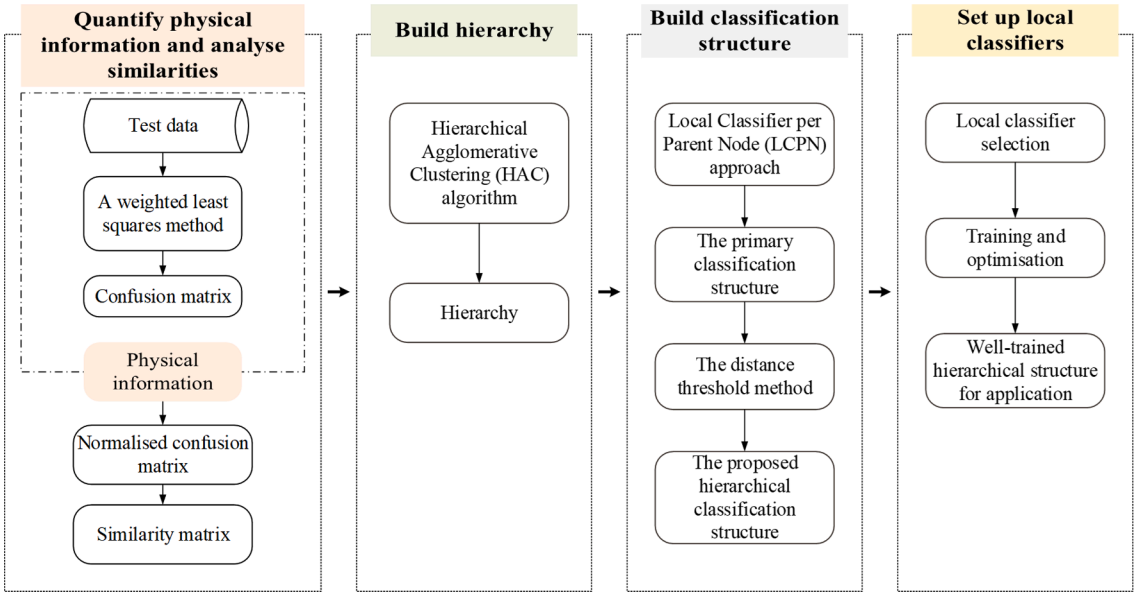


Fig. 2. The setup procedure of the hierarchical diagnostic system.

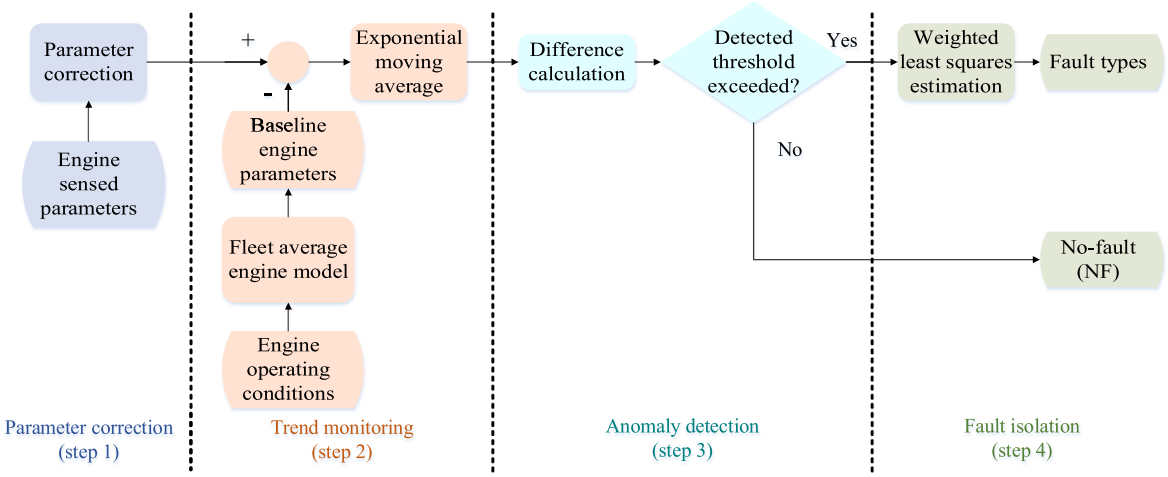


Fig. 3. The weighted least squares-based diagnostic method (adapted from reference [4]).

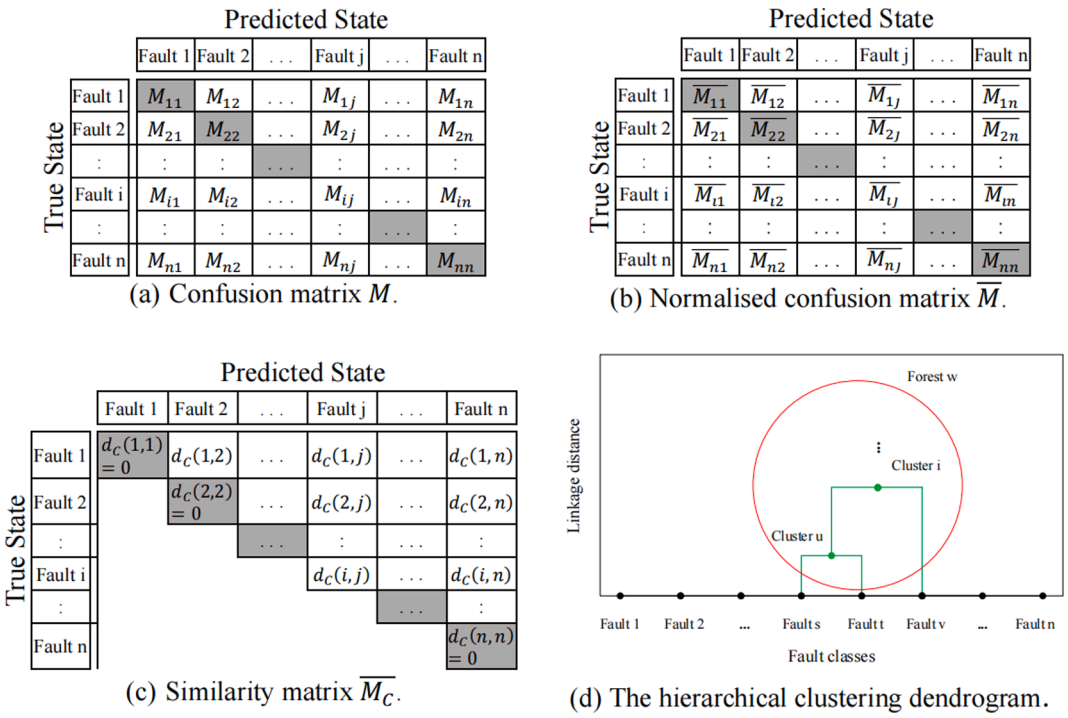


Fig. 4. Processes of building hierarchy from the confusion matrix.

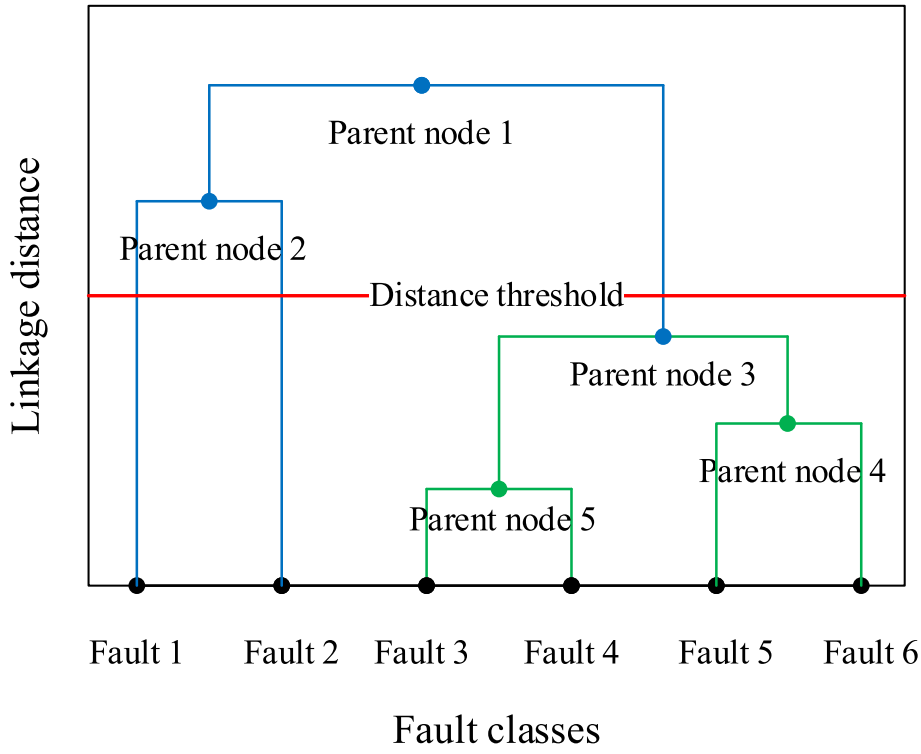
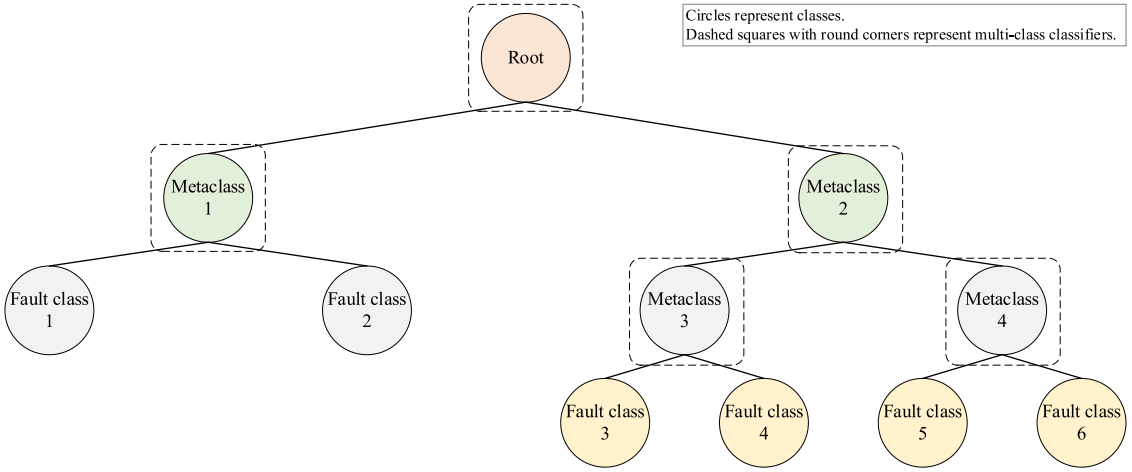


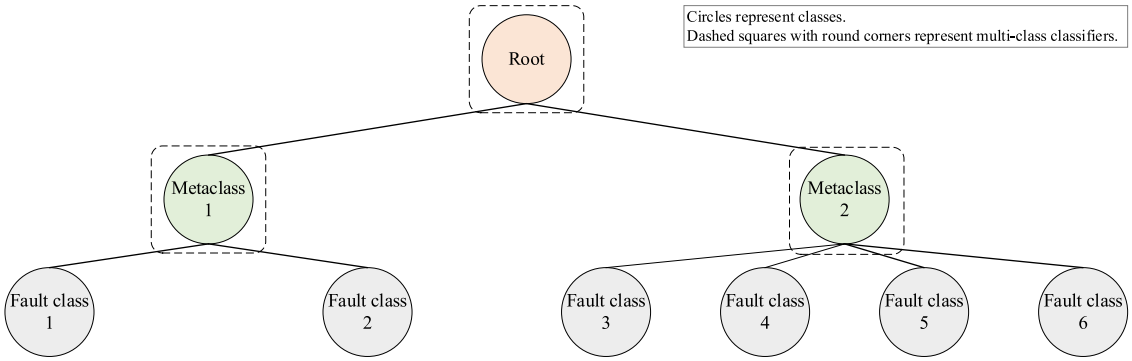
Fig. 5. An example of a class hierarchy represented as a dendrogram.

linkage distance criteria used.

