

An intelligent data filtering and fault detection method for gas turbine engines

Amare D. Fentaye^{1,*}, and *Konstantinos G. Kyprianidis*¹

¹School of Business, Society and Engineering, Mälardalen University, 883, SE-72123 Västerås, Sweden

Abstract. In a gas turbine fault diagnostics, the removal of measurement noise and data outliers prior to the fault analysis is very essential. The conventional filtering methods, particularly the linear ones, are not sufficiently accurate, which might possibly lead to the loss of critically important features in the fault analysis process. Conversely, the recorded accuracies obtained from the non-linear filters are promising. Recently, the focus has been shifted to the artificial neural network (ANN) based nonlinear filters due to their capability of providing a robust identity map between the input and output data, which can be efficiently exploited in the process of fault diagnosis. This paper aims to present combined auto-associative neural network (AANN) and K-nearest neighbor (KNN) based noise reduction and fault detection method for a gas turbine engine application. The performance of the developed method has been evaluated using data obtained from a model simulation. The test results revealed that the developed hybrid method is more effective and reliable than the conventional methods for the fault detection of the gas turbine engine with negligible false alarms and missed detections.

1 Introduction

Performance deterioration of a gas turbine engine strongly affects its operation. Gas path components are subjected to a very harsh operating conditions and environmental contaminants that may significantly degrade their performance [1]. Gas path component's performance deterioration can be expressed in terms of performance parameter (such as compressor flow capacity, compressor isentropic efficiency, turbine flow capacity and turbine isentropic efficiency) changes [2]. These changes would result in changes in gas path measurements (temperature, pressure, fuel flow, and shaft speed). Degradations are manifested by trend changes in gas path measurements from their baselines. Thus, in order to maintain a reliable engine operation, the abnormal operating conditions has to be detected successfully. However, as measurement uncertainties influence the detection accuracy, an effective data filtration prior to the fault detection task is also crucial.

Kalman filter (KF) is one of the conventional model based approaches used to undertake noise reduction and data validation tasks. For instance, multiple KF models were used for

* Corresponding author: amare.desalegn.fentaye@mdh.se

sensor and actuator fault detection and isolation (FDI) together with a component fault detection in an aircraft engine by Kobayashi and Simon [3]. A single FDI effectiveness of KF, artificial neural network (ANN) and hybrid ANN was investigated by Volponi et al. [4]. Although three of them could isolate more than 90% of the fault signatures, the result from the hybrid system was the best. Pourbabae et al. [5] also proposed a multi-model hybrid KF based engine sensor fault detection, isolation and identification (FDII) method. However, the effectiveness of the KF based techniques, particularly the linear filters, were highly influenced by the nonlinearity of the data and the level of measurement uncertainties [6]. To address the issue of nonlinearity, several authors like Lu et al. [7, 8] utilized the modified version of the linear KF. On the other hand, regardless of its limitation to deal with measurement uncertainty, gas path analysis (GPA) has also been widely utilized for engine performance diagnostics [9-12]. As far as artificial intelligence (AI) methods are concerned, Kramer [13, 14] introduced an AANN based technique for sensor validation and correction that is coping with the nonlinearity of the data. He used the network residuals to detect and estimate sensor faults. In the same year, Lu et al. [15] implemented an AANN for a gas turbine engine data screening and validation, and promising results were reported. Ganguli [16] also employed Median filters (MFs) and Fuzzy Logic (FL) for engine data filtering and fault detection. The MF scheme was used to filter measurement noises and outliers, while the FL based method used to detect engine component faults based on gas path measurement residuals. Recently, more attention has been given to ANNs due to their performance in noise reduction and FDI [17-22]. On the other hand, application of KNNs for a gas turbine diagnostic is also started lately. About a decade ago, the potential of KNN classifier for a two-spool turbofan engine fault classification was evaluated and compared with support vector machine (SVM), principal component analysis (PCA), and probabilistic neural network (PNN) methods by Donat et al. [23]. On another study, Loboda [24] compared the gas turbine fault classification performance of KNN, multilayer perceptron (MLP), and PNN. In general, combining two or more methods showed different advantages in gas turbine diagnostics than using the methods individually [25, 26].

In this paper, a combined AANN and KNN method is utilized for a gas turbine noise reduction and fault detection. The effectiveness of the method has been tested by applying on a two-shaft stationary gas turbine engine, which is thermodynamically similar to LM2500. The required training and test data samples are generated by simulating the performance model of the case engine under steady state operating conditions. In order to take into account measurement noises, Gaussian noise is added to the data associated with each measurement parameter.

2 Method

The proposed noise reduction and fault detection technique as shown in Figure 1 is developed by combining AANN and KNN approaches. The AANN structure is used to minimize the measurement noise and the KNN structure to distinguish the fault and no-fault data classes. Using the AANN structure prior to the fault detection may help to enhance the detection effectiveness. Because one of the major sources of a wrong detection is the occurrence of a high-level sensor noise due to the harsh engine operating environment. For an effective and practically acceptable fault detection, the noise effect should thus be minimized. During real-time application, first, the observed measurements will pass through the AANN structure for de-noising, next the residuals will be generated by taking the difference between the AANN output and the established baseline, then lastly those residuals will be fed into the classifier to decide whether the engine is healthy or unhealthy.

