The Aeronautical Journal (2026), **130**, pp. 2636–2661
doi:10.1017/aer.2026.10172

RESEARCH ARTICLE

An improved deep autoencoder framework for aviation piston engine unsupervised anomaly detection considering the real-life data characteristics

Guo Li[1], Tongge Xu[1] and Shuiting Ding[2]

[1]Beihang University, China
[2]Civil Aviation University of China, China
Corresponding author: Tongge Xu; Email: xutongge114@outlook.com

Received: 10 August 2025; **Revised:** 8 April 2026; **Accepted:** 8 April 2026

Keywords: aviation piston engine; deep autoencoder; real-life data; safety; unsupervised anomaly detection

Abstract

Anomaly detection in piston engines plays a pivotal role in maintaining the continuous airworthiness and operational safety of general aviation aircraft. However, real-life data present several challenges, including diverse operating conditions, intermixed normal and abnormal samples and complex fault phenomena, all of which complicate model training and manual labeling tasks. To address the challenges, an improved deep autoencoder-based framework under multiple working conditions (IMDAEF) is proposed to enhance the accuracy of anomaly detection and the performance evaluation process. First, in the proposed method, a clustering-guided approach is introduced for working condition identification. Piston engine parameters are used for K-means clustering, and aircraft flight phase information is incorporated to fine-tune the results. Second, during the deep autoencoder training process, a sample selection mechanism and an equivalent threshold method are conducted, considering the presence of outliers in normal operation data. Finally, during the model evaluation stage, recognising the complexity of real-life fault phenomena that hinder manual labeling, an outlier region division and selection mechanism is proposed to re-annotate subtle fault occurrences in the fault datasets, thereby improving model evaluation accuracy. Results from four representative fault datasets confirm that the proposed framework maintains strong detection performance against complex faults. Moreover, the experiments demonstrate its ability to mitigate the negative impact of real-life data characteristics while meeting real-time performance requirements, offering valuable insights for future research.

Nomenclature

AE	autoencoder
APE	aviation piston engine
CWCI	clustering method-guided working condition identification
DAE	deep autoencoder
DRE	dimensionless reconstruction error
ECOD	empirical cumulative distribution-based outlier detection
FN	false negative
FP	false positive
GA	general aviation
IMDAEF	improved deep autoencoder-based framework under multiple working conditions
LAD	local abnormal degree
LOF	local outlier factor
MDAE	deep autoencoder model for multiple working conditions
OAD	overall abnormal degree
OCSVM	one-class support vector machine

© The Author(s), 2026. Published by Cambridge University Press on behalf of Royal Aeronautical Society.

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

OR	outlier region
RE	reconstruction error
SVM	support vector machine
TN	true negative
TP	true positive
WCI	working condition identification

1.0 Introduction

Although the general aviation (GA) industry has been growing rapidly in China in recent years, the safety issue of GA aircraft still remains a challenge. As one of the most important propulsion systems used in GA aircraft, the performance state of an aviation piston engine (APE) is closely related to aircraft operation safety. In order to better manage APE health, an anomaly detection technique is of great practical significance in the process, which can promptly detect patterns in monitored performance data that do not conform to expected behaviour [1], thus ensuring efficient operation of APEs and preventing costly unplanned maintenance. Therefore, it is of great value to study how to improve the detection performance for APE anomaly detection.

In spite of the widespread use of APEs in GA, they are relatively less studied, and most literature focuses on other aero-engines, such as turbojets, turbofans, turboprops and turboshafts [2]. The methods used in aero-engine anomaly detection can be mainly divided into two categories: model-based methods and data-based methods [3]. The accuracy of model-based methods [4–7] mainly lies in the physical or mathematical expressions adopted in the process, which may not be proper for the real operation process in complex systems such as aero-engines. On the other hand, data-driven methods [8–10] do not require precise mathematical mechanism models and use a tremendous amount of data collected by multiple sensors for anomaly detection, which has a wider application range and better prospects.

With the rapid development of artificial intelligence, machine-learning models are increasingly dominating data-driven methods, which can be further classified into supervised and unsupervised learning. Supervised learning methods primarily involve classification models trained on labeled datasets. These datasets are typically sourced from well-labeled and verified public repositories, where abnormal samples are relatively abundant, contributing to the strong performance of supervised learning models. Nevertheless, due to the constraints of complex working environments and data collection conditions in other engineering application scenarios, analysts may struggle to obtain high-quality normal datasets and sufficient fault data. Additionally, fault phenomena can be highly complex. All of these aspects make data processing and labeling increasingly difficult, which may render supervised learning models less suitable for APE anomaly detection.

Given the aforementioned characteristics of real-life data, unsupervised learning models offer unique advantages and are regarded as the primary solution for APE anomaly detection. A practical advantage of these methods is their ability to perform anomaly detection with scarce or no labels, making them well-suited for handling the large volumes of real APE data. Traditional unsupervised learning models used in anomaly detection substantially include: distance-based or density-based methods, such as K-nearest neighbour models [11, 12], DBSCAN [13, 14], local outlier factor (LOF) [15]; support vector machine (SVM)-based methods, such as one-class SVM [16]; tree-based methods, such as isolated forest [17, 18], and autoencoder (AE)-based methods [19–21]. Besides, recent research has validated promising approaches such as transformer-based methods [22, 23], normalising flow-based methods [24, 25] and ensemble models [26], which offer distinct advantages for different anomaly detection scenarios. Despite the advancement of artificial intelligence, AEs are still considered to be an effective method for anomaly detection, and there are various modifications of AEs, such as sparse autoencoder [27], stacked autoencoder [28], variational autoencoder [29] and denoising autoencoder [30]. These types of AEs introduce different constraints and regularisation terms in the encoding–decoding processes to further improve the performance and generalisation ability of the methods. In general, recent advances in AE-based anomaly detection focus on two aspects: (1) architectural improvements to AEs [31] and

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

(2) hybrid models combining AEs with other effective models [32–34]. These developments enhance detection while maintaining computational efficiency for industrial deployment. Benefited from the continued development of AEs, these models can be applied to detect anomalies not only in tabular data, but also in graph data [35] and image/video data [36].

A traditional AE has three layers, including the encoder layer, the hidden layer and the decoder layer. With the development of deep learning skills, the AE model has also been transformed into the deep autoencoder (DAE) model. With a deeper network structure and better generalisation ability, DAEs are proper to solve high-dimensional, unlabeled datasets in engineering applications. The role of DAEs acted in anomaly detection can be mainly divided into two methods: first, the DAEs can be used as the feature extractor to extract lower-dimensional features from the original complex data, and the anomaly detection task is subsequently performed on the new feature space [37, 38]; another way assumes that abnormal instances are harder to reconstruct from reduced space than normal instances [39]. For the latter method, the threshold of the DAE model is of vital importance and determined in the normal datasets with lower reconstruction errors (REs). The unlabeled data points will be deemed as abnormal if their REs are larger than the set threshold. In this study, the second method is mainly adopted.

Challenges still exist in applying APE unsupervised learning with a DAE-based model in real-life scenarios. First, for engineering device anomaly detection, it is crucial to account for changes in working conditions in order to achieve optimal performance. For typical rotating machinery such as electric engines, working conditions could be classified by rotating speeds [40]. For APEs, the datasets are usually derived from the aircraft's real-time digital recording devices, indicating that the working conditions of APEs are closely related to aircraft flight phases. Nicoletta Fala [41] discussed flight phase identification skills, and clustering techniques were verified as effective methods. But there is a lack of research on APE working condition identification (WCI) using clustering-based techniques. Secondly, given that APE datasets from daily normal operations inherently contain outliers due to environmental interference (noise, vibration, et al.) and sensor limitations, such errors can consequently bias the DAE threshold determination and compromise final detection accuracy [42]. In order to reduce the impact of outliers in normal datasets during the DAE training phase, Song Fu et al. [43] developed an R-DAE structure in which a sample selection mechanism was performed during the training stage to screen out outliers in normal datasets; Vincent et al. [44] introduced a modified DAE model named denoising autoencoder, which intentionally injected noise into the input data during training to enhance the robustness; and inspired by robust principal component analysis, Chong Zhou et al. [20] proposed a robust deep autoencoder model, which split the input data and increased the robustness of standard deep autoencoders. However, the feasibility of these methods needs to be further verified on APE datasets. Additionally, it is certain that the DAE threshold determination is affected by the distribution of outliers existed in the normal datasets. Typically, the threshold is calculated through prior knowledge or non-parametric statistical methods such as the 3-sigma principle, control chart and so forth [45, 46]. Nevertheless, these methods are relatively rough and should be improved, considering the specific distribution of outliers in real-life APE datasets. Finally, there are many available public datasets such as the Tennessee Eastman process [47, 48] and the continuous stirred tank reactor [49]. Han et al. [50] also released 57 datasets collected from different fields. These benchmark datasets help researchers to focus on model performance improvement instead of expending a rich price on collecting and labeling data in the real world. But in APE anomaly detection, there is no public dataset, and the labeling task is relatively harder due to the discontinuous and inapparent fault phenomena. More specifically, this type of fault behaviour may easily lead to biased human judgements and even be neglected in the manual labeling process, thus leading to an invalid evaluation of the unsupervised learning model. As yet, there is a lack of relevant discussions on auxiliary annotation of the poorly labeled industrial datasets using a DAE-based framework.

In the face of the difficulties in existing unsupervised anomaly detection methods, inspired by the sample selection mechanism proposed in Ref. [43], this paper developed an improved DAE-based framework under multiple working conditions (IMDAEF) to achieve efficient unsupervised anomaly detection and provide a systematic solution to the complexities of APE real-life monitoring data. First, a clustering

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

method is adopted for the working condition classification based on APE multisensor data. Fine-tuning the technique is further employed to adjust the boundaries for the consistency between APE working conditions and aircraft flight phases. The fundamental model, MDAE, introduced in this paper, integrates the clustering-based method for identifying working conditions with corresponding multiple DAE models. Second, to address the presence of outliers in normal datasets, a sample selection mechanism is employed to remove abnormal samples from the original training set whenever possible, thereby minimising their negative impact on the MDAE training process. Third, in order to reasonably determine the threshold values in MDAE, a simple outlier detection method is applied to the REs calculated by MDAE instead of roughly judging by the 3-sigma principle or other statistic methods. Finally, during the model evaluation process, an outlier region division and selection mechanism is proposed, followed by re-annotation of poorly labeled fault datasets, thereby updating the model performance evaluation metrics at the end of the proposed framework.

The rest of the paper is organised as follows: Section 2 introduces the theoretical background and the details of the proposed framework; Section 3 illustrates the application of the framework on APE datasets; Section 4 presents the results, including a comparison with traditional unsupervised methods, an ablation study and robustness analysis; and finally, Section 5 concludes the paper.

2.0 Method

2.1 Clustering method-guided working condition identification for APEs

Engine condition identification assumes that data samples from the same working condition are closer in the feature space than those from different conditions. Therefore, clustering methods can be used for this purpose. The K-means algorithm is a representative clustering method that has been utilised for working condition classification [51, 52], and its main process can be summarised as follows: first, it randomly selects k objects from the whole dataset to serve as the initial clustering centres. Then, each remaining point is assigned to a cluster whose centroid is closest to it. Subsequently, recalculation of a specific centroid position is conducted for each newly obtained cluster by taking the mean of all data points within that cluster. This process is repeated until a cost function begins to converge. The flowchart is shown in Fig. 1.

