aerospace

MDPI

Article

Missile Fault Detection and Localization Based on HBOS and Hierarchical Signed Directed Graph

Hengsong Hu [1], Yuehua Cheng [1], Bin Jiang [1,*], Wenzhuo Li [1] and Kun Guo [2]

[1] College of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; huhengsong@nuaa.edu.cn (H.H.); chengyuehua@nuaa.edu.cn (Y.C.); liwenzhuo@nuaa.edu.cn (W.L.)
[2] Beijing Institute of Mechanical and Electrical Engineering, Beijing 100074, China; 2007guokun@163.com
* Correspondence: binjiang@nuaa.edu.cn

Abstract: The rudder surfaces and lifting surfaces of a missile are utilized to acquire aerodynamic forces and moments, adjust the missile's attitude, and achieve precise strike missions. However, the harsh flying conditions of missiles make the rudder surfaces and lifting surfaces susceptible to faults. In practical scenarios, there is often a scarcity of fault data, and sometimes, it is even difficult to obtain such data. Currently, data-driven fault detection and localization methods heavily rely on fault data, posing challenges for their applicability. To address this issue, this paper proposes an HBOS (Histogram-Based Outlier Score) online fault-detection method based on statistical distribution. This method generates a fault-detection model by fitting the probability distribution of normal data and incorporates an adaptive threshold to achieve real-time fault detection. Furthermore, this paper abstracts the interrelationships between the missile's flight states and the propagation mechanism of faults into a hierarchical directed graph model. By utilizing bilateral adaptive thresholds, it captures the first fault features of each sub-node and determines the fault propagation effectiveness of each layer node based on the compatibility path principle, thus establishing a fault inference and localization model. The results of semi-physical simulation experiments demonstrate that the proposed algorithm is independent of fault data and exhibits high real-time performance. In multiple sets of simulated tests with randomly parameterized deviations, the fault-detection accuracy exceeds 98% with a false-alarm rate of no more than 0.31%. The fault-localization algorithm achieves an accuracy rate of no less than 97.91%.

Keywords: hierarchical signed directed graph; fault detection; fault localization; compatible path; missile system

Citation: Hu, H.; Cheng, Y.; Jiang, B.; Li, W.; Guo, K. Missile Fault Detection and Localization Based on HBOS and Hierarchical Signed Directed Graph. *Aerospace* **2024**, *11*, 679. https://doi.org/10.3390/aerospace11080679

Academic Editor: Wim J. C. Verhagen

Received: 26 June 2024
Revised: 13 August 2024
Accepted: 16 August 2024
Published: 17 August 2024

1. Introduction

Missile systems comprise several complex components, which collectively form a sophisticated combat system through their coordination. These systems exhibit strong characteristics of nonlinearity, non-stationarity, and strong coupling [1]. The lifting surfaces of missiles generate lift during flight, enabling their aerial navigation and attitude maintenance. The rudder surfaces of missiles provide control moments to achieve rolling, yawing, and pitching maneuvers, ensuring the missile's ability to fly along its predetermined trajectory. Both are crucial for the missile system [2]. However, missile missions are subject to various uncertainties and disturbances from the atmospheric environment. Flight conditions can be extremely harsh [3–5], and the missile system may also face potential attacks from enemy anti-missile systems, making the lifting surfaces and rudder surfaces susceptible to fatal faults. The lack of corresponding fault detection and localization analyses renders it impossible to take effective measures after a failure, resulting in immeasurable financial and strategic losses. Therefore, timely and accurate fault detection and localization are crucial for ensuring missile safety. The purpose of fault diagnosis is to quickly and accurately obtain information about system faults, providing guidance for fault-tolerant

control. This enables fault-tolerant control to generate more appropriate and precise control laws based on the specific faults, which is of great significance.

