DESIGN OF A MULTI-SCALE FLIGHT PARAMETER DATA ANOMALY DETECTION ALGORITHM

Zichang Lan [1], Junpeng Bao [2], Yifeng Huang [3]*

[1] *School of computer science and technology, Xi'an Jiaotong University，Xi'an, China*
[2] *School of computer science and technology, Xi'an Jiaotong University，Xi'an, China*
[3] *Aeronautics Engineering College, Air Force Engineering University，Xi'an, China*
* *baojp@mail.xjtu.edu.cn*

Keywords: ANOMALY DETECTION, DEEP LEARNING, MULTI-SCALE

Abstract

Anomaly detection in flight parameter time-series data is of great significance for maintaining the regular operation of aircraft. With the development of deep learning, various anomaly detection algorithms have been designed. The DCdetector model uses a dual-attention contrastive structure which can represent input time series data from different perspectives. However, experimental validation on different types of datasets shows that while the DCdetector model exhibits excellent performance in detecting long-duration anomalous sequences, it performs poorly in detecting short-duration anomalies. This paper proposes two multi-scale optimized models based on the DCdetector architecture: a multi-scale segmentation model and a multi-scale concatenation model. In the multi-scale segmentation model, the input time-series data are divided into different scales by using pooling layers with varying sizes, then separately fed into the model for feature extraction and discrepancy representation. The multi-scale concatenation model also employs multi-scale pooling for division, but additionally concatenates the features from different scales before feeding them into the model to generate complementary representations. All representations are integrated for final discrepancy analysis. Experiments conducted on flight parameter datasets and other datasets demonstrate that the proposed multi-scale optimized models achieve strong detection performance for short-duration anomalous sequences while maintaining superior capability in identifying long-duration anomalies.

1 Introduction

1.1 Research Background and Content

Aircraft flight parameter time-series data (including altitude, airspeed, attitude, and engine status) serve as critical references for operational condition monitoring and flight safety. Although AI-based anomaly detection has significantly improved aviation safety systems by enhancing accuracy and efficiency, the complexity of these multidimensional datasets still poses major challenges. Anomalies may manifest as sudden value changes, inconsistencies, or complex inter-dimensional correlation patterns, while traditional single-point detection methods prove inadequate due to the temporal correlations inherent in complete anomaly sequences.

This paper proposes a multi-scale segmentation model and a multi-scale concatenation model, addressing the key limitations in short-term anomaly detection through an innovative multi-scale feature extraction framework. The multi-scale pooling module and novel model architecture with cross-scale fusion mechanisms enable the model to maintain long-sequence detection performance while significantly improving sensitivity to short-sequence anomalies, providing a balanced solution of computational efficiency and detection accuracy for practical aviation safety applications.

1.2 Related Work

Deep learning models have recently gained prominence in flight parameter anomaly detection. Convolutional Neural Networks (CNNs) are effective in extracting local features, while Temporal Convolutional Networks (TCNs), proposed by Schmidhuber's team [1], improve detection accuracy by using dilated convolutions to capture long-range dependencies. Long Short-Term Memory (LSTM) networks and their variants (e.g., BiLSTM) are widely adopted for modeling temporal dynamics; Sutskever et al. [2] demonstrated their strength in detecting gradual anomalies. The introduction of attention mechanisms, particularly the Transformer by Vaswani et al. [3], has enabled global feature modeling through self-attention, showing strong potential in identifying complex anomaly patterns.

However, combining CNNs with autoencoders often falls short in time-series tasks. To improve representation learning, Yin et al. [4] proposed a model combining CNNs with recurrent autoencoders and employed a two-stage sliding window for better data preprocessing. Addressing issues of subsequence modeling and hierarchical decomposition, Zhong et al. introduced MSD-Mixer [5], which uses multi-scale decomposition to represent different levels of input series. To capture evolving cross-series correlations in multivariate time series, Cai et al. developed MSGNet [6], which employs frequency-domain analysis and adaptive graph convolution.

Flight parameter anomaly detection fundamentally requires learning discriminative representations that clearly separate normal and anomalous sequences. Traditional reconstruction-based methods are common but often compromised by anomaly contamination, reducing detection reliability. Contrastive learning presents a robust alternative by focusing on differences between samples. In this context, Yang et al. [7] proposed DCdetector, which employs a dual-attention contrastive learning framework with an asymmetric dual-attention structure to create a permutation environment. It uses a pure contrastive loss to train the model, encouraging permutation-invariant and discriminative representations. The model maximizes similarity between the outputs of its two branches, causing anomalies to stand out due to inconsistent representations across views. Unlike reconstruction-based methods, DCdetector computes anomaly scores from the discrepancy in learned representations. Extensive experiments on both univariate and multivariate datasets validate its strong performance in detecting long-duration anomalies. However, its effectiveness diminishes with short-term anomalies, highlighting a need for further improvements.

In multivariate time series (MTS) forecasting, Transformer-based models have achieved success in capturing long-range dependencies. However, most focus primarily on temporal patterns, overlooking cross-variable dependencies. To address this, Zhang et al. proposed Crossformer [8], which introduces Dimension-Segment-Wise (DSW) embedding and Two-Stage Attention (TSA) layers to jointly model temporal and dimensional relationships. These are integrated within a Hierarchical Encoder-Decoder (HED) framework, yielding strong performance on different tasks. Standard Transformers often struggle with multi-scale temporal features. Pathformer, proposed by Chen et al. [9], introduces an adaptive multi-scale Transformer that processes data across varying resolutions and temporal distances using dual attention and dynamic pathway selection. Building on this, Pyraformer [10] applies a Pyramid Attention Module (PAM) with multi-resolution aggregation and intra-scale connections to capture long-range dependencies. Scaleformer [11] refines multi-scale representations through iterative adjustments, shared weights, and normalization. Triformer [12] introduces a triangular, variable-specific attention mechanism using lightweight patch attention layers and parameter sharing, enhancing computational efficiency and scalability.