However, the accuracy of the K-means algorithm largely depends on the centroid initialisation [53]. It may also fail to be consistent with the real APE working conditions and corresponding aircraft flight phases. Therefore, a clustering method-guided working condition identification (CWCI) is proposed to improve the identification of APE working conditions. First, APE multisensor data is collected, and the K-means algorithm is applied for identification. Next, the clustering results are used to preliminarily determine the boundaries between different working conditions for the sensor variables. Finally, since the datasets are collected from daily aircraft flight training, the boundaries of the various APE working conditions are fine-tuned to align with the actual flight phases.

2.2 Fundamentals of stacked DAE under multiple working conditions (MDAE)

The AE model is a crucial unsupervised learning model proposed by Hinton et al. [54], which is typically composed of the encoder layer, the hidden layer and the decoder layer, as shown in Fig. 2. The theory of the model is presented as follows. First, assuming the input vector $x \in R^n$, it is transformed to vector $z \in R^m$ with lower dimensions after the encoding process. Then the decoding layer is adopted to transform the vector z to the vector x^*. The process is shown in Equations (1)–(2):

$$z = f\left(W^{(1)}x + b^{(1)}\right) \tag{1}$$

$$x^* = g\left(W^{(2)}z + b^{(2)}\right) \tag{2}$$

where $W^{(i)}$ is the weight matrix, $b^{(i)}$ is the bias term of the network, $i = 1, 2$. $f(\cdot)$ and $g(\cdot)$ are activation functions for encoding and decoding layers, such as ReLU, LeakReLU and Tanh, etc.

The objective of an AE is to optimise the model by minimising the discrepancy between the input data x and the reconstructed data x^*. A loss function typically represents this discrepancy. The loss function

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

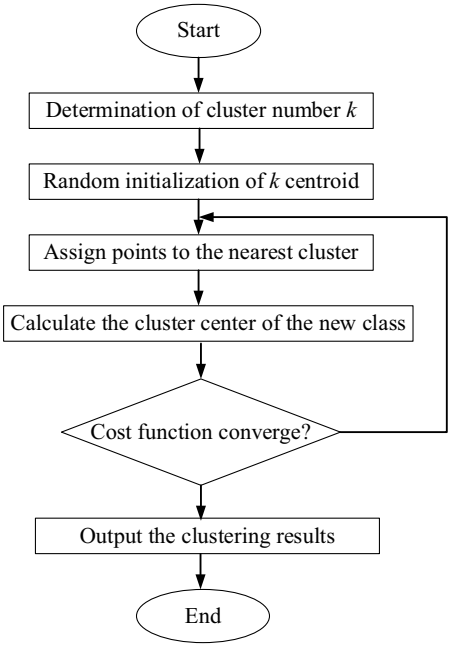


Figure 1. *K-means clustering algorithm flow chart.*

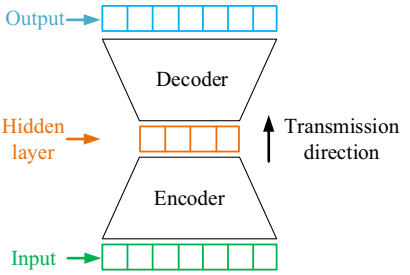


Figure 2. *Structure of an AE model.*

of the AE model is set as the sum of all squared errors between the input vector x and the output vector x^* for the training of the AE model, as shown in Equation (3):

$$L(x, x^*) = \frac{1}{N}\sum_{j=1}^{N} \left\| x_j - x_j^* \right\|^2 \tag{3}$$

where $L(\cdot)$ means the loss function, N indicates the size of the training set and $\|\cdot\|$ represents the Euclidean norm.

In this study, we further enhance the feature extraction capability and training performance of the model by increasing the depth of the AE's hidden layer, called DAE. Furthermore, a layer-wise pre-training technique is utilised to enhance the speed of iterative convergence, and the DAE model trained by this approach is designated as a stacked DAE, as shown in Fig. 3. The detailed pre-training technique is described as follows:

1. A typical three-layer AE is trained, using the mini-batch gradient descent method;
2. The encoder layer and the decoder layer in the first AE model are retained, and the hidden layer is extracted for the second AE training;

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

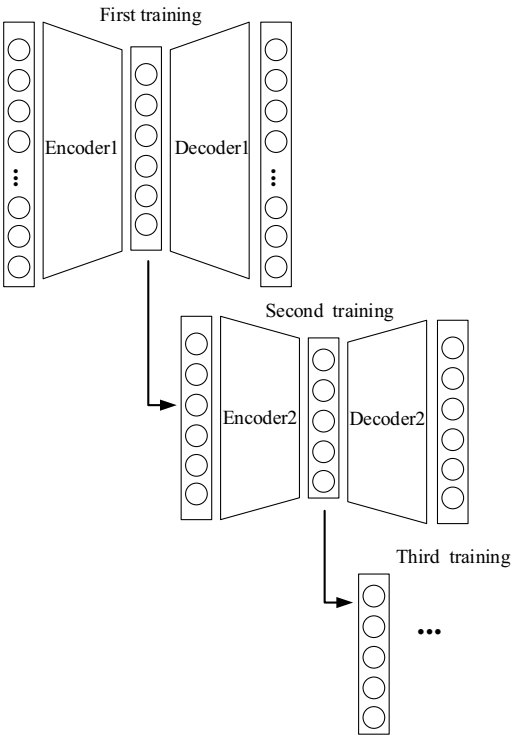


Figure 3. *Schematic diagram of the stack process for DAE training.*

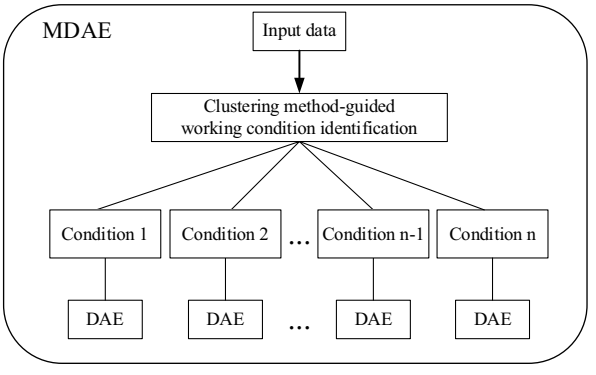


Figure 4. *Schematic diagram of the MDAE model.*

3. Iterate step 2 until the layers are deep enough;
4. Stack up the encoder and decoder layers trained in steps 2 and 3 to construct the final structure of a DAE model. All parameters of the DAE are initialised according to the pre-training results and then fine-tuned using the back propagation algorithm.

Owing to substantial variations in the state parameters of the APE system and diverse criteria for determining normality across various operating conditions, it is imperative to train a DAE specifically for each working condition. This approach enables more accurate detection of anomalies that occur under specific working conditions. The DAE-based model for multiple working conditions is called MDAE, as illustrated in Fig. 4. The structure incorporates a process for identifying working conditions of the input data, and the DAE models are trained specifically for each working condition.

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

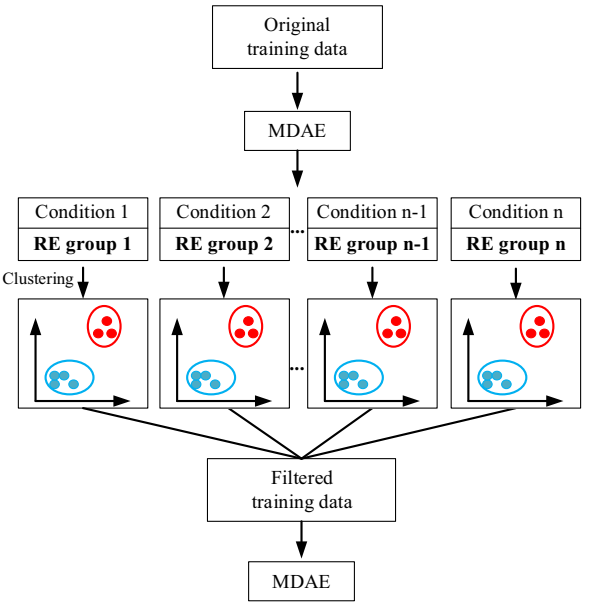


Figure 5. *Sample selection mechanism using MDAE.*

In order to filter the abnormal samples in the real-life datasets, a sample selection mechanism combined with the MDAE model is adopted. The technique is inspired by Ref. [43] and is further applied in MDAE, as shown in Fig. 5. Steps of the technique can be described as follows:

First, the original unlabeled data is used to train MDAE, and the REs of each original sample are calculated and divided into several datasets using the CWCI technique.

Second, the obtained REs in each working condition dataset are further clustered into two groups using the K-means algorithm. For DAE anomaly detection, outliers are harder to reconstruct compared to normal samples. Therefore, a higher RE for a sample increases the likelihood that it is an anomaly. Given the clustering results, the group with higher REs should be removed, and another group is further filtered to get the representative normal training set.

Finally, a new MDAE is trained using the filtered training set. Due to the sample selection mechanism, the new training set contains almost normal points and a few outliers. After the training of MDAE, the differences of REs between normal and abnormal samples are expected to be more significant than those of MDAE trained using original datasets. Apparently, the applied sample selection mechanism raises the training set quality effectively.

2.3 APE anomaly detection based on IMDAEF

The main structure of conducting unsupervised anomaly detection for APEs using IMDAEF is shown in Fig. 6, and the detailed process is described as follows.

First, the real-life multisensor data of a normal APE operation is collected and divided into two training sets, named Training set 1 and Training set 2. The first MDAE model is trained using Training set 1. After that, a sample selection mechanism is applied for data filtering. Points belonging to the cluster with large REs are removed, while samples closer to the centroid in another cluster are further selected to form a new training set. It can be inferred that the new dataset primarily consists of normal points, with virtually no outliers. After that, the MDAE is retrained using the new dataset as input.