The primary objective of fault detection is to promptly identify abnormal conditions within a system by continuously monitoring its measurable states in real time [6–8]. Meanwhile, fault-localization algorithms analyze the system's status information to trace back to the root cause of the fault, determining the specific component or location responsible for the system's abnormal behavior [9]. In other words, fault detection can determine whether a system is faulty and when the fault occurs, while fault localization can pinpoint the location of the faulty component. To ensure the overall algorithm's speed and real-time performance, it is customary to conduct system fault detection first. Subsequently, fault localization is carried out upon fault detection to avoid additional time consumption from executing fault-localization algorithms when the system is fault-free.

Currently, research methods for fault detection can be broadly categorized into two types: those based on traditional model approaches and those based on deep learning. The core idea of traditional model-based methods is to measure the system's abnormality by calculating the difference between normal data and the data under inspection using traditional models [10,11]. On the other hand, fault-detection algorithms based on deep learning utilize neural networks to directly map fault features to system abnormality in a nonlinear manner [12–18]. Traditional model-based fault-detection methods have advantages such as low computational complexity, excellent real-time performance, relatively low data requirements, and adaptability to sudden unknown faults. However, they struggle with handling large-scale high-dimensional data and are sensitive to the initial parameters of the model [7,19]. Fault-detection algorithms based on deep learning can achieve end-to-end learning, reducing the need for manual intervention, and can effectively handle large-scale high-dimensional data. However, they are highly dependent on data, have relatively long computation times, and cannot handle sudden unknown fault issues. Therefore, it is necessary to choose a suitable fault-detection algorithm based on the specific requirements of the detection object and its operational needs.

Fault-localization research methods can be categorized into two forms: model-based [20] and data-driven [21]. Model-based methods involve studying and analyzing the internal fault mechanisms of models to infer fault localization. Their advantage lies in their lower dependency on data, and models can better reflect the internal structure and characteristics of systems. However, accurately modeling complex systems poses a challenge. For instance, He Feng et al. [22] proposed a single-phase grounding fault-localization method for distribution networks, which utilizes a network description matrix to depict network topology, demonstrating its effectiveness in utilizing fault data for localization. Xu Yuxin et al. [23] introduced a method for large steam generator stator grounding fault localization, considering the distribution of winding potentials. They established equations for the magnitude and phase of zero-sequence voltage based on fault characteristics and winding potential distribution. Zhang Rongsheng et al. [24] presented a distributed power distribution network fault-localization method based on an improved bat algorithm, constructing a topological model for fault localization in distribution network zones using the principle of equivalent dual ports. On the other hand, data-driven methods do not require analysis and modeling of the internal structure of systems, thus avoiding complex nonlinear analysis and computation. Among these, deep learning-based methods are prominent, wherein deep learning networks are utilized to construct mapping models from system states to fault locations. For example, Zhang Zhuo et al. [25] conducted fault-localization experiments using three representative deep learning architectures: convolutional neural networks, recurrent neural networks, and multilayer perceptrons, concluding that convolutional neural networks are the most effective method for fault localization in the studied system architecture. M. Dashtdar et al. [26] employed artificial neural networks for fault diagnosis in transmission networks, demonstrating that extracting the maximum scale of horizontal components can reveal fault features suitable for training neural networks, including fault type, angle, and position information. Simulation results showed that

neural network-based fault-localization algorithms can estimate fault distance based on fault type under different conditions effectively.

Methods based on deep learning require high requirements for the format and accuracy of data [27,28], necessitating many data samples to obtain a network model. It is challenging to analyze systems with insufficient data samples or data loss. Additionally, deep learning networks are often considered black-box models, making it difficult to interpret their decision-making processes [29]. Model-based fault-localization methods require the establishment of a working model or reference model of the system. This typically necessitates a deep understanding of the system's structure and operation principles, and the modeling process may be complex. However, its advantages lie in its ability to fully utilize the internal structure and logical relationships of the system. It can provide relatively accurate fault-localization results without requiring many data samples while also offering good interpretability and adaptability.

