

Article

Remaining Useful Life Prediction of Aero-Engine Based on KSFA-GMM-BID-Improved Autoformer

Jiashun Wei, Zhiqiang Li, Yang Li and Ying Zhang *

College of Automobile and Traffic Engineering, Nanjing Forestry University, Nanjing 210037, China; 3239235297@njfu.edu.cn (J.W.); lzq990711@njfu.edu.cn (Z.L.); leey@njfu.edu.cn (Y.L.)

* Correspondence: zhangyingrms@njfu.edu.cn

Abstract: Addressing the limitation of traditional deep learning models in capturing the spatio-temporal characteristics of flight data and the constrained prediction accuracy due to sequence length in aero-engine life prediction, this study proposes an aero-engine remaining life prediction approach integrating a kernel slow feature analysis, a Gaussian mixture model, and an improved Autoformer model. Initially, the slow degradation features of gas path performance parameters over time are extracted through kernel slow feature analysis, followed by the establishment of a Gaussian mixture model to create a health state representation using Bayesian inferred distances for quantifying the aero-engine's health status. Moreover, a spatial attention mechanism is introduced alongside the autocorrelation mechanism of the Autoformer model to augment the global feature extraction capacity. Additionally, a multilayer perceptron is employed to further elucidate the degradation trends, which enhances the model's learning and predictive capabilities for extended sequences. Subsequently, experiments are conducted using authentic aero-engine operational data, comparing the proposed method with the standard Autoformer and Transformer models. The results demonstrate that the proposed method outperforms both models in swiftly and accurately predicting the remaining life of aero-engines with robustness and high prediction accuracy.

Keywords: aero-engine; kernel slow feature analysis; Gaussian mixture model; improved autoformer; life prediction



Citation: Wei, J.; Li, Z.; Li, Y.; Zhang, Y. Remaining Useful Life Prediction of Aero-Engine Based on KSFA-GMM-BID-Improved Autoformer. *Electronics* **2024**, *13*, 2741. <https://doi.org/10.3390/electronics13142741>

Academic Editor: Ahmed Abu-Siada

Received: 14 June 2024

Revised: 9 July 2024

Accepted: 10 July 2024

Published: 12 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The aero-engine serves as the vital component of an aircraft, responsible for generating thrust and propelling flight operations. Given its significant impact on aircraft safety and operational efficiency, the maintenance of aero-engines holds paramount importance [1]. To accurately evaluate engine reliability, the aerospace industry has introduced predictive maintenance and health management (PHM) techniques [2], with Remaining Useful Life (RUL) representing a pivotal aspect. Anticipating the engine's remaining operational lifespan enables the early detection of potential failure issues, facilitating timely maintenance interventions and mitigating resource wastage. This proactive approach not only minimizes costs but also substantially enhances flight safety.

Establishing an appropriate health indicator (HI) holds paramount importance in aero-engine remaining life prediction [3]. Gas path parameters are commonly selected as crucial indicators to assess the engine's health status. However, the extensive flight data encapsulate intricate nonlinear dynamics and fault details, complicating the relationship among different gas path parameters. Therefore, the accurate extraction of data features significantly impacts the precision of HI construction. While traditional methods like Principal Component Analysis [4] and Partial Least Squares [5] reduce high-dimensional data to a lower-dimensional space, practical implementation often challenges the underlying assumptions. Slow feature analysis (SFA) emerges as a novel technique capable of extracting features from time series data that evolve slowly over time, offering comprehensive

insights into industrial processes with minimal algorithmic complexity [6]. In the realm of dynamic correlation feature selection, Liu et al. [7] introduced an SFA algorithm to address high-dimensional data variables and strong correlation issues. To handle nonlinear data effectively, incorporating kernel techniques into slow feature analysis proves beneficial. Notably, Zhang et al. [8] devised a monitoring strategy based on bidirectional dynamic kernel slow feature analysis, utilizing Hotelling's T2 and SPE statistics for fault detection. By integrating multi-source information fusion to leverage extracted detailed features, a holistic approach is employed to construct health factors that characterize equipment status, forming the basis for predicting remaining service life.

The prediction of Remaining Useful Life based on constructed health indicators can be categorized into three main types: physical models, data-driven models, and hybrid models [9]. Data-driven prediction methods, in particular, stand out for their independence from prior knowledge and proficiency in handling complex degradation mechanisms without sacrificing generalization capabilities. Numerous scholars have delved into this area, classifying mainstream approaches into mathematical statistics and artificial intelligence. The mathematical statistics realm encompasses autoregressive models [10] and Markov models [11], among others. Notably, Diana et al. [12] developed three autoregressive models with exogenous variables to estimate the remaining service life of an aluminum plate during crack expansion. On the other hand, artificial intelligence algorithms typically encompass traditional machine learning methods and deep learning techniques, with support vector machines [13], random forests [14], and particle filtering [15] being commonly applied. For instance, W. Li et al. [16] introduced a new data-driven approach to the RUL prediction for metal forming processes under multiple contact sliding conditions. The data-driven approach took advantage of bidirectional long short-term memory (BLSTM) and convolutional neural networks (CNNs). A pre-trained lightweight CNN-based network, WearNet, was re-trained to classify the wear states of workpiece surfaces with high accuracy, and then the classification results were passed into a BLSTM-based regression model as inputs for RUL estimation. Fu et al. [17] introduced a novel approach that combines local-global cooperative learning with least squares support vector machines to enhance the accuracy of bearing service life predictions. Zhou et al. [18] developed an integral bearing fault prognosis framework informed by the well-designed time-domain signal preprocessing to conduct the bearing RUL prediction. This framework was built upon physical feature-oriented signal preprocessing and an associated wavelet neural network (WNN). The results consistently demonstrated prediction accuracy and performance robustness. Kim et al. [19] proposed a Transformer-based RUL estimation model that shows highly competitive RUL estimation performance and a novel Adaptive RUL-wise Reweighting (ARR) technique to tackle the unique data imbalance problem in RUL estimation. Wu et al. [20] proposed the traceless Kalman algorithm and refined the particle filtering algorithm by adjusting particle weights, thereby enhancing the precision of predicting the Remaining Useful Life of lithium-ion batteries.

Compared to traditional machine learning methods, deep learning offers significant advantages in handling complex data, particularly large-scale and high-dimensional datasets. Long short-term memory (LSTM), a popular algorithm and crucial variant of recurrent neural networks, incorporates gating units that effectively capture long-term dependencies, enhancing the modeling capacity for time series data. Yan et al. [21] employed frequency-domain features of the initial signals in gear computation to derive health indicators using ordered neuronal long- and short-term memory networks for Remaining Useful Life (RUL) prediction. Alongside LSTM, other relevant recurrent neural network (RNN) variants like gated recurrent unit (GRU) have emerged. With a more streamlined structure, improved gradient propagation, and memory capabilities, GRU finds broad applicability in various domains. Li et al. [22] developed a gated recurrent unit-depth autoregressive model for predicting rolling bearing failure time and probability. The evolution of deep learning led to the introduction of the Transformer model [23] for natural language processing tasks, subsequently expanding its application to life prediction scenarios. Leveraging the

self-attention mechanism, the Transformer model mines correlation information among sequences, enhancing the capacity to capture long-range dependencies. Liu et al. [24] utilized a multi-branch remain network to create Health Indexes (HIs) characterizing the turbopump bearings' health state, followed by accurate HI prediction using the Transformer model. While the Transformer-based time series prediction model has shown promise, certain deficiencies remain, particularly in the prediction of long-term series. Complex temporal patterns in long series pose challenges for the attention mechanism to extract reliable temporal dependencies, and the sparse attention mechanism may limit information utilization. To address these issues, Wu et al. [25] revolutionized the Transformer model with the introduction of the Autoformer. The Autoformer incorporates a deep decomposition architecture to extract predictive components from intricate time series data and integrates an autocorrelation mechanism to enhance information utilization, resulting in notable performance improvements across multiple sequential tasks.

To enhance the predictive capacity of the model for extended time series data, this paper introduces a method for predicting the remaining life of aero-engines using the KSFA-GMM-BID-improved Autoformer. This approach offers advancements over existing methods for predicting the remaining life of aero-engines in four key aspects:

- (1) The kernel slow feature analysis method is employed to extract depth features concerning the parameters of the aero-engine gas path. By extracting features that degrade slowly over time from the data on aero-engine gas path parameters, significant trends defining the health state of the aero-engine are uncovered. The utilization of the kernel technique enhances accuracy while simultaneously reducing computational complexity.
- (2) A Gaussian mixture model is utilized to establish the health state model, followed by the application of Bayesian inferred distance to precisely define the health state of the aero-engine. Additionally, a health indicator mapping method, which relies on a sparse self-encoder, is introduced to standardize the failure thresholds for predicting the remaining life of engines under various operational circumstances.
- (3) An improved Autoformer-based method is proposed for predicting the remaining life of aero-engines. In this method, a spatial attention mechanism is integrated alongside the autocorrelation mechanism in the Autoformer model to combine temporal and spatial features, thereby bolstering the model's global feature extraction capability. Moreover, the incorporation of a multilayer perceptron aids in extracting features from the trend terms input to the Autoformer decoder, facilitating the identification of degradation features and the overall enhancement of the model's learning and predictive capacities for extended sequences. Validation using real-world aircraft operational data demonstrates that the proposed approach significantly enhances the accuracy of predicting the remaining service life of aero-engines.
- (4) This study centers on the utilization of onboard fast storage recorder data obtained by the Civil Aviation Science and Technology Research Institute of China (CAST). It aims to tackle challenges associated with unrealistic aero-engine simulation data and incomplete experimental data.

The remainder of the paper is structured as follows: Section 2 outlines the principles of the pertinent algorithms, while the subsequent section details the aero-engine remaining service life prediction method introduced in this paper. Following that, Section 4 conducts the selection of gas path parameters and data preprocessing, and in Section 5, the accuracy of the proposed method is validated through experiments. Lastly, Section 6 presents the conclusion.

2. Algorithm Theory

2.1. Kernel Slow Feature Extraction Based on KSFA

2.1.1. Slow Feature Analysis

Slow feature analysis (SFA) is a technique designed for identifying features that evolve gradually over time within input signal vectors. By isolating the slow-changing elements of the signal, SFA effectively diminishes rapidly changing noise, enabling the discovery of

concealed features within intricate signals. The objective is to ascertain a mapping function that aligns the output variable with the progressively changing features observed over time.