Second, Training set 2 is adopted as the input to determine the thresholds of the IMDAEF. Considering the outliers present in the second training set, a simple detection algorithm called empirical cumulative distribution-based outlier detection (ECOD) is used to calculate the specific proportion

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

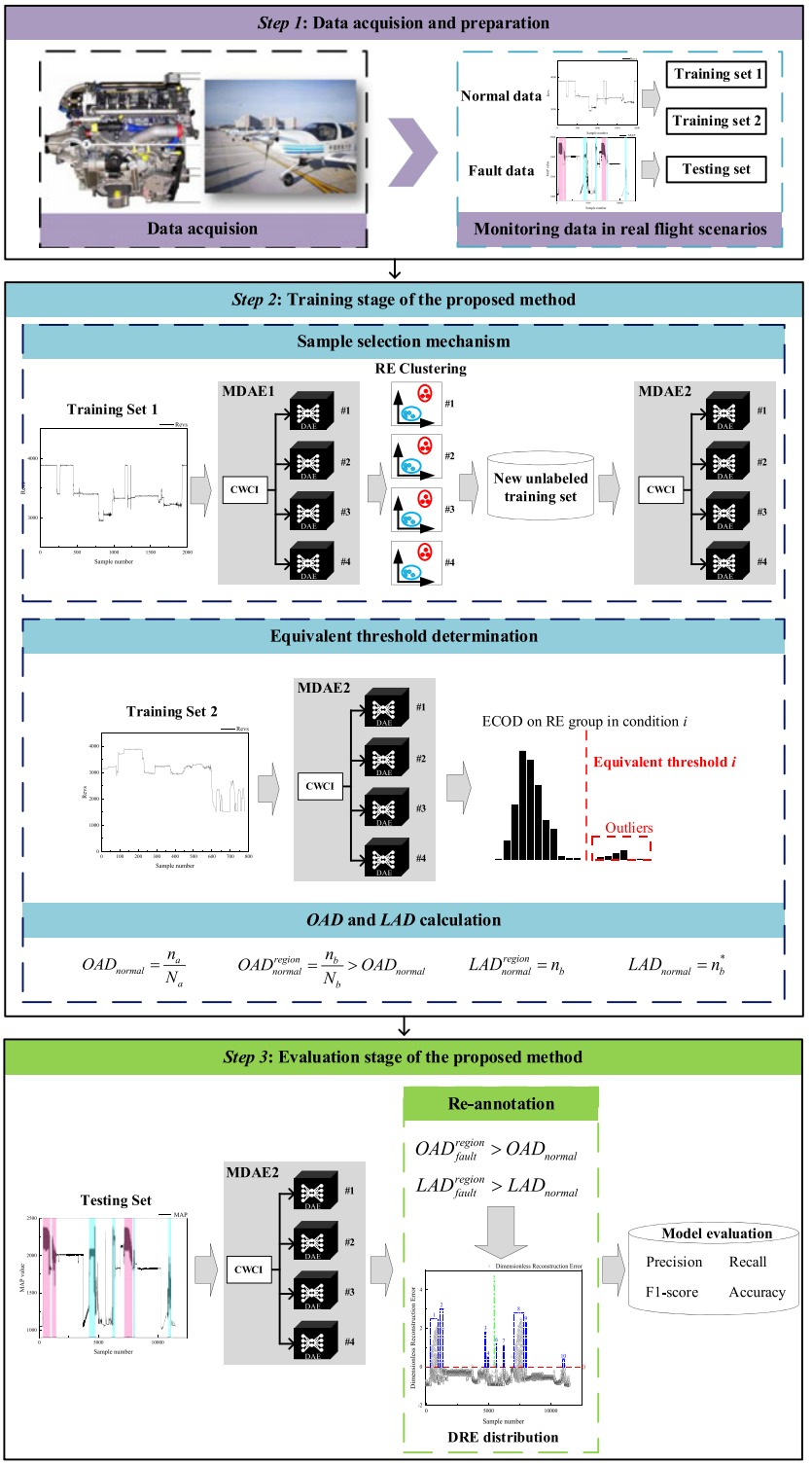


Figure 6. *Schematic diagram of IMDAEF.*

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

Algorithm 1 Outlier region division

Input: Dataset detection results vector $\mathbf{A}=(a_1,\ldots a_i,\ldots,a_m)$(normal: $a_i=0$; outlier: $a_i=1$), region serial number J, dataset overall abnormal degree OAD

Output: Region division results $\mathbf{T}=(t_1,\ldots,t_i,\ldots,t_m)$, ($t_i$ belongs to region J: $t_i=J$; t_i doesn't belongs to any region: $t_i=0$)

```
P=round(1/OAD)
for i∈[1,m], j∈[1,P] do
  if (A[i]=0) then
    T[i]←0
  elseif (A[i+j]=1) then
    T[i]←J
  elseif (A[i-j]=1) then
    T[i]←J
    J←J+1
  else
    T[i]←0
  end if
end for
```

of outliers in the APE normal dataset. Subsequently, equivalent thresholds are established for different working conditions by maintaining a consistent proportion of data points exceeding the corresponding threshold values. An explanation of this algorithm can be found in Ref. [55], as it will not be elaborated on here. Dimensionless operations are performed for each working condition to calculate the dimensionless REs (DREs), as shown in Equation (4). For IMDAEF, a given point is considered as an outlier if its DRE is above 0.

$$DRE=\frac{RE_{condition\ n}-Threshold_{condition\ n}}{Threshold_{condition\ n}} \tag{4}$$

Third, an outlier region (OR) division mechanism with two evaluation metrics called overall abnormal degree (OAD) and local abnormal degree (LAD) is proposed for the benefit of describing outlier clustering characteristics over time in a given dataset. The method is shown in Algorithm 1. The definition of OAD of the whole dataset is shown in Equation (5):

$$OAD_{normal}=\frac{n_a}{N_a} \tag{5}$$

where OAD_{normal} describes the overall abnormal distribution of the real-life APE normal dataset, n_a, N_a means the number of the outliers in the normal dataset and the size of the normal dataset.

After the OR division, similarly, the OAD of a specific region is defined as Equation (6):

$$OAD_{normal}^{region}=\frac{n_b}{N_b}>OAD_{normal} \tag{6}$$

where OAD_{normal}^{region} describes the abnormal degree of a specific region in the real-life normal dataset, and it is larger than OAD_{normal} to highlight the abnormality of ORs. n_b, N_b denote the number of abnormal samples in each outlier region and the size of the OR. The definition of LAD in a normal dataset is

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

shown in Equations (7a) and (7b):

$$LAD_{normal}^{region} = n_b \tag{7a}$$

$$LAD_{normal} = n_b^* \tag{7b}$$

where n_b^* is a value larger than that in most ORs of the same dataset. LAD_{normal}^{region} implies the outlier clustering in a specific region, while LAD_{normal} indicates the outlier clustering threshold value of the real-life normal dataset.

Finally, although public datasets for APE anomaly detection are limited, real-life multisensor datasets containing fault behaviours are extracted from daily flights and carefully labeled by maintenance staff for use as testing datasets. However, due to human limitations and the complex nature of fault behaviours in APE daily flights, the annotation task is particularly challenging, and the resulting labels may not always be entirely accurate. To address this, an OR selection technique, integrated with the previously described OR division mechanism, is introduced during the model evaluation process. The primary goal is to assist in re-labeling fault datasets using the *OAD* and *LAD* parameters obtained during the training process. By inputting the fault dataset into the MDAE in the framework, similar operations are performed, allowing the outlier distribution within the fault dataset to be determined. The corresponding *OAD* and *LAD* are presented in Equations (8) and (9):

$$OAD_{fault}^{region} = \frac{n_c}{N_c} > OAD_{normal} \tag{8}$$

$$LAD_{fault}^{region} = n_c > LAD_{normal} \tag{9}$$

where n_c, N_c denote the number of the abnormal samples in each OR and the size of the OR in a specific fault dataset.

The ORs selected in the testing fault-embedded datasets need to fulfil the following requirement: the *LAD* of the regions in the fault datasets should be larger than LAD_{normal} determined using the real-life normal dataset. Thus, the ORs are abnormal enough compared with the normal datasets, and if outliers in ORs are mislabeled as normal samples, they are supposed to be relabeled and recognized as anomalies, leading to the recalculation of the evaluation metrics. While manual verification remains mandatory during the final re-annotation stage, this method provides a relatively feasible criterion that can be used in the model evaluation process to help with the annotation task and improve the accuracy of the model performance evaluation metrics. It can also be applied in the subsequent unsupervised anomaly detection process to identify critical outlier clusters that are more anomalous and warrant attention.

Overall, Fig. 6 and the aforementioned description together illustrate the following module relationships in IMDAEF: the integration of the CWCI process and the DAE forms the MDAE model. A sample selection mechanism acts during MDAE training to optimise this model. Subsequently, the equivalent threshold determination module uses the trained MDAE to compute appropriate thresholds for anomaly detection, yielding the *OAD* and *LAD* parameters that are critical for the final re-annotation process. The proposed framework is considered applicable to various types of APEs. However, different engine types may exhibit distinct operational profiles and noise characteristics. Consequently, applying our framework to a new engine type would require adjustments to the framework parameters, with particular emphasis on recalibrating the working condition boundaries and equivalent thresholds based on the specific normal-operation data of that engine.

3.0 Experiment using IMDAEF

3.1 Data analysis and preprocessing

The experimental monitoring data were obtained from CD-135 APEs installed on the aircraft at the Flight College of Civil Aviation University of China (CAUC). These datasets are multidimensional time series data obtained by the full authority digital engine control (FADEC) system, as shown in

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

Table 1. *Typical APE sensor data obtained from FADEC system*

Date Time Unit	Revs rpm	Load /	MAP mbar	POil mbar	PRail bar	PBaro mbar	DisLoad /	LiMAPTar mbar	AdActVol m^3/min	PRaTar bar	WG_DC /
12/03/2023 01:34:39:00	3323	66.6	1741	4216	899	765	59.7	1744	41.05	917	84.4
12/03/2023 01:34:40:00	3326	66.6	1736	4151	899	765	59.8	1745	41.06	919	84.6
12/03/2023 01:34:41:00	3342	66.6	1741	4163	921	765	59.9	1749	41.09	925	85
						...					
12/03/2023 04:50:18:00	1495	0	1037	2712	347	987	5.3	1211	8.8	348	18

Table 2. *Main variables collected by the FADEC system*

	Variable abbreviation	Meaning	(Unit, range)
1	Revs	Crankshaft speed	(rpm, 0–8000)
2	Load	Position of throttle lever	(/, 0–100%)
3	MAP	Engine intake pressure	(mbar, 0–3000)
4	POil	Lubricating oil pressure	(mbar, 0–5000)
5	PRail	Common-rail fuel pressure	(bar, 0–2500)
6	PBaro	Ambient atmospheric pressure	(mbar, 0–2000)
7	DisLoad	Engine power	(/, 0–100%)
8	AdActVol	Fuel injection quantity	(m^3/min, 0–100)
9	WG_DC	Waste gate duty cycle	(/, 0–100%)
10	LiMAPTar	Target value of MAP	(mbar, 0–3000)
11	PRaTar	Target value of PRail	(bar, 0–2500)

Table 1. Due to practical limitations, the normal operation data used for training was mainly collected from a specific APE. To ensure coverage of all flight phases throughout the year, we selected datasets that encompass complete flight training profiles from multiple seasons. Moreover, normal operation data from the top one-third of the engine's service life were primarily selected to minimise the degradation effect on the detection results. And the testing datasets used for model evaluation are mainly fault-embedded datasets collected from multiple APEs with the same engine type. These fault datasets were downloaded by the maintenance staff and primarily consist of piston engine data collected during the three hours leading up to the aircraft's emergency landing. Therefore, the fault datasets inherently contain both normal and abnormal data.