This study considers the characteristics of missile objects and focuses on typical faults in missile control surfaces and lifting surfaces. The research aims to investigate suitable fault detection and fault-localization methods for these specific faults. The remainder of the study is organized as follows. First, Section 2 of this paper presents the problem statement, which includes Section 2.1, an introduction to the challenges and limitations of existing research in missile fault detection and localization, and Section 2.2, an analysis of typical fault types and mathematical models related to missile rudder surfaces and lifting surfaces. Section 2.3 provides an overview of the overall algorithm structure and discusses the contributions of the proposed fault detection and localization approach in this paper. Section 3 presents the principles and structure of the fault-detection algorithm, while Section 4 outlines the principles and structure of the fault-localization algorithm. In Section 5, a simulation-based verification and comparative experiment on a semi-physical missile platform is conducted to evaluate the fault detection and fault-localization algorithms. Finally, Section 6 provides a summary of the findings.

2. Problem Formulation and Overall Approach

2.1. Problem Formulation

Missiles, as critical strategic defense weapons, possess unique characteristics compared to other objects. Considering the specific features of their actual flight scenarios, the following requirements are posed for their fault detection and localization algorithms:

- **Real-time responsiveness**: Missiles have significant dynamic responses, causing faults to quickly affect the stability of the system. Therefore, real-time monitoring of the missile's state data is essential, requiring algorithms to have excellent real-time capabilities to provide monitoring results quickly.
- **Uncertainty**: Missiles encounter various uncertain factors during flight, including environmental conditions, aerodynamic characteristics, mass inertia, sensor measurements, wind speed, etc. Algorithms need to effectively handle uncertain data.
- **Limited fault samples**: It is challenging to collect many fault data samples under real-world conditions, with most of the data representing normal missile flight. Therefore, the dataset follows a long-tail distribution. Algorithms should not rely heavily on large datasets and should be able to achieve accurate fault detection and localization even in the presence of data imbalance.
- **Interpretability**: For specialized systems like missiles, which are crucial to national defense and security, it is essential that fault detection and localization results are understandable and that the detection and localization processes are traceable. Therefore, the corresponding fault detection and localization algorithms must possess a certain level of interpretability.

Due to the nonlinear nature of missile models, coupled with uncertainties and noise interference, it is challenging to establish accurate models. Existing model-based fault detection and localization algorithms struggle to be applicable. Data-driven approaches heavily rely on the quantity and quality of datasets, with current methods largely tailored

to simulated environments. However, in practical scenarios, fault data are scarce or difficult to obtain, with most data being non-faulty, leading to imbalanced dataset distributions. Therefore, existing data-driven methods face challenges in direct application. Currently, no research simultaneously addresses these points in fault detection and localization.

2.2. Typical Faults of Rudder Surfaces and Lifting Surfaces

Let the positive direction of the X-axis be along the missile body, pointing forward, and the positive direction of the Y-axis be perpendicular and pointing upward. According to the right-hand coordinate system, the positive direction of the Z-axis is horizontally pointing to the right. The rudder is designated as follows: 1, 2, 3, and 4 correspond to the right upper, left upper, left lower, and right lower positions, respectively. The sign convention for the deflection angles is defined as follows: when looking forward from the tail of the missile, a positive deflection angle is when the trailing edge of the control surface is deflected upward.

δ_x, δ_y and δ_z represent the desired roll, pitch, and yaw commands generated by the missile guidance controller. Their relationship with the four rudder surfaces is described by the following equations:

$$\begin{aligned} \delta_x &= \tfrac{1}{4}(\delta_1 - \delta_2 - \delta_3 + \delta_4) \\ \delta_y &= \tfrac{1}{4}(\delta_1 - \delta_2 + \delta_3 - \delta_4) \\ \delta_z &= \tfrac{1}{4}(\delta_1 + \delta_2 + \delta_3 + \delta_4) \end{aligned} \tag{1}$$

when a fault occurs in the missile's rudder surfaces, the resulting aerodynamic forces and moments may prevent the missile from responding to the desired commands generated by the control law, leading to mission failure. The missile's lifting surfaces primarily affect the magnitude of lift. When a fault occurs in the lift surfaces, the missile may fail to generate sufficient lift, which similarly impacts the missile's ability to complete its mission.