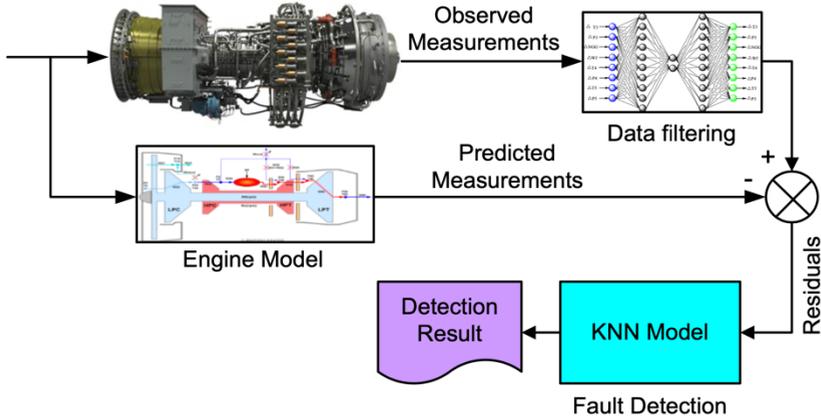


Figure 1. Schematic diagram of the noise filtering and fault detection scheme.

&1 AANN-based noise reduction

AANNs are feed forward types of neural networks (NN) that try to learn an approximation to the identity function in an unsupervised manner so as to give an output that is similar to its input [14]. Figure 2 shows the general topology of a five-layer AANN. The overall structure has two parts, compression and decompression. During the forward propagation, the compression part takes the high dimensional input information and compresses it into a smaller dimension with capturing important features. The decompression part then takes the compressed data from the bottleneck layer and tries to reconstruct the original information. It composed of an input layer, a mapping layer, a bottleneck layer, a de-mapping layer, and an output layer. The output layer is of the same size as the input layer. The bottleneck layer is located in the middle of the network, with the smallest number of neurons, where the important features of the data captured. The number of neurons in the bottleneck layer is determined based on the principal components required to extract the sufficiently important features of the input data. The sizes of mapping and de-mapping layers depend on the complexity of the network functions that can be generated during learning. According to the universal approximation theorem, the activation function for the mapping and de-mapping layers should be non-linear, whereas the remaining layers can use linear activation function. More details on AANN can be found in [13].

During training, noise-corrupted data is used as an input and noise-free data as an output. Using noisy data on both sides reduces the effectiveness of the de-noising network [27]. The network is then tried to reduce the noise and provide an output, which is as close as the desired noise-free output.

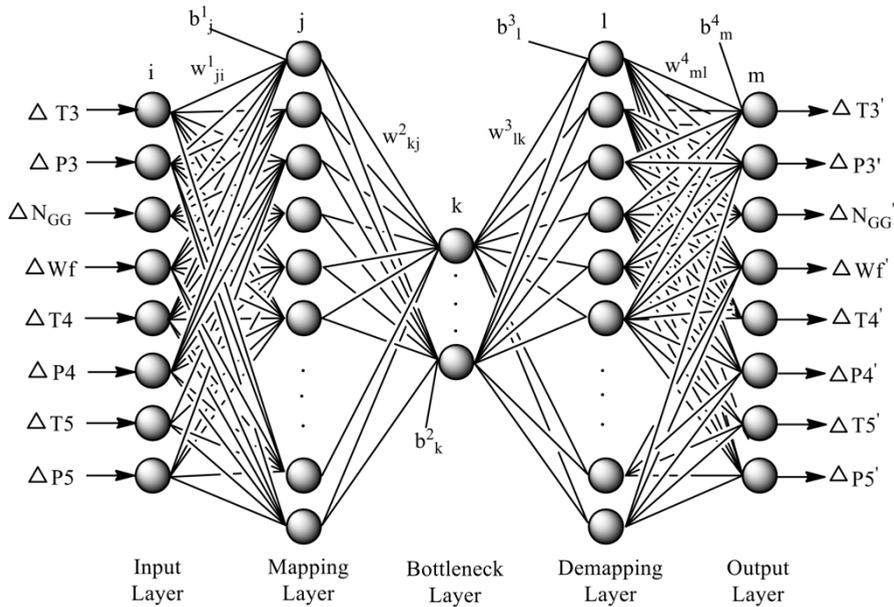


Figure. 2. The general five layer AANN architecture with 8 input/output parameters.

&2 Engi ne fault detection using KNN

Fault detection is the first and the very important step in fault diagnostics that used to decide whether the operating system is healthy or faulty. A detection system with a minimum false alarm and missed detection is thus very essential. In the current work, the fault detection problem is considered as a binary class classification problem. A KNN based classification technique is utilized to distinguish the healthy and faulty data classes of the engine. The process of developing a KNN classifier requires two steps: building the model using the training dataset and testing its performance on the test dataset. New data points are predicted based on their distance from the neighbor classes. Figures 3 and 4 show the conceptual model of a 2-class KNN classifier with different K values and the complete procedure used to develop the model, respectively.