Table 2 lists and describes key variables from the FADEC sensors, along with the units and ranges used in this study. The selection of these variables is grounded in engineering expertise [56], as they demonstrate high relevance for detecting anomalies in APEs. For variable 10, since the difference between the target value and the actual value may be more meaningful for detecting faults in APEs, the absolute value of the difference between variables LiMAPTar and MAP is selected as the final feature, instead of LiMAPTar. A similar feature processing method can also be performed on PRaTar. Therefore, the IMDAEF has an input variable dimension of 11. Notably, the flight phases of the APE-equipped aircraft can be judged from the variable PBaro. After feature selection, the normalisation of input data x is done using Equation (11) to get normalised x_{norm}.

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{11}$$

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

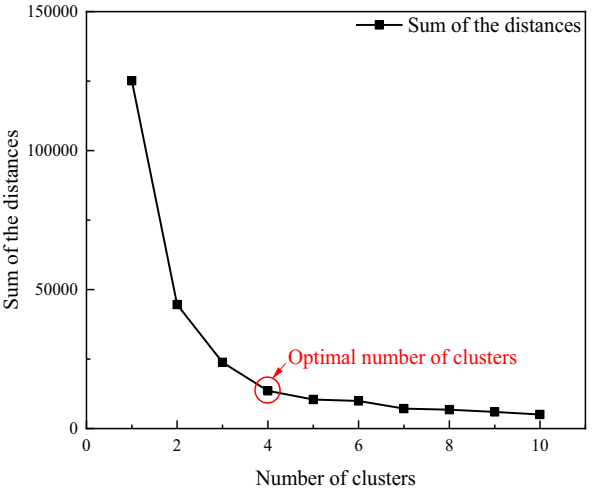


Figure 7. *Optimal cluster selection of K-means method.*

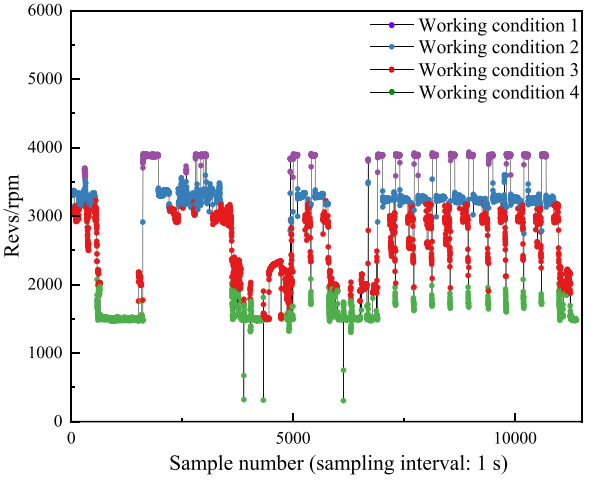


Figure 8. *Revs and corresponding clustering results.*

3.2 CWCI of APEs

As mentioned in Section 2, the APE normal operation dataset is divided into Training set 1 and Training set 2, respectively. The K-means algorithm is trained using Training set 1, which includes all working conditions. To determine the optimal number of clusters, the elbow method was applied, as illustrated in Fig. 7. The Y-axis indicates the sum of the distances from each sample to the corresponding cluster's centre point, while the X-axis is the number of the clusters. The results show that the optimal number of clusters is 4. All figures presented in the following sections were prepared using Origin.

Figure 8 is an example illustrating the trend of Revs in the selected normal operation dataset, while different colours show results of the K-means algorithm. In this paper, the term 'sample number' refers to a continuous sequence of samples taken at 1-second intervals. The four clusters in Fig. 8 correspond to distinct APE working conditions. Their clear stratification in the Revs plot, as evidenced by the colour distribution, indicates a strong correlation between operating condition and rotational speed. Here, Revs is used as an example to demonstrate the proposed identification method: the preliminary

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

Table 3. *APE working condition boundaries before and after fine-tuning*

Working condition boundary value of variable revs	Before fine-tuning	After fine-tuning
Ground idle	0–1950	0–1700
Descending	1950–3100	1700–3200
Cruising	3100–3600	3200–3750
Take-off and climbing	3600–4000	3750–4000

Table 4. *Typical parameters of identified APE working conditions*

Phase parameters	Typical range
Climbing/descending time	600–900 s
Climbing/descending altitude	2300–2800 m
Cruising altitude	2500–3000 m
Cruising time	1000–7000 s

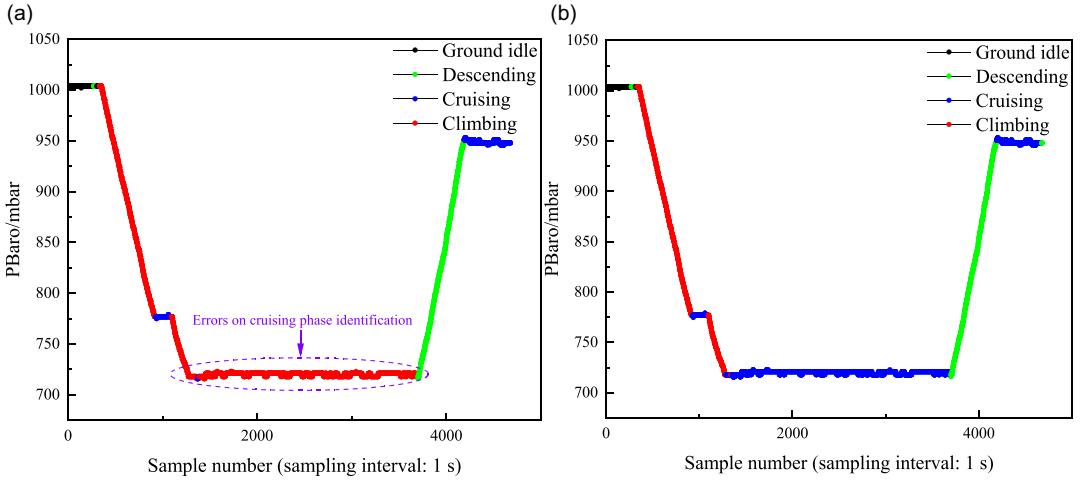


Figure 9. *Comparison of CWCI results. (a) K-means identification results. (b) Fine-tuned identification results.*

Revs boundaries of different working conditions can be determined from colour distributions in Fig. 8, and values are shown in Table 3. Nevertheless, due to the limitations of the clustering algorithm, the results may not always accurately reflect the actual states of APE-equipped aircraft, particularly as the APE working conditions should also align with the aircraft's flight phases. In this study, the flight phases are identified by the trend of PBaro over time, which can be divided into four phases: ground idle phase, take-off and climbing phase, cruising phase and descending phase. Thus, the values of APE working condition boundaries are further fine-tuned to ensure consistency with the PBaro distribution. This process can be seen in Fig. 9, and the final boundaries of Revs are presented in Table 3. Other variables can also be adjusted using a similar operation. Typical parameters of identified working conditions are presented in Table 4.

The effectiveness of the aforementioned CWCI approach is further verified on three datasets randomly selected from a normal operation flight record of APE-equipped aircraft. Results are shown in Fig. 10 and the identification accuracy is 95.8, 92.68 and 98.1%, demonstrating that the proposed method achieves sufficient accuracy for real-world APE working condition identification.

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

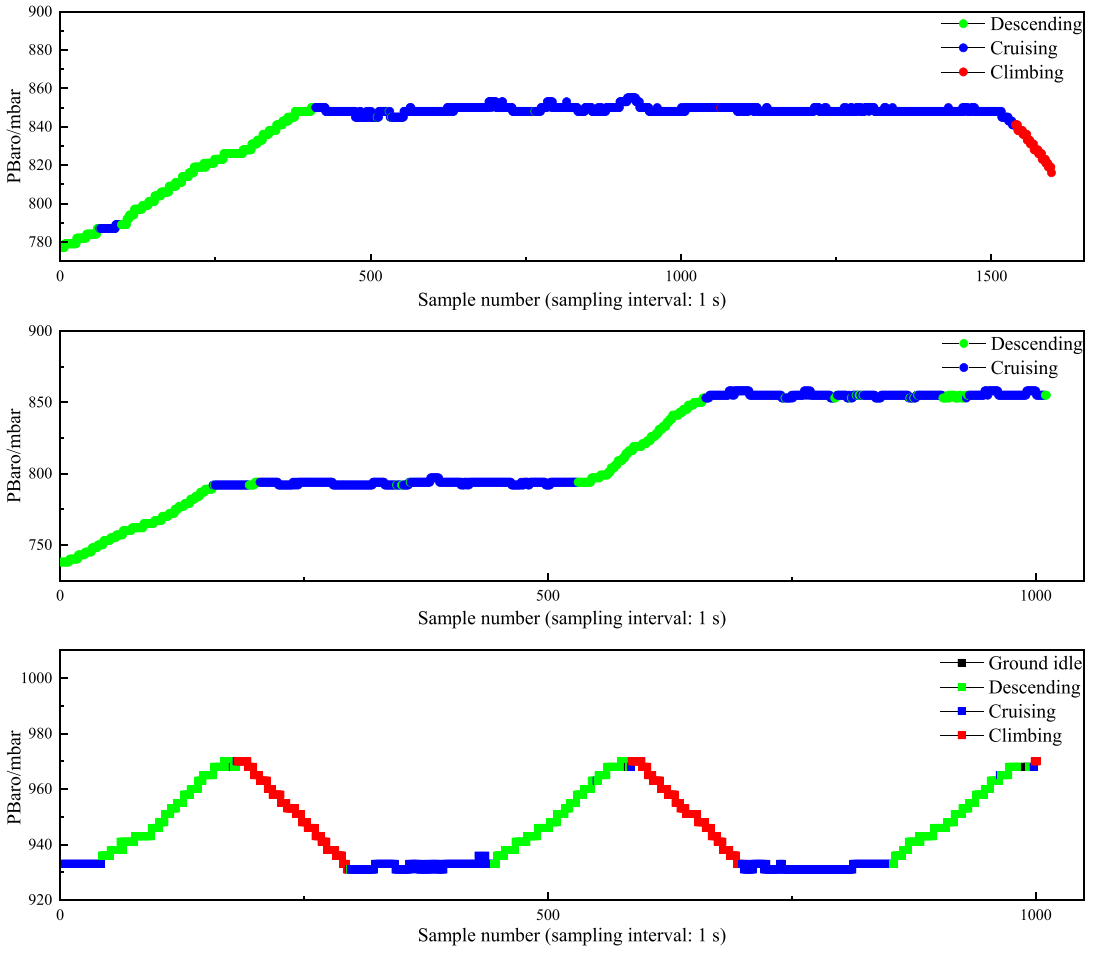


Figure 10. *Verification of CWCI on three sample datasets.*

3.3 Parameter setup of IMDAEF

For the training process of IMDAEF, Training sets 1 and 2 are used with dimensions of 11,377 × 11 and 11,325 × 11. The device is a high-performance CPU (AMD Ryzen 9 7945HX) with 16 GB of system memory, using the TensorFlow 2.11.0 framework. In IMDAEF, each DAE uses two hidden layers. Because the number of input variables is 11, the number of neurons in the input layer is 11. The learning rate is set to 0.001. The model employs Glorot uniform initialisation and L2 regularisation. The remaining hyperparameters were optimised through a tuning process. Here, hidden layer sizing and activation function selection for the climbing phase DAE are detailed below as a representative case study. Grid search results in Fig. 11(a) demonstrate decreasing REs with larger hidden layers, indicating improved fitting capability. A balance between performance and computational cost leads to the selection of 10 and 6 nodes for the respective hidden layers. Subsequent activation function selection is shown in Fig. 11(b), confirming the superior performance of LeakyReLU. The final optimised hyperparameters are presented in Table 5.