If a dimensional time series input signal is given as

$$x(t) = [x_1(t), x_2(t), \dots, x_M(t)] \quad (1)$$

then $t \in [t_0, t_1]$ denotes the time range.

Determine the mapping function, denoted as Equation (1), that ensures that the output variable in Equation (2) evolves gradually over time. This evolution typically mirrors the change rate, serving as an indicator of the mean of the squared first-order derivatives concerning time.

$$g(x) = [g_1(x), g_2(x), \dots, g_N(x)] \quad (2)$$

$$f(t) = [f_1(t), f_2(t), \dots, f_N(t)] \quad (3)$$

Construct optimization goals:

$$\min_{g_n} \Delta(f_n) = \left\langle (\dot{f}_n(t))^2 \right\rangle_t \quad (4)$$

where $\dot{f}_n(t)$ is the first-order derivative of $f_n(t)$, and $\langle \cdot \rangle_t$ is the computed mean.

Hypotheses (5), (6), and (7) guarantee that every variable maintains a mean of 0 and a variance of 1 and remains uncorrelated with the other variables. Moreover, each variable is stipulated to change at a slower rate than the subsequent variable in sequence.

$$\langle f_n(t) \rangle_t = 0 \quad (5)$$

$$\left\langle (f_n(t))^2 \right\rangle_t = 1 \quad (6)$$

$$\langle f_m(t)f_n(t) \rangle_t = 0, \quad \forall m < n \quad (7)$$

Each slow feature $f(t)$ undergoes a linear feature transformation to present it in a linear combination form of the original input signal. This transformation elucidates the intricate relationship between the slow feature and the original input signal, facilitating a clearer understanding. The resultant linear combination form is expressed as follows:

$$f_n(t) = w_n^T x \quad (8)$$

where w_n^T is the mapping vector.

In slow feature analysis, whitening is essential to optimize the problem solution by ensuring the independence of individual variables. This whitening process is commonly achieved through singular value decomposition (SVD).

The statistical properties of the input signal are characterized by defining a covariance matrix, depicted as follows:

$$P_1 = \left\langle x(t)x^T(t) \right\rangle_t \quad (9)$$

According to the theory of singular value decomposition, a matrix can be decomposed into a product of three matrices, i.e., the original matrix is equal to the left singular matrix multiplied by a diagonal matrix and multiplied by the transpose of the right singular matrix:

$$P_2 = U \Lambda U^T \quad (10)$$

where U and Λ denote the eigenmatrix and eigenvalue diagonal array, respectively. According to the whitening matrix $P = \Lambda^{-\frac{1}{2}} U^T$, the whitened data are

$$Z = \Lambda^{-\frac{1}{2}} U^T x = Px \quad (11)$$

Substituting Equation (10) into Equation (7) gives

$$f_n(t) - w_n^T x = w_n^T P^{-1} z = EZ \quad (12)$$

where $E = w_n^T P^{-1}$. Clearly $\langle ZZ^T \rangle_t = P \langle xx^T \rangle PT = I$ is obtained, which follows from the following constraints:

$$\langle f_n(t) f_n(t)^T \rangle_t = E \langle ZZ^T \rangle E^T = EE^T = I \quad (13)$$

Thus, E is an orthogonal matrix, and the objective function $\left\langle (\dot{f}_n(t))^2 \right\rangle_t$ can be expressed as

$$\left\langle (\dot{f}_n(t))^2 \right\rangle_t = r_n^T \left\langle \dot{z} \dot{z}^T \right\rangle_t r_n \quad (14)$$

Decomposition based on singular values is expressed as follows:

$$\left\langle \dot{z} \dot{z}^T \right\rangle_t = E^T B_1 E \quad (15)$$

where $B_1 = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$ is the eigenvalue matrix with sequentially increasing eigenvalues, with the objective function $\left\langle (\dot{f}_n(t))^2 \right\rangle_t = \lambda_i$. The mapping matrix w_n^T consisting of the mapping vector $W_1 = [w_1, w_2, \dots, w_M]^T$ is given by

$$W_1 = EP = E\Lambda^{-\frac{1}{2}}U^T \quad (16)$$

Therefore the optimization problem of SFA is to choose the feature vector λ_1 corresponding to the minimum feature value w_1 as the mapping vector to solve for $\left\langle (\dot{f}_1(t))^2 \right\rangle_t$ and so on to obtain other slow features.

2.1.2. Kernel Slow Feature Analysis

Aero-engines exhibit significant nonlinearities in the gas path parameters during operation. The kernel technique proves to be a successful approach in handling nonlinear dependencies and has been employed for addressing nonlinear challenges in slow feature characterization [26]. Kernel slow feature analysis (KSFA) utilizes kernel functions instead of the conventional polynomial expansion method, offering improved capabilities in handling nonlinear dependencies. The fundamental concept involves mapping the original data from a lower-dimensional space to a higher-dimensional space and utilizing kernel functions to perform the inner product operation in this expanded space to effectively process the nonlinear dependencies.

Various types of commonly employed kernel functions exist, and this paper utilizes the prevalent Gaussian kernel function known for its robust generalization and effective smoothing estimation properties. The Gaussian kernel function facilitates the transformation of original data into a higher-dimensional space, enabling the capture of nonlinear characteristics through the computation of inner products within this expanded space.

$$G_{ij} = G(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{c}\right) \quad (17)$$

Here, x_i and x_j are arbitrary samples.

Substituting the polynomial expansion with the Gaussian kernel function, specifically the transformed data denoted as $\dot{G}$ instead of the second-order polynomial expansion, alters the objective function of the mapping function in Equation (4) to

$$\min \Delta(s_j(t)) = \min w_j^T \left\langle \begin{matrix} \dot{G}^T & \dot{G} \end{matrix} \right\rangle_t w_j \quad (18)$$

The linear combination form in Equation (8) can be recast as

$$A_g W = B_g W_2 \Omega \quad (19)$$

where $A_g = \left\langle \begin{matrix} \dot{G}^T & \dot{G} \end{matrix} \right\rangle_t$ is the covariance matrix of $\dot{G}$, and $B = \langle G^T G \rangle_t$ is the covariance matrix of G .

Solving Equation (19) yields the transformation matrix W . The kernel slow feature can be expressed as

$$S_g = W_2 x \quad (20)$$

2.2. Health Indicator Construction

2.2.1. GMM-BID

The Gaussian mixture model (GMM) is a clustering algorithm that relies on a model encompassing multiple Gaussian probability density functions. It leverages the Expectation Maximization (EM) algorithm for training a Gaussian distribution model [27]. GMM serves as a straightforward extension of the Gaussian model by incorporating various combinations of Gaussian distributions to replicate a data distribution, as denoted by the following:

$$p(x_1) = \sum_{m=1}^M \pi_m p(x_1 | \theta_m) \quad (21)$$

where M represents the number of single Gaussian models, π_m is the weight coefficient of the single Gaussian model in the mixture model, and $\sum \pi_m = 1$ denotes the $p(x | \theta_m)$ Gaussian distribution function with a mean of it and covariance matrix. Noting $\phi = \{\pi_1, \dots, \pi_m; \mu_1, \dots, \mu_m; S_1, \dots, S_m\}$, with these parameters in mind, the equation can be expressed as follows:

$$p(x_2 | \phi) = \sum_{m=1}^M \pi_m p(x_2 | \theta_m) \quad (22)$$

Following the modeling process described above, the research establishes a Gaussian mixture model (GMM) as a benchmark for the health state using normal data. In this study, a Bayesian Inference-based Distance (BID) is utilized as a fusion metric to quantitatively evaluate the health condition of the aero-engine.

Given K Gaussian classifications, C_k is the k th component among them, and the probability of occurrence of each component is denoted as α_k , and at this time, for the test point belonging to the k th component, the probability of C_k can be denoted as $p(C_k | x_t)$:

$$p(C_k | x_t) = \frac{\alpha_k p(x_t | C_k)}{p(x_t)} = \frac{\alpha_k p(x_t | C_k)}{\sum_{i=1}^K \alpha_i p(x_t | C_i)} \quad (23)$$

where α_k is the a priori probability, derived from the modeling data. $p(x_t | C_k)$ can be expressed as

$$p(x_t | C_k) = \frac{1}{(2\pi)^{\frac{1}{2}} |S_k|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (x_t - \mu_k)^T S_k^{-1} (x_t - \mu_k) \right] \quad (24)$$

where the mean of the k th Gaussian component is μ_k , and the covariance matrix is S_k . The distance from x_t to each component C_k can be expressed as

$$D_{C_k}(x_t) = (x_t - \mu_k)^T S_k^{-1} (x_t - \mu_k) \quad (25)$$

The BID metric represents a weighted sum of the distance of each component for a single test point and can be expressed as

$$BID = \sum_{k=1}^K p(C_k|x_t) D_{C_k}(x_t) \quad (26)$$

2.2.2. Kalman Filtering

Kalman Filtering (KF) is a recursive and unbiased estimation algorithm that aims to minimize the linear mean square deviation. It relies on the system's state space equations to estimate the hidden states of a linear system [28]. By filtering the metrics post-feature fusion, the KF algorithm can effectively capture the degradation trend of the engine. The algorithm comprises two key steps: prediction and updating.

(1) Prediction

Let us assume that the state transfer equation and measurement equation of a system are defined as follows:

$$\begin{cases} c_j = f(c_{j-1}, w_{j-1}) \\ z_j = h(c_j, v_j) \end{cases} \quad (27)$$

where c_j and z_j are the state values and measurements of the system at the moment j ; $f(\cdot)$ and $h(\cdot)$ are the state transfer function and the measurement function of the system.

Calculate the one-step state prediction of the system:

$$\hat{c}_{j|j-1} = T\hat{c}_{j-1} + Bu_j \quad (28)$$

where $\hat{c}_{j-1}$ is the state estimate of the system at moment $j-1$; $\hat{c}_{j|j-1}$ is the one-step estimate of $\hat{c}_{j-1}$; T is the state transfer matrix; B is the state control matrix; and u_j is the state control vector.