The brief description of the major steps involved in the process of developing the classifier is given as follows:

Step 1. The feature vectors associated with the healthy and faulty classes are prepared. The whole data is then divided into two subsets, one to construct the model and the rest to test its classification accuracy. The class labels are defined for the training dataset and saved in the database together with the unlabeled test feature vector.

Step 2. The model is constructed based on the known features and labels of the training dataset to visualize the two classes.

Step 3. The unlabeled test data samples are used to evaluate the classification performance of the model. In order to determine the class where the test data belongs to, the distances between the new data points and all the training data points need to be computed. Euclidian, Minkowski, Manhattan, Cosine distance, and Correlation distance are the most commonly used distance functions for KNN.

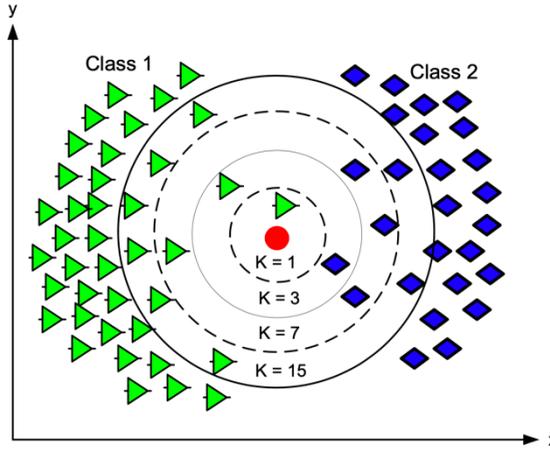


Figure 3. Conceptual model of KNN based classification.

Step 4. The computed distances corresponding to each class were sorted in an ascending order, and the first k nearest neighbors were selected based on their distance. When k is 1, the class of the new data point will be determined based on the first nearest point. It is preferable when a very large training data is used. Too big k values may result over generalization, while too small k values usually cause overfitting and unstable decision boundaries. k should always be an odd number to avoid tie in the vote.

Step 5. After selecting the appropriate k nearest neighbors, the class of the test data can be decided according to the majority vote, by counting the number of nearest neighbors belonging to each class or weighted-sum voting. If weighted distances are used to make the vote, the degree of contribution of the nearest neighbors should be computed applying different distance weighting functions. The weight value of each neighbor is added to the total weight of the corresponding class and the class with the largest weight value wins. One of the following distance functions can be used to weight the contribution of neighbors: equal (weight = 1), inverse (weight = 1/d), and square inverse (weight = 1/d²).

Step 6. In order to select the most appropriate classification model, the cross-validation technique was applied. There are different cross-validation techniques available for machine learning algorithms. Two of the most widely used techniques are holdout cross-validation and k-fold cross-validation. The holdout cross-validation is accomplished by dividing the sample dataset into two sub-sets, training and test datasets. Whereas, in the latter case, the sample data is divided into k number of equal-sized groups or folds. The training is done on the k-1 folds while remaining fold is left for validation. The latter approach was applied in the current work.

Step 7. The detection accuracy was then evaluated based on the following Equations:

$$RTP = \frac{TPs}{TPs + FN_s} \times 100\% \tag{1}$$

$$RFP = \frac{FP_s}{FP_s + TN_s} \times 100\% \tag{2}$$

$$Missed\ Detection = [100 - RTP]\% \tag{3}$$

$$False\ Alarm = [100 - RFP]\% \tag{4}$$

$$ODA = \frac{TP_s + TN_s}{TP_s + FP_s + FN_s + TN_s} \times 100\% \tag{5}$$

where:

- RTP = rate of true positive, RTN = rate of true negative, RFP = rate of false positive, RFN = rate of false negative, and ODA = overall detection accuracy.
- TPs (True positives) = number of correctly detected faulty patterns.
- FPs (False positives) = number of no-fault patterns wrongly detected as faulty.
- TNs (True Negatives) = number of correctly detected no-fault patterns.
- FNs (False Negatives) = number of fault patterns wrongly detected as no-fault.