Therefore, missile rudder surfaces and lifting surfaces are susceptible to faults due to adverse flight conditions and potential attacks from enemy anti-missile systems. Such faults can immediately affect the stability of the missile flight attitude, posing challenges to the successful completion of designated missions. Typical faults include rudder stuck, rudder loose, rudder damage, and lifting surface damage [30].

Rudder stuck fault refers to the motor shaft being stuck at a certain angle, preventing the rudder surfaces from reaching the desired angle [31,32].

$$\delta_a = \delta_a^p \quad p \in (-30^\circ, 30^\circ) \tag{2}$$

where a represents the number of the fault surface, δ_a^p represents the rudder effect generated when surface a is stuck at angle p, considering a maximum deflection angle of 30° for the rudder surface.

Rudder loose fault refers to the loosening of the motor shaft and rudder surface shaft, complete failure of the hinge torque, and the complete disappearance of the rudder effect of the faulty rudder surface [33,34].

$$\delta_a = 0 \; a = 1, 2, 3, 4 \tag{3}$$

The structural damage in the rudder and lifting surfaces results in rudder damage faults and lifting damage faults, leading to the loss of rudder effectiveness on the faulty surfaces.

$$\delta_a = \eta \delta_a' \quad a = 1, 2, 3, 4 \tag{4}$$

$$\delta_L = \eta \delta_L' \quad L = \text{left, right} \tag{5}$$

where η represents the degree of rudder surface deficiency. δ_a' represents the rudder effectiveness when rudder a has no deficiencies. δ_L' represents the effectiveness of the control surface when the left or right lifting surfaces have no deficiencies.

The missile measures its current state through sensors and uses guidance control algorithms to steer toward the desired trajectory. The states in the guidance control loop generally include missile attitude, position, velocity, etc., and typically do not include rudder surface information (such as rudder deflection angle, motor current, etc.). Therefore, if only rudder surface information is used for fault detection and localization, disturbances affecting the sensors collecting rudder surface information could lead to erroneous information being provided, resulting in misdiagnosis by the fault detection and localization algorithms. To avoid this scenario, an algorithm based on the missile's flight state for fault detection and localization is needed.

2.3. Overall Approach

Based on the aforementioned requirements, this paper adopts the Histogram-based Outlier Score method for fault detection. The HBOS algorithm offers simplicity and speed, meeting the real-time requirements of the algorithm. Additionally, HBOS employs a dynamic histogram model that adapts according to the data distribution, making it effective in handling uncertain and long-tailed data distributions [35]. The dynamic histograms fitted by HBOS exhibit clear and visible histogram structures, ensuring the interpretability of the fault-detection algorithm.

A fault-localization method for missile systems based on hierarchical directed graphs and adaptive thresholds is proposed. The hierarchical directed graph reasoning model operates at high speed and provides high real-time performance. The adaptive threshold is established based on extreme value distribution theory and adjusts adaptively according to the data distribution, demonstrating good adaptability to uncertain data. Moreover, the hierarchical directed graph relies on the missile's dynamic and kinematic models, offering good interpretability. The overall process of the fault detection and localization algorithm proposed in this paper is shown in Figure 1.