3.4 Equivalent thresholds and OAD calculations

Following the implementation of the sample selection mechanism on Training set 1, the MDAE model in IMDAEF undergoes training, and the REs for each data point in Training set 2 are subsequently

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

Table 5. *Main hyperparameters of the IMDAEF*

	IMDAEF			
Hyperparameter	Ground idle	Descending	Cruising	Take-off and climbing
Size of the DAE layers	11-10-8-10-11	11-10-8-10-11	11-10-8-10-11	11-10-6-10-11
Activation function	LeakyReLU	LeakyReLU	LeakyReLU	LeakyReLU
Batch size	10	10	10	10
Epochs	10	10	10	10

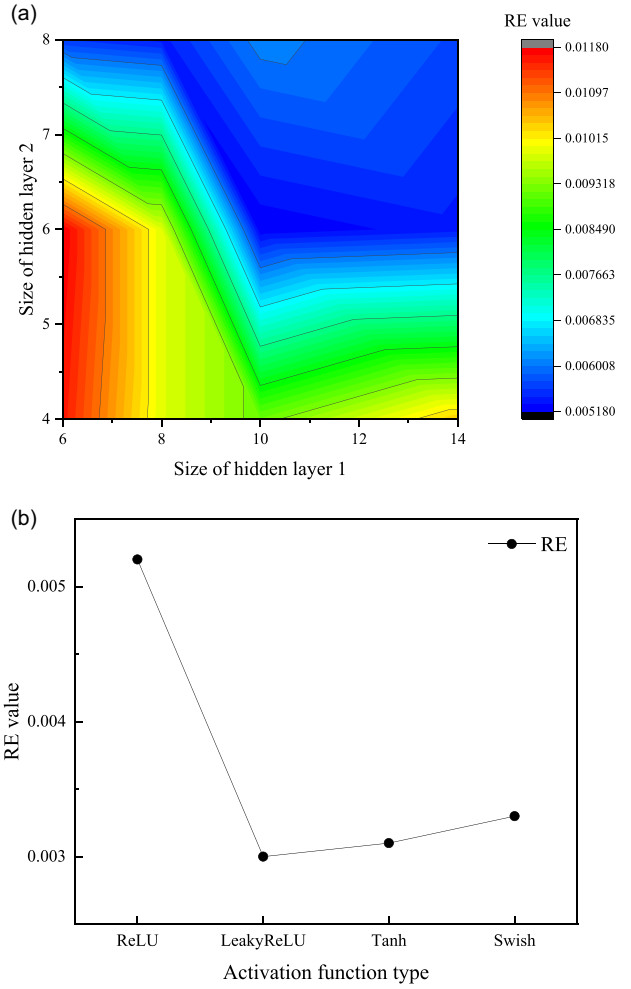


Figure 11. *Partial results of the tuning process. (a) Grid search of optimal hidden layer sizes. (b) Activation function selection.*

computed using the trained model. Ideally, the established threshold value should exceed all REs derived from the normal operation dataset. Nevertheless, considering the inherent characteristics of real-life operational data, the determination of thresholds must be more precise, allowing the REs of outliers in the normal dataset to exceed the thresholds. Therefore, it is necessary to calculate the specific proportion of outliers (OAD_{normal}) in the dataset. For this purpose, a simple algorithm called ECOD is introduced,

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

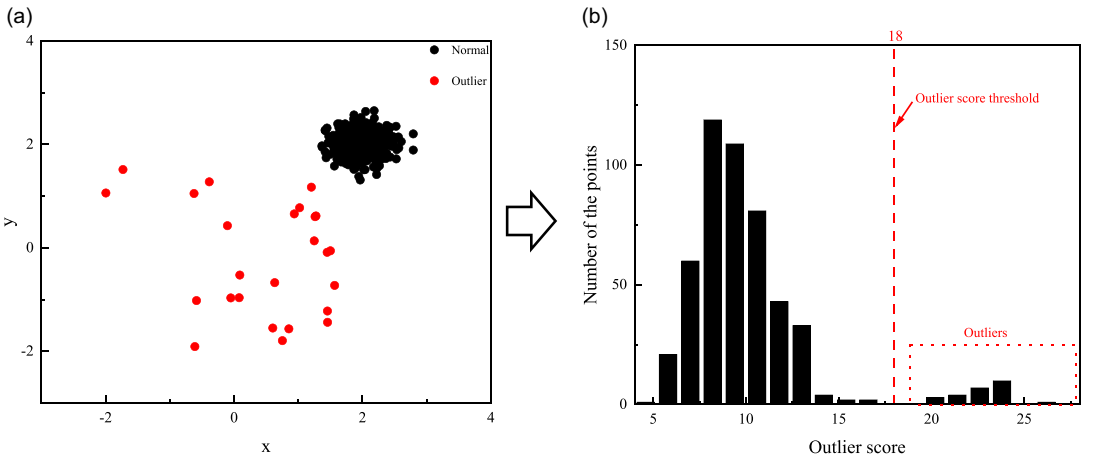


Figure 12. *Example of OAD calculation using ECOD algorithm.*

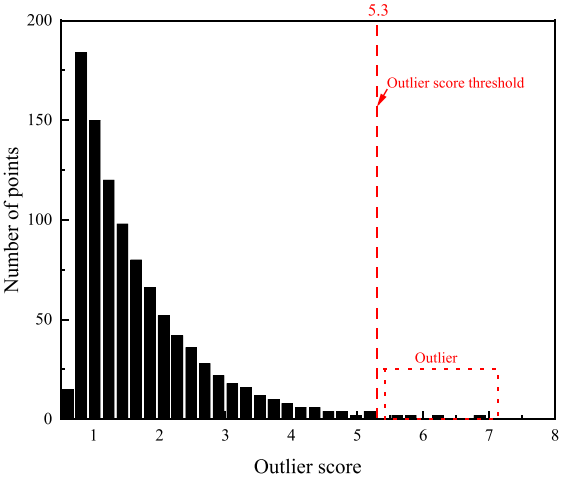


Figure 13. *OAD calculation for a real-life normal dataset using ECOD algorithm (take-off and climbing phase).*

and an example is shown in Fig. 12 to illustrate this process. The sample dataset comprises 500 data points, including 5% outliers, as shown in Fig. 12(a). After using ECOD, the outlier scores of each point are obtained and the corresponding distribution is shown in Fig. 12(b). A significant gap is observed and the outlier score threshold is set as 18. Points with outlier scores greater than 18 are considered outliers, and the number is 25. Then $OAD_{normal} = 25/500 = 0.05$, which is the same proportion as the initially set outlier points.

Similar operations can also be performed separately for REs under different working conditions, which are generated by trained MDAE using Training set 2 as the input. The results of the climbing phase are presented as representative outcomes in Fig. 13, and the outlier score threshold is set as 5.3. The calculated OAD_{normal} of the four working conditions are very close, mainly around 0.098 to 0.011. Assuming that the distribution of outliers in the normal dataset is unrelated to the working conditions of the piston engine, the final OAD_{normal} of the APE normal dataset is 0.01(1%). Therefore, for each working condition, the equivalent RE threshold requires ensuring that the REs of 1% points are above that. The

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

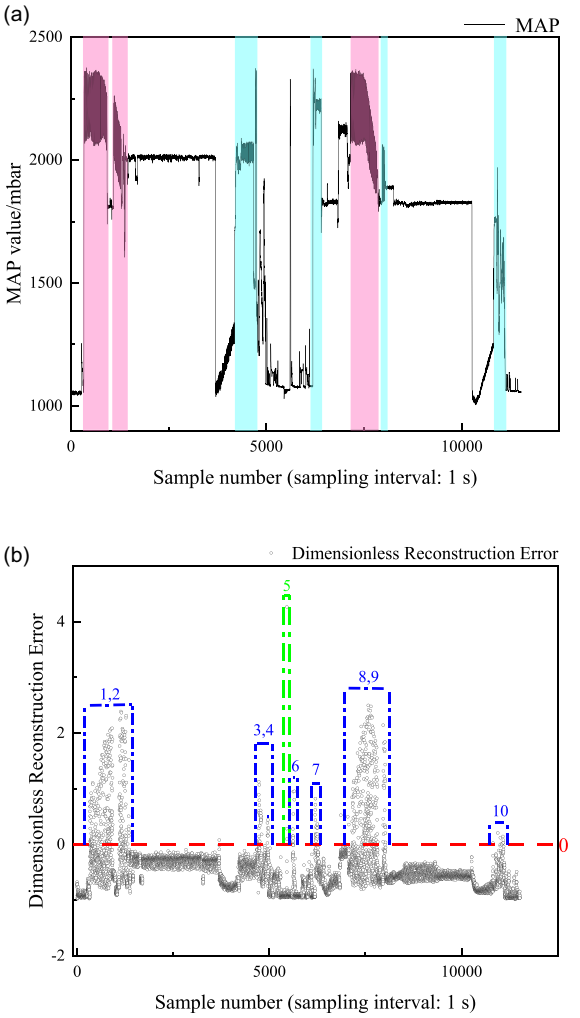


Figure 14. *Example of OR division and selection mechanism using Faultset1. (a) MAP of Faultset1. (b) Distribution of dimensionless reconstruction errors.*

final equivalent RE thresholds of ground idle phase, take-off and climbing phase, cruising phase and descending phase are 0.01, 0.015, 0.013, 0.017.

3.5 OR division, selection and re-annotation during model evaluation stage

For a real-life normal dataset, after the determination of equivalent RE thresholds, the dimensionless operation is applied, and the REs are converted to DREs. The advantage of dimensionless operation is that it allows outliers under different working conditions to be visualised and assessed in a single figure. The OR division mechanism introduced in Section 2.3 is then performed on the DREs to obtain the LAD_{normal} indicating the local outlier clustering threshold. Results show that LAD_{normal}^{region} in over 90% of the divided regions are less than 8. Therefore, the final LAD_{normal} is set as 8.

During the anomaly detection model evaluation process, the OR division mechanism and re-annotation process are executed on fault-embedded datasets. These datasets are manually labeled and Faultset1 is selected as an example to introduce the key processes. Faultset1 is a dataset recording the malfunction of an old-style solenoid valve in the CD-135 piston engine. The main phenomenon is the regular fluctuations of MAP and WG_DC parameters, as shown in Fig. 14(a). This fault is a

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

Table 6. *Evaluation metric changes of Faultset1 before and after re-annotation*

Model	Precision	Recall	F1-score	Accuracy
IMDAEF (without re-annotation)	84.46%	38.56%	52.94%	90.77%
IMDAEF (re-annotation)	89.59%	42.24%	57.41%	91.08%

Table 7. *Brief description of fault datasets*

Fault dataset number	Description of the fault occurrence
Faultset1	Malfunction of an old-style solenoid valve
Faultset2	Low MAP value and high WG_DC
Faultset3	PRail sensor failure
Faultset4	Wire breakage of PRail sensor with power fluctuation

representative sample to show the complex behaviour of faults in APEs, in which the obvious phenomena are marked in red and the subtle phenomena are marked in blue. During the manual labeling process of the fault-embedded datasets, these subtle phenomena can easily lead to biased human judgements and even be neglected. To demonstrate the difficulty of labeling tasks for these phenomena, the red zones are initially labeled as anomalies and the rest are labeled as normal.