In more detail, let n fault classes be assigned to n single clusters, called original clusters. Euclidean Distance $d_C(i,j)$ between any two fault classes i and j can be obtained from the similarity matrix as shown in Fig. 4 (c). $d[i,j]$ corresponds to the linkage distance between cluster i and j, for the original clusters, $d[i,j]=d_C(i,j)$. Comparing all the $c_n^2 = n(n-1)/2$ linkage distances between the original clusters, the two clusters with the shortest linkage distance (for example, s and t) are merged. In the HAC method, the two clusters s and t are combined into a single cluster named u, then, s and t are removed from the forest, u is added to the forest, as is shown in Fig. 4 (d). After the first iteration, there are two original clusters, i. e., s and t in cluster u and $n-2$ remaining original clusters. Correspondingly, the distance matrix is updated to reflect the changes of the clusters in the forest and the distance of the newly formed



(a) A schematic of the classification structure based on the original hierarchy.



(b) A schematic of the classification structure after introducing the distance threshold.

Fig. 6. Schematics of the classification structures.

cluster u from the remaining clusters in the forest.

The above process will continue iteratively until only one cluster, the root cluster, remains in the forest, indicating that the hierarchy building process is completed. An example of the hierarchy graphically represented as a dendrogram is shown in Fig. 5, where there are five parent nodes and six leaf nodes.

2.3. Build classification system

Once the hierarchy is built, the fault classification system consisting of a tree of classifiers can be developed following the hierarchy and the Local Classifier per Parent Node (LCPN) approach [53]. An example of a fault classification system built on the hierarchy is demonstrated in Fig. 6 (a), where five local classifiers are obtained. The LCPN approach trains a multi-class classifier for every non-leaf node (or parent node), including the root, to distinguish between its child nodes. As a result, a set of fault classifiers at the bottom of the tree is obtained. Then, to classify a new sample, the tree of classifiers is traversed in a top-down manner from the root until a leaf is reached.

The top-down prediction approach is used in the hierarchical diagnostic system, such as that shown in Fig. 6 (a). In other words, for each new sample in the test data set, the system firstly predicts its first level (most generic) class, then it uses the predicted class to narrow the choices of classes to be predicted at the second level, and then to the next level until the final prediction is made.

A disadvantage of the top-down approach is the error accumulation effect where an error at a certain class level will be propagated downwards to the next level through the hierarchy. Therefore, a distance threshold method is introduced to the class hierarchy to reduce the error accumulation. The idea of such a method is shown in Fig. 5 where a distance threshold is introduced in the hierarchical dendrogram. All links connecting nodes with distances greater than or equal to the linkage distance threshold are reserved; otherwise, the nodes are merged. In other words, parent node 1 and parent node 2 are reserved, whereas parent nodes 3, 4 and 5 are merged into one single parent node. As a result of using the threshold, the five local classifiers are merged into three local classifiers in the classification system, as shown in Fig. 6 (b).

Different distance thresholds may lead to different classification structures and result in different classification accuracies measured by the classification accuracy metric presented in Appendix A. To obtain the best distance threshold that corresponds to the best classification accuracy, a grid search optimisation algorithm [17] is applied where a search space may be defined as a grid of distance values, and every position in the grid is evaluated and the best one is selected.

2.4. Set up local classifiers

MLPs are chosen as the local classifiers throughout the classification system. MLPs are also often called feed-forward back-propagation

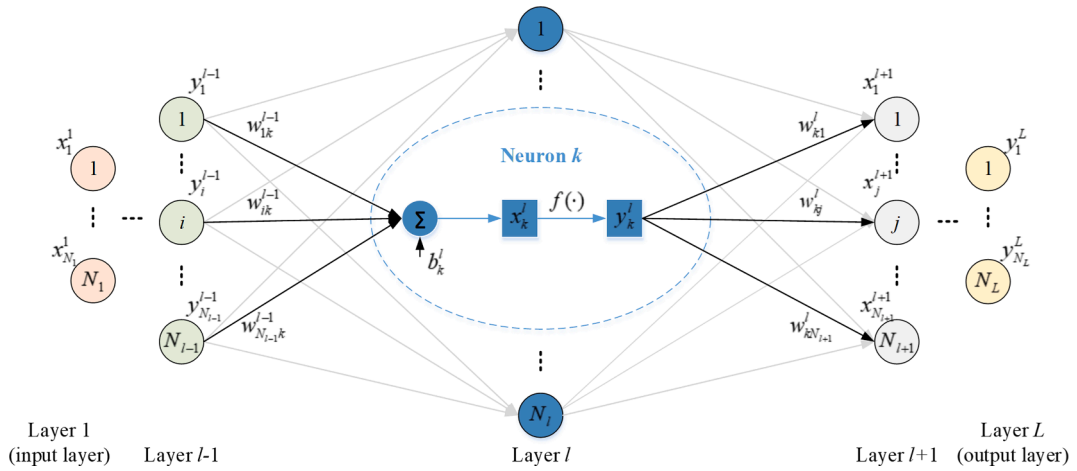


Fig. 7. Typical MLP hidden layers.

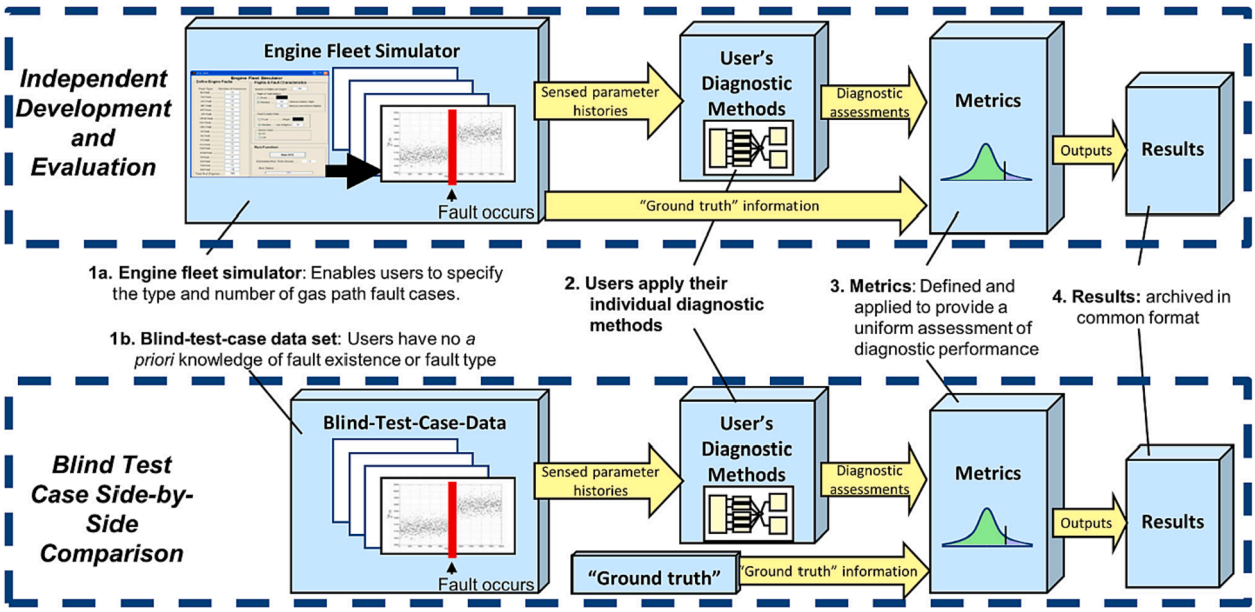


Fig. 8. ProDiMES public benchmarking process [4].

(FFBP) neural networks [54], most widely used in ML gas turbine diagnostics. An MLP typically consists of three principal layers: the input layer that receives measurements as input, hidden layers that process the information, and the output layer that provides the predicted fault classes. As is shown in Fig. 7, a typical hidden layer carries out multiplication operations and further proceed with nonlinear activation functions. The output y_k^l of the k^{th} neuron at layer l is mathematically represented by Eq. (3).

$$y_k^l = f\left(x_k^l\right) \text{ where } x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} w_{ik}^{l-1} y_i^{l-1} \tag{3}$$

Where $f(\,)$ is the nonlinear activation function, x_k^l is the input, b_k^l is the bias of the k^{th} neuron at layer l, and y_i^{l-1} is the output of the i^{th} neuron at layer $l-1$. w_{ik}^{l-1} is the kernel (i. e. weight) from the i^{th} neuron at layer $l-1$ to the k^{th} neuron at layer l. $l=1$ indicates the input layer and $l=L$ the output layer.

A large number of samples are required to train and test the MLP. The quality of the MLP is assessed by a mean absolute error (MAE) of the test samples represented by Eq. (4).

$$\mathrm{E} = \mathrm{E}\left(y_1^L, \cdots, y_{N_L}^L\right) = \frac{\sum_{i=1}^{N_L} \left|y_i^L - t_i\right|}{N_L} \tag{4}$$

where the output of the MLP corresponding to the test samples are the predicted fault class $[y_1^L, \cdots, y_{N_L}^L]$ and $[t_1, \cdots, t_{N_L}]$ are the true fault class.

An iterative training process of an MLP can be stated as follows:

a) Initialise weights and biases of the MLP network.
b) For each FFBP iteration, do as follows:
i. FP: Forward-propagate all training sample input from the input layer to the output layer to find outputs of each neuron at each layer.
ii. BP: Calculate the errors at the output layer and back-propagate them to the hidden layer to calculate the MAEs for the training samples.
iii. Update: Update the weights and biases to reduce the training sample errors.
iv. Test: Use the test samples to check the MAEs for the test samples.

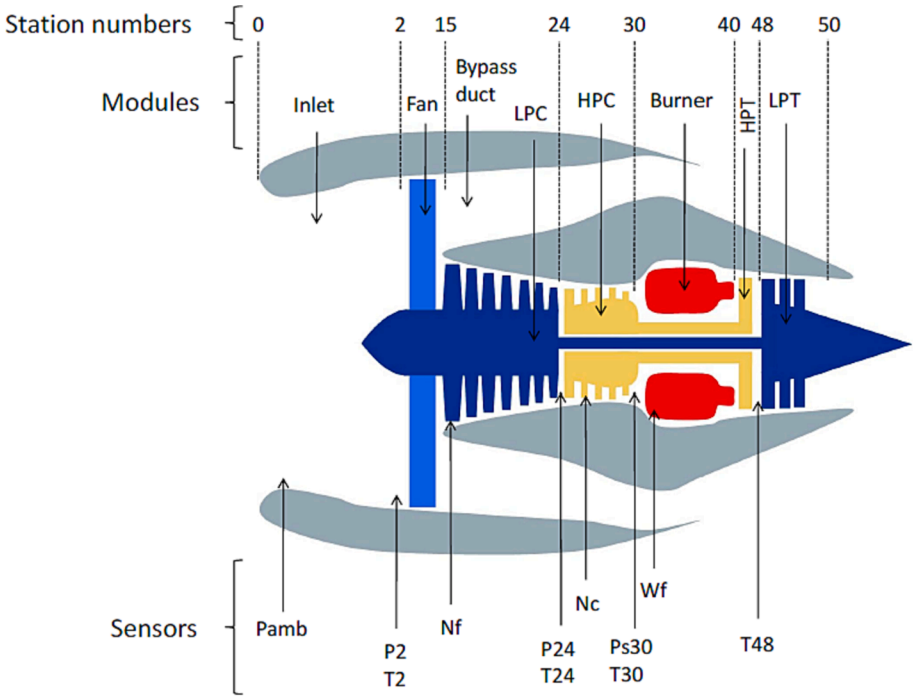


Fig. 9. C-MAPSS engine model station numbers, modules, and sensors [4].