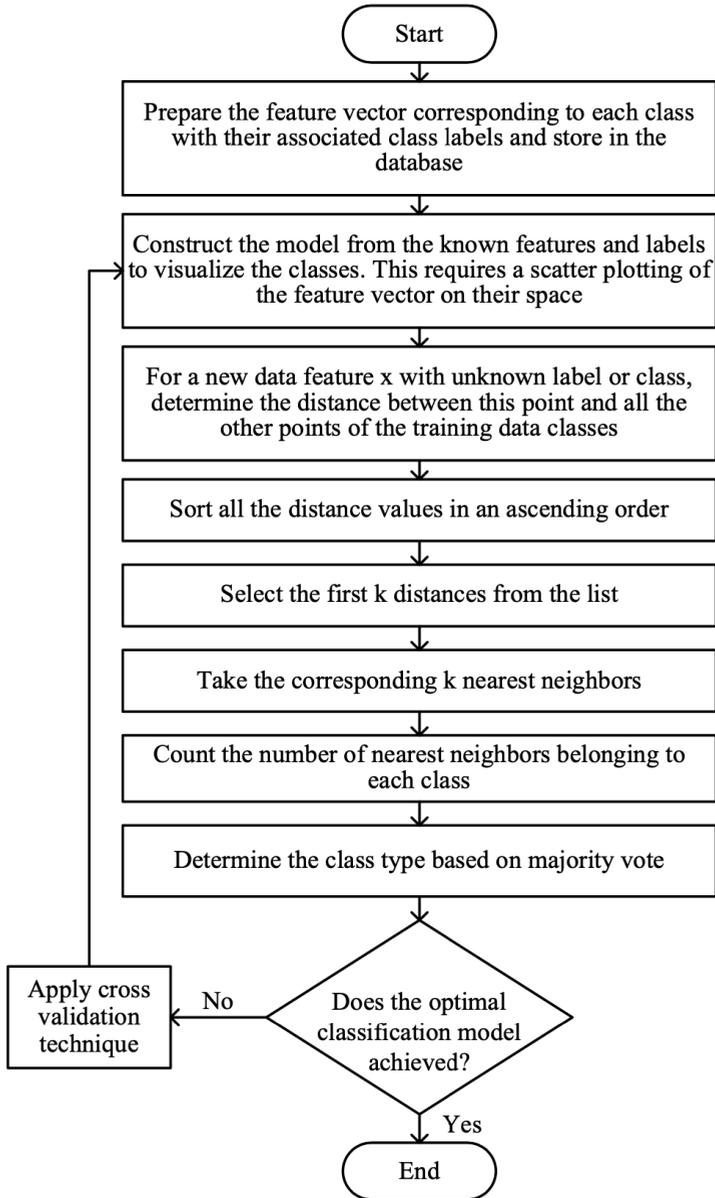


Figure. 4. Flowchart of the majority vote based KNN classification procedure.

Case studies

Seven component faults in a two-shaft industrial gas turbine engine were considered as shown in Table 1. In order to generate the required data for training and test, at first, the performance prediction model was developed using a simulation program, tuned to represent a gas turbine which is thermodynamically similar to GE LM2500 engine. It consists of a 16-stage axial compressor (C), a 2-stage compressor turbine (CT) and a 6-stage power turbine (PT) that can produce 23.3 MW power output. Figure 5 illustrates the schematics of the basic components along with the gas-path measurements used for the detection.

Table 1. Description of the considered component faults

Fault Type	Description
Fault 1	Compressor fouling (CF)
Fault 2	CT erosion (CTE)
Fault 3	PT erosion (PTE)
Fault 4	CF and CTE
Fault 5	CF and PTE
Fault 6	CTE and PTE
Fault 7	CF, CTE, and PTE

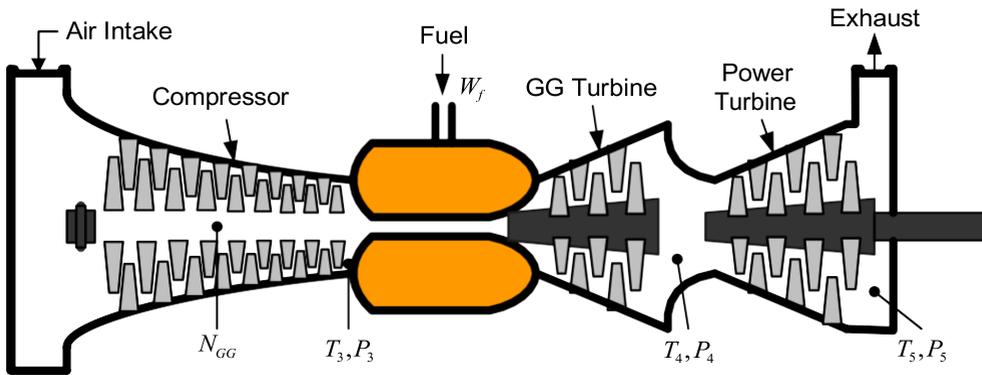


Figure 5. Two-shaft gas turbine schematics.

The necessary clean condition data is generated taking into account ambient condition variations ranging from $-45\text{ }^{\circ}\text{C}$ to $50\text{ }^{\circ}\text{C}$ for ambient temperature and from 0 to 3% for ambient pressure. In order to generate the test data that can represent a deteriorated engine operating condition, artificial fault case patterns derived from the assumed seven component fault scenarios were implanted into the model and simulated. Overall 20955 data samples (7620 no-fault and 13335 fault data samples (1905 samples drawn from each fault type) were used to train and test the system. For compressor fouling, flow capacity losses ranging from 0 to -5% and isentropic efficiency losses ranging from 0 to -2.5% were considered in different ratios. Similarly, for turbine erosion, flow capacity changes ranging from 0 to 5% and isentropic efficiency changes ranging from 0 to -2.5% were considered. In the end, in order to satisfy the real measurement situations, a zero mean Gaussian white noise is superimposed to the data. Table 2 presents the maximum noise-level considered in standard deviation values of each gas path measurement.