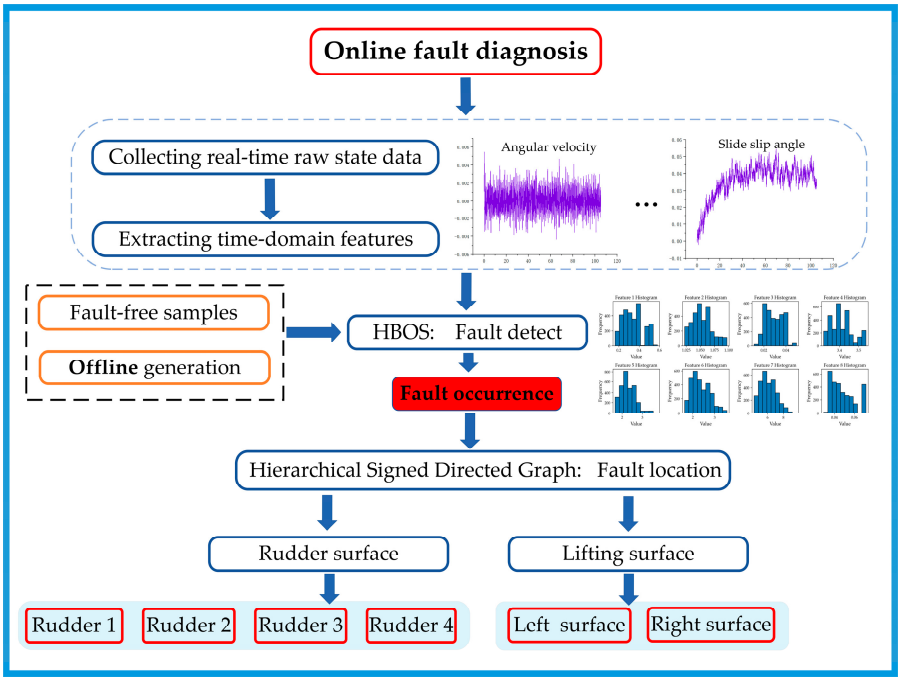

Figure 1. The overall algorithmic flow for fault detection and localization.

The fault detection and localization algorithms proposed in this paper, based on flight state data, exhibit good real-time performance, effectively handle uncertain data, and do not rely heavily on a large number of data samples, thus meeting practical requirements. The contributions of this article include the following points:

- The missile's flight dynamics and kinematic model are used to obtain the fault propagation and mutual influence relationships between different missile flight states. These relationships are cleverly expressed using a hierarchical directed graph model while also leveraging relevant mechanisms to constrain the data.
- In fault detection and fault-localization algorithms, the utilization of adaptive thresholds for data processing is considered. This approach not only leverages their adaptive characteristics to adapt to the differences in missile flight state balancing under various parameter deviations but also effectively distinguishes between noise and faults.
- The fault-detection model can be constructed using only normal data, while the fault-localization model is built based on the fault propagation mechanisms and the mutual influence relationships among flight states. Both fault detection and fault localization achieve high accuracy and good real-time performance without relying on fault data. They can adapt well to practical scenarios.

This article proposes fault detection and fault-localization algorithms specifically designed for typical faults in missile control surfaces and lifting surfaces. By combining flight state propagation mechanisms and data-driven approaches, the algorithms eliminate the need for complex mechanistic models while utilizing a relatively concise parameter propagation influence mechanism to constrain the data model. This removes the dependency on fault data samples. The algorithms aim to provide timely alerts and accurately locate faults once they occur, which is of great importance and significance in ensuring the safety of missile flights.

3. Online Fault-Detection Algorithm

3.1. The Principle of HBOS Anomaly Detection

The HBOS anomaly detection algorithm first performs histogram probability density modeling. This modeling method does not model the dependence between data features but constructs histograms (i.e., binning) for each data dimension separately, making each dimension independent of each other. Subsequently, the algorithm computes the relative frequencies (bin heights) of data falling into bins for each dimension of the input data. The relative frequency represents an estimate of the density of the data for that dimension, with higher density resulting in lower anomaly scores for data points. If the input data for a dimension is categorical, the relative frequency is obtained by calculating the frequency of each input value.

The commonly used method for constructing histograms and computing relative frequencies is the dynamic width histogram method. In the dynamic width histogram method, the input data samples are first sorted, and a fixed number of N/k consecutive values are placed into each bin, where N is the total number of instances (i.e., the number of input data samples), and k is the number of bins. Therefore, the sum of the areas of all bins in the histogram equals the number of instances N. By setting the area of each bin equal, the width of the bin is determined by the first and last data values falling into that bin, which, in turn, determines the height of the bin, yielding the relative frequency. After establishing histograms for each dimension and determining the bin heights, normalization is performed to ensure that the maximum bin height is 1, resulting in a normalized fitted probability density model. Dynamic histograms can adaptively adjust the width and number of bins based on the distribution of the data, which enhances their robustness and generality when dealing with uncertain and long-tailed distributed data.