Table 2
EFS fault types [4].

Fault ID	Fault description	Fault magnitude	γ:η	σ [%]	Takeoff average σ [actual units]	Cruise average σ [actual units]
0	No-fault	-	-	-	-	-
1	Fan fault	1 - 7%	2 - 1	-	-	-
2	LPC fault	1 - 7%	2 - 1	-	-	-
3	HPC fault	1 - 7%	2 - 1	-	-	-
4	HPT fault	1 - 7%	–0.5 - –1	-	-	-
5	LPT fault	1 - 7%	–0.5 - –1	-	-	-
6	VSV fault	1 - 7%	-	-	-	-
7	VBV fault	1 - 19%	-	-	-	-
8	Nf sensor fault	± 1 - 10σ	-	0.25	5.59 rpm	4.67 rpm
9	Nc sensor fault	± 1 - 10σ	-	0.17	15.04 rpm	13.22 rpm
10	P24 sensor Fault	± 1 - 10σ	-	0.5	0.134 psia	0.040 psia
11	Ps30 sensor Fault	± 1 - 10σ	-	0.2	0.896 psia	0.224 psia
12	T24 sensor fault	± 1 - 10σ	-	0.16	1.01 °R	0.816 °R
13	T30 sensor fault	± 1 - 10σ	-	0.16	2.45 °R	1.92 °R
14	T48 sensor fault	± 1 - 10σ	-	0.5	10.07 °R	7.48 °R
15	Wf/WF36 sensor fault	± 1 - 10σ	-	0.6	0.035 pps	0.0069 pps
16	P2 sensor fault	± 1 - 10σ	-	0.15	0.023 psia	0.0077 psia
17	T2 sensor fault	± 1 - 10σ	-	0.16	0.838 °R	0.707 °R
18	Pamb sensor fault	± 1 - 19σ	-	0.15	0.022 psia	0.0052 psia

v. Check: Compare the MAE against a pre-defined threshold. If the errors are smaller than the threshold, declare the completion of the training. Otherwise, repeat the training iterations.

Once the training process is completed, the well-trained classification system is ready to be used for diagnostics.

3. Application

3.1. Case study description

A case study was carried out to demonstrate the application of the introduced hierarchical fault classification method for gas turbine fault diagnostics. The ProDiMES software, which provides a standard benchmark enabling users to develop, evaluate, and compare diagnostic methods, is used in this study to generate required engine data. An introduction to ProDiMES is contained in reference [4], and the detailed instructions on its applications can be found in reference [25]. Based on the consideration that the case study is only used to evaluate the fault diagnostic performance of the proposed hierarchical fault classification method, data processing such as measurement correction, noise filtering, and gradual deterioration are simplified in this paper. The data are processed with the processing methods in ProDiMES [4] to ensure a fair comparison. Fig. 8 illustrates the benchmarking process of this approach.

3.2. Data generation and processing

3.2.1. Data generation

An Engine Fleet Simulator (EFS) in ProDiMES based on the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) for a model high-bypass two-spool turbofan engine working at steady state conditions is used to generate the data for this study. The engine model is shown in Fig. 9. The EFS produces simulated “snap-shot” engine

Table 3
Fault symbols and descriptions.

Symbol	Fault description	Fault type
NF	No-fault	-
Fan	Fan fault	Component
LPC	LPC fault	Component
HPC	HPC fault	Component
HPT	HPT fault	Component
LPT	LPT fault	Component
VSV	VSV fault	Actuator
VBV	VBV fault	Actuator
Nf	Physical fan speed fault	Power setting sensor
Nc	Physical core speed fault	Sensor
P24	Total pressure at LPC outlet fault	Sensor
Ps30	Total temperature at LPC outlet fault	Sensor
T24	Static pressure at HPC outlet fault	Sensor
T30	Total temperature at HPC outlet fault	Sensor
T48	Total temperature at HPT outlet fault	Sensor
Wf/WF36	Fuel flow rate fault	Sensor
P2	Total pressure at fan inlet fault	Ambient sensor
T2	Total temperature at fan inlet fault	Ambient sensor
Pamb	Ambient pressure fault	Ambient sensor

Table 4
Training data set.

Dataset description	Value
Number of engine health conditions	19
Number of flights	94,800
Fault initiation	Random
Minimum initiation flight	11
Fault evolution	Random
Rapid fault evolution rate (minimum)	9
Rapid fault evolution rate (maximum)	9
Sensor noise	On

measurements, with relevant measurement noise, as if collected from the engine over multiple flights.

Engine operating conditions, deterioration profiles, fault magnitudes, and sensor noise are randomly generated to emulate realistic engine behaviour. Firstly, a unique operating history is randomly assigned to the model engine. The city pairs that the engine to alternate takeoffs between, and the calendar date that data collection to commence are assigned randomly. Pressure altitude, Mach number, ambient temperature, and power setting parameters at each flight's takeoff and cruise operating points are randomly generated based on defined distributions representing commercial aircraft operations. At takeoff, the power setting parameter is Power Lever Angle (PLA), either 100 percent, or a fixed de-rated setting of 90 or 80 percent as commonly applied in commercial aircraft operations. At cruise, net thrust is indicated by the power setting parameter. Secondly, a unique deterioration profile to the engine is randomly assigned. Deterioration effects are simulated by adjustments to ten health parameters within the engine simulation to degrade over time. It is noted that, in the EFS case generator, a maximum of 5000 flight cycles can be defined for the engine. In this case, the engine's starting health condition will be randomly assigned somewhere along the deterioration profiles. Thirdly, the fault initiation, the fault magnitude, and the fault evolution rate for the engine are also defined. A summary of the fault types and their uniformly distributed fault magnitudes are provided in Table 2. The component faults (i.e. Fan, LPC, HPC, HPT and LPT faults with fault ID 1 through 5, respectively) are simulated by simultaneously adjusting the faulty health parameters corresponding to the efficiency, η and the flow capacity, γ. Component fault magnitude distributions described in Table 2 are based on the root-sum-square value of the combined η and γ deviations. $\gamma : \eta$ is the uniformly distributed ratio of flow capacity to efficiency adjustment. Actuator faults in the variable stator vanes, VSV, or the variable bleed valve, VBV (fault ID 6 and 7) will result in a mis-scheduling between the commanded and actual actuator positions. Sensor fault (fault ID 8 through 18) magnitudes are implemented in the standard deviation units, σ, of measurement noise at takeoff and cruise. The description of the symbols is given in Table 3.

Two sets of data were created using the EFS. The first one is the training data set with noisy measurements for the training and validation of the diagnostic algorithms. It includes data from 1896 different engines, each 50 flights in length, including fault-free (nominal) engines and engines experiencing single fault events containing abrupt and rapid faults. A sample contains the data collected from a flight. In this case study, cruise data are selected. Thus, the training data set contains 94,800 training samples. Inputs of a training sample are the eleven measurements as shown in Fig. 9. The output of a training sample is the fault ID as shown in Table 2. The description of the training data is shown in Table 4. The training data size could be modified during the training optimisation process. Similarly, the second data set is the test data set that includes a fleet of 9983 engines conducting 50 flights each. The test data set contains 499,150 test samples. Except for the two sets of data, a blind test case data set (i.e., a data set where the true fault state of the engines contained in the data set is unknown to the end-users) from NASA is used. The total number of blind test samples is 499,550.

3.2.2. Data processing

Because the data processing methods are not investigated in this paper, the datasets are processed with the processing methods in ProDiMES [4]. In ProDiMES, as an initial step of data processing, all engine measurement data are corrected to standard ISA conditions at sea level to eliminate the impact of ambient conditions on the measurement data. Then, a gradual deterioration trend monitoring approach [4] is applied to capture the gradual performance changes in the form of residuals, or measurement deltas, relative to a fleet average engine model or 50 percent deteriorated engine. In ProDiMES, this model was generated by running the C-MAPSS Steady-State engine model over a range of altitude, Mach number, and corrected fan speed settings while setting module health parameters to represent a 50 percent deteriorated engine. Finally, an exponential moving average (EMA) algorithm is applied to smooth the residual values in preparation for further analysis. Seven of the eleven measurements, Nc, P24, Ps30, T24, T30, T48, WF36, are preserved as the inputs of a sample after the data processing.

3.3. Hierarchical diagnostic method

3.3.1. Similarities between classes

The predictions based on the test data set given by the weighted least squares-based diagnostic method are processed using the method described in Section 2.1. A confusion matrix is generated and shown in Fig. 10 (a), where the number of each actual fault class and the corresponding predicted fault class is provided. Then the normalised confusion matrix is obtained and shown in Fig. 10 (b). After that, the similarity matrix can be obtained using Eq. (2), and the results are shown in Fig. 10 (c).

3.3.2. Hierarchy

The class hierarchy is built by applying the HAC algorithm described in Section 2.2. The original linkage distances are the similarities between classes as shown in Fig. 10 (c), and the Complete Distance is used as the linkage criterion for calculating the newly formed distances. The hierarchical clustering results are graphically represented as a dendrogram, as shown in Fig. 11.

3.3.3. Classification

The hierarchy is merged using the threshold approach described in Section 2.3. The pre-defined distance threshold is set to be 1.2. As a result, four connecting nodes with distances greater than 1.2 are reserved, and the other nodes are merged. Consequently, the classification structure using the LCPN approach is generated and shown in

	NF	Fan	LPC	HPC	HPT	LPT	VSV	VBV	Nf	Nc	P24	Ps30	T24	T30	T48	WF36	P2	T2	Pamb
NF	261465	39	48	7	18	28	18	102	42	112	41	78	66	59	76	45	105	66	45
Fan	2561	10476	0	0	0	0	0	3	190	0	0	2	1	0	1	0	3	2	12
LPC	5932	0	6064	2	0	13	0	809	0	3	58	0	6	1	0	3	105	0	28
HPC	1182	0	4	11762	0	0	0	0	0	9	0	7	0	1	0	0	2	1	5
HPT	477	3	0	0	12808	0	0	0	0	0	0	14	0	0	0	7	0	0	0
LPT	915	0	12	0	0	11793	1	2	4	0	5	1	0	1	0	0	0	1	117
VSV	766	0	0	0	0	0	11966	0	0	84	0	0	2	0	0	0	0	0	0
VBV	6547	0	903	0	0	2	0	5654	0	2	20	1	61	0	2	0	26	8	2
Nf	2531	303	0	0	0	1	0	3	10028	0	0	6	0	1	0	0	5	0	72
Nc	9396	0	0	1	0	0	44	1	0	3836	0	1	5	20	0	0	0	0	0
P24	2591	3	82	0	0	3	0	28	0	1	10585	8	0	0	0	0	42	0	8
Ps30	5049	23	1	5	28	3	0	0	2	1	4	8087	0	0	0	31	2	12	3
T24	2758	2	6	0	0	4	0	48	0	7	0	0	10427	1	0	0	1	23	0
T30	2740	0	1	33	1	2	0	0	0	42	0	0	10	10360	0	0	8	1	0
T48	2173	0	0	0	3	0	0	0	0	0	0	0	0	0	11210	0	0	0	0
WF36	2767	0	1	0	5	0	0	1	2	1	0	7	1	1	3	10416	0	0	1
P2	7335	11	173	5	0	0	0	11	1	1	16	3	1	6	0	0	5401	23	2
T2	2970	5	0	2	0	0	1	30	1	0	0	5	33	5	0	0	22	10148	1
Pamb	11808	146	30	1	0	54	0	2	46	2	5	4	2	1	3	0	0	1	995

(a) Confusion Matrix M.