In addition to these, other frameworks have shaped the landscape of time-series anomaly detection. USAD [13] adopts adversarial training for unsupervised MTS detection, while TranAD [14] optimizes Transformer-based detection for industrial-scale data. Schmidl et al. [15] conducted a comparative study of nine deep models, aiding method selection. Deng et al. [16] applied graph neural networks to model inter-variable dependencies, and Li et al.'s MAD-GAN [17] pioneered the use of GANs in time-series anomaly detection. These works offer critical foundations and benchmarks for the multi-scale modeling approach explored in this research.

2. Methodology

Detecting anomalies in flight data is critical for aviation safety, but current methods like DCdetector excel only at identifying long anomalies while performing poorly on short ones. To bridge this gap, we propose a multi-scale detection algorithm that enhances sensitivity to transient anomalies through hierarchical feature extraction. Building on DCdetector's dual-attention framework, we develop two enhanced models: a multi-scale segmentation model that independently processes multi-resolution time-series data before feature integration, and a multi-scale concatenation model that adds cross-scale fusion for richer feature representation.

From a technical perspective, this work introduces a multi-scale decomposition methodology that employs hierarchical pooling operations to process time-series data at varying resolutions while maintaining dimensional consistency. The proposed approach features two key innovations: a segmentation model that independently processes and concatenates multi-scale representations, and an enhanced concatenation model that fuses cross-scale features to generate more discriminative representations. By preserving DCdetector's dual-attention contrastive structure - which extracts both patch-wise and in-patch temporal features through shared-weight branches - while incorporating multi-scale analysis, the method achieves simultaneous capture of global trends and local details for improved detection of anomalies across durations.

2.1 Multi-scale Model Structure

Fig 1 illustrates the complete architecture of the multi-scale segmentation model initially optimized based on the DCdetector framework. The model's core innovation involves introducing a multi-scale feature extraction mechanism, where the input multidimensional time-series data first passes through a multi-scale pooling processing layer. This layer employs average pooling windows of different scales to perform multi-level downsampling on the original time-series signals, generating multiple subsequences sets with varying temporal resolutions. The subsequence data from each scale independently enters a preprocessing module for standardization and dimension alignment before being fed into the dual-branch feature extraction network. The patch-wise branch focuses on analyzing global dependencies between different time segments, capturing long-term temporal patterns through cross-segment self-attention mechanisms, while the in-patch branch concentrates on local feature interactions within individual time segments, utilizing intra-segment attention networks to extract fine-grained temporal features. Both branches employ shared-weight multi-head self-attention mechanisms for feature transformation. The scale-specific representation vectors processed by both branches then enter the difference representation module to generate discriminative features. Finally, the model performs dimension alignment and concatenation on all scale-specific representations to form a comprehensive multi-scale feature representation, which is input to the anomaly determination module to obtain the final anomaly detection results. This architectural design enables the model to simultaneously capture macroscopic trends and microscopic anomalies in time-series data, significantly improving its adaptability to anomaly patterns of different durations.

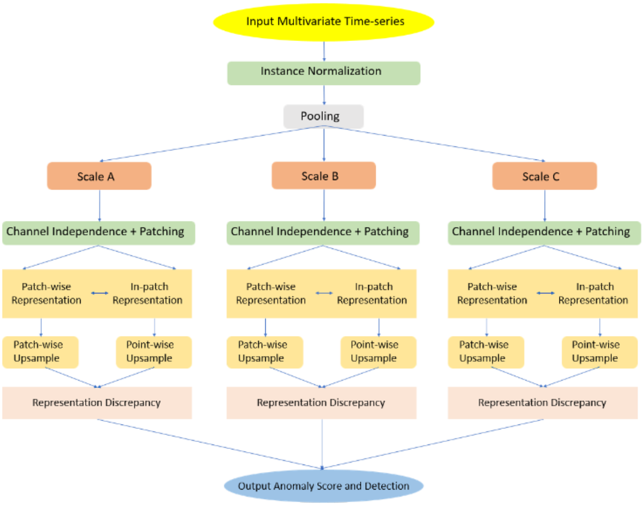
Input Multivariate Time-series
Instance Normalization
Pooling
Scale A
Scale B
Scale C
Channel Independence + Patching
Channel Independence + Patching
Channel Independence + Patching
Patch-wise Representation
In-patch Representation
Patch-wise Representation
In-patch Representation
Patch-wise Representation
In-patch Representation
Patch-wise Upsample
Point-wise Upsample
Patch-wise Upsample
Point-wise Upsample
Patch-wise Upsample
Point-wise Upsample
Representation Discrepancy
Representation Discrepancy
Representation Discrepancy
Output Anomaly Score and Detection

Fig. 1. Multi-scale segmentation model.

Building upon the multi-scale segmentation model illustrated in Fig. 1, we have developed an enhanced multi-scale concatenation model as shown in Fig. 2. This advanced architecture maintains the fundamental pooling-based multi-scale decomposition approach, where input time-series data is partitioned into different temporal scales, each processed independently to generate scale-specific representations. The critical innovation lies in the feature integration phase: following the patch-wise and in-patch feature extraction across all scales, the model performs comprehensive concatenation of these multi-scale features. These aggregated features, combined with the original scale-specific representations, undergo processing through both the dual-attention contrastive module and difference representation module to produce a unified feature representation. This final representation is then evaluated by the anomaly determination module to generate detection outcomes.

The model's architecture achieves superior performance through its unique capacity to simultaneously analyze both individual scale characteristics and their cross-scale interactions. By preserving the original multi-scale analysis framework while introducing this concatenation-based feature fusion mechanism, the approach demonstrates enhanced capability in detecting anomalies that manifest across different temporal resolutions. This integration proves particularly effective for capturing complex anomaly patterns that exhibit both short-term variations and long-term trends, addressing a fundamental limitation of conventional single-scale detection methods. The concatenation operation serves as a crucial bridge between granular feature analysis and holistic pattern recognition, enabling more robust and comprehensive anomaly detection across diverse operational scenarios.