Calculate the one-step prediction error covariance matrix $P_{j|j-1}$:

$$P_{j|j-1} = TP_{j-1}T^T + Q_{j-1} \quad (29)$$

where P_{j-1} is the error covariance matrix at the moment of $j-1$; $P_{j|j-1}$ is the one-step prediction value of P_{j-1} ; and Q_{j-1} is the process noise covariance at the moment of $j-1$.

(2) Update

Calculate the Kalman filter gain matrix J_j :

$$J_j = P_{j|j-1}H^T (HP_{j|j-1}H^T + R_j)^{-1} \quad (30)$$

where J_j is the Kalman gain; H is the measurement matrix; and R_j is the measurement noise covariance at the moment of j .

Update the state vector estimate $\hat{c}_j$:

$$\hat{c}_j = \hat{c}_{j|j-1} + J_j(z_j - H\hat{c}_{j|j-1}) \quad (31)$$

Update the error covariance matrix P_j :

$$P_j = (I - J_jH)P_{j|j-1} \quad (32)$$

where I is the unit matrix.

2.2.3. Sparse Autoencoder-Based Health Indicator Mapping

Sparse Autoencoders (SAEs) are unsupervised neural network algorithms derived from the Autoencoder (AE) structure. They incorporate sparsity constraints to encourage sparse unit activations within the hidden layer during training [29]. The autoencoder comprises two essential components: the encoder and the decoder. When encoding the input data, the encoder function processes it, converting the input data $\{L_1, L_2, L_3, \dots, L_n\}$ into hidden layer vectors, as computed below:

$$h_{w,b}(L) = s(W^{(1)}L + b^{(1)}) \quad (33)$$

where s is the sigmoid excitation function, $W(1)$ is the weight, and $b(1)$ is the bias term coefficient.

In the decoder, the output data are reconstructed by decoding the hidden layer vector via the decoding function. This reconstruction process generates an error, aiming to minimize it to ensure consistency between the input and output data. The formula for error calculation is as follows:

$$J(W, b) = \frac{1}{n} \sum_{i=1}^n \|L_i - \hat{L}_i\|^2 + \left[\frac{\theta}{2} \sum_{i=1}^{m_i} \sum_{j=1}^{s_i} \sum_{j=1}^{s_{j-1}} (W_{ji}^{(1)})^2 \right] \quad (34)$$

where $\hat{L}_i$ is the output data after reconstruction; n is the number of samples; and θ is the weight attenuation coefficient.

After decoding by the decoder, the network weights are adjusted using the back-propagation algorithm to minimize the error. The sparse autoencoder aims to enhance neural network performance by incorporating sparse constraints during training. This ensures that certain neuron nodes in the hidden layer are suppressed while maintaining activity in nodes relevant to input data. The network is commonly constrained using KL sparsity as follows:

$$KL(p||\hat{p}) = \beta \sum_{j=1}^{s_2} p \log \frac{p}{\hat{p}_j} \quad (35)$$

where β is the weight value of the penalty factor; p is the sparsity constant. The total cost function is calculated by the following equation:

$$J_s(W, b) = J(W, b) + \beta \sum_{j=1}^{s_2} KL(p||\hat{p}_j) \quad (36)$$

When employing the backpropagation algorithm for fine-tuning, the primary objective is to iteratively optimize the network's parameters by computing gradients and applying the gradient descent method to make incremental adjustments to weights and bias terms based on the gradient direction. This iterative process aims to better align the network with the training data, thereby enhancing learning effectiveness.

To standardize the engine failure thresholds across various failure scenarios, an improved sigmoid function is employed as the mapping function to transform the reconstruction error into health indicators within the (0, 1) interval as outlined below:

$$HI(E(l, \bar{l})) = \frac{1}{1 + e^{gE(l, \bar{l}) - i}} \quad (37)$$

where g is the shape factor; i is the bias constant.

2.3. Remaining Life Prediction Based on Improved Autoformer

2.3.1. Autoformer

The Transformer model, renowned for its robust capacity to capture long-range dependencies and extract features, as well as its capability to offer some level of interpretability for model decisions, has found extensive applications in natural language processing. Its integration into the realm of time series analysis has led to the development of several advanced models for time series forecasting, such as the Reformer, Informer, and Autoformer. Particularly, the Autoformer innovatively merges the autocorrelation mechanism with the sequence decomposition module to effectively process time series data, depicted in Figure 1 below.

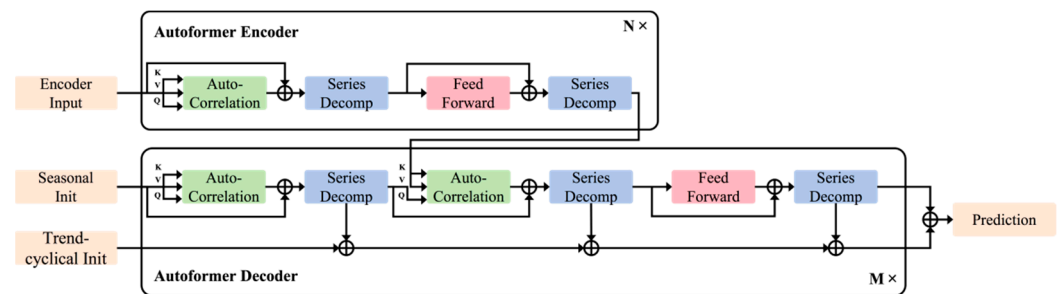


Figure 1. The model architecture of the Autoformer.

(1) Autocorrelation Mechanism

The autocorrelation mechanism was introduced to enhance the self-attention mechanism for capturing autocorrelation details within long-term time series data while also decreasing computational complexity. This mechanism comprises two key components: cycle-based dependence, utilized for autocorrelation computation, and time-delayed information aggregation, employed for merging akin subsequence details. Figure 2 below illustrates the schematic diagram of the autocorrelation mechanism.

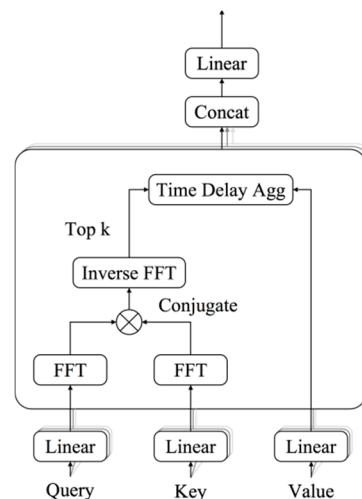


Figure 2. Auto-correlation mechanism.

Period-based dependence enables the identification of subsequences positioned in the same phase across different periods. Guided by stochastic process theory, the autocorrelation coefficient $R_{XX}(\tau)$ for the discrete time series $\{X_t\}$ is computed using the following formula:

$$R_{XX}(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^{N-1} X_t X_{t-\tau} \quad (38)$$

where N is the length of the sequence; τ is the lag moment; and $R_{XX}(\tau)$ is the similarity between the sequence $\{X_t\}$ and its τ lag sequence $\{X_{t-\tau}\}$.

During the aforementioned computation process, the relationship among cycles indicates the subsequences sharing the same phase within the underlying cycle, characterized by its sparsity. For improved efficiency in calculating the period dependence, $R_{XX}(\tau)$ can be determined using the Fast Fourier Transform (FFT) following the Wiener–Khinchin theory. The detailed operational procedure is as follows:

$$S_{XX}(f) = F(X_t)F^*(X_t) = \int_{-\infty}^{\infty} X_t e^{-i2\pi f t} dt \int_{-\infty}^{\infty} X_t e^{-i2\pi f t} dt \quad (39)$$

$$R_{XX}(\tau) = F^{-1}(S_{XX}(f)) = \int_{-\infty}^{\infty} S_{XX}(f) e^{i2\pi f \tau} df \quad (40)$$

where F is the FFT; F^* is the conjugate operation; F^{-1} is the inverse operation; and $S_{XX}(f)$ is a function in the frequency domain.

The figure demonstrates the process of cycle-dependent operation, after inputting the original sequence of length L , three vectors $Query(Q)$, $Key(K)$, and $Value(V)$ can be obtained after linear transformation. The FFT operation is performed on Q , the conjugate operation of FFT is performed on K , and then the inverse operation is performed on the $S_{XX}(f)$ obtained by multiplying the results to find the final $R_{XX}(\tau)$.

The time delay information aggregation adjusts the sequence based on the chosen time delay to synchronize similar subsequences at identical phase positions within the cycle. Subsequently, the softmax function normalizes the confidence levels to combine these subsequences. Within the single-head autocorrelation mechanism, the initial sequence of length L undergoes the subsequent algorithmic procedure to derive the Q , K , and V vectors:

$$\tau_1, \dots, \tau_k = \arg\text{Topk}(R_{Q,K}(\tau))_{\tau \in \{1, \dots, L\}} \quad (41)$$

$$R_{Q,K}(\tau_1), \dots, R_{Q,K}(\tau_k) = \text{SoftMax}(R_{Q,K}(\tau_1), \dots, R_{Q,K}(\tau_k)) \quad (42)$$

$$\text{Auto-Correlation}(Q, K, V) = \sum_{i=1}^k \text{Roll}(V, \tau_i) R_{Q,K}(\tau_i) \quad (43)$$

In the formula, $\arg\text{Topk}$'s function is to take the maximum k autocorrelation coefficients; $R_{Q,K}$ is the autocorrelation coefficient of Q and K ; $\text{Roll}(X, \tau)$ is to put the sequence X for a rolling time delay τ .

In the multi-head autocorrelation mechanism, the Q , K , and V matrices of the i th head are Q_i , K_i , V_i , and $i \in \{1, \dots, h\}$, respectively, and the process is as follows:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= W_{\text{output}} \times \text{Concat}(\text{head}_1, \dots, \text{head}_h) \\ \text{head}_i &= \text{Auto-Correlation}(Q_i, K_i, V_i) \end{aligned} \quad (44)$$

where W_{output} is the trainable matrix. The computational complexity of the autocorrelation mechanism is $O(\log L)$ since a sequence of $O(\log L)$ of length L is aggregated in the time delay information aggregation module.