	NF	Fan	LPC	HPC	HPT	LPT	VSV	VBV	Nf	Nc	P24	Ps30	T24	T30	T48	WF36	P2	T2	Pamb
NF	1.00	1E-04	2E-04	3E-05	7E-05	1E-04	7E-05	4E-04	2E-04	4E-04	2E-04	3E-04	3E-04	2E-04	3E-04	2E-04	4E-04	3E-04	2E-04
Fan	2E-01	0.79	0	0	0	0	0	2E-04	1E-02	0	0	2E-04	8E-05	0	8E-05	0	2E-04	2E-04	9E-04
LPC	5E-01	0	0.47	2E-04	0	1E-03	0	6E-02	0	2E-04	4E-03	0	5E-04	8E-05	0	2E-04	8E-03	0	2E-03
HPC	9E-02	0	3E-04	0.91	0	0	0	0	0	7E-04	0	5E-04	0	8E-05	0	0	2E-04	8E-05	4E-04
HPT	4E-02	2E-04	0	0	0.96	0	0	0	0	0	0	1E-03	0	0	0	5E-04	0	0	0
LPT	7E-02	0	9E-04	0	0	0.92	8E-05	2E-04	3E-04	0	4E-04	8E-05	0	8E-05	0	0	0	8E-05	9E-03
VSV	6E-02	0	0	0	0	0	0.93	0	0	7E-03	0	0	2E-04	0	0	0	0	0	0
VBV	5E-01	0	7E-02	0	0	2E-04	0	0.43	0	2E-04	2E-03	8E-05	5E-03	0	2E-04	0	2E-03	6E-04	2E-04
Nf	2E-01	2E-02	0	0	0	8E-05	0	2E-04	0.77	0	0	5E-04	0	8E-05	0	0	4E-04	0	6E-03
Nc	7E-01	0	0	8E-05	0	0	3E-03	8E-05	0	0.29	0	8E-05	4E-04	2E-03	0	0	0	0	0
P24	2E-01	2E-04	6E-03	0	0	2E-04	0	2E-03	0	7E-05	0.79	6E-04	0	0	0	0	3E-03	0	6E-04
Ps30	4E-01	2E-03	8E-05	4E-04	2E-03	2E-04	0	0	2E-04	8E-05	3E-04	0.61	0	0	0	2E-03	2E-04	9E-04	2E-04
T24	2E-01	2E-04	5E-04	0	0	3E-04	0	4E-03	0	5E-04	0	0	0.79	8E-05	0	0	8E-05	2E-03	0
T30	2E-01	0	8E-05	3E-03	8E-05	2E-04	0	0	0	3E-03	0	0	8E-04	0.78	0	0	6E-04	8E-05	0
T48	2E-01	0	0	0	2E-04	0	0	0	0	0	0	0	0	0	0.84	0	0	0	0
WF36	2E-01	0	8E-05	0	4E-04	0	0	8E-05	2E-04	8E-05	0	5E-04	8E-05	8E-05	2E-04	0.79	0	0	8E-05
P2	6E-01	8E-04	1E-02	4E-04	0	0	0	8E-04	8E-05	8E-05	1E-03	2E-04	8E-05	5E-04	0	0	0.42	2E-03	2E-04
T2	2E-01	4E-04	0	2E-04	0	0	8E-05	2E-03	8E-05	0	0	4E-04	2E-03	4E-04	0	0	2E-03	0.77	8E-05
Pamb	9E-01	1E-02	2E-03	8E-05	0	4E-03	0	2E-04	4E-03	2E-04	4E-04	3E-04	2E-04	8E-05	2E-04	0	0	0.00	0.08

(b) Normalised Confusion Matrix $\overline{M}$.

	NF	Fan	LPC	HPC	HPT	LPT	VSV	VBV	Nf	Nc	P24	Ps30	T24	T30	T48	WF36	P2	T2	Pamb
NF	0.00	1.13	0.72	1.28	1.36	1.30	1.32	0.66	1.11	0.41	1.13	0.87	1.11	1.11	1.18	1.11	0.60	1.09	0.12
Fan		0.00	0.96	1.21	1.26	1.22	1.23	0.95	1.08	0.99	1.12	1.01	1.11	1.11	1.15	1.12	0.97	1.10	1.06
LPC			0.00	1.08	1.15	1.10	1.12	0.54	0.94	0.61	0.95	0.77	0.95	0.95	1.00	0.95	0.62	0.93	0.65
HPC				0.00	1.32	1.29	1.30	1.08	1.20	1.13	1.21	1.13	1.21	1.20	1.24	1.21	1.10	1.20	1.22
HPT					0.00	1.33	1.34	1.15	1.25	1.21	1.26	1.19	1.25	1.25	1.28	1.26	1.17	1.25	1.30
LPT						0.00	1.31	1.10	1.21	1.15	1.22	1.14	1.22	1.22	1.25	1.22	1.12	1.21	1.24
VSV							0.00	1.12	1.22	1.17	1.23	1.16	1.23	1.23	1.26	1.23	1.14	1.22	1.26
VBV								0.00	0.94	0.56	0.95	0.76	0.94	0.94	1.00	0.94	0.60	0.92	0.60
Nf									0.00	0.97	1.11	1.00	1.10	1.10	1.14	1.11	0.95	1.09	1.05
Nc										0.00	0.99	0.75	0.97	0.97	1.04	0.98	0.53	0.95	0.36
P24											0.00	1.02	1.12	1.12	1.15	1.12	0.97	1.10	1.06
Ps30												0.00	1.01	1.01	1.06	1.01	0.76	0.99	0.81
T24													0.00	1.11	1.15	1.11	0.96	1.10	1.05
T30														0.00	1.15	1.11	0.96	1.10	1.05
T48															0.00	1.15	1.02	1.14	1.12
WF36																0.00	0.96	1.10	1.05
P2																	0.00	0.93	0.54
T2																		0.00	1.03
Pamb																			0.00

(c) Similarity Matrix $\overline{M_C}$.

Fig. 10. Calculation of the similarity between classes from the confusion matrix.

Fig. 12. It can be seen that there are four binary-class and one multi-class local classifier in the structure. Once the classification structure is decided, the local classifier selection process can be conducted. In this particular case, MLPs are chosen for all the five local classifiers. Following the local classifier selection, the grid search optimisation method [17] is used to select the optimal hyperparameters. Details of the grid search optimisation method are described in Section 2.3. Optimisation results for each local classifier are shown in Table 5. The training data size after optimisation is 8,955,200. Consequently, the well-trained classification system is ready to be used for fault diagnostics.

4. Results and discussion

4.1. Blind test results

The blind test results are presented and discussed in this section. To demonstrate the advantages of the introduced hierarchical diagnostic structure, three other diagnostic structures developed by other researchers were also used to do the same diagnostic analysis as comparisons. The three diagnostic structures are:

- The flat structure employs a flat MLP classifier for fault classification, as shown in Fig. 13.

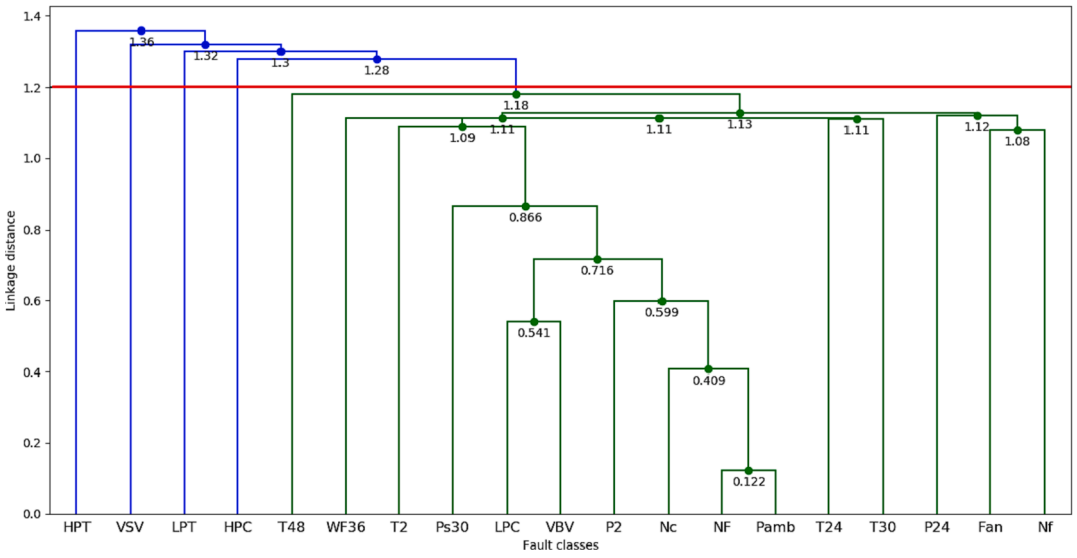


Fig. 11. Hierarchical clustering dendrogram.

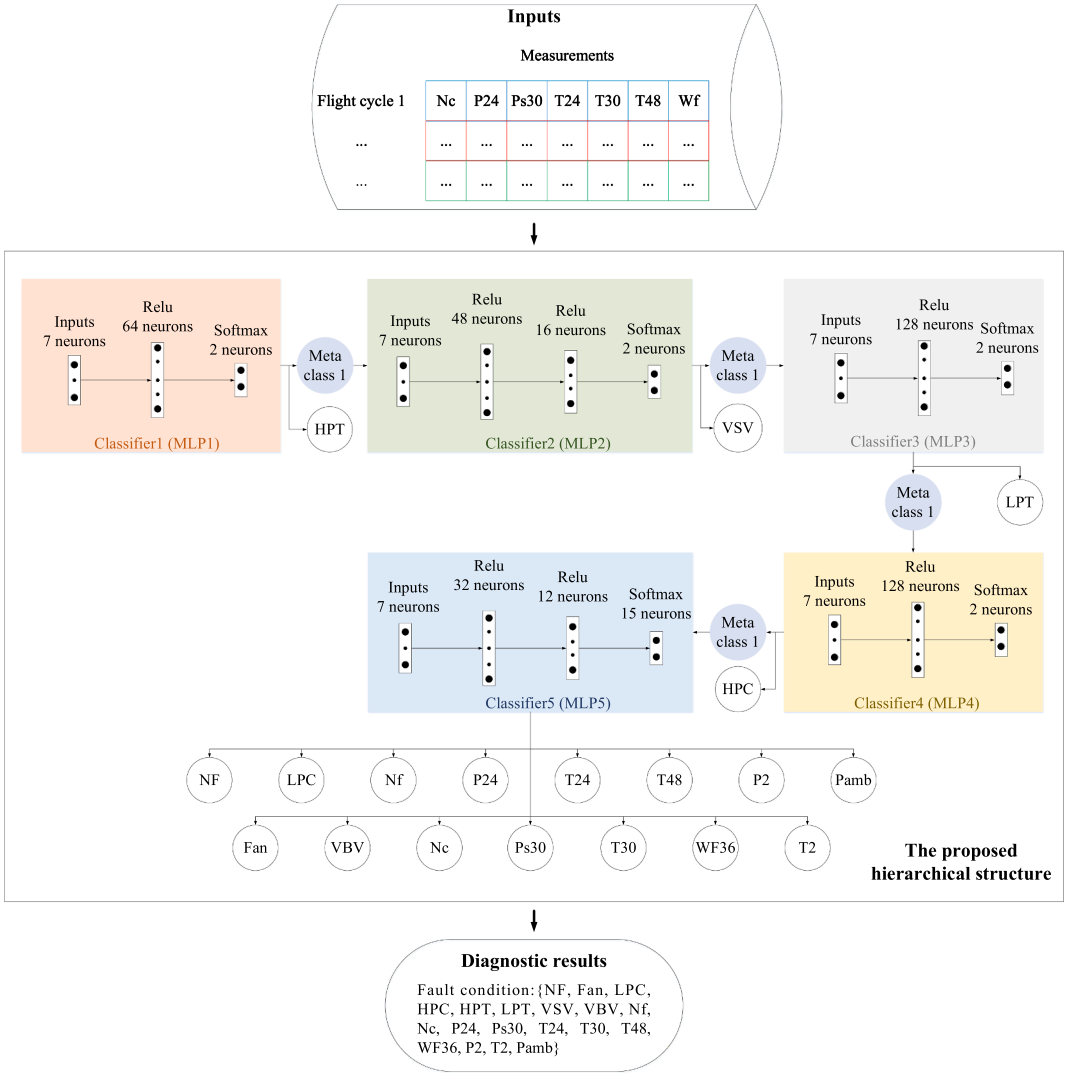


Fig. 12. The proposed hierarchical diagnostic structure.