Table 2. The sensors' noise standard deviations in %

Sensor	T3	P3	N _{GG}	Wf	T4	P4	T5	P5
Noise ($\pm\sigma$)	0.4	0.25	0.05	0.5	0.4	0.25	0.4	0.25

Results and discussion

3.1 Noise reduction

In order to select the optimal AANN structure, hidden layers with different number of neurons were considered (for the mapping and de-mapping layers from 8 to 50 neurons and for the bottleneck layer from 1 to 7 neurons). A large number of neurons were required in the mapping and de-mapping layers because these layers are quite complex non-linear functions. In the training process the learning data sample was divided into three groups: training, validation, and test datasets in different ratios. Taking into account the model complexity and training computational efficiency, the model structure 8:30:4:30:8 was selected as the best structure among the other potential structures. The selected structure was able to provide a noise reduction of over 85% for most of the measurements.

Owing to space limitations only the denoising results for the temperature measurements are presented as shown in Figures 6-8. The difference between noisy input data and network output data refers to the level of noise reduction by the network. In general, one can see that a significant level of noise has been removed from the data corresponding to each measurement although the effectiveness of the network is not uniform between different measurements. Eq. (6) shows the formula that has been used to compute the percentage noise reduction for each parameter.

$$\text{Noise reduced (\%)} = \frac{1}{N} \sum_{i=1}^N \frac{|(X_n)_i - (X_o)_i|}{|(X_n)_i - (X_t)_i|} \times 100 \quad (6)$$

Where X_n is the noisy data input, X_o is the de-noised AANN output, X_t is the target, i is the i^{th} sample data, and N is the number of samples.

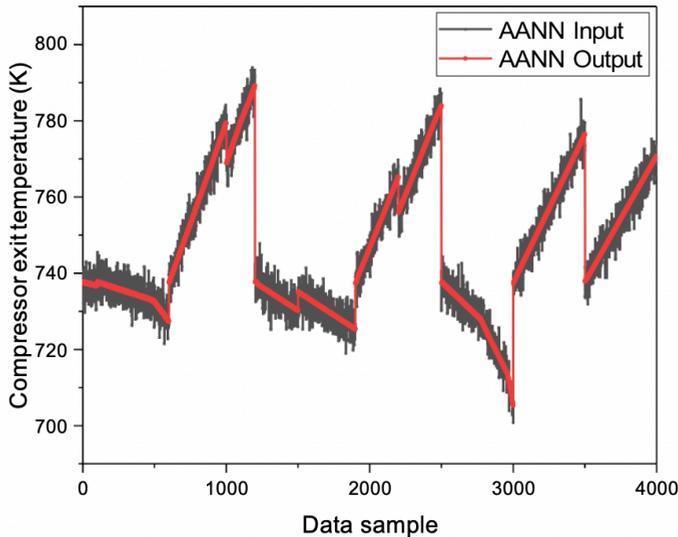


Figure 6. Noise removal results of 4000 data examples for compressor exit temperature.

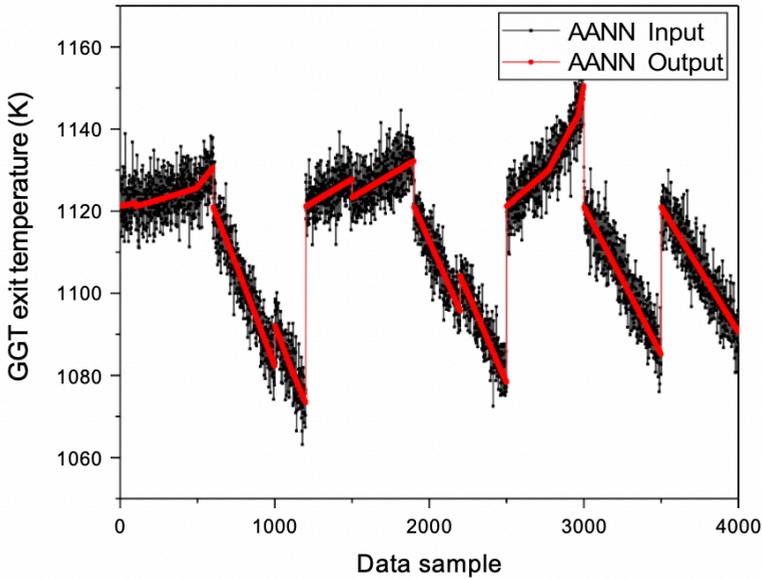


Figure 7. Noise removal results of 4000 data examples for gas generator turbine exit temperature.

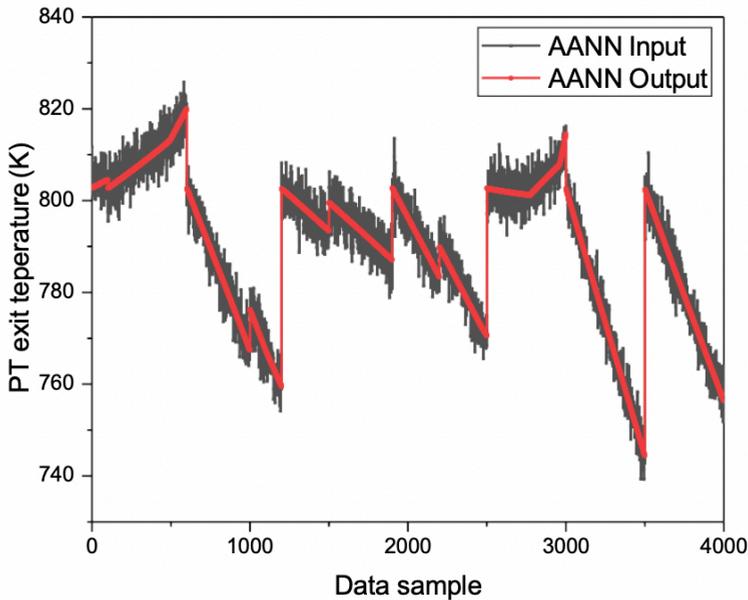


Figure 8. Noise removal results of 4000 data examples for power turbine exit temperature.