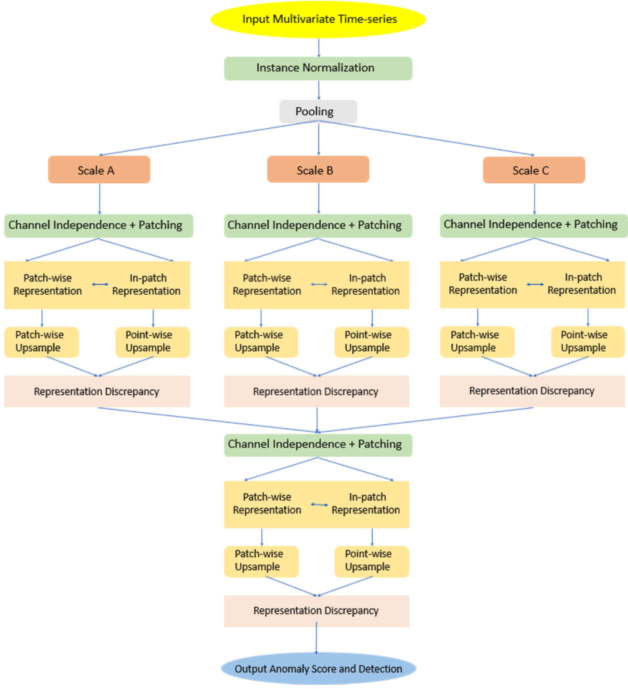
Input Multivariate Time-series
Instance Normalization
Pooling
Scale A
Scale B
Scale C
Channel Independence + Patching
Channel Independence + Patching
Channel Independence + Patching
Patch-wise Representation
In-patch Representation
Patch-wise Representation
In-patch Representation
Patch-wise Representation
In-patch Representation
Patch-wise Upsample
Point-wise Upsample
Patch-wise Upsample
Point-wise Upsample
Patch-wise Upsample
Point-wise Upsample
Representation Discrepancy
Representation Discrepancy
Representation Discrepancy
Channel Independence + Patching
Patch-wise Representation
In-patch Representation
Patch-wise Upsample
Point-wise Upsample
Representation Discrepancy
Output Anomaly Score and Detection

Fig. 2. Multi-scale concatenation model.

2.2 Implementation of Multi-scale Methodology

Multi-scale partitioning is the core optimization method in this paper. The following subsections will elaborate on the implementation details of the multi-scale partitioning model and the multi-scale concatenation model, respectively. In the multi-scale partitioning model, the input time series data is segmented using the Average Pooling method [18]. The core idea of this approach is to perform hierarchical downsampling of the original time series signal using sliding windows of various scales, thereby capturing feature patterns at different temporal resolutions. The implementation process is as follows: for each scale (a positive integer), a pooling window is defined where both the window size and the stride are equal to the scale. This window slides over the input time series with the given stride. For each windowed region, the average value of all data points within the window is computed and used to represent that region, forming a new downsampled time series. This multi-scale pooling process produces a set of subsequences with different temporal resolutions. Pooling windows with larger scales are capable of capturing more global trend features, while those with smaller scales are better suited for extracting local detail variations.

After the pooling operation is completed, in order to avoid information loss and maintain consistent data dimensions, this paper uses an upsampling method to supplement the pooled time series data. Specifically, for each scale, each pooled result is duplicated so that the number of repeated values matches the size of the pooling window. The final time series data input to the model is:

$$Input[i] = \text{Upsampling}(Pooled[i]) \tag{1}$$

For each pooling scale i, the result of the pooled time series data is denoted as $Pooled[i]$. Each is individually fed into the model, where feature extraction is performed through slicing operations. These features are then processed by a dual-attention mechanism to obtain dual-branch representations for each scale. All representations are subsequently flattened into one-dimensional tensors. Finally, these one-dimensional tensors are concatenated to form the model's output, which is then passed into the anomaly decision module to produce the detection results. The overall representation output by the multi-scale partitioning model is denoted as:

$$Rep_D = \text{Concat}(Rep[1], \, \dots \, , Rep[n]) \tag{2}$$

For each scale, the obtained representation is denoted as $Rep[i]$, with a total of n scales. Here, Concat refers to the concatenation operation. The multi-scale concatenation model builds upon the multi-scale partitioning model by adding a concatenation operation for the time series features. The input time series data is also processed through average pooling, segmented into inputs of different scales, and individually fed into the model for feature extraction. Dual-branch representations are then obtained via a dual-attention mechanism. The difference lies in that, after feature extraction for each scale is completed, the feature extraction results from all scales are concatenated together. This concatenated representation is then passed through another dual-attention mechanism, and a new representation is obtained via a differential representation module, denoted as:

$$Rep_{ALL} = Disc(Attn(\text{Concat}(Feature[1], \dots, Feature[n]))) \tag{3}$$

Here, $Feature[i]$ represents the time series feature corresponding to the scale, and there are scales in total. and denote the differential representation module and the dual-attention module, respectively. The representation obtained from the differential representation module is concatenated with the representations corresponding to all individual scales to form the overall representation of the input time series data. The overall representation of the input time series data can be expressed as:

$$Rep_C = \text{Concat}(Rep[1], \, \dots \, , Rep[n], \, Rep_{ALL}) \tag{4}$$

Here, $Rep[i]$ represents the differential representation corresponding to the scale. By feeding the overall representation into the anomaly decision module, the anomaly detection result of the multi-scale concatenation model is obtained.

2.3 The remaining components of the model

The remaining components of our model largely follow the DCdetector architecture. This study employs a dual-attention contrastive structure that extracts time-series representations through two complementary perspectives: patch-wise (inter-patch relationships) and in-patch (intra-patch patterns). The input time series is decomposed into multiple patches characterized by patch quantity, length, and channel dimensions. These patches undergo parallel processing through multi-head self-attention mechanisms - one capturing dependencies between different patches, while another modeling relationships within individual patches. Both attention pathways utilize weight-shared networks with identical computational procedures: embedding transformation followed by scaled dot-product attention and concatenation of multiple attention heads. This dual-path design enables comprehensive temporal feature learning while maintaining parameter efficiency through shared representations between the two perspectives.