Using Faultset1 as input for IMDAEF, the calculated DREs of samples in Faultset1 are shown in Fig. 14(b). The rectangular box represents the ORs after applying the region division mechanism. The LAD_{normal}^{region} of 10 regions in Faultset1 are 219, 100, 29, 9, 3, 10, 19, 284, 24 and 8. Compared with the marks in Fig. 14(a), the positions of regions 1, 2 and 8 highly coincide with the red-marked zones, which are initially labeled as anomalies. Although samples in the other regions are labeled as normal, the positions of regions 3, 4, 7, 9 and 10 largely coincide with the zones marked in blue, suggesting that these regions represent phenomena that were not initially labeled due to their subtlety. The LAD_{normal} equal 8, thus the region 5 marked in green is abandoned and samples in regions 3, 4, 6, 7, 9 and 10 tend to be re-annotated as anomalies. Changes in the evaluation metrics due to the re-annotation process are shown in Table 6. The entire process has been fully automated using MATLAB. Notably, since LAD_{normal} is set heuristically, manual verification of the identified regions remains necessary to ensure reliability. Overall, the OR division and selection mechanism provide an effective foundation for labeling complex fault phenomena, proving valuable for both model evaluation and subsequent detection stages.

4.0 Result analysis

4.1 Fault dataset introduction

To evaluate the anomaly detection performance of different models, four fault-embedded operational datasets of the APEs are summarised in Table 7. These fault sets were documented by CAUC Flight School staff as representative of APE failure modes, based on their routine maintenance experience. Specific parameter changes within these fault datasets are illustrated in Fig. 15, where significant fault sections, diagnosed and highlighted in red through human analysis, are emphasised. Among these datasets, Faultset1 and Faultset2 represent complex faults characterised by discontinuous and subtle phenomena, while the others exhibit relatively simple faults with pronounced phenomena occurring near the end of the datasets.

4.2 Evaluation metric selection

Four typical evaluation metrics called *Precision*, *Recall*, *F1-score* and *Accuracy* are introduced in this paper for the model evaluation stage. Definitions of these metrics are shown in Equations (12)–(15):

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

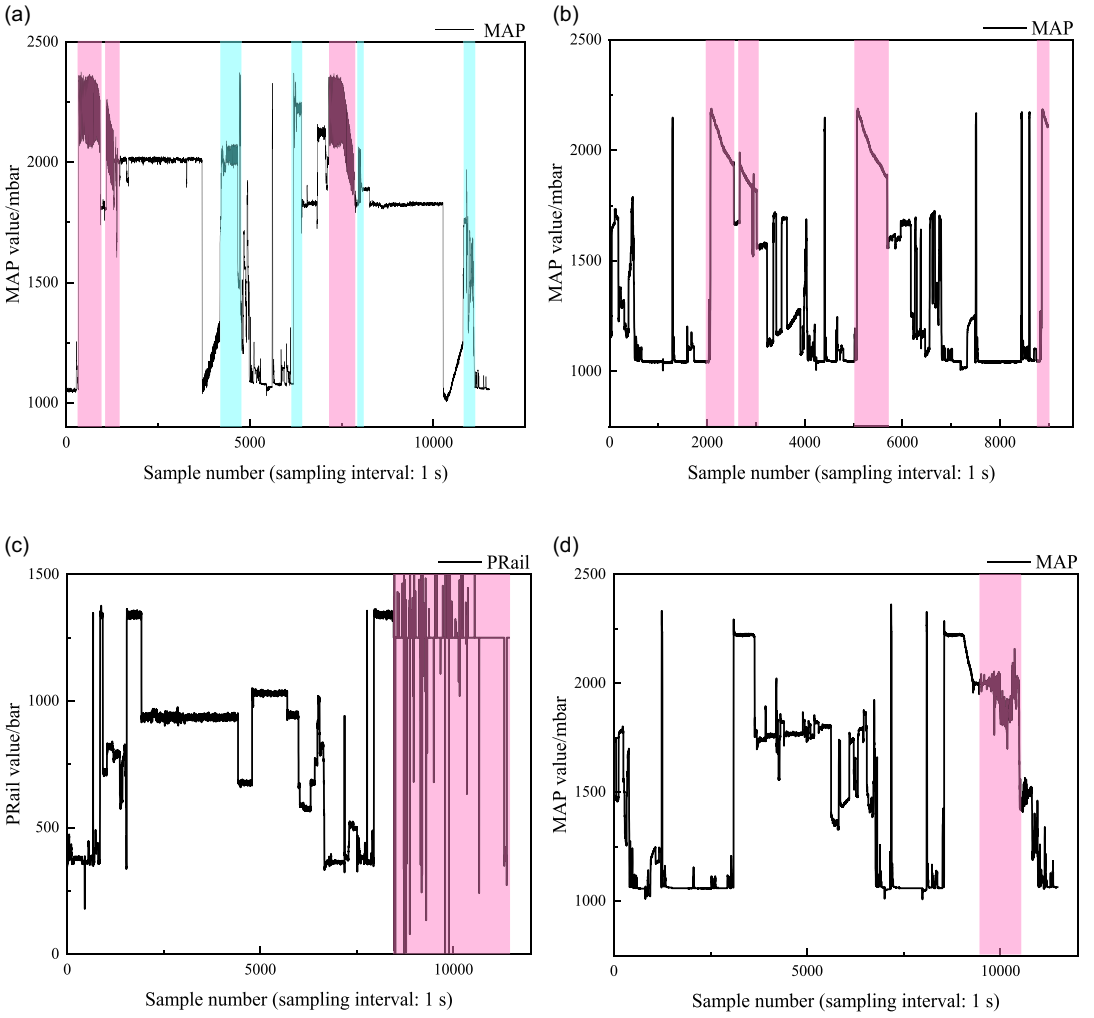


Figure 15. *Fault phenomena embedded in representative datasets. (a) MAP of Faultset1. (b) MAP of Faultset2. (c) Prail of Faultset3. (d) MAP of Faultset4.*

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

where True Positive (*TP*) represents the number of samples that are predicted to be abnormal and confirmed as abnormal; False Positive (*FP*) represents the number of samples that are predicted to be abnormal but are actually normal; False Negative (*FN*) represents the number of samples that are predicted to be normal but are actually abnormal; and True Negative (*TN*) represents the number of samples predicted as normal and confirmed as normal.

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

Several considerations must be addressed when using these metrics for APE unsupervised anomaly detection tasks:

- **Sample Imbalance**: Real-world APE operation datasets are highly imbalanced, meaning that even if a model fails to detect abnormal samples, the *Accuracy* metric might still appear favourable.
- **Complex Fault Behaviours**: The intricate nature of APE fault behaviours and the challenges of manual labeling can result in some fault samples being overlooked. This can lead to an increase in *FP* and a decrease in *TP*, ultimately causing deviations in all four metrics.
- **Importance of Recall**: *Recall* is especially critical in applications where missing abnormal instances can be costly. Therefore, *Recall* holds greater significance in APE anomaly detection tasks. Additionally, abnormal samples are typically labeled based on clear phenomena and robust engineering expertise, lending higher credibility to the calculations of *TP*, *FN* and *Recall*.

4.3 Performance evaluation

Three groups of comparative experiments are conducted to demonstrate the superiority of our framework. Three independent runs with different random seeds are conducted for each model to ensure statistical robustness. Table 8 summarises the corresponding mean values and deviations of the evaluation metrics, with the best results underlined. For brevity, the per cent signs are omitted in the table. While more in-depth statistical tests are suggested, further elaboration is reserved for subsequent studies due to space limitations.

Here is a detailed analysis of the three groups of comparative experiments in this study. The first experiment group compares DAE and MDAE to assess the effectiveness of the CWCI process prior to anomaly detection. As the results show in Table 8, the single DAE model cannot detect the anomalies in Faultset1 and Faultset2, while the MDAE can. For Faultset3 and Faultset4, the *Precision* metrics of MDAE are lower than DAE. One of the reasons may be that the subtle fault phenomena are mistakenly labeled as normal samples, leading to an increase of *FP* and the decrease of *Precision* in MDAE. Besides, if the model has low *Precision* and high *Recall*, the *F1-score*, which balances the *Precision* and *Recall* of the model, can be applied for the final decision. Therefore, the overall performance of the MDAE is superior to the DAE, and the CWCI process is necessary during APE anomaly detection.

The second experiment group is the comparison between the basic model MDAE and our new proposed framework IMDAEF. From the results, it is obvious that the performance of IMDAEF is better than the MDAE since all the evaluation metrics of IMDAEF are much better than the MDAE, which indicates the effectiveness of the sample selection mechanism and the equivalent threshold methods during the training process. The framework achieves the best average *Accuracy* of 94.21% across the four faultsets. The best overall performance of IMDAEF is on Faultset3, and all of the metrics are over 94%. High *Recall* (97.91%), *F1-score* (96.35%) and *Accuracy* (98.02%) are obtained by our framework on Faultset3, while lower performances are observed on Faultset1 and Faultset2, indicating that there is still room for improvement of our framework for complex faults.

The final experimental group benchmarks our proposed IMDAEF against several baselines, including Multi-condition One-Class Support Vector Machine (M-OCSVM) and two recent deep learning models, Transformer-based Anomaly Detection (TranAD) [57] and UnSupervised Anomaly Detection (USAD) [58]. The results demonstrate that IMDAEF achieves the best performance across all evaluation metrics on Faultset1 and Faultset2. While TranAD and USAD perform better on Faultset3, IMDAEF attains a comparable level of performance to these models on Faultset4, with all three significantly outperforming M-OCSVM. Overall, the comparative results confirm that IMDAEF is a feasible and effective framework for APE anomaly detection, particularly demonstrating superior performance over baseline methods in complex fault scenarios.