In order to see the influence of noise level on the effectiveness of the de-noising network, noise values of $\pm 1\sigma$, $\pm 2\sigma$, and $\pm 3\sigma$ were considered. Table 3 shows the percentage noise reductions obtained. In fact, as the level of the noise increases, the magnitude of the noise to be reduced from each measurement decreases. Nevertheless, for all the measurements a significant amount of noise has been removed. Since the AANN output is the input of the KNN classifier, the remaining percentage of noise in the input data helps to enhance the flexibility/robustness of the classifier when unexpected data features come into the system.

Table 3. Effectiveness of the AANN at different noise-levels.

Noise Level	Percentage noise reduction								
	T3	P3	NgG	Wf	T4	P4	T5	P5	Average
$\pm 1\sigma$	87.6	85.8	86.7	84.5	87.5	84.6	86.7	86.3	86.21
$\pm 2\sigma$	81.5	80.7	82.8	81.3	81.4	80.6	79.3	81.5	81.14
$\pm 3\sigma$	77.3	76.4	73.7	74.9	77.5	73.5	74.5	77.1	75.61

(. & : U`hXYhVWjcb

In this section, the fault detection performance of the KNN technique was evaluated using the generated performance data that can represent both the healthy and faulty engine operating conditions. In order to select the appropriate classifier model, effects of the major controlling factors such as the distance function, the number of nearest neighbors, and training data size were investigated. A total of 20955 patterns (7620 no-fault and 13335 fault patterns) were used to train and test the model. The healthy operating conditions were assigned as positive (+), while the faulty conditions were assigned as negative (-). The detection accuracy (i.e., how accurately the two data classes are distinguished) was computed based on the standard detection decision matrix approach.

To analyse the effectiveness of different KNN functions and select the most suitable function for our data, medium KNN, coarse KNN, cosine KNN, cubic KNN, and weighted KNN functions were considered. These classifiers are different, to some extent, in their classification algorithm, computational speed, memory requirement, flexibility, and interpretability. Table 4 illustrates the classification results obtained. The maximum accuracy is provided by the Fine KNN model (98.3%), while the minimum by Coarse KNN (93.9%). The classification accuracy using Weighted KNN and Cosine KNN is 0.6% and Medium KNN and Cubic KNN is 1% lower than the maximum accuracy. As per this table, the Fine KNN was chosen for further investigation. Since a very low-level faults were also considered, as can be seen in the table, some patterns were recognized by the classifier as a fault-free patterns. An attempt to minimize the number of missed detection may increase false detections, and vice versa. However, since too early maintenance is not economical, it is better to have missed detections than false alarms.

Table 4. Classification performance of different KNN functions.

KNN Type	TPs	FPs	TNs	FNs	MD	FA	ODA
Fine KNN	12959	0	7620	375	2.8	0	98.3
Medium KNN	12858	0	7620	477	3.6	0	97.3
Coarse KNN	12294	0	7620	1044	7.8	0	93.9
Cosine KNN	12948	0	7620	387	2.9	0	97.7
Cubic KNN	12872	0	7620	463	3.6	0	97.3
Weighted KNN	12947	0	7620	388	2.9	0	97.7

4.2.1 Effect of distance matrix and weight

In order to investigate whether the classification effectiveness is influenced by the distance function and the distance weight or not, five distance matrices with three different weight functions were considered. For this purpose, the Fine KNN model, the KNN with better classification performance as reported in Table 4, was selected. Table 5 presents the overall detection accuracy of this model for the possible combinations of the distance matrix and weight functions. As seen in this result, the influence of the distance matrix and the weight

function was insignificant. Therefore, the Euclidian function was chosen as the distance matric for our model, and it is also one of the most commonly used functions in the literature.

Table 5. Effect of the distance matric and distance weight on the classification performance.

Distance Matric	Distance Weight	ODA
Euclidian	Equal	98.3
	Inverse	98.3
	Square inverse	98.3
Minkowski	Equal	97.7
	Inverse	98.0
	Square inverse	98.1
Manhattan	Equal	98.2
	Inverse	98.3
	Square inverse	98.3
Cosine	Equal	98.2
	Inverse	98.2
	Square inverse	98.2
Correlation	Equal	97.8
	Inverse	98.0
	Square inverse	98.1

4.2.2 Effect of training data size

The accuracy of machine learning methods is influenced by the number of samples used for training. Small training data samples may cause overfitting and high prediction errors. Conversely, regardless of memory and computational requirements, for practical concerns, a more training data size often provides a better testing accuracy. However, after a certain limit, the effect of increasing the amount of training data samples will be negligible [28]. A model that is overfitting will be less good for a new dataset than a model with a good fit. One way to have a look into generalizability is to have a training set and test set. In this paper, one of the most widely used approaches known as the k fold cross-validation technique was applied. According to this approach, the learning dataset is divided into k equal groups. During the learning process, the training is done on the k-1 data groups and test on the rest group, and so on for all possible combinations. Accordingly, to examine the sensitivity of the KNN classifier model towards the number of folds, six different cases (5, 10, 15, 20, 25 and 30 folds) were taken into account. Table 6 presents the fault detection results obtained. In general, since the number of cross-validation samples was high enough, in all the six cases the classifier provided over 98% accuracy. Consequently, the 10 fold cross-validation technique, which is also the most commonly used validation approach in different KNN applications [29], has been selected.