The patch-wise and in-patch branches extract features from the same time series by modeling inter-patch and intra-patch dependencies, respectively. The former captures relationships across patches at identical positions, while the latter focuses on interactions within each patch. These dual representations reflect complementary views of the input under structural reordering. The model enforce consistency between the patch-wise and in-patch branches using bidirectional Kullback–Leibler divergence losses. Each loss consists of two asymmetric KL terms with stop-gradient operations, encouraging alignment while preserving independent learning in both branches. During model testing, the anomaly score is defined as [7]:

$$AnoScore(X) = \sum_{i} \sum_{j} [D_{KL}(N_{ij}, \text{Stopgrad}(P_{ij})) + D_{KL}(\text{Stopgrad}(P_{ij}) \, , N_{ij})] \tag{5}$$

Where represents the input time series data, D_{KL} denotes the Kullback-Leibler (KL) divergence, and Stopgrad indicates the gradient stop module. The model's final anomaly detection rule is derived as follows [7]:

$$Y_i = \begin{cases} 1, & AnoScore(X_i) \geq Threshold \\ 0, & AnoScore(X_i) < Threshold \end{cases} \tag{6}$$

Here, X_i represents the i-th time point in the input data, and Y_i denotes the corresponding anomaly detection result for X_i , where $Y_i = 1$ indicates this point is anomalous and $Y_i = 0$ indicates this point is normal.

3 Experiments

3.1 Dataset and Experimental Environment

The paper initially employs a dataset of throttle lever displacement and rotational speed as the preliminary experimental flight parameter dataset, which includes four channels of data: time, low-pressure rotor speed, high-pressure rotor speed, and throttle lever displacement. Both the low-pressure rotor speed and high-pressure rotor speed exhibit a positive correlation with throttle lever displacement. Each entry in this dataset is augmented with an anomalous sequence for detection experiments, encompassing two types of anomalies: numerical anomalies and relational anomalies. Additionally, classic time series anomaly detection datasets such as MSL, UCR, and UCR_AUG are utilized. In the UCR and UCR_AUG datasets, normal data points exhibit regular fluctuations, while anomalous points manifest as irregular abrupt changes, with anomaly lengths ranging from several hundred to several thousand data points. By varying the lengths and types of anomalous sequences, this study achieves comparative experimental results. Notably, the UCR and UCR_AUG datasets are single-channel datasets, each containing approximately 250 entries, while the MSL dataset consists of a single 55-channel dataset.

The Python environment for the model experiments is configured as follows: Python 3.12.9, torch 2.6.0 with CUDA 12.4, pandas 2.2.3, numpy 1.26.4, and einops 0.8.1. The GPU used for the experiments is a vGPU-32GB (32GB) * 1, with 32GB of video memory, and the operating system is Ubuntu 22.04.

3.2 The Detection Performance of the DCdetector Model

A series of experiments were conducted to validate DCdetector's strong performance in detecting long anomaly sequences. Using flight throttle lever displacement and rotational speed data, numerical anomalies were introduced by lowering a segment of low-pressure rotor speed to about 40% below normal. Results in Fig. 3 show the model successfully detected this anomaly. The controlled manipulation preserved relationships between parameters, enabling precise evaluation. These results confirm DCdetector's effectiveness in identifying extended anomalies in complex flight data.

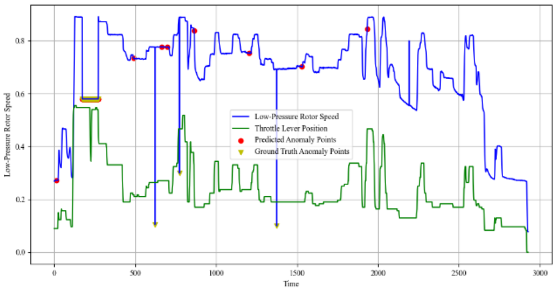

Fig. 3. Numerical anomaly detection results for flight parameter data

The paper then verifies DCdetector's effectiveness in detecting relational anomalies using flight throttle lever displacement and rotational speed data. In normal aircraft system operation, rotor speed and throttle lever displacement exhibit a clear positive correlation where increased throttle displacement leads to higher rotor speed and vice versa. Based on this principle, two test scenarios were established: anomalous cases with isolated sudden changes in low-pressure rotor speed, and normal cases with synchronized directional changes in both low-pressure rotor speed and throttle lever displacement (simultaneous increases or decreases). The anomaly detection results for these scenarios are presented in Fig. 4 and Fig. 5.

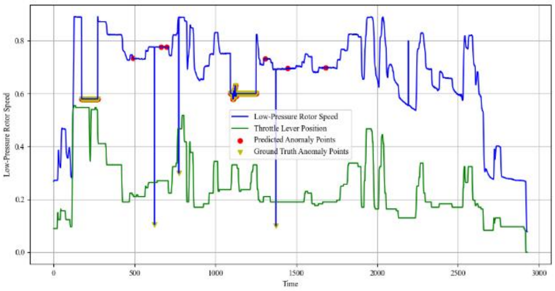

Fig. 4. Detection of isolated anomalies in low-pressure rotor speed

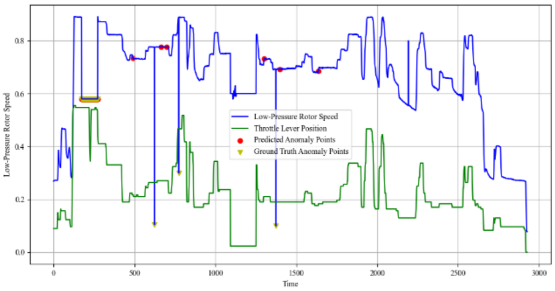

Fig. 5. Detection of co-directional anomalies in low-pressure rotor speed and throttle lever displacement

The detection results in Fig. 4 clearly show the model's ability to identify abnormal conditions when the low-pressure rotor speed experiences an abrupt decrease while other parameters remain stable. Specifically, when the rotor speed drops by more than 12% within 8 seconds without corresponding throttle lever movement, the system successfully raises an alert. In contrast, Fig. 5 presents a normal operational scenario where coordinated decreases in both low-pressure rotor speed and throttle lever displacement are correctly recognized as non-anomalous behavior, demonstrating the model's understanding of proper parameter correlations. These findings confirm DCdetector's dual capability in detecting both isolated numerical deviations and complex relational anomalies in aircraft operational data.