Table 9 presents a comprehensive evaluation of computational efficiency, comparing training costs and real-time performance across different models. The training time of the USAD model is the largest,

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

Table 8. *Evaluation results of present models*

	Faultset1				Faultset2			
Method	*Prec.*	*Rec.*	*F1*	*Acc.*	*Prec.*	*Rec.*	*F1*	*Acc.*
DAE	/	/	/	/	/	/	/	/
MDAE	83.43(±1.01)	27.92(±0.71)	41.84(±0.33)	89.55(±0.12)	94.27(±2.27)	58.04(±1.24)	71.84(±1.04)	87.93(±0.48)
M-OCSVM	42.78(±0.31)	5.15(±0.06)	9.21(±0.11)	86.30(±0.01)	93.51(±0.02)	13.91(±0.06)	24.22(±0.09)	76.91(±0.01)
TranAD	87.63(±1.40)	36.76(±0.49)	51.80(±0.73)	90.95(±0.12)	/	/	/	/
USAD	/	/	/	/	97.22(±0.75)	48.51(±0.39)	64.72(±0.50)	85.97(±0.10)
IMDAEF	89.66(±1.64)	42.34(±2.06)	57.52(±1.74)	91.10(±0.20)	98.06(±1.12)	64.05(±1.05)	77.48(±0.90)	89.88(±0.41)
	Faultset3				Faultset4			
Method	*Prec.*	*Rec.*	*F1*	*Acc.*	*Prec.*	*Rec.*	*F1*	*Acc.*
DAE	99.37(±0.32)	23.09(±0.10)	37.47(±0.15)	79.78(±0.04)	87.63(±0.91)	29.72(±0.73)	44.39(±0.92)	92.21(±0.10)
MDAE	81.45(±0.21)	70.40(±0.02)	75.53(±0.09)	88.03(±0.04)	81.30(±0.96)	55.04(±1.41)	65.64(±1.08)	93.98(±0.15)
M-OCSVM	99.02(±0.00)	23.55(±0.00)	38.04(±0.00)	79.88(±0.00)	89.20(±0.14)	45.38(±0.66)	60.15(±0.61)	93.72(±0.07)
TranAD	98.66(±0.10)	99.15(±0.07)	98.91(±0.02)	99.43(±0.01)	73.41(±0.23)	1.00(-0.25)	84.66(±0.10)	96.21(±0.01)
USAD	97.59(±0.30)	98.98(±0.23)	98.28(±0.14)	99.11(±0.07)	86.34(±0.58)	92.31(±0.25)	89.22(±0.52)	97.47(±0.11)
IMDAEF	94.85(±0.21)	97.91(±0.39)	96.35(±0.11)	98.02(±0.06)	92.50(±1.15)	88.44(±1.13)	90.42(±0.76)	97.84(±0.17)

M-: the model considers the CWCI process.

Table 9. *Comparison of training and prediction time*

Model	DAE	MDAE	M-OCSVM	TranAD	USAD	IMDAEF
Training time	53.05 s	56.97 s	13.03 s	75.41 s	478.21 s	110.70 s
Prediction time	0.08 ms	0.40 ms	0.21 ms	1.86 ms	4.71 ms	0.53 ms

Table 10. *Ablation experimental results*

	Faultset1	Faultset2	Faultset3	Faultset4
Method	*F1-score*			
IMDAEF	57.41%	77.45%	96.34%	90.36%
w/o CWCI process	29.51%	28.59%	73.05%	44.25%
w/o sample selection mechanism	52.79%	75.01%	92.21%	86.17%
w/o equivalent threshold determination	41.91%	39.77%	93.43%	85.65%
w/o re-annotation process	52.94%	75.63%	95.40%	85.06%

while the M-OCSVM is the least. Although IMDAEF requires a comparatively longer training time, its inference time is significantly less than that of TranAD and USAD, highlighting its practical advantage. Additionally, a comparison of runtime memory usage under the same conditions demonstrates that our model consumes 8–20 MB less than both TranAD and USAD. All the above results demonstrate that our model maintains the required accuracy with modest computational demands, making it well suited for real-time APE anomaly detection tasks.

4.4 Ablation study

To investigate the relative contribution of each core component within the proposed framework, we systematically perform ablation studies by sequentially excluding individual modules and quantitatively evaluating their impact on overall model performance metrics during the evaluation process. The comparative analysis is conducted through examination of *F1-score* variations across selected datasets. First, the CWCI process is removed, and the core model is replaced by a simple DAE model instead. Second, we consider the framework without the sample selection mechanism but instead with a pure MDAE. Third, we consider the model without the equivalent threshold determination, i.e. we assume the thresholds are larger than all of the REs in the normal dataset (excluding rare extremes). Finally, we study the model without the re-annotation process while retaining all other components.

The results are summarised in Table 10 and provide the following findings: when we remove the CWCI process and ignore the effect of various working conditions, the average drop is the maximum, achieving 36.54%. The removal of equivalent threshold determination represents the second-most impactful alteration to the framework, following only the CWCI process. Furthermore, its absence most severely impairs the detection performance on the more complex faults such as Faultset1 and Faultset2. When we remove the sample selection mechanism, the maximum decrease in *F1-scores* is 4.62%, demonstrating the need for the filtering process for real-life APE datasets. Replacing the re-annotation process has the highest performance drop of nearly 5.3% for Faultset4. However, the drop is not significant for Faultset3, demonstrating that the effect of the re-annotation process may also vary based on the complexities of the targeted fault datasets.

4.5 Robustness analysis

In this study, a test is further conducted to evaluate the noise robustness of IMDAEF. Based on APE engineering experience, Gaussian noise with a mean of 0 and a standard deviation ranging from 0 to 3% of full scale is randomly added to 20% of the dataset for the training. Notably, the Gaussian noise

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

Table 11. *Comparative results of robustness analysis*

Method	Ours	Ours+noise
F1-score (Faultset4)	90.36%	88.13%

assumption is commonly used in engine health management and has been extensively employed and validated, as supported by numerous studies [59–61]. The results are shown in Table 11. Experimental results demonstrate that the proposed framework retains stable performance under noisy conditions, confirming its robustness against noise interference.

5.0 Conclusion

Obtaining accurate and representative annotated data is difficult, and there exhibits complex behaviours in both normal and faulty states, which results in poor detection performance of existing anomaly detection methods for APEs. To address this issue, a novel unsupervised anomaly detection method named IMDAEF is investigated to achieve accurate APE unsupervised anomaly detection. In comparison with the current unsupervised anomaly detection methods, the proposed approach offers three distinct advantages, which are elaborated below.

First, a clustering method-guided working condition identification is proposed to determine the boundaries between different working conditions more reasonably. Unlike traditional clustering methods, the developed identification approach refines the clustering results by incorporating aircraft flight phase information, making it more rational and comprehensive.

Second, to address the presence of outliers in real-life normal operation datasets of the APE, this study employs two approaches during the model training stage: firstly, a sample selection mechanism is implemented to automatically remove abnormal samples during the training process of the MDAE model. This ensures that the trained MDAE effectively reconstructs normal samples while struggling with anomalous ones; secondly, a simple outlier detection algorithm is utilised for equivalent threshold determination, as the distribution of outliers in normal operation datasets may not strictly follow the 3σ principle. Consequently, the thresholds obtained are more accurate, further enhancing the performance of the proposed model.

Finally, an OR division and selection mechanism is developed to address the challenge posed by the lack of well-annotated datasets during the model evaluation stage. The *OAD* and *LAD* parameters, which characterise the overall and local abnormality of the datasets, play a key role in the OR division and selection process by identifying samples with distinct clustering features in fault-embedded datasets. This technique provides valuable evidence for improving the annotation of subtle phenomena in existing fault datasets. Additionally, it can be applied in the unsupervised anomaly detection process to highlight regions that exhibit greater abnormality and warrant further attention.

The superiority of the proposed unsupervised anomaly detection framework has been validated using various real-life fault-embedded datasets. The experimental data were sourced from monitoring records provided by the FADEC system of the aircraft. Our framework achieves the best *F1-score* and *Accuracy* on three datasets (Faultset1, Faultset2, Faultset4) and attains the highest average *Accuracy* (94.21%) across all four sets, demonstrating superior overall performance. The experimental results indicate that the proposed framework systematically mitigates the adverse effects of real-life data on model training and performance evaluation to a certain extent. Compared to the baseline DAE, our framework yields a substantial 58.88% gain in *F1-score*, specifically on Faultset3. Ablation study and robustness study are also explored. Overall, the framework is suitable for real-time APE anomaly detection, offering valuable insights for APE safety analysis and continuous airworthiness research.

Our primary contribution lies in demonstrating the empirical effectiveness of the proposed framework, though the study has limitations. First, in the process of conducting anomaly detection research on APEs, two categories of challenging anomalies are identified: those that manifest exclusively under specific working conditions, and those indicated by subtle frequency-domain changes in sensor data.

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

Accordingly, our future work will therefore focus on enhancing the model's ability to detect these anomalies. Second, this study did not fully investigate operational factors and performance degradation. Future research will address these aspects, particularly since performance degradation may elevate noise levels, a problem where adaptive techniques may offer a solution.

Acknowledgements. The work was funded by the National Natural Science Foundation of China and Civil Aviation Administration of China [Grant U2233213]. The work was supported by the Innovation Team of Complex System Safety and Airworthiness of Aero-Engine from the Aircraft/Engine Integrated System Safety Beijing Key Laboratory, Beijing 100191, China.

Competing interests. The authors declare none.