Table 5
Local classifiers optimisation results.

Hyperparameters	Searching ranges	Optimal result	
		Classifier 1, 2, 3, 4	Classifier 5
Hidden layer number	(1, 4, 1)	1, 2, 1, 1	2
Units per hidden layer	(1, 1000, 2)	64, 48–16, 128, 128	32–12
Optimizer	(rmsprop, sgd, adam)	adam	adam
Learning rate	default	default	default
Activations	(relu, sigmoid)	relu	relu
Last-layer activation	softmax	softmax	softmax
Loss function	(binary_crossentropy, categorical_crossentropy)	binary_crossentropy	categorical_crossentropy
Epochs	1000	1000	1000
Batch size	(1, 2000, 1)	120, 96, 76, 140	72
Early stopping	(yes, no)	yes	yes
Evaluation protocols	(hold-out validation, K-fold cross-validation, iterated K-fold validation)	K-fold cross-validation	K-fold cross-validation

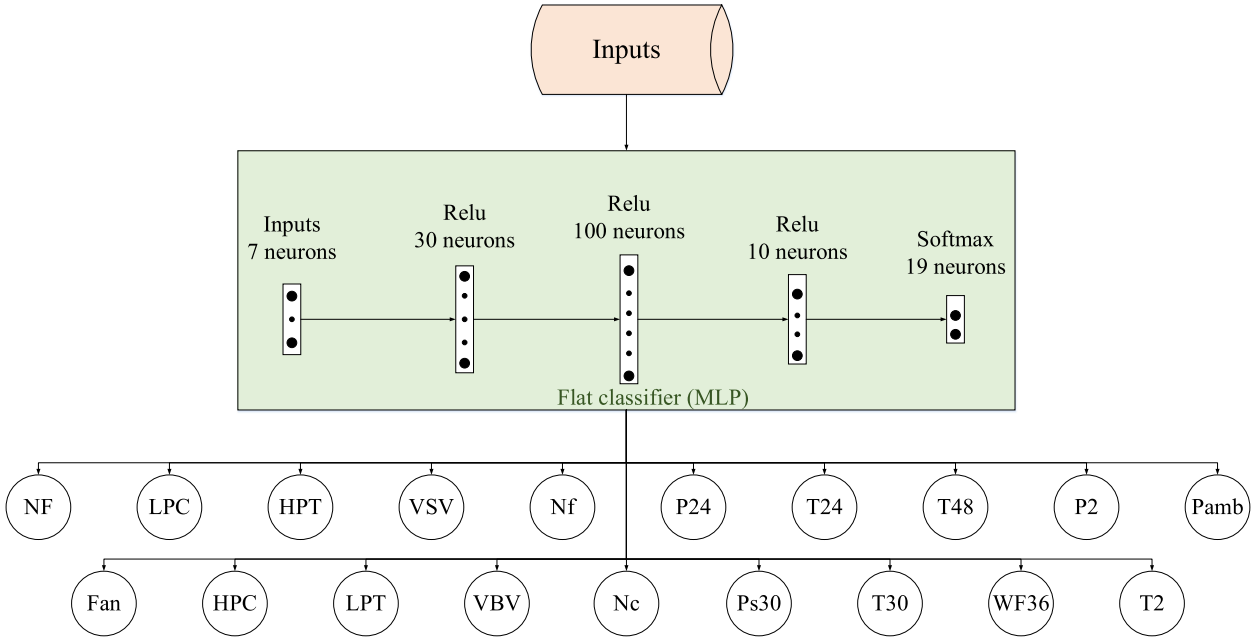


Fig. 13. A Flat MLP classification structure.

- The nested structure based on the nested NNs proposed by Ogaji [34]. The nested structure was built using the priori knowledge about the fault classes, as shown in Fig. 14. In this study, the fault detection system was nested based on actuator faults, sensor faults and component faults.
- The hierarchical structure based on the original hierarchy where no thresholds were introduced and no nodes were merged. As is shown in Fig. 15, the hierarchical structure based on the original hierarchy employs sixteen local MLP classifiers to predict the fault classes.

All four diagnostic methods were applied to the 499,550 blind test samples, and the diagnostic results were submitted to NASA for evaluation. The performance of the diagnostic methods is assessed by the evaluation metrics as shown in Appendix A, and the evaluation results of the method are shown in Fig. B.1 of Appendix B. The detailed discussions and comparisons of the results are given as follows.

4.1.1. Comparison of detection performances

The False Positive Rate (FPR), False Alarm Rate (FAR), True Positive Rate (TPR), as well as Detection Latency for all the samples are presented in Table 6.

A target FAR of no more than one false alarm per 1000 flights is specified in reference [4] to maintain uniformity in the applied diagnostic methods. All these methods satisfy the requirement.

TPR of the proposed hierarchical diagnostic method is 60.3 %, better than all the other three methods by at least 1.7 %.

In most cases, some latency is associated with the correct detection of a fault resulting in missed detections within the first few flights after the fault occurs. As shown in Table 6, the Detection Latency of the proposed diagnostic method is 3.19, which exhibits better diagnostic latency than the other diagnostic methods.

Based on the above, it can be seen that the proposed diagnostic method performs better than other methods in detecting fault events.

4.1.2. Comparison of Correct classification Rate and Misclassification Rate

The average Correct Classification Rate (CCR) and the average Misclassification Rate (MCR) for all fault types, fault magnitudes, and fault evolution rates are shown in Table 7.

CCR and MCR are two metrics evaluating fault classification performance. CCR of the proposed hierarchical diagnostic method is 58.5 %, and the proposed method produces at least 1.7 % more correct fault classifications than the compared diagnostic methods. The MCR results illustrate that the proposed method produces 1.73 % misclassifications, 0.09 % higher than the flat structure and ranks second among all the

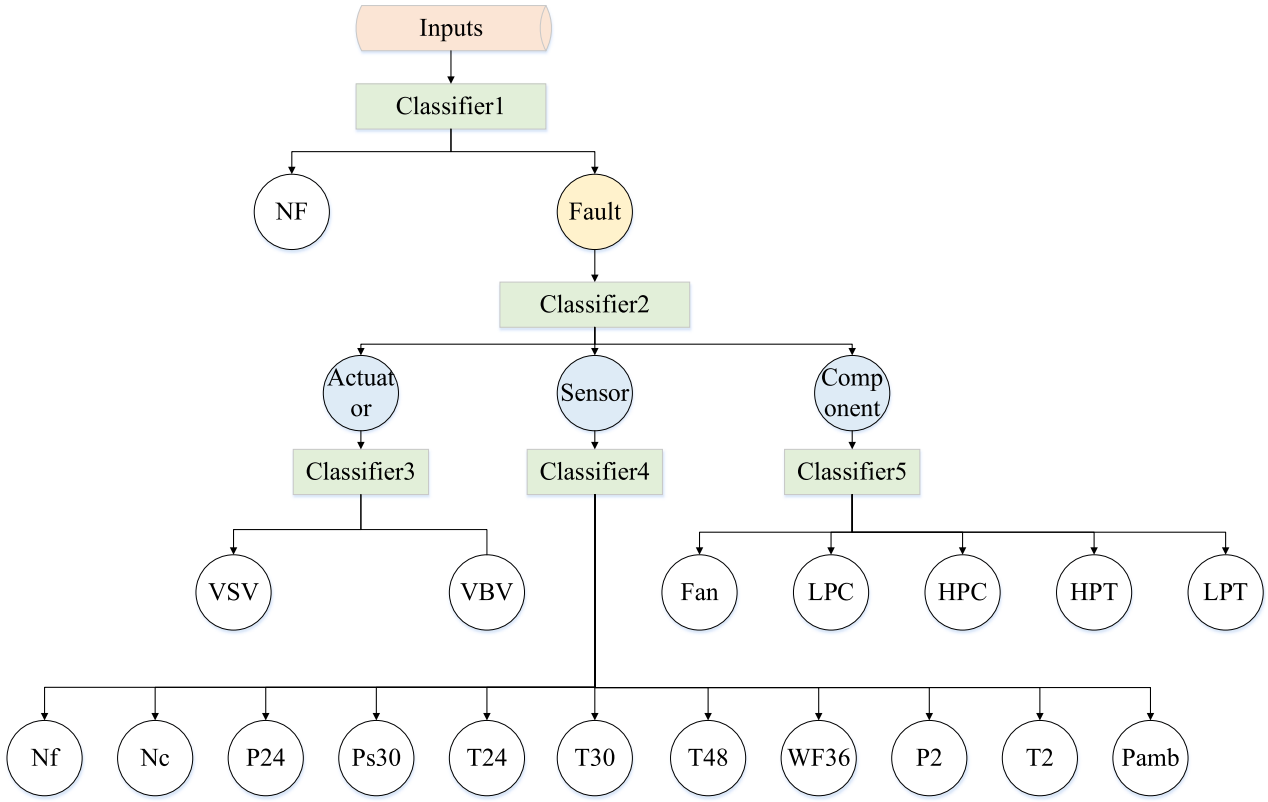


Fig. 14. Nested classification structure (adapted from reference [34]).

compared diagnostic methods.

4.1.3. Comparison of Kappa Coefficients

The average Kappa Coefficients concerning all fault types, fault magnitudes, and fault evolution rates are presented in Table 8.

Kappa Coefficient is an overall reflection of fault classification performance. Kappa Coefficient of the proposed hierarchical diagnostic method is 0.693 and is at least 0.018 higher than the compared diagnostic methods. The proposed diagnostic method has the highest Kappa Coefficient, followed by the flat structure, the hierarchical structure based on the original hierarchy, and the nested structure. The proposed diagnostic method can provide the best fault classification performance.

4.2. Discussion

A summary of the blind test case results of four different diagnostic methods is given in Table 9, which shows the ranking of the diagnostic methods for each metric. The hierarchical fault classification method proposed in this paper ranks first in TPR, CCR, Detection Latency and Kappa Coefficient evaluation metrics, and second in MCR. Therefore, the hierarchical diagnostic method shows the best diagnostic performance compared with the three other structures based on the comparison of the indicator of fault classification performance, the average Kappa Coefficient. This is due to the following reasons:

- Firstly, the hierarchy structure built from the similarity matrix can effectively decompose the complicated multi-class diagnostic problem into several relatively simple sub-problems.
- Secondly, the threshold method described in Section 2.3 can reduce the accumulation of sequential classification errors.
- Thirdly, the local classifier selection enables different classifiers to be used, and the best algorithm for each local classifier can be selected, which can provide the best possible overall prediction performance.

The flat and nested structures can be seen as two special cases of hierarchical structures. Compared to the nested and hierarchical structures based on the original hierarchy, the flat structure shows slightly better performance as the error accumulation effect is smaller. The proposed hierarchical structure in this research has more or equal local classifiers (five local classifiers) than those of the flat and nested structures (one and five local classifiers, respectively) but much smaller number of local classifiers than that of the hierarchical structure based on the original hierarchy (sixteen local classifiers), which is a trade-off between complexity and accuracy.