(2) Sequence Decomposition Module

The sequence decomposition module involves breaking down the original time series into trend and periodic components to facilitate learning the intricate temporal patterns in long-term forecasting scenarios. The trend component captures the gradual evolution of the time series, while the periodic component captures cyclic variations. To address future series' unpredictability, this module integrates into the Autoformer framework to decompose known series by adjusting moving averages to dampen cyclical fluctuations,

extracting the trend component; subsequently, it subtracts the trend to derive the periodic term. The detailed process unfolds as follows:

$$X_t = \text{AvgPool}(\text{Padding}(X_i)) \quad (45)$$

$$X_s = X - X_t \quad (46)$$

where X_i is the original sequence; $X_{en}^l = \text{Encoder}()$ is the decomposed trend term; X_s is the decomposed period term; $\text{Padding}(\cdot)$ is the padding operation on the first and last of the sequence with neighboring sequences; and $\text{AvgPool}(\cdot)$ is the moving average operation.

(3) Encoder

In the Autoformer's architecture, the encoder is stacked by three modules, the autocorrelation mechanism, the sequence decomposition module, and the feed-forward network layer, and the encoder can also be stacked multiple times. The encoder mainly focuses on extracting the periodic features of the sequence, and the sequence decomposition module can extract the deeper implicit periodic features after multiple stacks, which will also be used as cross-information to help the decoder further refine the prediction results after the output of the encoder. Assuming that an encoder has an encoder layer, the computational equation of the l th encoder layer is $X_{en}^l = \text{Encoder}(X_{en}^{l-1})$, and the computational process of the encoder layer is specified as follows:

$$S_{en,-}^{l,1} = \text{SeriesDecomp}\left(\text{Auto} - \text{Correlation}\left(X_{en}^{l-1}\right) + X_{en}^{l-1}\right) \quad (47)$$

$$S_{en,-}^{l,2} = \text{SeriesDecomp}\left(\text{FeedForward}\left(S_{en}^{l,1}\right) + S_{en}^{l,1}\right) \quad (48)$$

where “ $-$ ” is the trend term part; $S_{en,i}^{l,i} \in \{1,2\}$ is the periodic term part of the two sequence decomposition modules of the l th encoder layer, respectively; and $X_{en}^l = S_{en}^{l,2}$, $l \in \{1, \dots, N\}$ is the output of the l th encoder layer.

(4) Decoder

The decoder follows a structure similar to the encoder, with an additional autocorrelation mechanism and sequence decomposition module compared to the encoder. Within the decoder, the first autocorrelation mechanism refines predictions, the second autocorrelation mechanism interacts with the encoder output to leverage historical periodic patterns, and the trend terms from three sequence decomposition modules are aggregated and integrated with periodic terms to yield the ultimate prediction outcome. Assuming a decoder with decoder layers, the computation equation of the l th decoder layer is denoted as $X_{de}^l = \text{Decoder}(X_{de}^{l-1}, X_{en}^N)$. The computational process of the decoder layer is outlined as follows:

$$S_{de}^{l,1}, T_{de}^{l,1} = \text{SeriesDecomp}\left(\text{Auto} - \text{Correlation}\left(X_{de}^{l-1}\right) + X_{de}^{l-1}\right) \quad (49)$$

$$S_{de}^{l,2}, T_{de}^{l,2} = \text{SeriesDecomp}\left(\text{Auto} - \text{Correlation}\left(S_{de}^{l,1}, X_{en}^N\right) + S_{de}^{l,1}\right) \quad (50)$$

$$S_{de}^{l,3}, T_{de}^{l,3} = \text{SeriesDecomp}\left(\text{FeedForward}\left(S_{de}^{l,2}\right) + S_{de}^{l,2}\right) \quad (51)$$

$$T_{de}^l = T_{de}^{l-1} + W_{l,1} * T_{de}^{l,1} + W_{l,2} * T_{de}^{l,2} + W_{l,3} * T_{de}^{l,3} \quad (52)$$

where $S_{de,i}^{l,i}, T_{de,i}^{l,i} \in \{1,2,3\}$ is the period term and trend term decomposed by the i th sequence decomposition module in the l th decoder layer; $X_{de}^l = S_{de}^{l,3}$, $l \in \{1, \dots, M\}$ is the output of the l th decoder layer.

2.3.2. Spatial Attention Module

Various spatial locations within input data frequently exhibit distinct contributions to the model in which they reside. The spatial attention module (SAM) serves as a technique aiding in the analysis of spatial information to discern regions of heightened contribution to the model [30]. At its essence, SAM allocates attention weights to different input data locations, facilitating the model in prioritizing information critical to the current task at hand.

In the realm of aero-engine remaining life prediction, the spatial attention mechanism enhances the focus on spatial location information pertaining to the health indicator sequence, assigning a higher weight to crucial spatial data. This approach ultimately enhances the precision of the prediction model. The principles and structure of the spatial attention mechanism (SAM) are illustrated in Figure 3 below.

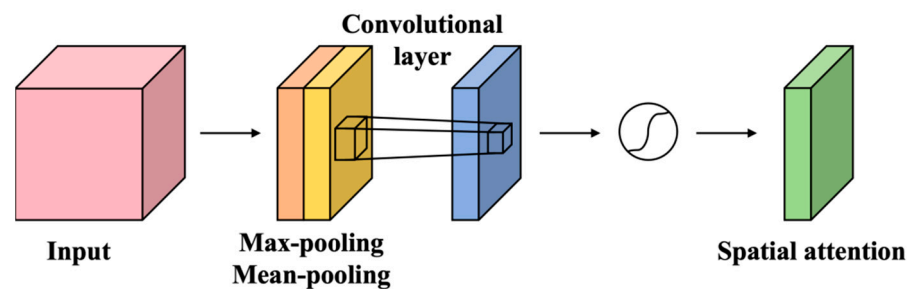


Figure 3. Spatial attention module.

The spatial attention mechanism is initiated by analyzing input features through a combination of maximum pooling and average pooling layers, aimed at capturing location-specific importance. Subsequently, the outcomes are consolidated, and a singular convolution kernel is employed to conduct convolutional operations on the merged vectors, followed by processing through a sigmoid activation function to derive weight coefficients for varied sequence positional information. These weight coefficients are then utilized to perform inner product operations with the initial input features, and the resultant product is added back to the original input features to generate the ultimate output features.

2.3.3. Multilayer Perceptron

The multilayer perceptron (MLP) is a feedforward artificial neural network comprising multiple single-layer perceptrons [31]. In an MLP, the output layer of one perceptron serves as the input layer for the subsequent perceptron, thus propagating forward until the final layer of the network is reached, and the resulting output reflects the MLP output. Consequently, the fundamental components of a multilayer perceptron can be categorized into three structures: input layer, hidden layer, and output layer, each fully interconnected. Figure 4 depicts a straightforward multilayer perceptron model with a sole hidden layer.

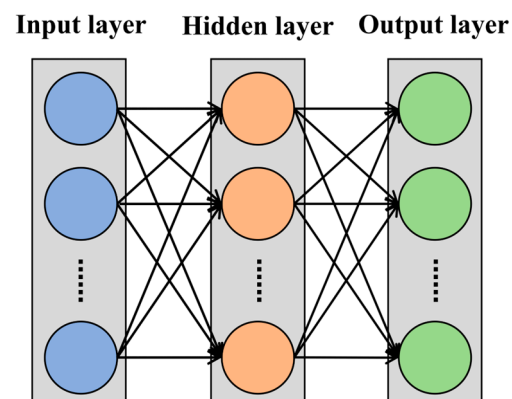


Figure 4. Multilayer perceptron with single hidden layer.

The initial layer, known as the input layer, primarily serves to receive input data and ensures that each input value correlates with a neuron to transmit it to the subsequent layer. Subsequent to the input layer is the hidden layer, responsible for processing the output from the preceding layer via weighted combination operations and nonlinear functions, forwarding it to succeeding layers until it reaches the ultimate output layer.

Each layer within the neural network of the multilayer perceptron comprises numerous neurons, with the functionalities of individual neurons detailed in Figure 5 below.

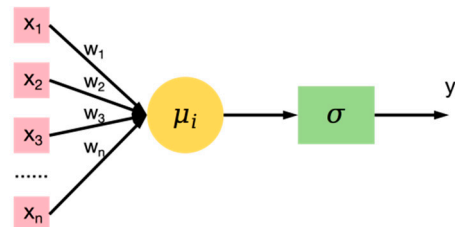


Figure 5. The operation of neurons.

In the figure, $i_1, i_2, \dots, i_n$ is the input to the neuron, $r_1, r_2, \dots, r_n$ is the corresponding weight, and the matrix form of the two is

$$I = [i_1, i_2, i_3, \dots, i_n]^T \quad (53)$$

$$R = [r_1, r_2, r_3, \dots, r_n]^T \quad (54)$$

μ_i is a weighted summation of the input vectors:

$$z = \sum_{i=1}^n R_i I_i + b_k \quad (55)$$

where b_k is the bias of the neuron node. After weighted summation, the output value is obtained after the activation function:

$$\text{output} = g(R_i I_i + b_k) \quad (56)$$

2.3.4. Improvement of Autoformer

The standard Autoformer model effectively captures temporal correlations among various time steps within a sequence through the stacking of modules like the autocorrelation mechanism and sequence decomposition, thereby extracting features from lengthy sequences. In this study, an improved Autoformer model is introduced, which integrates a spatial attention mechanism into the standard model. This enhancement incorporates a parallel structure to amalgamate temporal and spatial features, thus bolstering the model's global feature extraction capacity. Moreover, a multilayer perceptron is utilized to further extract trend-related features, enhancing the predictive accuracy of the model. The comprehensive architecture of the improved Autoformer model is illustrated in Figure 6 below.

(1) Encoder

Within the encoder segment of the improved Autoformer model, the initial autocorrelation mechanism adeptly captures temporal correlations across various time steps in the engine health indicators. However, it overlooks the spatial aspects of the sequence. To address this gap, a spatial attention mechanism is concurrently introduced alongside the autocorrelation mechanism module to encompass both temporal and spatial characteristics of the original health indicator sequences. The resultant feature values from these mechanisms are then integrated to yield a deeper understanding of the original sequences. Subsequently, the outcome is fragmented into multiple sub-sequences by the sequence decomposition module, emphasizing the extraction of high-order periodic term features while minimizing computational complexity. Furthermore, the feed-forward network layer in the encoder elevates the model's capacity to grasp intricate feature representations.