For each dimension of the fault-free data, construct the corresponding histogram to estimate the probability density function of the normal sample data. The probability of the detection data being normal can be calculated as follows:

$$P(p) = P_1(p)P_2(p)\dots P_i(p)\dots P_d(p) \tag{6}$$

where d is the total number of features of the detection data, P_i is the probability density function of the nth dimension. Taking the logarithm of Equation (5) yields Equation (6).

$$\log(P(p)) = \sum_i^d \log(P_i(p)) \tag{7}$$

The larger the probability density, indicating a higher likelihood of the sample being normal data, the smaller the anomaly score. Therefore, we take the negative of Equation (6) to construct the anomaly score (i.e., HBOS) as shown in Equation (7):

$$\text{HBOS}(p) = -\log(P(p)) = \sum_i^d \frac{1}{\log(P_i(p))} \tag{8}$$

The HBOS value indicates the degree of anomaly for a given sample point. A higher value represents a higher degree of anomaly. Additionally, the HBOS algorithm classifies sample points based on their HBOS values using an anomaly threshold. A label of "0" indicates that the sample point is in a normal state, while a label of "1" indicates that the sample point is in an abnormal state.

3.2. Missile Online Fault Detection

3.2.1. Feature Selection

Due to the numerous flight states of missiles, extracting time-domain features for each state as inputs to HBOS can result in information redundancy. This not only increases the computational burden and affects the real-time performance of the algorithm but also allows less influential features to impact fault detection, therefore reducing the accuracy of the algorithm. Therefore, both flight states and time-domain indicators need to be filtered or selected carefully to mitigate these issues. Combined with the missile object, its flight status data is measured by sensors. Focus on typical faults of missile actuators: rudder stuck, rudder loose, rudder damage, lifting surface damage.

As shown in Figure 2, the feature selection approach in this paper involves selecting the missile flight states and time-domain indicators that are most affected by faults. The time-domain indicators for each flight state are computed as features. By utilizing information entropy, the feature with the maximum difference between the total entropy and the conditional entropy corresponding to each feature is selected, as shown in Equation (8). The selected features are the most advantageous for fault detection.

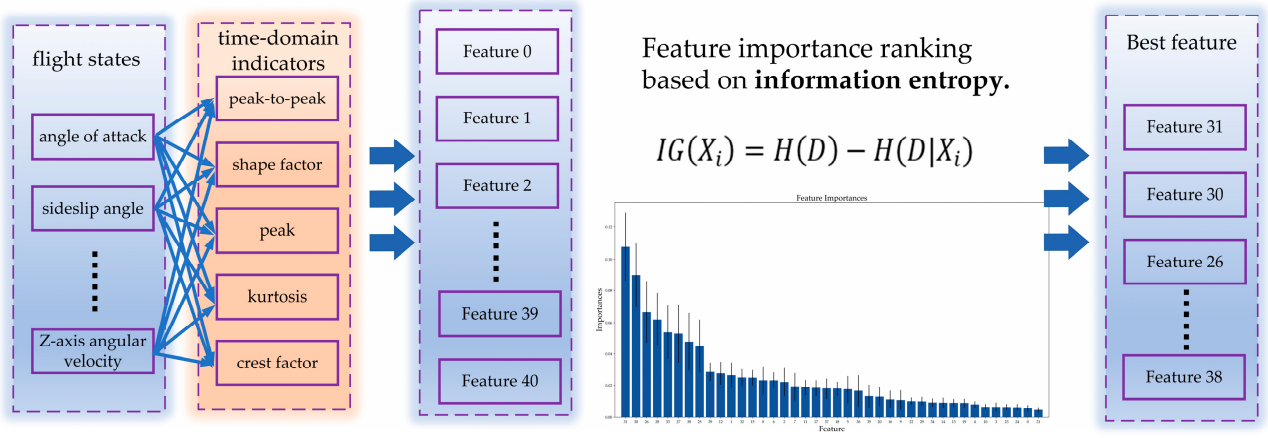

Figure 2. The process of feature extraction.

Select the most prominent time-domain indicators representing faults from the flight status data as multidimensional feature data.