Concerning computation burden, the experiments were conducted on a laptop with an NVIDIA GTX 1660 Ti GPU processor running on Windows 10. In the case study, with GPU acceleration, the training time for the local classifiers in the proposed hierarchical structure is 54 s, which is 15 s less than the training time for a flat classifier. The proposed method can reduce the computation required by the system.

With regards to the model interpretability and diagnostic visibility, instead of a black box, the proposed method provides physical insight and graphical representations of the diagnostic system. The hierarchy dendrogram and the hierarchical diagnostic structure provide a physical representation of the diagnostic system. For example, a great correlation among Fan and Nf fault as well as among LPC and VBV fault can be observed in the dendrogram. That is the reason why an individual classifier is applied to Fan and Nf faults as well as LPC and VBV faults. The whole diagnostic structure can be visualized and interpreted with the proposed method.

Table 10 summarises the blind-test-case metric results from other known diagnostic methods using the ProDiMES software for evaluation. Regularized Extreme Learning Machines-Sparse Representation Classification (RELM-SRC) method [55] has the best diagnostic performance, which provides a Kappa Coefficient of 0.685. The Kappa Coefficient of the proposed method in this paper is 0.693, which is 0.008, larger than the Kappa Coefficient provided by the RELM-SRC method.

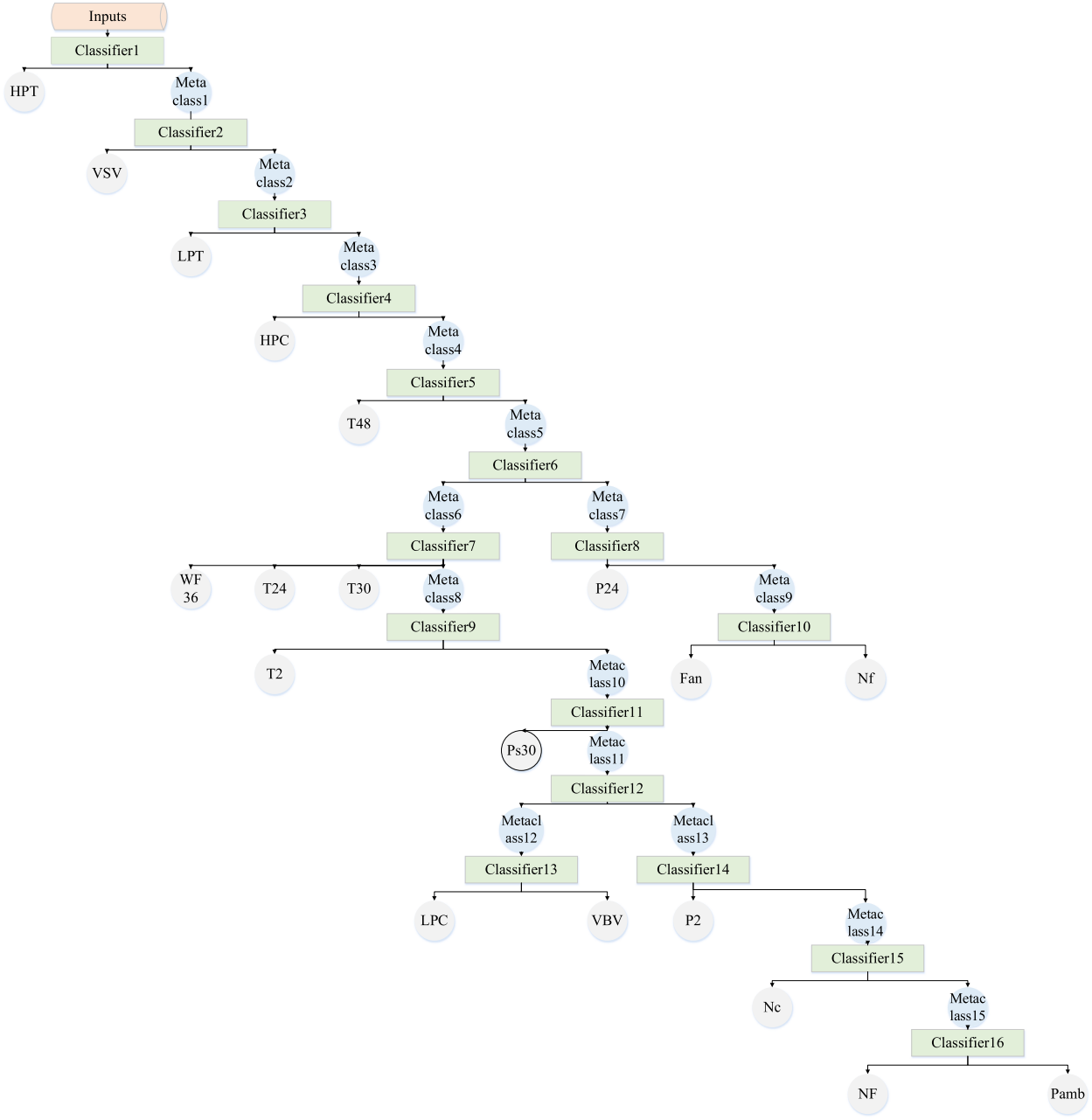


Fig. 15. Hierarchical diagnostic structure based on original hierarchy.

Table 6
Detection performance.

Diagnostic method	FPR [%]	FAR [average number of flights per false alarm]	TPR [%]	Detection Latency [average number of flights]
The flat structure	0.04735	2112	58.4	3.29
The nested structure	0.06456	1549	58.3	3.34
The hierarchical structure based on the original hierarchy	0.08117	1232	58.6	3.30
The proposed hierarchical diagnostic structure	0.09240	1082	60.3	3.19

Table 7
Correct Classification Rate and Misclassification Rate.

Diagnostic method	CCR [%]	MCR [%]
The flat structure	56.8	1.64
The nested structure	56.0	2.33
The hierarchical structure based on the original hierarchy	56.5	2.05
The proposed hierarchical diagnostic structure	58.5	1.73

Table 8
Kappa Coefficients.

Diagnostic method	Kappa Coefficient
The flat structure	0.675
The nested structure	0.669
The hierarchical structure based on the original hierarchy	0.673
The proposed hierarchical diagnostic structure	0.693

Table 9
Ranking of different diagnostic methods.

Diagnostic method	TPR	Detection Latency	CCR	MCR	Kappa Coefficient
The flat structure	3rd	2nd	2nd	1st	2nd
The nested structure	4th	4th	4th	4th	4th
The hierarchical structure based on the original hierarchy	2nd	3rd	3rd	3rd	3rd
The proposed hierarchical diagnostic structure	1st	1st	1st	2nd	1st

Table 10
Blind test case results from other known diagnostic methods.

Diagnostic method	TPR [%]	Detect latency	CCR [%]	MCR [%]	Overall Kappa
Weighted Least Squares[25]	44.7	4.86	43.4	1.35	0.588
PNNs[25]	44.7	4.86	43.7	1.04	0.590
Extended Kalman Filter[25]	50.9	4.02	46.7	4.15	0.627
Generalised Observer[25]	51.9	4.24	45.2	6.78	0.617
PNNs[16]	48.5	4.70	45.2	3.29	0.595
k-Nearest Neighbour[16]	48.5	4.70	46.1	2.37	0.605
PNN and Adaptive Engine Model Fusion[16]	48.5	4.70	45.0	3.51	0.595
Support Vector Machine (SVM)[56]	52.5	3.30	–	–	0.660
Upper and Lower Singleton Type-2 Fuzzy Logic System (ULST2-FLS)[57]	52.2	4.45	49.4	2.72	0.647
Regularized Extreme Learning Machines- Sparse Representation Classification (RELM-SRC) [55]	55.7	3.30	55.4	0.30	0.685

5. Conclusions

In the context of Intelligent Engines, this study proposes a novel hierarchical aero-engine gas path diagnostic method for improved model interpretability, model flexibility, diagnostic visibility, diagnostic accuracy, and diagnostic reliability. The proposed method is evaluated with NASA's ProDiMES software.

The conclusions drawn from this study are as follows:

- Instead of a black box, the proposed method provides physical insight and graphical representations of the diagnostic system. It improves model interpretability and diagnostic visibility for enhanced diagnostic information and efficiency in learning and representation.
- The proposed diagnostic method recommends a hierarchical structure that enables local classifier selection, training and global application. The divide-and-conquer approach improves model flexibility, which facilitates fine-tuning and control.
- The proposed diagnostic method is superior in diagnostic accuracy to other methods in the open literature. Kappa Coefficient of the proposed hierarchical diagnostic method is 0.693 and is at least 0.008 higher than the other diagnostic methods.

Overall, the proposed method can quantify the dependence relationships between the fault classes for enhanced diagnostic information, recommend the best diagnostic structure for reduced complexity, and solve the error accumulation problem for improved diagnostic accuracy. The proposed method can potentially contribute to intelligent condition monitoring systems by effectively exploiting physical and data-based information, which will positively enhance the availability, reliability, and efficiency of Intelligent Engines.

As an ML method, the proposed method has limitations in terms of the need for a large amount of labelled training data, case-dependency and offline training for the applications in aero-engine diagnostics. Fortunately, emerging technologies such as Digital Twin, Intelligent Engine, and Incremental Online Learning provide potential opportunities for ML applications. As for future work, more studies to address these application limitations will be worth developing.

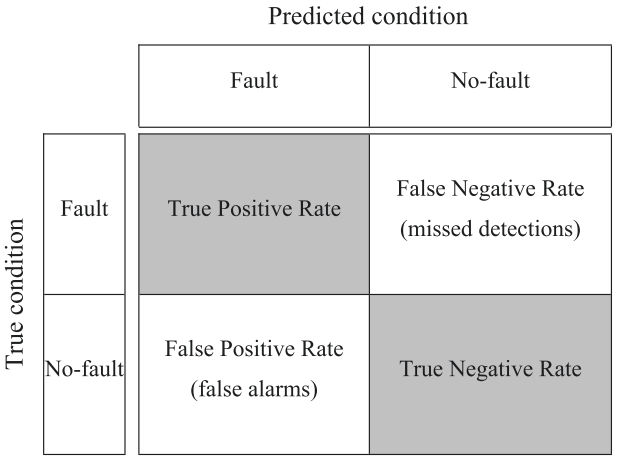


Fig. A.2. Normalised detection decision matrix.

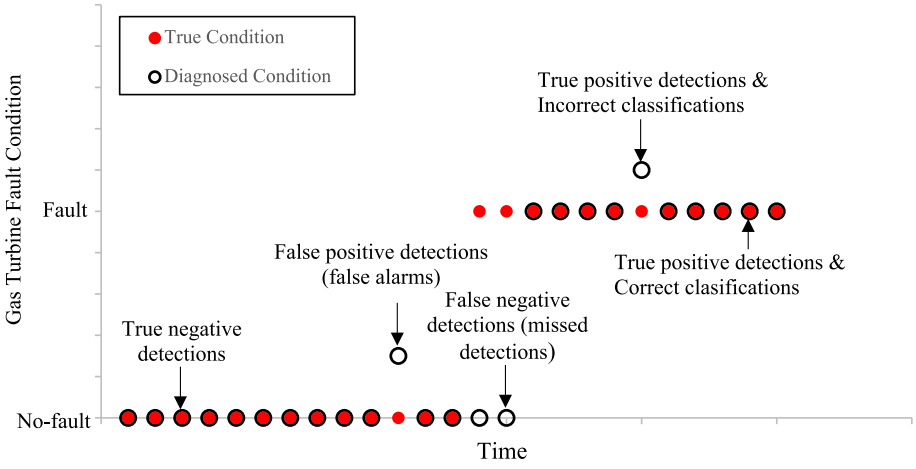


Fig. A.1. Example of the evaluation results used in diagnostic assessment.