The detection performance is illustrated in Fig. 4, where the model successfully captures the anomalous event characterized by an isolated sudden drop in low-pressure rotor speed (from 85% to 72% within 5 seconds). Conversely, Fig. 5 demonstrates the model's capability to distinguish normal operational scenarios, correctly classifying the synchronized decrease of both low-pressure rotor speed and throttle lever displacement as non-anomalous behavior. These findings validate DCdetector's dual capability in identifying numerical deviations in extended sequences, and recognizing complex relational anomalies through multi-parameter correlation analysis in flight data systems.

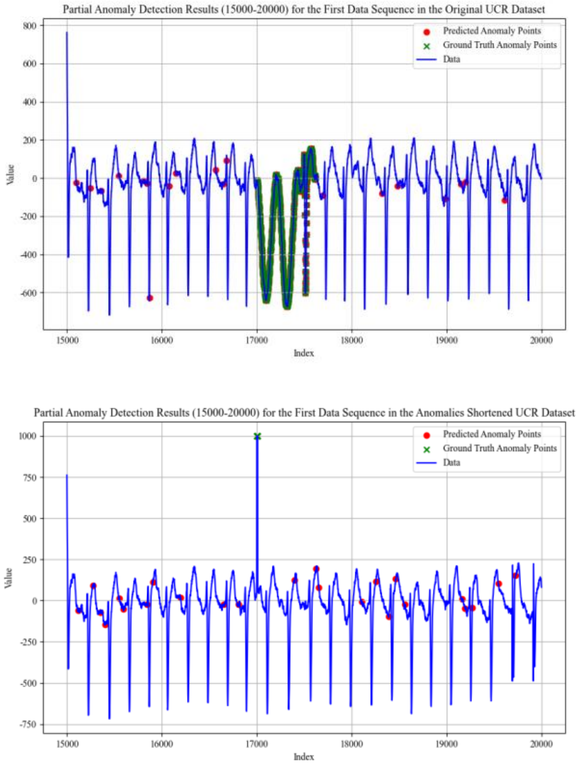

Fig. 6. Example of Short-Sequence Anomaly Construction Method

When investigating the impact of different anomaly segment lengths on model detection performance using throttle lever displacement and rotational speed datasets, we incidentally discovered that anomaly segments shorter than a certain threshold length would completely evade detection. To determine DCdetector's minimum detectable anomaly length threshold, we conducted systematic experiments by varying both anomaly segment length and anomaly magnitude in the throttle lever displacement and rotational speed datasets. The experimental results are presented in Table I.

TABLE I. PRECISION STATISTICS FOR SHORT-SEQUENCE ANOMALY DETECTION ON FLIGHT PARAMETER DATASET

Anomaly	Anomaly Length						
Amplitude	3	5	10	15	20	25	30
0.5	0.00%	0.00%	0.00%	0.00%	86.96%	88.47%	90.91%
0.6	0.00%	0.00%	0.00%	0.00%	86.96%	88.47%	90.91%
0.7	0.00%	0.00%	0.00%	0.00%	86.96%	88.47%	90.91%
0.8	0.00%	0.00%	0.00%	0.00%	86.96%	88.47%	90.91%
1.2	0.00%	0.00%	0.00%	0.00%	86.96%	88.47%	90.91%
1.3	0.00%	0.00%	0.00%	0.00%	86.96%	88.47%	90.91%

Systematic analysis of short anomaly sequence detection on the flight parameter dataset in Table I reveals that DCdetector maintains stable recall rates across anomaly magnitudes from 0.5 to 1.3 times normal values, showing insensitivity to magnitude changes. However, performance strongly depends on anomaly sequence length: recall is consistently 0% for sequences shorter than 15 timesteps, indicating total detection failure, while sequences of 20 timesteps or more see recall rapidly improve to 86.96%-90.91%, highlighting a clear limitation in detecting short-duration anomalies.

To eliminate potential dataset-specific effects of throttle lever displacement and rotational speed data on DCdetector's short anomaly detection performance, we conducted additional experiments on UCR, UCR_AUG and MSL datasets by systematically varying anomaly sequence lengths. Analysis of original test results from these three benchmark datasets enables comprehensive evaluation of DCdetector's anomaly detection performance under standard test conditions. The experiments preserved each dataset's native anomaly length distributions (approximately 300-500 timesteps for UCR/UCR_AUG and 200-400 timesteps for MSL), reflecting real-world anomaly patterns. Table II presents the model's detection performance on unmodified UCR, UCR_AUG and MSL datasets, where average precision and recall rates represent means across all test sequences.

TABLE II. ANOMALY DETECTION PERFORMANCE ACROSS ORIGINAL DATASETS

	Average	Average	Average	Average
Dataset	Anomaly	Prediction	Precision	Recall
	Length	Count		Rate
UCR	460	697	64.13%	100%
UCR	10	212.66	0.56%	14.0%
UCR_AUG	2434.4	2407.73	91.22%	84.53%
UCR_AUG	10	207.26	2.13%	13.3%
MSL	7716	8285	92.17%	98.33%
MSL	10	665	0%	0%

As shown in Table II, while DCdetector demonstrates strong detection performance for long anomaly segments (hundreds to thousands of timesteps) in the original UCR, UCR_AUG, and MSL datasets, its performance deteriorates dramatically when detecting shorter anomalies. When the anomaly segment length was reduced to 10 timesteps across all three datasets, the model's precision and recall rates both dropped to near 0%. For example, on the UCR dataset, the model not only failed to detect the shortened anomalies but also produced numerous scattered false alarms, revealing a systematic limitation in its short-term feature extraction capability. This comparative analysis clearly demonstrates DCdetector's critical deficiency in handling short-duration anomalies despite its effectiveness with longer anomaly patterns.