Table 6. Effect of training data size on the classification accuracy.

Number of folds	Training time (s)	MD	FA	ODA
5	2	2.9	0	98.3
10	2	2.9	0	98.3
15	2	2.9	0	98.3
20	3	2.9	0	98.3
25	2	2.9	0	98.3
30	4	2.9	0	98.3

To summarize, as we can see from the sensitivity analysis, the effect of most of the controlling factors on the classification performance of the KNN model was insignificant. The fault detection accuracy mainly depends on the relationship between the low-level faults

and the measurement noise values. As a matter of fact, when the fault signatures associated with these conditions are similar, there will be a possibility of miss detection and/or false alarms. However, in the present work, due to the de-noising process prior to the fault detection, a reasonably high-degree of accuracy was obtained.

5 Conclusions

In this paper, a combined AANN and KNN technique is devised for a gas turbine engine noise reduction and fault detection application. The AANN structure is used to minimize the measurement noise prior to the fault detection process. Whereas the KNN technique is used to determine whether the engine is healthy or unhealthy. The data required to demonstrate and evaluate the performance of the developed method is generated from a model simulation, tuned to represent a two-shaft gas turbine which is thermodynamically similar to LM2500 engine. As gas turbine physical problems, compressor fouling and turbine erosion, which are the major and usual causes of gas turbine gas path component(s) performance deterioration, has been considered. White Gaussian noise is added to the performance data in order to account the measurement noise. The test results showed that the AANN algorithm was able to reduce over 85% of the measurement noise while the KNN classifier was able to detect 98.3% of the faulty conditions successfully. The proposed method can, therefore, be employed for gas turbine gas path fault detection.

Nomenclature

AANN	Autoassociative neural network
ANN	Artificial neural network
C	Compressor
CF	Compressor fouling
CT	Compressor turbine
CTE	Compressor turbine erosion
EKF	Extended Kalman filter
FD	False detection
FDI	Fault detection and isolation
FDII	Fault detection isolation and identification
FL	Fuzzy logic
FN	False negative
FP	False positive
GPA	Gas path analysis
KF	Kalman filter
MD	Missed detection
MFs	Membership function
MLP	Multilayer perceptron
ODA	Overall detection accuracy
PNN	Probabilistic neural network
PT	Power turbine
PTE	Power turbine erosion

RFP	Rate of false positive
RMS	Root mean square
RTP	Rate of true positive
SVM	Support vector machine
TN	True negative
TP	True positive

Acknowledgments

This research was partially funded by the Knowledge Foundation (KKS) under the project DIAGNOSIS.

References

1. A. D. Fentaye, A. T. Baheta, S. I. Gilani, and K. G. Kyprianidis, "A Review on Gas Turbine Gas-Path Diagnostics: State-of-the-Art Methods, Challenges and Opportunities," *Aerospace*, vol. 6, p. **83**, (2019).
2. Y. Qingcai, S. Li, Y. Cao, and N. Zhao, "Full and Part-Load Performance Deterioration Analysis of Industrial Three-Shaft Gas Turbine Based on Genetic Algorithm," In Proceedings of the ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition, Harbin, China, (13–17 June 2016); p. V006T005A016.
3. T. Kobayashi and D. L. Simon, "Evaluation of an enhanced bank of Kalman filters for in-flight aircraft engine sensor fault diagnostics," *J ENG GAS TURB POWER*, vol. 127, pp. 497-504, (2005).
4. A. J. Volponi, H. DePold, R. Ganguli, and C. Daguang, "The Use of Kalman Filter and Neural Network Methodologies in Gas Turbine Performance Diagnostics: A Comparative Study," *J ENG GAS TURB POWER*, vol. 125, pp. 917-924, (2003).
5. B. Pourbabaee, N. Meskin, and K. Khorasani, "Sensor Fault Detection, Isolation, and Identification Using Multiple-Model-Based Hybrid Kalman Filter for Gas Turbine Engines," *IEEE Transactions on Control Systems Technology*, (2015).
6. R. Ganguli, *Gas Turbine Diagnostics: Signal Processing and Fault Isolation*: CRC press: Boca Raton, FL, USA, (2012).
7. F. Lu, Y. Chen, J. Huang, D. Zhang, and N. Liu, "An integrated nonlinear model-based approach to gas turbine engine sensor fault diagnostics," *Proc IMechE Part G: J Aerospace Engineering*, vol. **228**, pp. 2007-2021, (2014).
8. F. Lu, T. Gao, J. Huang, and X. Qiu, "A novel distributed extended Kalman filter for aircraft engine gas-path health estimation with sensor fusion uncertainty," *Aerospace Science and Technology*, vol. **84**, pp. 90-106, (2019).
9. Q. Yang, S. Li, and Y. Cao, "Multiple model-based detection and estimation scheme for gas turbine sensor and gas path fault simultaneous diagnosis," *JMST*, vol. 33, pp. 1959-1972, (2019).
10. V. Zaccaria, M. Stenfelt, I. Aslanidou, and K. G. Kyprianidis, "Fleet Monitoring and Diagnostics Framework Based on Digital Twin of Aero-Engines," in ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition, Oslo, Norway (June 11-15, 2018).
11. M. Rahman, V. Zaccaria, X. Zhao, and K. Kyprianidis, "Diagnostics-Oriented Modelling of Micro Gas Turbines for Fleet Monitoring and Maintenance Optimization," *Processes*, vol. **6**, p. 216, (2018).