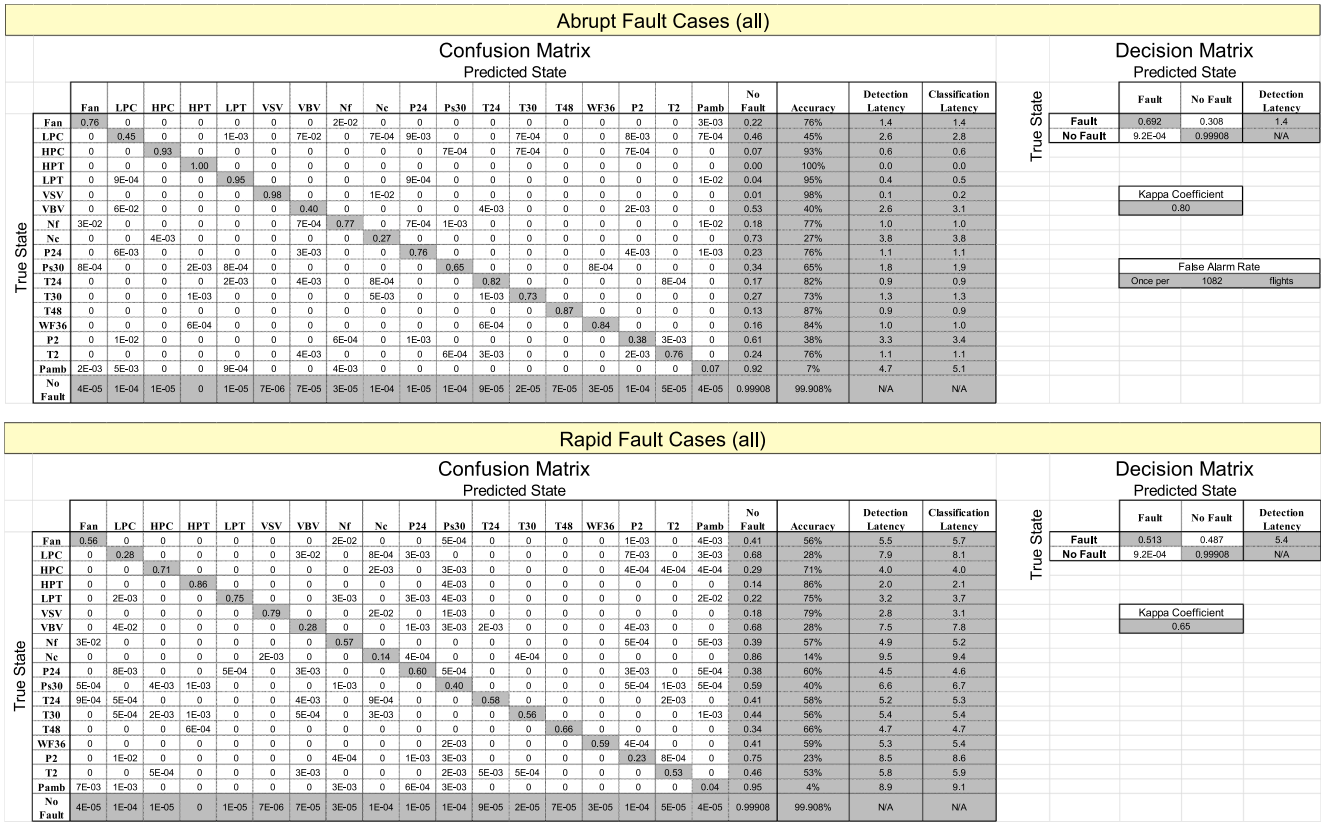


Fig. B.1. Evaluation metrics for the hierarchical diagnostic method.

CRediT authorship contribution statement

Junjie Zhao: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Yi-Guang Li:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision. **Suresh Sampath:** Writing – review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Evaluation metrics

The evaluation metrics are presented as follows. A schematic of an example that applies the evaluation metrics in diagnostic assessment is shown in Fig. A.1.

True Positive Rate (TPR) or Sensitivity refers to the ability of a diagnostic method to correctly identify the faulty cases that do have faults. Mathematically, TPR is calculated by Eq. (A.1).

$$TPR = \frac{The\ number\ of\ faulty\ cases\ correctly\ identified}{The\ total\ number\ of\ faulty\ cases} \tag{A.1}$$

True Negative Rate (TNR) or Specificity refers to the ability of a diagnostic method to correctly identify the healthy (no-fault) cases that do not have faults. Mathematically, TPR is calculated by Eq. (A.2).

$$TNR = \frac{The\ number\ of\ healthy\ cases\ correctly\ identified}{The\ total\ number\ of\ healthy\ cases} \tag{A.2}$$

False Negative Rate (FNR) or Missed Detection Rate is the ratio between the number of healthy (no-fault) cases wrongly identified as faulty cases and the total number of faulty cases, i.e., $FNR = 1 - TPR$.
False Positive Rate (FPR) is the ratio between the number of faulty cases wrongly identified as healthy cases and the total number of healthy (no-fault) cases, i.e., $FPR = 1 - TNR$.
False Alarm Rate (FAR) is the ratio between the total number of healthy (no-fault) cases and the number of faulty cases wrongly identified as healthy cases, i.e., $FAR = 1/FPR$.
Correct Classification Rate (CCR) is the ratio between the number of cases correctly classified as faulty cases and the total number of faulty cases. CCR for each fault class is listed on the main diagonal elements of the normalised confusion matrix, as shown in Fig. 4 (b).
Misclassification Rate (MCR) is the ratio between the number of cases incorrectly classified as faulty cases and the total number of faulty cases. MCR is displayed along the off-diagonal elements of the normalised confusion matrix, as shown in Fig. 4 (b).
Classification Accuracy is the ratio between the total number of correctly identified cases and the total number of all cases, as shown in Eq. (A.3).

$$Classification\ Accuracy = \frac{The\ total\ number\ of\ correctly\ identified\ cases}{The\ total\ number\ of\ cases} \tag{A.3}$$

Detection Latency is the average number of flights a fault must persist before a true positive detection is detected by a diagnostic method.
Classification Latency is the average number of flights where a fault must persist before the fault is correctly classified by a diagnostic method.
Kappa Coefficient provides a measure of a diagnostic algorithm's ability to correctly classify a fault. It is a robust measure because the possibility of the results occurring by chance is taken into account. It is calculated from the elements of an un-normalised confusion matrix M shown in Fig. 4 (a) by Eq. (A.4). A Kappa Coefficient is a number between −1 and 1, its value being equal one generally means a diagnostic method achieving perfect fault classification performance while its value lower than zero is considered its worse classification performance than that expected by chance.

$$k = \frac{N(correctly\,classified) - N(expected\ correctly\ \ classified\,by\,chance)}{N(total) - N(\text{expected correctly}\ \ classified\,by\,chance)} \tag{A.4}$$

Where

$$N(correctly\,classified) = \sum_{i=1}^{n} M_{ii} \tag{A.5}$$

$$N(total) = \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij} \tag{A.6}$$

$$N(expected\ correctly\ \ classified\,by\,chance) = \sum_{i=1}^{n}\left\{\sum_{j=1}^{n}\frac{M_{ij}}{N(total)}\cdot\sum_{j=1}^{n} M_{ji}\right\} \tag{A.7}$$

Where the two subscript indices i and j represent the row and column numbers corresponding to individual confusion matrix elements; $\sum_{j=1}^{n}\frac{M_{ij}}{N(total)}$ is the summation of the ratios between the total number of true cases for fault class i and the total number of all cases, referring to the probability of correctly classified cases by chance for fault class i; $\sum_{j=1}^{n} M_{ji}$ is the the toal number of predicted cases for fault class i, and $\sum_{j=1}^{n}\frac{M_{ij}}{N(total)}\cdot\sum_{j=1}^{n} M_{ji}$ is the number of correctly classified cases by chance for fault class i.

Items 1 to 5 in the evaluation metrics list are used to form a normalised detection decision matrix as presented in Fig. A.2. Items 6 to 8 in the evaluation metrics list are derived from a classification confusion matrix, as shown in Fig. 4 (a).

Appendix B. The blind test evaluation results from NASA

References

[1] Lipowsky H, Staudacher S, Bauer M, Schmidt KJ. Application of bayesian forecasting to change detection and prognosis of gas turbine performance. J Eng Gas Turbines Power 2010;132:1–8. https://doi.org/10.1115/1.3159367.
[2] Chen Y-Z, Tsoutsanis E, Xiang H-C, Li Y-G, Zhao J-J. A dynamic performance diagnostic method applied to hydrogen powered aero engines operating under transient conditions. Appl Energy 2022;317:1–20. https://doi.org/10.1016/j.apenergy.2022.119148.
[3] Li J, Ying Y. Gas turbine gas path diagnosis under transient operating conditions: A steady state performance model based local optimisation approach. Appl Therm Eng 2020;170:1–14. https://doi.org/10.1016/j.applthermaleng.2020.115025.
[4] Simon DL. NASA/TM—2010-215840: Propulsion Diagnostic Method Evaluation Strategy (ProDiMES) User's Guide. NASA; 2010.
[5] Sarkar S, Rao C, Ray A. Estimation of multiple faults in aircraft gas-turbine engines. Proc. 2009 Am. Control Conf., Hyatt Regency Riverfront, St. Louis, MO, USA, 10-12 June 2009, Piscataway, NJ: IEEE; 2009, p. 216–21. 10.1109/ACC.2009.5159981.
[6] Kang DW, Kim TS. Model-based performance diagnostics of heavy-duty gas turbines using compressor map adaptation. Appl Energy 2018;212:1345–59. https://doi.org/10.1016/j.apenergy.2017.12.126.
[7] Tahan M, Tsoutsanis E, Muhammad M, Abdul Karim ZA. Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review. Appl Energy 2017;198:122–44. https://doi.org/10.1016/j.apenergy.2017.04.048.
[8] Bai M, Liu J, Chai J, Zhao X, Yu D. Anomaly detection of gas turbines based on normal pattern extraction. Appl Therm Eng 2020;166:1–18. https://doi.org/10.1016/j.applthermaleng.2019.114664.
[9] Tang S, Tang H, Chen M. Transfer-learning based gas path analysis method for gas turbines. Appl Therm Eng 2019;155:1–13. https://doi.org/10.1016/j.applthermaleng.2019.03.156.
[10] Escher PC. Pythia: An object-orientated gas path analysis computer program for general applications. PhD thesis. Cranfield University, 1995.
[11] Li YG, Nilkitsaranont P. Gas turbine performance prognostic for condition-based maintenance. Appl Energy 2009;86:2152–61. https://doi.org/10.1016/j.apenergy.2009.02.011.