References

[1] Chandola, V., Banerjee, A. and Kumar, V. Anomaly detection: a Survey, *ACM Comput. Surv.*, 2009, **41**, (3), pp 1–58.
[2] Khattak, W.R., Salman, A., Ghafoor, S. and Latif, S. Multi-modal LSTM network for anomaly prediction in piston engine aircraft, *Heliyon*, 2024, **10**, (3), p e25120.
[3] Jaw, L.C. Recent advancements in aircraft engine health management (EHM) technologies and recommendations for the next step, *Proceedings of ASME Turbo Expo 2005: Power for Land, Sea, and Air*, Nevada, USA, 2005.
[4] Zhou, X., Lu, F. and Huang, J. Fault diagnosis based on measurement reconstruction of HPT exit pressure for turbofan engine, *Chin. J. Aeronaut.*, 2019, **32**, (5), pp 1156–1170.
[5] Bouzid, M.B. and Champenois, G. An efficient simplified physical faulty model of a permanent magnet synchronous generator dedicated to stator fault diagnosis part II: automatic stator fault diagnosis, *IEEE. T. Ind. Appl.*, 2017, **53**, (3), pp 2762–2771.
[6] Cha, J., Ko, S., Park, S. and Jeong, E. Fault detection and diagnosis algorithms for transient state of an open-cycle liquid rocket engine using nonlinear Kalman filter methods, *Acta. Astronaut.*, 2019, **163**, pp 147–156.
[7] Li, Y.G. Gas turbine performance and health status estimation using adaptive gas path analysis. *J. Eng. Gas. Turb. Power.*, 2010, **132**, (4), p 041701.
[8] Fast, M., Palme,́T. and Genrup, M. A novel approach for gas turbine condition monitoring combining CUSUM technique and artificial neural network, *Proceedings of the ASME Turbo Expo*, Florida, USA, 2009.
[9] Zhou, D., Zhang, H. and Weng, S. A new gas path fault diagnostic method of gas turbine based on support vector machine, *J. Eng. Gas. Turb. Power.*, 2015, **137**, (10), p 102605.
[10] Zhao, L., Mo, C., Sun, T. and Huang, W. Aero engine gas-path fault diagnose based on multimodal deep neural networks, *Wirel. Commun. Mob. Comput.*, 2020.
[11] Ying, S., Wang, B., Wang, L., Li, Q., Zhao, Y., Shang, J., Huang, H., Cheng, G., Yan, Z. and Geng, J. An improved KNN-based efficient log anomaly detection method with automatically labeled samples, *ACM Trans. Knowl. Discov. Data.*, 2021, **15**, (3), pp 1–22.
[12] Li, Y. and Guo, L. TCM-KNN scheme for network anomaly detection using feature-based optimizations, Acm Symposium on Applied Computing, Brazil, 2008.
[13] Jain, P., Bajpai, M.S. and Pamula, R. A modified DBSCAN algorithm for anomaly detection in time-series data with seasonality, *Int. Arab. J. Inf. Techn.*, 2022, **19**, (1), pp 23–28.
[14] Christopher, R.D.E., Song, S., Lee, H., Choi, D., Lim, J., Bok, K. and Yoo J. Anomaly detection based on GCNs and DBSCAN in a large-scale graph, *Electronics*, 2024, **13**, (13), p 2625.
[15] Adesh, A., Shobha, G., Shetty, J. and Xu, L. Local outlier factor for anomaly detection in HPCC systems, *J. Parallel. Distr. Com.*, 2024, **192**, p 104923.
[16] Pang, J., Pu, X. and Li, C. A hybrid algorithm incorporating vector quantization and one-class support vector machine for industrial anomaly detection, *IEEE T. Ind. Inform.*, 2022, **18**, (12), pp 8786–8796.
[17] Xu, D., Wang, Y., Meng,Y. and Zhang, Z. An improved data anomaly detection method based on isolation forest, *10th International Symposium on Computational Intelligence and Design*, Hangzhou, China, 2017.
[18] Li, N., Liu, X., Liu, Z., Mao, L., Zhao, L. and Wang, X. Anomaly detection in power grid IoT system based on isolated forest, *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Melbourne, Australia, 2022.
[19] Sakurada, M. and Yairi, T. Anomaly detection using autoencoders with nonlinear dimensionality reduction, *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, Australia, 2014.
[20] Zhou, C., Paffenroth, R.C. Anomaly detection with robust deep autoencoders, *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax, Canada, 2017.
[21] Han, Y., Ma, Y., Wang, J. and Wang, J. Research on ensemble model of anomaly detection based on autoencoder, 2020 *IEEE 20th International Conference on Software Quality, Reliability and Security*, Macau, China, 2020.
[22] Song, J., Kim, K., Oh, J., Cho, S., Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M. and Levine, S. Memto: memory-guided transformer for multivariate time series anomaly detection, *Advances in Neural Information Processing Systems*, New Orleans, USA, 2023.
[23] Fentaye, A.D. and Kyprianidis, K.G. Gas turbine prognostics via Temporal Fusion Transformer, *Aeronaut. J.*, 2024, **128**, (1325), pp 1594–1609.

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

[24] Miao, Z., Chen, G., Liao, X., Dai, J. and He, Y. Self-supervised dual-layer 2D normalizing flow method for industrial anomaly detection, *Appl. Soft. Comput.*, 2024, **163**, p 111928.
[25] Dai, E. and Chen, J. Graph-augmented normalizing flows for anomaly detection of multiple time series, arXiv preprint, 2022.
[26] Zhang, X., Wang, C., Zhou, W., Xu, J. and Han, T. Trustworthy diagnostics with out-of-distribution detection: a novel max-consistency and min-similarity guided deep ensembles for uncertainty estimation, *IEEE Internet. Things. J.*, 2024, **11**, (13), pp 23055–23067.
[27] Han, X., Liu, Y., Zhang, Z., Lü, X. and Li, Y. Sparse auto-encoder combined with kernel for network attack detection, *Comput. Commun.*, 2021, **173**, pp 14–20.
[28] Lin, Y., Liang, Y., Yoshida, S., Feng, X. and Guan, R. A hybrid algorithm of extreme learning machine and sparse auto-encoder, *International Conference on Smart Computing and Communication*, Shenzhen, China, 2016.
[29] San Martin, G., Droguett, E.L., Meruane, V. and Moura, M. Deep variational auto-encoders: a promising tool for dimensionality reduction and ball bearing elements fault diagnosis, *Struct. Health. Monit.*, 2019, **18**, (4), pp 1092–1128.
[30] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. and Manzagol, P. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, *J. Mach. Learn. Res.*, 2010, **11**, pp 3371–3408.
[31] Hu, D., Zhang, C., Yang, T. and Fang, Q. A deep autoencoder with structured latent space for process monitoring and anomaly detection in coal-fired power units, *Reliab. Eng. Syst. Saf.*, 2025, **261**, p 111060.
[32] Wang, X. and Chen, Y. Unsupervised detection of multivariate geochemical anomalies using a high-performance deep autoencoder Gaussian mixture model, *J. Geochem. Explor.*, 2025, **271**, p 107671.
[33] Hooshmand, M.K., Huchaiah, M.D., Alzighaibi, A.R., Hashim, H., Atlam, E. and Gad, I. Robust network anomaly detection using ensemble learning approach and explainable artificial intelligence (XAI), *Alex. Eng. J.*, 2024, **94**, pp 120–130.
[34] Lin, S., Zhang, M., Cheng, X., Shi, L., Gamba, P. and Wang, H. Dynamic low-rank and sparse priors constrained deep autoencoders for hyperspectral anomaly detection, *IEEE Trans. Instrum. Meas.*, 2024, **73**, pp 1–18.
[35] Ding, K., Li, J., Bhanushali, R. and Liu, H. Deep anomaly detection on attributed networks, *Proceedings of the 2019 SIAM International Conference on Data Mining*, Calgary, Canada, 2019.
[36] Xu, D., Ricci, E., Yan, Y., Song, J. and Sebe, N. Learning deep representations of appearance and motion for anomalous event detection, *British Machine Vision Conference*, Swansea, UK, 2015.
[37] Moslemi, A. and Jamshidi, M. Unsupervised feature selection using sparse manifold learning: auto-encoder approach, *Inf. Process. Manag.*, 2025, **62**, (1), p 103923.
[38] Rifai, S., Vincent, P., Muller, X., Glorot, X. and Bengio, Y. Contractive auto-encoders: explicit invariance during feature extraction, *Proceedings of the 28th International Conference on International Conference on Machine Learning*, Bellevue, USA, 2011.
[39] Pang, G., Shen, C., Cao, L. and Hengel, A. Deep learning for anomaly detection: a review, *ACM Comput. Surv.*, 2021, **54**, (2), pp 1–38.
[40] Lin J. and Zhao, M. A review and strategy for the diagnosis of speed-varying machinery, 2014 *International Conference on Prognostics and Health Management*, Cheney, USA, 2014.
[41] Fala, N., Georgalis, G. and Arzamani, N. Study on machine learning methods for general aviation flight phase identification, *J. Aerosp. Inf. Syst.*, 2023, **20**, (10), pp 636–647.
[42] Yin, X., He, Q., Peng, S., Chen, D., Zhang, H. and Zhang, B. A health state prediction model for aeroengine based on multi-attribute belief rule base with considering monitoring error, *Int. J. Fuzzy. Syst.*, 2025, **27**, pp 1072–1095.
[43] Fu, S., Zhong, S., Lin L. and Zhao, M. A re-optimized deep auto-encoder for gas turbine unsupervised anomaly detection, *Eng. Appl. Artif. Intell.*, 2021, **101**, (12), p 104199.
[44] Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P. Extracting and composing robust features with denoising autoencoders, *Proceedings of the 25th international conference on Machine learning*, Helsinki, Finland, 2008.
[45] Schneider, S., Antensteiner, D., Soukup, D. and Scheutz, M. Autoencoders - a comparative analysis in the realm of anomaly detection, 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, New Orleans, USA, 2022.
[46] Ahasan, M.R., Haque, M.S., Akram, M.R., Momen, M.F. and Alam, M.G.R. Deep learning autoencoder based anomaly detection model on 4G network performance data, 2022 *IEEE World AI IoT Congress*, Seattle, USA, 2022.
[47] Tong, C.D., Palazoglu, A. and Yan, X.F. An adaptive multimode process monitoring strategy based on mode clustering and mode unfolding, *J. Process. Contr.*, 2013, **23**, pp 1497–1507.
[48] Song, B., Ma, Y. and Shi, H. Multimode process monitoring using improved dynamic neighborhood preserving embedding, *Chemom. Intell. Lab. Syst.*, 2014, **135**, pp 17–30.
[49] Song, B., Yan, H., Shi, H. and Tan, S. Multisubspace elastic network for multimode quality-related process monitoring, *IEEE Trans. Ind. Inf.*, 2019, **16**, pp 5874–5883.
[50] Han, S., Hu, X., Huang, H., Jiang, M. and Zhao, Y. ADBench: anomaly detection benchmark, ArXiv, 2022.
[51] Liu, B., He, J. and Liu, G. A study of working conditions based on cluster analysis, 2010 *Asia-Pacific Power and Energy Engineering Conference*, Chengdu, China, 2010.
[52] Du, G., Meng, Q. and Yu, H. Research on modeling of cement denitrification under typical working conditions, 2018 *IEEE 4th Information Technology and Mechatronics Engineering Conference*, Chongqing, China, 2018.
[53] Esteves, R.M., Hacker, T. and Rong, C. Competitive K-means, a new accurate and distributed K-means algorithm for large datasets, 2013 *IEEE 5th International Conference on Cloud Computing Technology and Science*, Bristol, UK, 2013.
[54] Hinton, G.E. and Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks, *Science*, 2006, **313**, pp 504–507.

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press

[55] Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C. and Chen, G.H. ECOD: unsupervised outlier detection using empirical cumulative distribution functions, *IEEE Trans. Knowl. Data. Eng.*, 2022, **35**, (12), pp 12181–12193.
[56] Continental Aerospace Technologies GmbH. Operation & Maintenance Manual, 2020.
[57] Tuli, S., Casale, G. and Jennings, N. TranAD: deep transformer networks for anomaly detection in multivariate time series data, *Proc. VLDB Endow.*, 2022, **15**, (6), pp 1201–1214.
[58] Audibert, J., Michiardi, P., Guyard, F., Marti, S. and Zuluaga, M.A. USAD: unsupervised anomaly detection on multivariate time series, *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, USA, 2020.
[59] Li, N., Yang, J., Zhou, R. and Liang, C. Determination of knock characteristics in spark ignition engines: an approach based on ensemble empirical mode decomposition, *Meas. Sci. Technol.*, 2016, **27**, (4), p 045109.
[60] Longwic, R., Sen, A.K., Górski, K., Lotko, W. and Litak, G. Cycle-to-cycle variation of the combustion process in a diesel engine powered by different fuels, *J. Vibroeng.*, 2011, **13**, (1), pp 120–127.
[61] Verma, R. and Ganguli, R. Denoising jet engine gas path measurements using nonlinear filters, *IEEE/ASME Trans. Mechatron.*, 2005, **10**, (4), pp 461–464.

Cite this article: Li G., Xu T. and Ding S. (2026). An improved deep autoencoder framework for aviation piston engine unsupervised anomaly detection considering the real-life data characteristics. *The Aeronautical Journal*, **130**, 2636–2661. https://doi.org/10.1017/aer.2026.10172

https://doi.org/10.1017/aer.2026.10172 Published online by Cambridge University Press