The missile sensor can measure various states, including angle of attack, sideslip angle, acceleration, and angular acceleration. Attitude angles, velocity, latitude, longitude, altitude, etc., can be derived from the measured states. Attitude control serves as the

inner loop control of the missile and is most significantly affected by faults. Therefore, it is considered to utilize the time-domain feature information of the control inner loop states for fault detection. The selected states include angle of attack, sideslip angle, pitch angle, roll angle, yaw angle, X-axis angular velocity, Y-axis angular velocity, and Z-axis angular velocity.

In the time-domain analysis technique, different traditional time-domain statistical indicators (TDSIs) are used for signal analysis, which involves peak, peak-to-peak, mean, root mean square, crest factor, skewness, kurtosis, clearance factor, shape factor [36] and so on.

Considering that during fault-free stages, the missile flight state tends to be stable, and faults typically manifest as spikes in the signal. Therefore, it is essential to choose time-domain features sensitive to spike signals [37,38]. As a result, the selected time-domain features include peak-to-peak, shape factor, peak, kurtosis, and crest factor.

Information entropy has extensive applications in information theory and probability statistics [39], primarily for measuring the uncertainty or information content of random variables. It is commonly used to assess the importance of features.

Suppose the sample set is denoted by D, consisting of N samples, each containing K features. Let X_i represent the i-th feature of the sample, and y represent the label. The mathematical expression for the entropy $H(D)$ is [40]:

$$H(D) = -\sum_{k=1}^{|y|} \frac{|y_k|}{N} log_2 \frac{|y_k|}{N} \tag{9}$$

Here, $|y_k|$ denotes the number of samples belonging to class y_k, and $|y|$ represents the total number of classes. Then, compute the conditional entropy $H(D|X_i)$ for feature X_i over the sample set D:

$$H(D|X_i) = \sum_{j=1}^{|X_i|} \frac{|D_j|}{N} H(D_j) \tag{10}$$

where $|X_i|$ denotes the number of possible values for feature X_i, $|D_j|$ represents the number of samples with the value X_i^j for feature X_i, and $H(D_j)$ is the entropy of the sample set when feature X_i takes the value X_i^j. Finally, calculate the information gain $IG(X_i)$ for feature X_i:

$$IG(X_i) = H(D) - H(D|X_i) \tag{11}$$

By computing the information gain for all features, we can obtain the importance ranking of each feature. For the selected eight flight states of the missile, time-domain features are extracted for each state, resulting in a total of 40-dimensional features. The aim is to select the most important features based on information entropy. The importance ranking of the 40 features is shown in Figure 3.

Based on Figure 3, the importance of the top 8 features is significantly higher than that of the other features, specifically Features 31, 30, 26, 28, 33, 27, 38, and 25. These features are most susceptible to faults. Therefore, we select these eight features as inputs for HBOS. These features correspond to the angle of attack—peak-to-peak, angle of attack—shape factor, roll angle—peak-to-peak, pitch angle—peak value, pitch angle—kurtosis, Y-axis angular velocity—kurtosis, Y-axis angular velocity—crest factor, and Y-axis angular velocity—peak value.

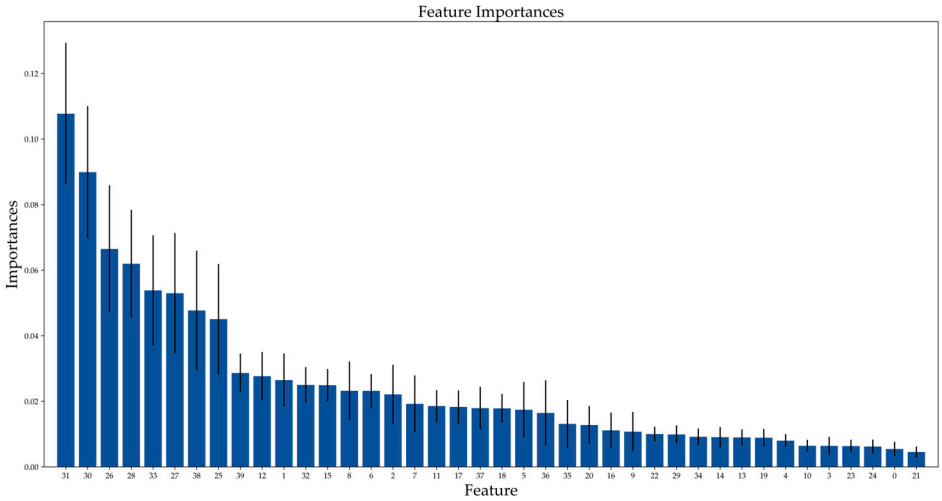

Figure 3. Chart of feature importance ranking.