[12] Provost MJ. The use of optimal estimation techniques in the analysis of gas turbines. PhD thesis. Cranfield University, 1994.
[13] Li YG, Pilidis P. GA-based design-point performance adaptation and its comparison with ICM-based approach. Appl Energy 2010;87:340–8. https://doi.org/10.1016/j.apenergy.2009.05.034.
[14] Doel DL. An assessment of weighted-least-squares based gas path analysis. Proc. ASME 1993 Int. Gas Turbine Aeroengine Congr. Expo., vol. 116, Cincinnati, Ohio, 24-27 May 1993, New York: ASME; 1993, p. 366–73. 10.1115/93-GT-119.
[15] Kong C. Review on advanced health monitoring methods for aero gas turbines using model based methods and artificial intelligent methods. Int J Aeronaut Sp Sci 2014;15:123–37. https://doi.org/10.5139/IJASS.2014.15.2.123.
[16] Koskoletos AO, Aretakis N, Alexiou A, Romesis C, Mathioudakis K. Evaluation of aircraft engine gas path diagnostic methods through ProDiMES. J Eng Gas Turbines Power 2018;140:1–13. https://doi.org/10.1115/1.4040909.
[17] F, Chollet Deep Learning with Python. vol. 10. Shelter Island: Manning Publications; 2018. 10.1016/s1474-4422(11)70007-2.
[18] Orozco DJR, Venturini OJ, Escobar Palacio JC, del Olmo OA. A new methodology of thermodynamic diagnosis, using the thermoeconomic method together with an artificial neural network (ANN): a case study of an externally fired gas turbine (EFGT). Energy 2017;123:20–35. https://doi.org/10.1016/j.energy.2016.12.099.
[19] LK, Sambasivan, VB, Kini, S, Ryali, J, Mukherjee, D. Mylaraswamy Comparison of a few fault diagnosis methods on sparse variable length time series sequences. Proc. ASME Turbo Expo 2007 Power Land, Sea, Air, Montreal, Canada, 14–17 May 2007: 2007, p. 713–20. 10.1115/GT2007-27843.
[20] Zhou D, Zhang H, Weng S. A new gas path fault diagnostic method of gas turbine based on support vector machine. J Eng Gas Turbines Power 2015;137:1–6. https://doi.org/10.1115/1.4030277.
[21] LC, Jaw, YJ, Lee Engine diagnostics in the eyes of machine learning. Proc. ASME Turbo Expo 2014, vol. 6, Düsseldorf, Germany, 16–20 June 2014, New York: ASME; 2014, p. 1–8. 10.1115/GT2014-27088.
[22] M, Maragoudakis, E, Loukis, PP, Pantelides Random forests identification of gas turbine faults. Proc. 19th Int. Conf. Syst. Eng. (ICSEng 2008), Las Vegas, NV, USA, 19-21 August 2008: 2008, p. 127–32. 10.1109/ICSEng.2008.81.
[23] Loboda I, Olivares Robles MA. Gas turbine fault diagnosis using probabilistic neural networks. Int J Turbo Jet Engines 2015;32:175–91. https://doi.org/10.1515/tjj-2014-0019.
[24] Ogaji SOT, Singh R, Probert SD. Multiple-sensor fault-diagnoses for a 2-shaft stationary gas-turbine. Appl Energy 2002;71:321–39. https://doi.org/10.1016/S0306-2619(02)00015-6.
[25] Simon DL, Borguet S, Léonard O, Zhang X. Aircraft engine gas path diagnostic methods: public benchmarking results. J Eng Gas Turbines Power 2014;136:1–10. https://doi.org/10.1115/1.4025482.
[26] Lu PJ, Zhang MC, Hsu TC, Zhang J. An evaluation of engine faults diagnostics using artificial neural networks. J Eng Gas Turbines Power 2001;123:340–6. https://doi.org/10.1115/1.1362667.
[27] MJ, Roemer Testing of a real-time health monitoring and diagnostic system for gas turbine engines. Proc. 34th AIAA/ASME/SAE/ASEE Jt. Propuls. Conf. Exhib., Cleveland, OH, USA,13 July 1998: 1998. 10.2514/6.1998-3603.
[28] Tamilselvan P, Wang P. Failure diagnosis using deep belief learning based health state classification. Reliab Eng Syst Saf 2013;115:124–35. https://doi.org/10.1016/j.ress.2013.02.022.
[29] H, Luo, S, Zhong Gas turbine engine gas path anomaly detection using deep learning with Gaussian distribution. Proc. 2017 Progn. Syst. Heal. Manag. Conf. (PHM-Harbin 2017), Harbin, China, 9-12 July 2017: 2017, p. 1–6. 10.1109/PHM.2017.8079166.
[30] Bai M, Yang X, Liu J, Liu J, Yu D. Convolutional neural network-based deep transfer learning for fault detection of gas turbine combustion chambers. Appl Energy 2021;302:1–24. https://doi.org/10.1016/j.apenergy.2021.117509.
[31] J, Zhao, Y, Li, Abrupt fault detection and isolation for gas turbine components based on a 1d convolutional neural network using time series data. Proc. AIAA Propuls. Energy 2020 Forum, Virtual, Online, 24-28 August 2020: 2020, p. 1–19. 10.2514/6.2020-3675.
[32] Zhou D, Yao Q, Wu H, Ma S, Zhang H. Fault diagnosis of gas turbine based on partly interpretable convolutional neural networks. Energy 2020;200:1–17. https://doi.org/10.1016/j.energy.2020.117467.
[33] M, Zedda, R, Singh, Fault diagnosis of a turbofan engine using neural networks: a quantitative approach. Proc. 34th AIAA/ASME/SAE/ASEE Jt. Propuls. Conf. Exhib., vol. 6, Cleveland, OH, USA, 13-15 July 1998: 1998. 10.2514/6.1998-3602.
[34] Ogaji SOT. Advanced gas-path fault diagnostics for stationary gas turbines. PhD Thesis. Cranfield University, 2003.
[35] Joly RB, Ogaji SOT, Singh R, Probert SD. Gas-turbine diagnostics using artificial neural-networks for a high bypass ratio military turbofan engine. Appl Energy 2004;78:397–418. https://doi.org/10.1016/j.apenergy.2003.10.002.
[36] A, Courdier, YG, Li, Power setting sensor fault detection and accommodation for gas turbine engines using artificial neural networks. Proc. ASME Turbo Expo 2016 Turbomach. Tech. Conf. Expo., June 13 – 17, 2016, Seoul, South Korea: 2016, p. 1–10.
[37] Fentaye AD, Ul-Haq Gilani SI, Baheta AT, Li YG. Performance-based fault diagnosis of a gas turbine engine using an integrated support vector machine and artificial neural network method. Proc Inst Mech Eng Part A J Power Energy 2019;233:786–802. https://doi.org/10.1177/0957650918812510.
[38] N, Batayev Gas turbine fault classification based on machine learning supervised techniques. Proc. 14th Int. Conf. Electron. Comput. Comput. (ICECCO 2018), Kaskelen, Kazakhstan, 29 November - 1 December 2018: IEEE; 2019, p. 206–12. 10.1109/ICECCO.2018.8634719.
[39] Allen CW, Holcomb CM, De Oliveira M. Gas turbine machinery diagnostics: A brief review and a sample application. Proc. ASME Turbo Expo 2017 Turbomach. Tech. Conf. Expo., vol. 6, Charlotte, NC, USA, 26-30 June 2017: 2017, p. 1–12. 10.1115/GT2017-64755.
[40] Wang XX, Ma LY. A compact K nearest neighbor classification for power plant fault diagnosis. J Inf Hiding Multimed Signal Process 2014;5:508–17.
[41] Zhao YP, Wang JJ, Li XY, Peng GJ, Yang Z. Extended least squares support vector machine with applications to fault diagnosis of aircraft engine. ISA Trans 2020;97:189–201. https://doi.org/10.1016/j.isatra.2019.08.036.
[42] Rabenoro T, Lacaille J, Cottrell M, Rossi F. Interpretable aircraft engine diagnostic via expert indicator aggregation. Trans Mach Learn 2014;7:39–64.
[43] Lu J, Huang J, Lu F. Kernel extreme learning machine with iterative picking scheme for failure diagnosis of a turbofan engine. Aerosp Sci Technol 2020;96:1–12. https://doi.org/10.1016/j.ast.2019.105539.
[44] GR, Matuck, JR, Barbosa, C, Bringhenti, I, Lima Multiple Faults Detection of Gas Turbine by MLP Neural Network. Proc. ASME Turbo Expo 2009 Power Land, Sea Air-GT2009, June 8-12, 2009, Orlando, Florida, USA: 2009, p. 1–7.
[45] I, Loboda Gas turbine fault classification using probability density estimation. Proc. ASME Turbo Expo 2014 Turbine Tech. Conf. Expo., Düsseldorf, Germany, 16–20 June 2014, New York: 2014, p. 1–13.
[46] Muthuraman S, Twiddle J, Singh M, Connolly N. Condition monitoring of SSE gas turbines using artificial neural networks. Insight Non-Destructive Test Cond Monit 2012;54:436–9. https://doi.org/10.1784/insi.2012.54.8.436.
[47] Sina Tayarani-Bathaie S, Sadough Vanini ZN, Khorasani K. Dynamic neural network-based fault diagnosis of gas turbine engines. Neurocomputing 2014;125:153–65. https://doi.org/10.1016/j.neucom.2012.06.050.
[48] Sina Tayarani-Bathaie S, Khorasani K. Fault detection and isolation of gas turbine engines using a bank of neural networks. J Process Control 2015;36:22–41. https://doi.org/10.1016/j.jprocont.2015.08.007.
[49] Yan LP, Dong XZ, Wang T, Gao Q, Tan CQ, Zeng DT, et al. A fault diagnosis method for gas turbines based on improved data preprocessing and an optimisation deep belief network. Meas Sci Technol 2020;31:1–12. https://doi.org/10.1088/1361-6501/ab3862.
[50] X, Fu, H, Chen, G, Zhang, T. Tao A New Point Anomaly Detection Method about Aero Engine Based on Deep Learning. Proc. 2018 Int. Conf. Sensing, Diagnostics, Progn. Control (SDPC 2018), Xi'an, China, 15-17 August 2018: IEEE; 2018, p. 176–81. 10.1109/SDPC.2018.8664977.
[51] Cao Q, Chen S, Zheng Y, Ding Y, Tang Y, Huang Q, et al. Classification and prediction of gas turbine gas path degradation based on deep neural networks. Int J Energy Res 2021;45:10513–26. https://doi.org/10.1002/er.6539.
[52] F, Nielsen Chapter 8: Hierarchical Clustering. Introd. to HPC with MPI Data Sci., vol. 7, Springer; 2016, p. 221–39.
[53] Silla CN, Freitas AA. A survey of hierarchical classification across different application domains. Data Min Knowl Discov 2011;22:31–72. https://doi.org/10.1007/s10618-010-0175-9.
[54] I, Goodfellow, Y, Bengio, A, Courville Deep Learning. MIT Press; 2016.
[55] Pérez-Ruiz JL, Tang Y, Loboda I. Aircraft engine gas-path monitoring and diagnostics framework based on a hybrid fault recognition approach. Aerospace 2021;8:1–26. https://doi.org/10.3390/aerospace8080232.
[56] I, Loboda, JL, Pérez-Ruiz, S, Yepifanov A benchmarking analysis of a data-driven gas turbine diagnostic approach. Proc. ASME Turbo Expo 2018 Turbomach. Tech. Conf. Expo., Oslo, Norway, 11-15 June 2018: 2018, p. 1–13. 10.2355/tetsutohagane1955.60.4_S8.
[57] Calderano PHS, Ribeiro MGC, Amaral RPF, Vellasco MMBR, Tanscheit R, de Aguiar EP. An enhanced aircraft engine gas path diagnostic method based on upper and lower singleton type-2 fuzzy logic system. J Brazilian Soc Mech Sci Eng 2019;41:1–14. https://doi.org/10.1007/s40430-019-1567-4.

https://dspace.org/

A hierarchical structure built on physical and data-based information for intelligent aero-engine gas path diagnostics

Zhao, Junjie; Li, Yiguang; Sampath, Suresh

203-02-15
Elsevier
https://doi.org/10.1016/j.apenergy.2022.120520
Zhao J, Li Y-G, Sampath S. (2023) A hierarchical structure built on physical and data-based information for intelligent aero-engine gas path diagnostics. Applied Energy, Volume 332, February 2023, Article number 120520

Downloaded from a DSpace repository. https://dspace.org/