(2) Decoder

In the decoder section of the improved Autoformer model, the module structure closely mirrors that of the encoder. Similarly, a spatial attention mechanism is concurrently incorporated alongside the autocorrelation mechanism, enabling the fusion of feature values from both mechanisms and feeding them into the sequence decomposition module. Additionally, trend-related features are extracted using a multilayer perceptron, and the resulting features are combined with the processed periodic term outcomes. Following successive module operations, the outputs from the periodic and trend components are aggregated to derive the ultimate prediction results.

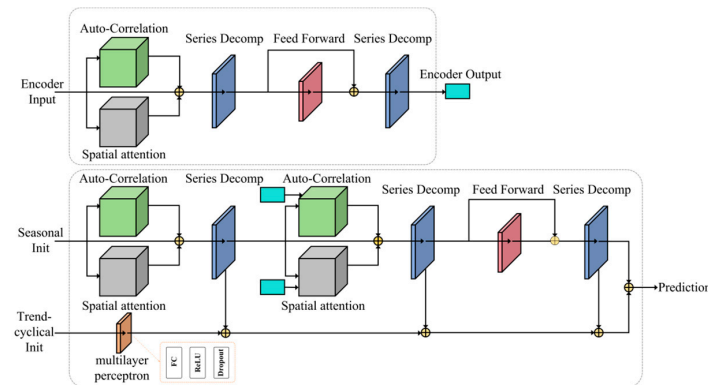


Figure 6. The model architecture of the improved Autoformer.

3. Aero-Engine Remaining Life Prediction Model Based on KSFA-GMM-BID-Improved Autoformer

Following the aforementioned analysis, this paper introduces an aero-engine remaining life prediction methodology leveraging the KSFA-GMM-BID-improved Autoformer, with a visual representation of the process depicted in Figure 7.

As illustrated in Figure 7, the proposed method for predicting the remaining service life of aero-engines consists of the following steps:

(1) Feature extraction

The nuclear slow feature analysis method is employed for feature extraction. During the offline phase, health state training data are chosen to build the nuclear slow feature analysis model and derive its parameters. Subsequently, in the online phase, the nuclear slow feature analysis is applied to test data to extract the slow change features.

(2) Feature fusion

The GMM-BID method is utilized for feature fusion. During the offline phase, health state training data are chosen to build the Gaussian mixture model (GMM) and determine its parameters. Subsequently, in the online phase, the global distance between the test data and the health state model is assessed using Bayesian inferred distance.

(3) HI mapping

The health indicator mapping is achieved through a sparse autoencoder and an improved sigmoid function. Initially, the health indicators undergo data reconstruction using the sparse Autoencoder. Subsequently, the reconstruction error is computed by comparing the reconstructed data with the original health indicators. Lastly, the reconstruction error is transformed to the (0, 1) interval by utilizing the improved sigmoid function.

(4) Determination of degradation starting point

The box-and-line diagram technique is employed to pinpoint the initiation of aero-engine degradation. This is achieved by computing the upper and lower boundaries of the box-and-line diagram for the health index, identifying the point at which the health index

surpasses the boundary for the first time consecutively over n points, signifying the onset of degradation.

(5) RUL prediction

A novel approach for aero-engine Remaining Useful Life (RUL) prediction utilizing the improved Autoformer model is introduced. The spatial attention mechanism is concurrently incorporated alongside the autocorrelation mechanism segment within the model, enabling the fusion of temporal and spatial features, which are then input into the sequence decomposition module. Additionally, a multilayer perceptron is employed for feature extraction on the trend term input to the Autoformer decoder, facilitating a further extraction of degradation characteristics and, subsequently, seamless integration with the processed results of the periodic term.

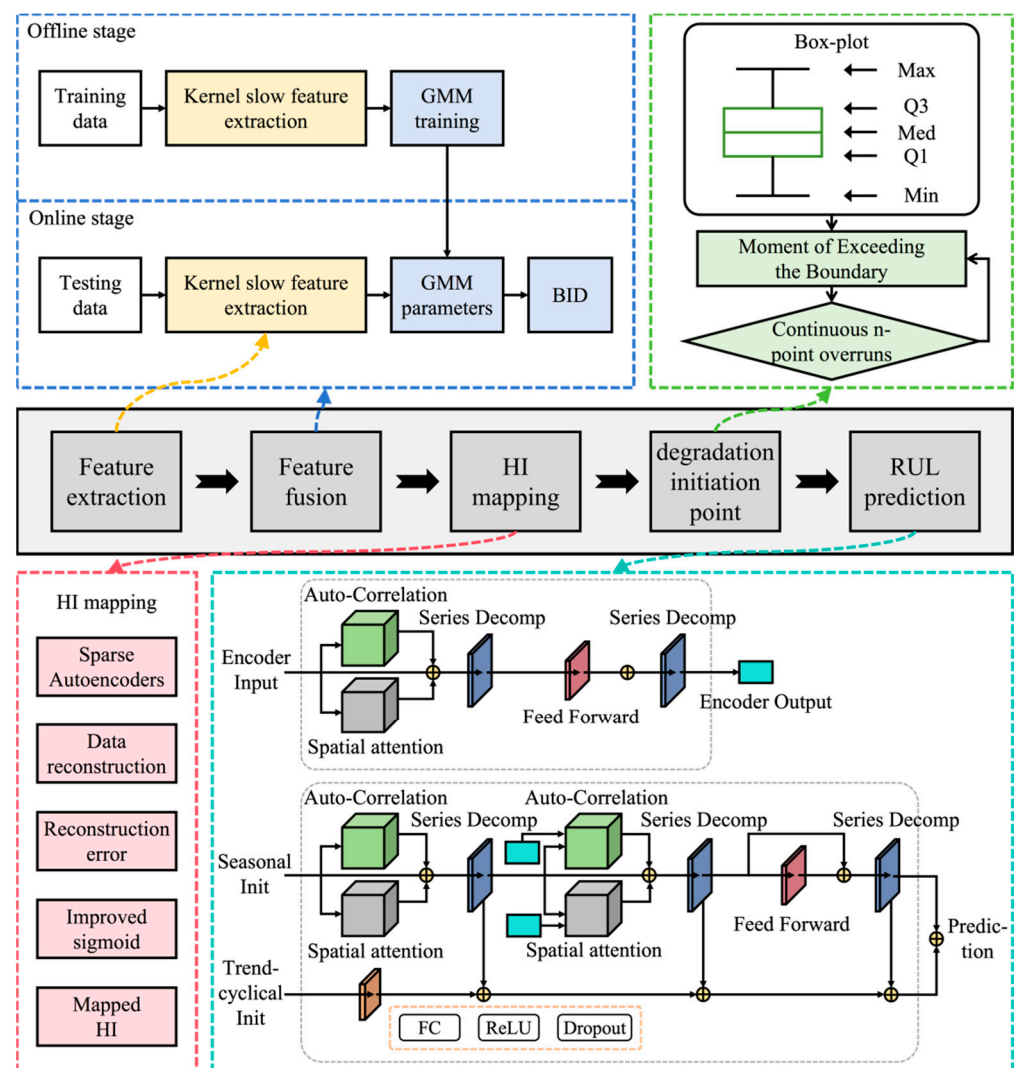


Figure 7. Flowchart of aero-engine Remaining Useful Life prediction.

4. Aero-Engine Gas Path Parameter Selection and Pre-Processing

4.1. Experimental Dataset Selection

The study utilizes data from the CFM56-7B engine of the Boeing 737 aircraft, provided by the National Natural Science Foundation of China and the Civil Aviation Joint Fund Key Project. The subsequent research will be centered on validating and analyzing over-temperature anomalies in the exhaust temperature of the engine based on Quick Access Recorder (QAR) data from three specific cases. The first case involves flight data for a B737 aircraft engine spanning from 20 March 2019 to 28 June 2019. The second case comprises

flight data from 25 March 2019 to 10 June 2019, for a B737 aircraft engine. The third case involves flight data from 7 February 2020 to 13 June 2020, for a B737 aircraft engine.

The Quick Access Recorder (QAR) data comprise multiple monitoring parameters that depict the aircraft's flight status. Inputting the complete set of parameters into the life prediction model escalates computational costs and introduces redundancy, which can significantly perturb the prediction outcomes. Hence, this study filters parameters based on two criteria: (1) ensuring the selected parameters possess complete records in the QAR data and (2) establishing a close correlation between the chosen parameters and the aero-engine's health state.

Employing the aforementioned criteria, coupled with the engineer's expertise and the engine's operational principles, six essential gas path parameters from Table 1 were chosen for engine residual life prediction. These parameters—engine exhaust gas temperature (EGT), fuel flow (FF), low-pressure rotor speed (N1), high-pressure rotor speed (N2), low-pressure pressurized engine outlet temperature (T25), and high-pressure pressurized engine outlet temperature (T3)—were selected to form a predictive parameter vector for engine residual life assessment.

Table 1. The selected aero-engine gas path performance parameters.

Name of the Parameter	Unit of the Parameter
EGT	Degree Celsius ($^{\circ}\text{C}$)
FF	Pounds per hour (lb/h)
N1	Revolutions per minute (r/min)
N2	Revolutions per minute (r/min)
T25	Degree Celsius ($^{\circ}\text{C}$)
T3	Degree Celsius ($^{\circ}\text{C}$)

Within an aircraft's various flight stages, the cruise phase represents a sustained, nearly constant flight state characterized by continuous propulsion. During this phase, the engine's gas path condition monitoring parameters exhibit stability, offering a more reliable indicator of the aero-engine's health [32]. Figure 8 illustrates the variation in exhaust gas temperature for a specific flight.