To analyze this phenomenon further and visually compare detection before and after anomaly modification, representative samples from UCR, UCR_AUG, and MSL were selected. Fig. 6 shows UCR results: while the original data had some scattered false alarms, longer anomalies were detected correctly. However, in the shortened-anomaly version, the model failed to detect anomalies, showing the same scattered false alarms. The original anomalies had irregular fluctuations, but the modified data featured sharp transient spikes. Visualization clearly shows the model completely missed these short spikes, with anomaly scores remaining flat. UCR_AUG results are shown in Fig. 7 and Fig. 8.

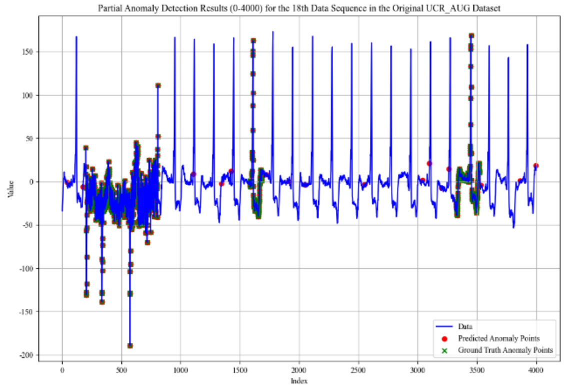

Fig. 7. Detection Results on Original UCR_AUG Dataset Samples

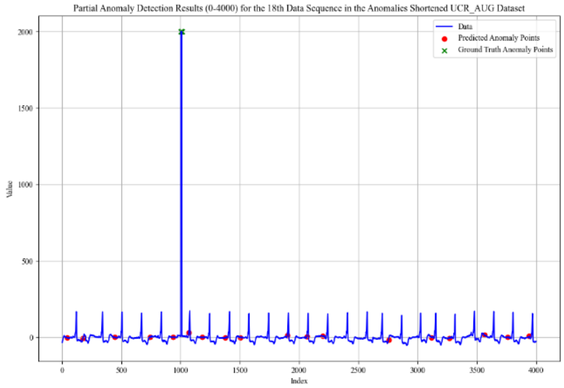

Fig. 8. Detection Results on Shortened UCR_AUG Anomaly Segments

Fig. 7 and Fig. 8 respectively present partial detection results from the UCR_AUG dataset before and after shortening the anomaly segments. The results demonstrate similar detection patterns to the UCR dataset: while the model correctly identified longer anomaly segments in the original data, it failed to accurately detect anomalies in the shortened version while maintaining scattered false alarms. The MSL dataset's detection results before and after modifying anomaly segment length are shown in Fig. 9 and Fig. 10.

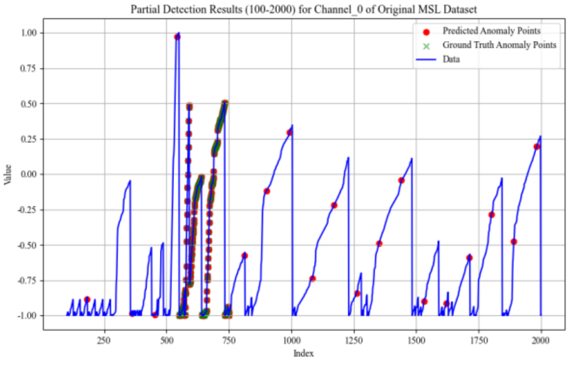

Fig. 9. Detection Results on Original MSL Dataset Sample

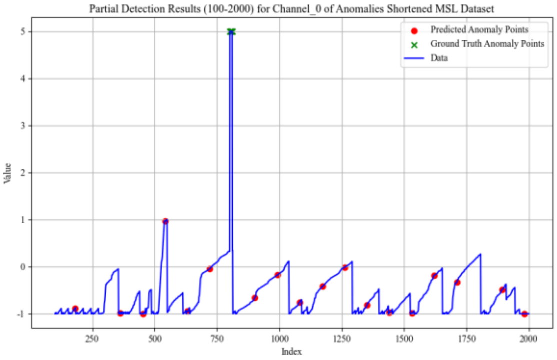

Fig. 10. Detection Results on Shortened MSL Anomaly Segments

The MSL dataset contains 55 channels with only one data record. To achieve clear visualization, we selected partial data from channel 0 for display. Fig. 9 and Fig. 10 present the detection results for MSL's channel 0 before and after shortening the anomaly sequence length. The results align with those from UCR and UCR_AUG: longer anomaly sequences in the original dataset were correctly detected, while shortened anomalies were missed, with scattered false alarms persisting. These experimental results across UCR, UCR_AUG, and MSL datasets conclusively demonstrate DCdetector's poor performance in short-sequence anomaly detection, effectively ruling out dataset-specific effects on the model's short-sequence detection capability.

To assess parameter effects on DCdetector's short anomaly detection, we conducted controlled experiments on UCR, UCR_AUG and MSL datasets with fixed short anomaly lengths while varying window (35-105) and batch sizes (32-128). UCR results are shown in Table III.

TABLE III. DETECTION PERFORMANCE OF DIFFERENT MODEL PARAMETERS ON UCR DATASET

Window Size	Batch Size	Anomaly Length	Average Prediction Count	Average Precision	Average Recall Rate
105	128	10	231	0.00%	0.00%
105	64	10	231	0.00%	0.00%
105	32	10	231	0.00%	0.00%
70	128	10	243	0.00%	0.00%
70	64	10	243	0.00%	0.00%
70	32	10	243	0.00%	0.00%
35	128	10	221.5	0.00%	0.00%
35	64	10	221.5	0.00%	0.00%
35	32	10	221.5	0.00%	0.00%

As evidenced by Table III, DCdetector demonstrates consistently poor detection performance (essentially 0%) on the UCR dataset across varying window sizes (ranging from 105 to 35) and batch sizes (ranging from 128 to 32). None of the tested parameter combinations yielded any improvement in the model's ability to detect short anomaly sequences. To further validate this observation, parallel experiments conducted on the UCR_AUG dataset produced comparable results as shown in Table IV.