12. D. Zeng, D. Zhou, C. Tan, and B. Jiang, "Research on Model-Based Fault Diagnosis for a Gas Turbine Based on Transient Performance," *Applied Sciences*, vol. **8**, p. 148, (2018).
13. M. A. Kramer, "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," *AIChE Journal*, vol. **37**, p. 233, (1991).
14. M. A. Kramer, "Neutral network applications in chemical engineeringAutoassociative neural networks," *COMPUT CHEM ENG*, vol. **16**, pp. 313-328, 1992/04/01 (1992).
15. P. J. Lu and T. C. Hsu, "Application of autoassociative neural network on gas-path sensor data validation," *J PROPUL POWER*, vol. **18**, pp. 879-888, (2002).
16. R. Ganguli and B. Dan, "Trend Shift Detection in Jet Engine Gas Path Measurements Using Cascaded Recursive Median Filter With Gradient and Laplacian Edge Detector," *J ENG GAS TURB POWER*, vol. **126**, pp. 55-61, (2004).
17. S. Sina Tayarani-Bathaie and K. Khorasani, "Fault detection and isolation of gas turbine engines using a bank of neural networks," *J PROCESS CONTR*, vol. **36**, pp. 22-41, 12// (2015).
18. M. Amozegar and K. Khorasani, "An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines," *Neural Networks*, vol. **76**, pp. 106-121, 4// (2016).
19. Z. N. Sadough Vanini, N. Meskin, and K. Khorasani, "Multiple-Model Sensor and Components Fault Diagnosis in Gas Turbine Engines Using Autoassociative Neural Networks," *J ENG GAS TURB POWER*, vol. **136**, p. 091603, (2014).
20. I. Loboda and M. A. Olivares Robles, "Gas turbine fault diagnosis using probabilistic neural networks," *INT J TURBO JET ENG.*, vol. **32**, pp. 175-191, (2015).
21. A. D. Fentaye, A. T. Baheta, and S. I. U.-H. Gilani, "Gas turbine gas-path fault identification using nested artificial neural networks," *Aircraft Engineering and Aerospace Technology*, vol. **90**, pp. 992-999, (2018).
22. D. Amare, T. Aklilu, and S. Gilani, "Gas path fault diagnostics using a hybrid intelligent method for industrial gas turbine engines," *J. Braz. Soc. Mech. Sci. & Eng*, vol. **40**, p. 578, (2018).
23. W. Donat, K. Choi, W. An, S. Singh, and K. Pattipati, "Data visualization, data reduction and classifier fusion for intelligent fault diagnosis in gas turbine engines," *J ENG GAS TURB POWER*, vol. **130**, p. 041602, (2008).
24. I. Loboda, "*Gas turbine fault classification using probability density estimation*," in *Proceedings of ASME Turbo Expo: Turbine Technical Conference and Exposition, GT2014-27265*, Germany, Dusseldorf, (2014).
25. A. Fentaye, V. Zaccaria, M. Rahman, M. Stenfelt, and K. Kyprianidis, "*Hybrid model-based and data-driven diagnostic algorithm for gas turbine engines*," *Proceedings of ASME Turbo Expo 2020: Turbomachinery Technical Conference and Exposition, GT2020-14481*, (2020).
26. A. D. Fentaye, S. I. Ul-Haq Gilani, A. T. Baheta, and Y.-G. Li, "Performance-based fault diagnosis of a gas turbine engine using an integrated support vector machine and artificial neural network method," *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. **233**, pp. 786-802, (2019).
27. P. J. Lu, M. C. Zhang, T. C. Hsu, and J. Zhang, "An evaluation of engine faults diagnostics using artificial neural networks," *J ENG GAS TURB POWER*, vol. **123**, pp. 340-346, (2001).

28. R. Bettocchi, M. Pinelli, P. R. Spina, and M. Venturini, "Artificial Intelligence for the Diagnostics of Gas Turbines—Part I: Neural Network Approach," *J ENG GAS TURB POWER*, vol. **129**, pp. 711-719, (2006).
29. S. Borra and A. D. Ciaccio, "Measuring the prediction error. A comparison of cross validation, bootstrap and covariance penalty methods," *Computational Statistics & Data Analysis*, vol. **54**, pp. 2976-2989, (2010).