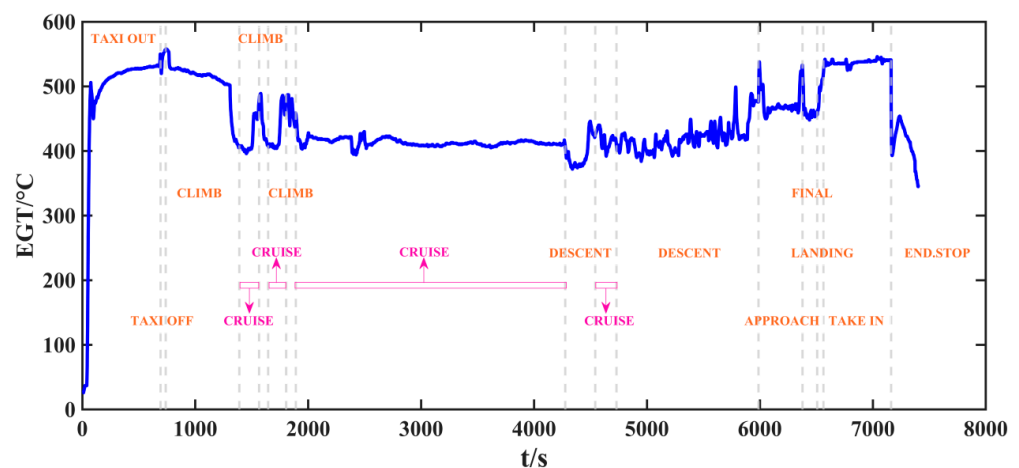


Figure 8. Exhaust gas temperature changes for a given flight.

The variation of exhaust gas temperature for an aero-engine during a flight cycle on 9 June 2019 is depicted in Figure 8, and it can be seen that the value of this parameter fluctuates a lot in the stages of takeoff, climb, descent, and landing, with fluctuations of varying degrees in the range of 300–600 $^{\circ}\text{C}$, whereas in the cruise stage, the state is relatively smooth, being maintained at about 400 $^{\circ}\text{C}$. Therefore, this paper selects the six parameters screened in the QAR data in the cruise phase for subsequent processing and

modeling. Since in real life, the operation route of the same flight may change, and thus the corresponding cruise phase mileage is different when the airplane is flying different routes, this paper carries out the residual life prediction based on the data of the cruise phase, which can obtain the residual flight time of the engine in the cruise phase, and can then project the residual service life of the engine in the whole phase under different routes.

4.2. Data Preprocessing

The six gas path parameters obtained from the Quick Access Recorder (QAR) data exhibit high volume during the cruise phase, hindering subsequent model construction. To address this, the dataset is aggregated by averaging every 60 data points, converting it to one point per minute, thereby reducing data bulk. Furthermore, considering the original parameters' disparate magnitudes, the maximum/minimum value standardization method is applied to maintain data distribution integrity. The standardization formula is as follows:

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (57)$$

5. Experimental Process and Results

5.1. Kernel Slow Feature Extraction

Utilizing the engine from Case 1 as an illustration, Figure 9 demonstrates the evolution of raw monitoring data for chosen parameters over the course of the flight.

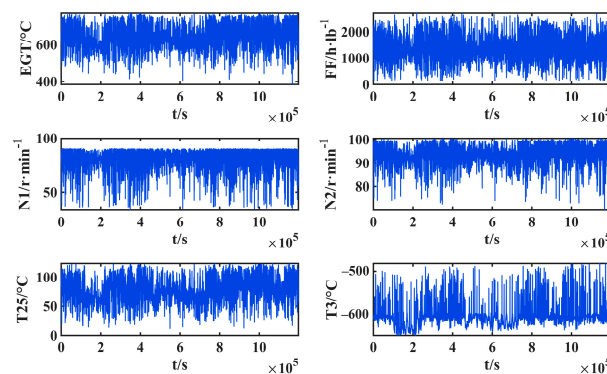


Figure 9. Variation curves of six gas path parameters for the engine of Case 1.

The uniform trend of each monitoring parameter for this engine is not readily discernible in Figure 9 due to the abundance of data. As a solution, the data were averaged every 60 points, and the parameters underwent feature extraction via kernel slow feature analysis. The outcomes are depicted in Figure 10.

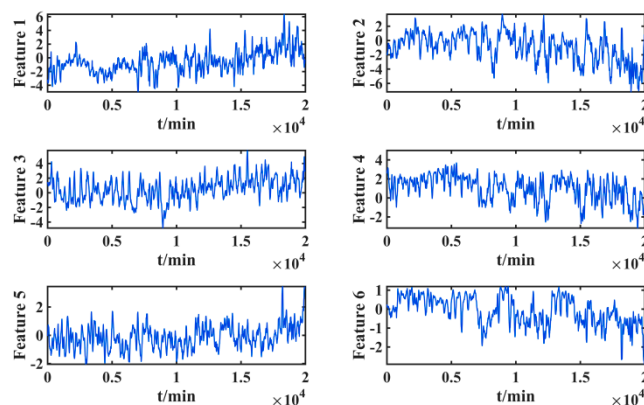


Figure 10. Slow feature extraction results for the engine of Case 1.

Figure 10 illustrates that the derived nuclear slow features remain relatively stable in the initial 15,000 min, followed by diverse degradation trends during advanced usage stages. A comparison with the original monitoring data highlights the enhanced ability of the kernel slow features to depict the engine's health status.

5.2. Health Indicator Construction

Based on the obtained kernel slow features, it is evident that various features exhibit diverse degrees of engine performance degradation. Relying on a single feature is insufficient to provide a comprehensive representation of the engine's health status. Therefore, multi-feature fusion is essential to formulate a more holistic health index.

This study applies the GMM-BID method to merge the nuclear slow features of the three examined engines individually. The top 10% of data from each engine are chosen as the health state data. Following iterations, a GMM with three Gaussian components effectively models the standard data. Subsequent to training the GMM model for the normal state, all test data are fed into the model to obtain a respective BID value for each input. The outcomes illustrating the establishment of health indicators for the three engines are presented in Figure 11.

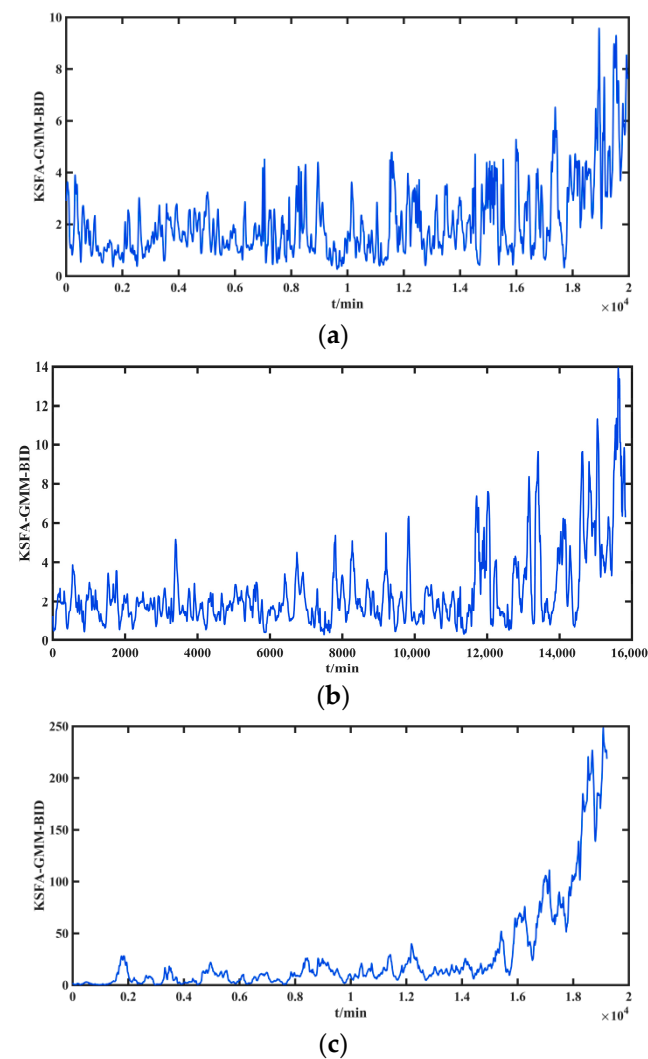


Figure 11. Results of constructing health indicators for three engines. (a) Engine for Case 1, (b) engine for Case 2, and (c) engine for Case 3.

The health indicators generated by the GMM-BID algorithm, from the amalgamation of the derived kernel slow features of the three engines, exhibit distinct degradation

trends corresponding to each engine's data. Despite the variability in trends, all indicators effectively mirror the performance degradation across the engines.

To predict the remaining life of the engine accurately, a consistent failure threshold must be established, necessitating the mapping of health indicators from the three engines to a standardized range. Prior to mapping, the KSFA-GMM-BID health indicators, denoting the degradation trend of all engines, are filtered using the Kalman filter algorithm, as illustrated in Figure 12.

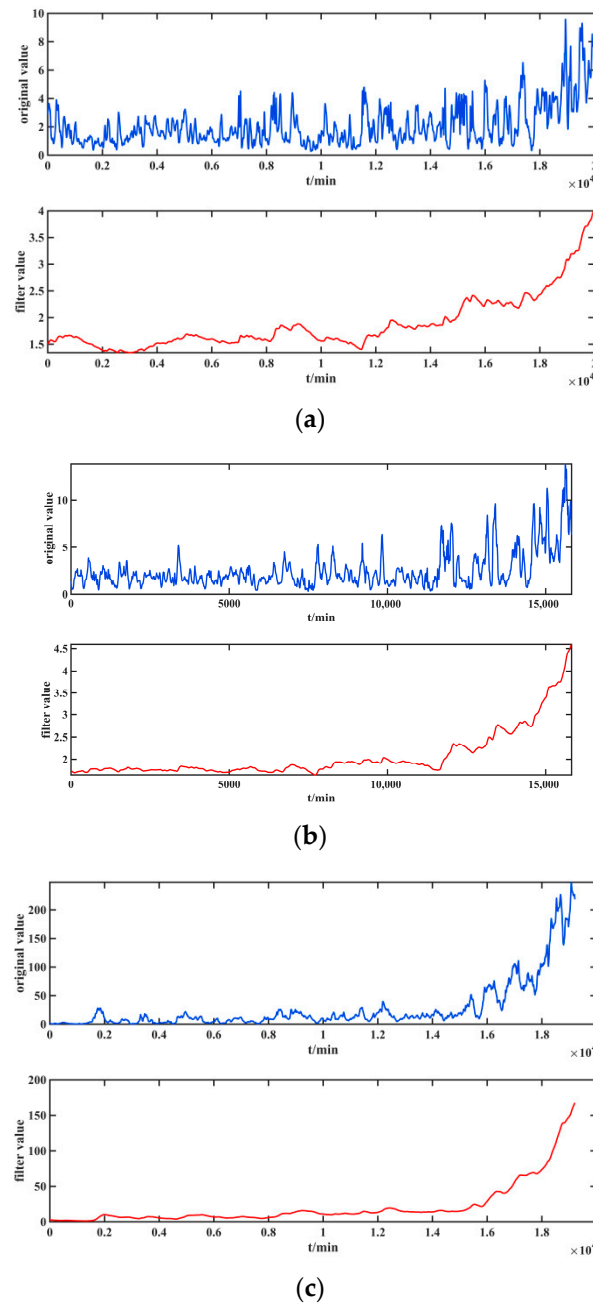


Figure 12. Health indicator filtering results for the three engines. (a) Engine for Case 1, (b) engine for Case 2, and (c) engine for Case 3.