TABLE IV. DETECTION PERFORMANCE OF DIFFERENT MODEL PARAMETERS ON UCR_AUG DATASET

Window Size	Batch Size	Anomaly Length	Average Prediction Count	Average Precision	Average Recall Rate
60	128	10	207.26	2.13%	13.3%
60	64	10	207.26	2.13%	13.3%
60	32	10	207.26	2.13%	13.3%
30	128	10	162.08	3.23%	13.3%
30	64	10	162.08	3.23%	13.3%
30	32	10	162.08	3.23%	13.3%
15	128	10	168.4	0.76%	11.1%
15	64	10	168.4	0.76%	11.1%
15	32	10	168.4	0.76%	11.1%

Table IV results demonstrate DCdetector's consistently poor performance (near 0%) on UCR_AUG across all tested window sizes (35-105) and batch sizes (32-128), with no parameter combination improving short anomaly detection. Comprehensive experiments confirm DCdetector's fundamental limitations in detecting short anomaly sequences under various test conditions.

3.3 Performance Validation of the Multi-Scale Segmentation Model

Subsection A showed DCdetector's poor detection of short anomaly sequences. This subsection demonstrates that the multi-scale partitioning model greatly improves short anomaly detection. Table V presents a comparative analysis of the DCdetector model's anomaly detection performance across varying anomaly lengths before and after the integration of multi-scale modules on UCR_1 dataset.

TABLE V. COMPARATIVE PERFORMANCE OF DCDETECTOR MODEL WITH/WITHOUT MULTI-SCALE MODULES ON UCR_1

Anomaly Length	Original Model Precision	Original Model Recall	Multiscale Model Precision	Multiscale Model Recall
200	46.30%	100%	46.95%	100%
150	39.58%	100%	39.06%	100%
120	33.99%	100%	35.09%	100%
100	31.85%	100%	30.79%	100%
80	26.67%	100%	25.48%	100%
50	17.61%	100%	18.80%	100%
45	14.66%	100%	14.98%	100%
40	14.20%	100%	14.37%	100%
38	13.92%	100%	13.71%	100%
35	0.00%	0%	13.26%	100%
33	0.00%	0%	12.41%	100%
30	0.00%	0%	0.00%	0%
28	0.00%	0%	0.00%	0%
25	9.54%	100%	9.65%	100%
23	0.00%	0%	8.95%	100%
20	0.00%	0%	8.03%	100%
18	0.00%	0%	7.11%	100%
15	0.00%	0%	6.15%	100%
13	0.00%	0%	0.00%	0%
10	0.00%	0%	0.00%	0%
8	0.00%	0%	0.00%	0%
5	0.00%	0%	0.00%	0%

Fig. 11 compares results between the original DCdetector (without multi-scale pooling) and the multi-scale partitioning model on the UCR_1 dataset, across varying anomaly segment lengths.

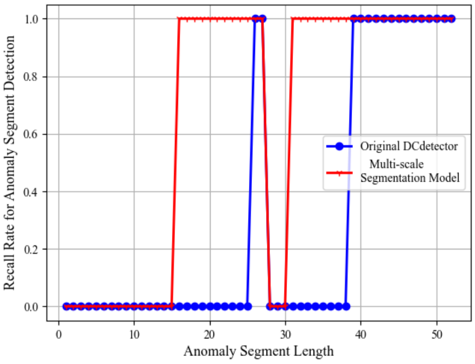

Fig. 11. Comparison of Anomaly Detection Performance With/Without Multi-scale Pooling Module

From Fig. 11, it can be concluded that the model incorporating pooling operations can detect shorter anomaly sequences compared to the original DCdetector model. This clearly indicates that the addition of multi-scale pooling operations significantly improves detection performance for short anomaly sequences, proving the effectiveness of multi-scale pooling for short-sequence anomaly detection tasks.

In the multi-scale partitioning model, different combinations of pooling scales also impact the model's detection performance. Fig. 12 illustrates the detection results under varying pooling scale combinations while maintaining a fixed window size of 105.

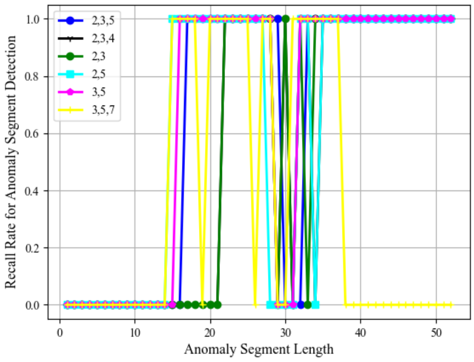

Fig. 12. Comparative Performance of Multi-scale Segmentation Model at Different Scales

Fig. 12 shows significant variations in detection performance across different pooling scale combinations ([2,3,5], [2,3,4], [2,3], [2,5], [3,5], [3,5,7]) in the multi-scale partitioning model. The model achieves slightly better performance with [2,5] or [3,5,7] scale combinations, detecting anomaly sequences as short as 14 timesteps - a marginal improvement over other scales. Similarly, detection performance for each scale exhibits fluctuations when anomaly segments reach approximately 30 timesteps in length.

3.4 Performance Validation of the Multi-Scale Concatenation Model

The multi-scale concatenation model enhances the partitioning model's architecture by processing time-series data at multiple scales through pooling operations. Each scale undergoes dual-attention and difference representation modules, with all scale features subsequently concatenated and processed to generate a comprehensive final representation for anomaly detection. Comparative experiments (Fig. 13) evaluate the concatenation model against the partitioning model and original DCdetector on UCR 1 across varying anomaly lengths.

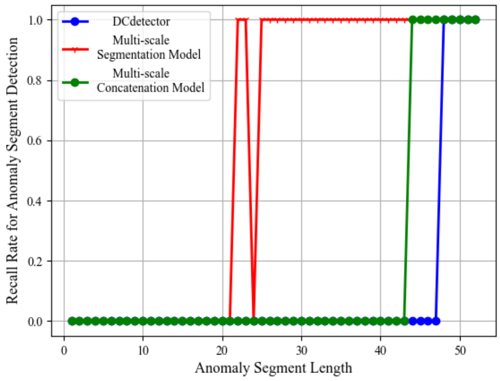

Fig. 13. Comparative Performance of Different Anomaly Detection Models

Fig. 13 presents the anomaly detection results of all three models on UCR_1 using a window size of 90 and pooling scale combination [5,6] for varying anomaly lengths (note the model parameters differ from Fig. 18 due to increased GPU memory requirements after multi-scale concatenation optimization). The results demonstrate that both the multi-scale partitioning model and multi-scale concatenation model outperform the original DCdetector in detecting short anomaly sequences, with the multi-scale partitioning model achieving better minimum detectable anomaly length under these parameters.

Since single-scale evaluation cannot fully characterize the multi-scale concatenation model's detection capability, we conducted additional experiments with fixed window size (90) but varying scale combinations. The experimental results are shown in Fig. 14.

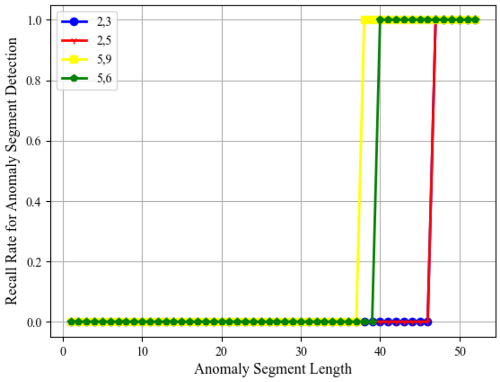

Fig. 14. Comparative Performance of Multi-scale Concatenation Model at Different Scales

Fig. 14 demonstrates the multi-scale concatenation model's detection performance under different scale combinations ([2,3], [2,5], [5,9], [5,6]). The results reveal varying detection capabilities across scales, with the [5,9] combination achieving detection of anomaly segments as short as 38 timesteps. However, overall analysis indicates the multi-scale concatenation model's performance for short anomaly sequences remains slightly inferior to the multi-scale partitioning model.

4 Conclusion

This study develops a comprehensive multi-scale deep learning framework for flight parameter anomaly detection to address critical challenges in aviation safety. Building on the limitations of existing approaches, we propose a novel dual-model architecture that combines multi-scale segmentation and feature concatenation techniques. The framework preserves the strengths of conventional methods in long-sequence detection while introducing innovative mechanisms for improved short-duration anomaly identification. Through systematic theoretical analysis and methodological innovation, we establish an advanced solution that integrates multi-resolution processing with cross-scale feature fusion. The approach demonstrates significant improvements in detecting transient anomalies while maintaining robust performance across varying operational scenarios. This research contributes both theoretical advancements in time-series analysis and practical engineering solutions for aviation safety monitoring, offering a sophisticated yet adaptable framework for flight parameter anomaly detection.

Building upon the current research achievements, several promising directions warrant further exploration in flight parameter anomaly detection. Future studies could investigate dynamic scale-adaptive mechanisms to replace the fixed-scale pooling windows, potentially incorporating meta-learning frameworks or attention-based dynamic weighting for intelligent scale selection. The development of wavelet-based time-frequency joint analysis methods may offer enhanced sensitivity for transient anomaly detection. Algorithm optimization should prioritize real-time performance and lightweight architectures through techniques like model compression, incremental learning, and edge computing deployment. Expanding into multimodal data fusion represents another critical direction, integrating analysis of unstructured data like cockpit voice recordings and surveillance videos with traditional flight parameters. These advancements, combined with the proposed multi-scale optimization approach, could significantly improve aviation system stability by enabling more precise identification of subtle anomalies while maintaining operational efficiency. The continued evolution of deep learning techniques promises to further strengthen the connection between artificial intelligence and aviation safety applications, ultimately leading to more robust and intelligent monitoring systems.

6 References

[1] Bai, S., J. Z. Kolter, and V. Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." arXiv:1803.01271, 2018.

[2] Hochreiter, S., and J. Schmidhuber. "Long short-term memory." Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[3] Vaswani, A., et al. "Attention is all you need." Advances in Neural Information Processing Systems, pp. 5998-6008, 2017.

[4] Yin, C., et al. "Anomaly detection based on convolutional recurrent autoencoder for IoT time series." IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 1, pp. 112-122, 2020.

[5] Zhong, S., et al. "A multi-scale decomposition mlp-mixer for time series analysis." arXiv:2310.11959, 2023.

[6] Cai, W., et al. "MSGNet: Learning multi-scale inter-series correlations for multivariate time series forecasting." Proceedings of the AAAI Conference on Artificial Intelligence, pp. 11141-11149, 2024.

[7] Yang, Y., et al. "DCdetector: Dual attention contrastive representation learning for time series anomaly detection." Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3038-3047, 2023.

[8] Zhang, Y., and J. Yan. "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting." The Eleventh International Conference on Learning Representations, 2023.

[9] Chen, P., et al. "Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting." arXiv:2402.05956, 2024.

[10] Liu, S., et al. "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting." International Conference on Machine Learning, pp. 12345-12356, 2022.

[11] Shabani, A., et al. "Scaleformer: Iterative multi-scale refining transformers for time series forecasting." arXiv:2206.04038, 2022.

[12] Cirstea, R.-G., et al. "Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting." arXiv:2204.13767, 2022.

[13] Audibert, J., et al. "USAD: Unsupervised anomaly detection on multivariate time series." Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3395-3404, 2020.

[14] Tuli, S., G. Casale, and N. R. Jennings. "TranAD: Deep transformer networks for anomaly detection in multivariate time series data." arXiv:2201.07284, 2022.

[15] Schmidl, S., P. Wenig, and T. Papenbrock. "Anomaly detection in time series: A comprehensive evaluation." Proceedings of the VLDB Endowment, vol. 15, no. 9, pp. 1779-1797, 2022.

[16] Deng, A., and B. Hooi. "Graph neural network-based anomaly detection in multivariate time series." Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4027-4035, 2021.

[17] Li, D., et al. "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks." International Conference on Artificial Neural Networks, pp. 703-716, 2019.

[18] Gholamalinezhad, H., and H. Khosravi. "Pooling methods in deep neural networks, a review." arXiv:2009.07485, 2020