Next, the processed health indicators are standardized to the (0, 1) range using a sparse autoencoder. An HI value nearing 1 indicates a healthy engine while approaching 0 signifies engine inactivity. Figure 13 displays the HI trends against flight time for the three engines.

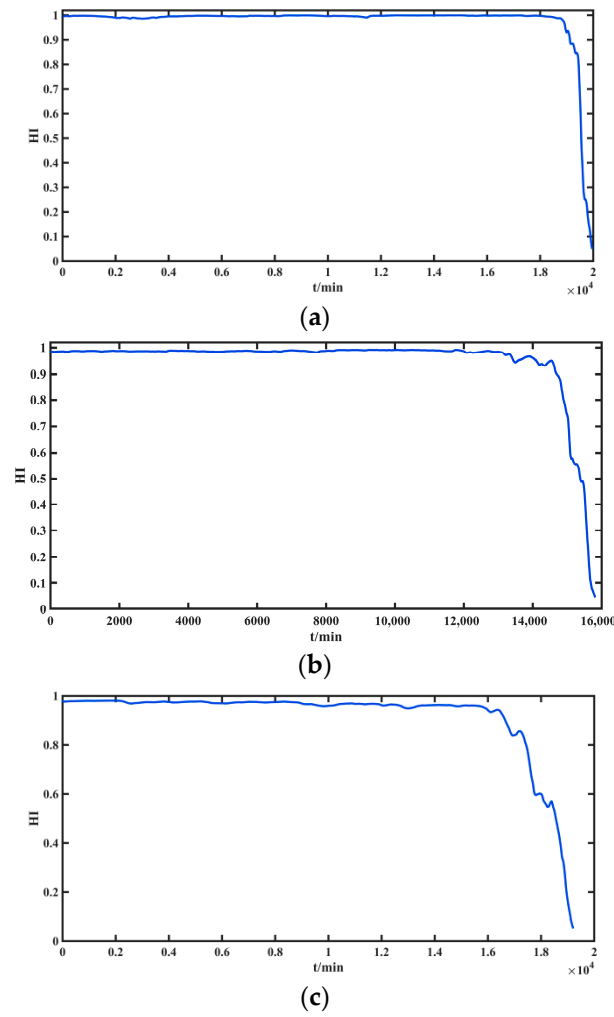


Figure 13. Health indicators after mapping of three engines. (a) Engine for Case 1, (b) engine for Case 2, and (c) engine for Case 3.

The mapped HI illustrates that the health status of the three engines distinctly transitions through two stages. Initially, they operate smoothly and healthily, followed by a later stage of gradual degradation leading to failure. During the initial phase, HI stabilizes near 1, signifying normal operation with relatively smooth performance. Subsequently, HI gradually declines from 1 to 0, indicating the engines' deterioration towards failure.

5.3. Life Prediction Evaluation Index

In order to evaluate the prediction performance of the model, the root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are used in this paper [33]. The calculation formula is as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (\hat{y}_n - y_n)^2} \quad (58)$$

$$MAE = \frac{1}{N} \sum_{n=1}^N |\hat{y}_n - y_n| \quad (59)$$

$$MAPE = \frac{1}{N} \sum_{n=1}^N \frac{|\hat{y}_n - y_n|}{y_n} \times 100\% \quad (60)$$

where y_n is the real value of the engine; $\hat{y}_n$ is the predicted value of the engine. The smaller the value of these three evaluation indexes, the smaller the difference between the predicted value and the real value, and the more accurate the prediction result of the model.

5.4. Life Prediction Results and Analysis

This study establishes the upper and lower boundaries of engine health indicators through box-and-line diagrams, defining the onset of engine degradation when 1000 consecutive health indicator values surpass these limits. Consequently, the degradation initiation points for the three engines are determined, as presented in Table 2.

Table 2. The determination result of a degradation starting point based on a boxplot.

The Number of the Engine	Degradation Start Moment/Minute
Engine for Case 1	17,896
Engine for Case 2	13,344
Engine for Case 3	16,549

The improved Autoformer model introduced in this study determined the configuration settings for each parameter via grid search experiments, as detailed in Table 3.

Table 3. Results of model parameter selection.

The Name of the Parameter	Value	The Name of the Parameter	Value
e_layers	2	train_epochs	15
d_layers	1	n_heads	8
seq_len	96	d_model	512
label_len	48	features	MS
pred_len	24	optimizer	Adam
batch_size	16	loss function	MSE
learning_rate	0.001	activation function	GeLU

We conduct HI prediction utilizing predefined hyperparameters. Training on datasets from the three engines involves 80% degradation stage data to achieve progressive multi-step forecasting. Given the standardized HI within the (0, 1) range, the aero-engine's failure threshold is established at 0.1. The remaining life prediction outcomes for all engines are depicted in Figure 14.

Figure 14 presents prediction outcomes for three engines trained with 80% of data from the degradation phase. The blue curve represents the original HI curve, while the red curve depicts the predictive curve generated by the improved Autoformer. A red dashed line at a value of 0.1 signifies the failure threshold, with the time interval between the light blue dotted lines indicating the remaining flight duration at the corresponding prediction point. The first engine undergoes actual failure at 19,899 min, whereas the predictive failure is at 19,786 min, resulting in an actual RUL of 401 min and a predicted RUL of 288 min. For the second engine, the actual and predicted failure times occur at 15,682 min and 15,638 min, providing an actual RUL of 468 min and a predicted RUL of 424 min, respectively. The third engine experiences actual failure and predicted failure at 19,103 min and 19,051 min, respectively, entailing an actual RUL of 511 min and a predicted RUL of 459 min.

To validate the efficacy of the prediction model suggested in this study, a comparison is made with the Transformer and Autoformer models. The parameters of the comparison model align closely with those of the proposed model. The comparison outcomes are detailed in Table 4 below.

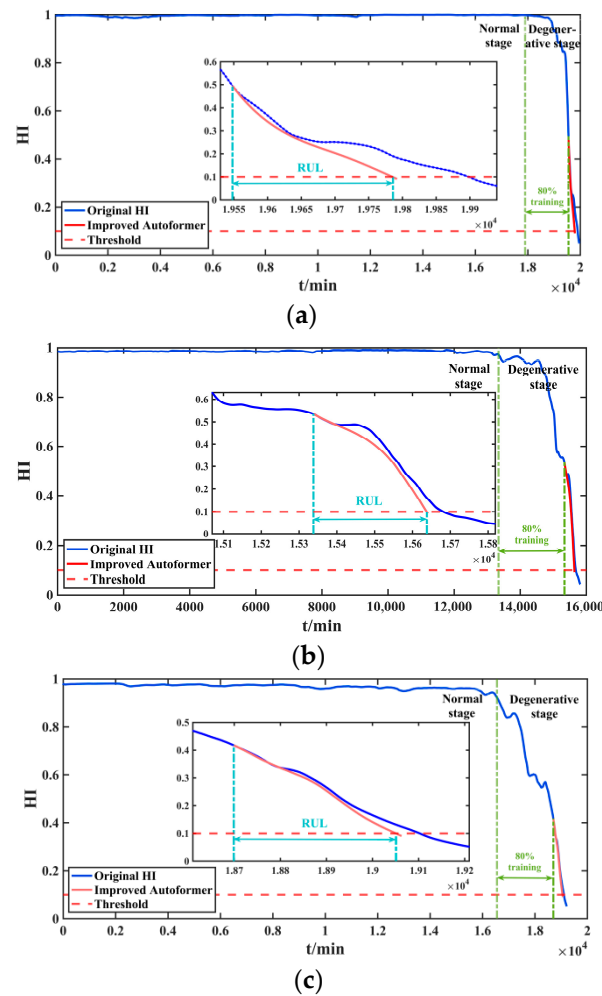


Figure 14. The results of the Remaining Useful Life prediction for the three engines. (a) Engine for Case 1, (b) engine for Case 2, and (c) engine for Case 3.

Table 4. Comparison of different prediction models.

The Number of the Engine	Predictive Model	RMSE	MAE	MAPE/%
Engine for Case 1	Transformer	0.0511	0.0517	13.3205
	Autoformer	0.0427	0.0464	12.1433
	Improved Autoformer	0.0353	0.0297	9.5347
Engine for Case 2	Transformer	0.0624	0.0639	19.5728
	Autoformer	0.0591	0.0558	19.3289
	Improved Autoformer	0.0523	0.0419	17.0177
Engine for Case 3	Transformer	0.0188	0.0142	8.0317
	Autoformer	0.0192	0.0217	9.1402
	Improved Autoformer	0.0159	0.0125	6.7310

The results in Table 4 reveal a substantial enhancement in prediction accuracy with the improved Autoformer model proposed in this paper. The average RMSE, MAE, and MAPE values across the three engine datasets are 0.0345, 0.0280, and 11.0945%, respectively. These figures indicate a 22%, 35%, and 19% reduction compared to the Transformer model, showcasing the effectiveness of incorporating both temporal and spatial features of aero-engine gas path parameters for enhancing predictive performance. Moreover, the improved Autoformer model exhibits a 15%, 32%, and 18% decrease in the evaluation indexes in contrast to the standard Autoformer model, underlining the advantage of considering spatial and temporal characteristics in aero-engine parameter analysis. Notably, the model's lower

prediction errors on the engine datasets affirm its robustness and accuracy, highlighting its efficacy in predicting engine residual life across various failure scenarios and offering innovative insights for aero-engine lifespan forecasting.

6. Conclusions

This paper proposes an aero-engine remaining life prediction method based on the KSFA-GMM-BID-improved Autoformer. The findings of the research are as follows:

- (1) The KSFA-GMM-BID fusion technique introduced herein can aptly formulate a health metric representing the engine's condition, subsequently standardizing it within the (0, 1) interval via a sparse self-encoder mapping approach to establish a consistent failure threshold for life prediction.
- (2) The improved Autoformer prediction approach considered in this study comprehensively integrates spatio-temporal characteristics of aero-engine gas path performance parameters, bolstering the predictive capability of the model. By incorporating a spatial attention mechanism parallel to the Autoformer encoder's autocorrelation segment, this model merges temporal and spatial features to enhance global feature extraction. Additionally, a multilayer perceptron is leveraged for the trend term feature extraction in the Autoformer decoder, maximizing the utilization of degenerative aspects. Compared to the Transformer and Autoformer models, the method proposed herein significantly enhances the accuracy of aero-engine remaining service life prediction.

Author Contributions: Y.Z.: Conceptualization, Methodology, Validation, Writing—review and editing; J.W.: Software, Investigation, Project administration, Writing—original draft; Y.L.: Validation, Writing—review and editing; Z.L.: Validation, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The experimental data used in this paper are derived from the Boeing 737 aircraft CFM56-7B engine data provided by the key project of the National Natural Science Foundation of China and the Civil Aviation Joint Fund. Due to the confidentiality of the data, so raw data is not available to the public.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, X.; Chen, Y.; Ni, H.; Zhang, D. Aero-engine remaining useful life prediction based on a long-term channel self-attention network. *Signal Image Video Process.* **2023**, *18*, 637–645. [\[CrossRef\]](#)
2. Lu, X.; Pan, H.; Zhang, L.; Ma, L.; Wan, H. A dual path hybrid neural network framework for remaining useful life prediction of aero-engine. *Qual. Reliab. Eng. Int.* **2024**, *40*, 1795–1810. [\[CrossRef\]](#)
3. Peng, C.; Chen, Y.; Gui, W.; Tang, Z.; Li, C. Remaining useful life prognosis of turbofan engines based on deep feature extraction and fusion. *Sci. Rep.* **2022**, *12*, 6491. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Peng, D.; Yin, S.; Li, K.; Luo, H. An SW-ELM Based Remaining Useful Life Prognostic Approach for Aircraft Engines. *IFAC Pap.* **2020**, *53*, 13601–13606. [\[CrossRef\]](#)
5. Huang, C.; Du, J.; Nie, B.; Yu, R.; Xiong, W.; Zeng, Q. Feature selection method based on partial least squares and analysis of traditional chinese medicine data. *Comput. Math. Methods Med.* **2019**, *2019*, 9580126. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Huang, J.; Sun, X.; Yang, X.; Shardt, Y.A. Active nonstationary variables selection based just-in-time co-integration analysis and slow feature analysis monitoring approach for dynamic processes. *J. Process Control* **2022**, *117*, 112–121. [\[CrossRef\]](#)
7. Liu, X.; Zhang, Y.; Zhang, L.; Yang, Y. A novel process monitoring method based on dynamic related ReliefF-SFA method. *IEEE Access* **2020**, *8*, 41673–41683. [\[CrossRef\]](#)
8. Zhang, H.; Deng, X.; Zhang, Y.; Hou, C.; Li, C. Dynamic nonlinear batch process fault detection and identification based on two-directional dynamic kernel slow feature analysis. *Can. J. Chem. Eng.* **2020**, *99*, 306–333. [\[CrossRef\]](#)
9. Wang, L.; Chang, D.; Li, Z. MSCNN-BLSTM based Prediction of the Remaining Useful Life of Aeroengine. *J. Phys. Conf. Ser.* **2022**, *2361*, 012019. [\[CrossRef\]](#)
10. Vega-Nieva, D.J.; Briseño-Reyes, J.; López-Serrano, P.M.; Corral-Rivas, J.J.; Pompa-García, M.; Cruz-López, M.I.; Cuahutle, M.; Ressler, R.; Alvarado-Celestino, E.; Burgan, R.E. Autoregressive Forecasting of the Number of Forest Fires Using an Accumulated MODIS-Based Fuel Dryness Index. *Forests* **2023**, *15*, 42. [\[CrossRef\]](#)

11. Zhu, Y.; Chen, J.; Wang, K.; Liu, Y.; Wang, Y. Research on Performance Prediction of Highway Asphalt Pavement Based on Grey–Markov Model. *Transp. Res. Rec.* **2022**, *2676*, 194–209. [\[CrossRef\]](#)
12. Barraza-Barraza, D.; Tercero-Gómez, V.G.; Beruvides, M.G.; Limón-Robles, J. An adaptive ARX model to estimate the RUL of aluminum plates based on its crack growth. *Mech. Syst. Signal Process.* **2017**, *82*, 519–536. [\[CrossRef\]](#)
13. Li, Y.; Huang, X.; Zhao, C.; Ding, P. A novel remaining useful life prediction method based on multi-support vector regression fusion and adaptive weight updating. *ISA Trans.* **2022**, *131*, 444–459. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Wang, H.; Li, D.W.; Li, D.J.; Liu, C.; Yang, X.; Zhu, G. Remaining Useful Life Prediction of Aircraft Turbofan Engine Based on Random Forest Feature Selection and Multi-Layer Perceptron. *Appl. Sci.* **2023**, *13*, 7186. [\[CrossRef\]](#)
15. Liu, X.; Chen, G.; Cheng, Z.; Wei, X.; Wang, H. Convolution neural network based particle filtering for remaining useful life prediction of rolling bearing. *Adv. Mech. Eng.* **2022**, *14*, 16878132221100631. [\[CrossRef\]](#)
16. Li, W.; Zhang, L.-C.; Wu, C.-H.; Wang, Y.; Cui, Z.-X.; Niu, C. A data-driven approach to RUL prediction of tools. *Adv. Manuf.* **2024**, *12*, 6–18. [\[CrossRef\]](#)
17. Fu, L.; Li, P.; Gao, L.; Miao, A. Local-global cooperative least squares support vector machine and prediction of remaining useful life of rolling bearing. *Meas. Control* **2023**, *56*, 358–370. [\[CrossRef\]](#)
18. Zhou, K.; Tang, J. A wavelet neural network informed by time-domain signal preprocessing for bearing remaining useful life prediction. *Appl. Math. Model.* **2023**, *122*, 220–241. [\[CrossRef\]](#)
19. Kim, G.; Choi, J.G.; Lim, S. Using transformer and a reweighting technique to develop a remaining useful life estimation method for turbofan engines. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108475. [\[CrossRef\]](#)
20. Wu, T.; Zhao, T.; Xu, S. Prediction of Remaining Useful Life of the Lithium-Ion Battery Based on Improved Particle Filtering. *Front. Energy Res.* **2022**, *10*, 863285. [\[CrossRef\]](#)
21. Yan, H.; Qin, Y.; Xiang, S.; Wang, Y.; Chen, H. Long-term gear life prediction based on ordered neurons LSTM neural networks. *Measurement* **2020**, *165*, 108205. [\[CrossRef\]](#)
22. Li, J.; Wang, Z.; Liu, X.; Feng, Z. Remaining Useful Life Prediction of Rolling Bearings Using GRU-DeepAR with Adaptive Failure Threshold. *Sensors* **2023**, *23*, 1144. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Zhang, X.; Sun, J.; Wang, J.; Jin, Y.; Wang, L.; Liu, Z. PAOLTransformer: Pruning-adaptive optimal lightweight Transformer model for aero-engine remaining useful life prediction. *Reliab. Eng. Syst. Saf.* **2023**, *240*, 109605. [\[CrossRef\]](#)
24. Liu, Y.; Chen, J.; Wang, T.; Li, A.; Pan, T. A variational transformer for predicting turbopump bearing condition under diverse degradation processes. *Reliab. Eng. Syst. Saf.* **2023**, *232*, 109074. [\[CrossRef\]](#)
25. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22419–22430.
26. Corrigan, J.; Zhang, J. Developing accurate data-driven soft-sensors through integrating dynamic kernel slow feature analysis with neural networks. *J. Process Control* **2021**, *106*, 208–220. [\[CrossRef\]](#)
27. Feng, S.; Wang, A.; Cai, J.; Zuo, H.; Zhang, Y. Health State Estimation of On-Board Lithium-Ion Batteries Based on GMM-BID Model. *Sensors* **2022**, *22*, 9637. [\[CrossRef\]](#)
28. Lu, Z.; Wang, N.; Dong, S. Improved Square-Root Cubature Kalman Filtering Algorithm for Nonlinear Systems with Dual Unknown Inputs. *Mathematics* **2023**, *12*, 99. [\[CrossRef\]](#)
29. Yang, N.; Zhang, Z.; Yang, J.; Hong, Z. Mineralized-Anomaly Identification Based on Convolutional Sparse Autoencoder Network and Isolated Forest. *Nat. Resour. Res.* **2022**, *32*, 1–18. [\[CrossRef\]](#)
30. Chen, C.; Wang, T.; Liu, Y.; Cheng, L.; Qin, J. Spatial attention-based convolutional transformer for bearing remaining useful life prediction. *Meas. Sci. Technol.* **2022**, *33*, 114001. [\[CrossRef\]](#)
31. Fan, X.; Li, X.; Yan, C.; Fan, J.; Chen, L.; Wang, N. Converging Channel Attention Mechanisms with Multilayer Perceptron Parallel Networks for Land Cover Classification. *Remote Sens.* **2023**, *15*, 3924. [\[CrossRef\]](#)
32. Pan, W.; Feng, Y.; Liu, J. Parameter-Influencing Analysis of Aeroengine Operation Reliability. *J. Aerosp. Eng.* **2023**, *36*, 04023030. [\[CrossRef\]](#)
33. Kayaalp, K.; Metlek, S.; Ekici, S.; Şöhret, Y. Developing a model for prediction of the combustion performance and emissions of a turboprop engine using the long short-term memory method. *Fuel* **2021**, *302*, 121202. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.