3.2.2. Drifting Peak Adaptive Threshold Based on Extreme Value Theory

Due to the presence of uncertainties during normal missile flights, the same parameters exhibit different curve characteristics in each flight, making it challenging to determine the state of nodes in the case of minor faults. Therefore, the adaptive threshold method is adopted to generate real-time thresholds for various real-time flight parameters under different fault types, ensuring accurate node state determination. This method adjusts thresholds in real time to maintain precision, operates at a fast speed, and exhibits strong robustness. Hence, the Drift Streaming Peaks-Over-Threshold (DSPOT) method [41,42] based on the Extreme Value Theory is utilized to handle the HBOS anomaly scores of missile online samples.

1. Extreme Value Theory

The objective of Extreme Value Theory (EVT) is to analyze and synthesize the occurrence patterns of extreme events based on historical data or events. The distribution representing these extreme events is referred to as Extreme Value Distributions (EVDs), which can be mathematically expressed as:

$$G_\gamma : x \to \exp\left(-(1+\gamma x)^{-\frac{1}{\gamma}}\right), \gamma \in R, 1+\gamma x > 0 \tag{12}$$

where γ represents the extreme value index; its value depends on the initial distribution pattern. All commonly observed standard extreme value distributions adhere to the above equation.

2. Peak-Over Threshold (POT) method

The second theorem of EVT, also known as the Peak-Over Threshold (POT) method, is as follows: It holds if and only if $\sigma(t)$ exists, satisfying:

$$\overline{F}_t(x) = P(X - t > x | X > t)_{t \to \tau} \sim \left(1 + \frac{\gamma x}{\sigma(t)}\right)^{-\frac{1}{\gamma}} \quad x \in R \; s.t. \; 1 + \gamma x > 0 \tag{13}$$

where X represents the data points, and the part exceeding the threshold is denoted as $X - t$. $X - t$ follows a Generalized Pareto Distribution with shape parameter γ and scale parameter σ, which need to be estimated.

3. Maximum Likelihood Estimation

Compared to traditional methods, such as the Method of Moments and the Probability Weighted Moments [19], the Maximum Likelihood Estimation method [43] exhibits higher efficiency and robustness. The estimation forms for γ and σ are as follows:

$$log\lambda(\gamma,\sigma) = -N_t log\sigma - (1+\frac{1}{\gamma})\sum_{i=1}^{N_t}\log\left(1+\frac{\gamma}{\sigma}Y_i\right) \tag{14}$$

where $Y_i = X_i - t$, $Y_i > 0$ denotes the exceedance of X_i over the threshold and N_t is the number of exceedances of the threshold. To address this numerical optimization problem, Grimshaw's trick is employed to transform the two variables in the equation into a single-variable problem. Let $\zeta(\gamma,\sigma) = logL(\gamma,\sigma)$, if ζ has a maximum, then the solution for the system $\nabla\zeta(\gamma,\sigma) = 0$ exists, denoted as (γ^*,σ^*) At this point, $x^* = \gamma^*/\sigma^*$ is the solution to the scalar equation $u(x)v(x) = 1$, which is expressed as:

$$u(x) = \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{1}{1+xY_i}, v(x) = 1+\frac{1}{N_t}\sum_{i=1}^{N_t}\log(1+xY_i) \tag{15}$$

From the above equation, $\gamma^* = v(x^*) - 1, \sigma^* = \gamma^*/x^*$ can be solved. The adaptive threshold th based on Maximum Likelihood Estimation is given as follows:

$$th = t + \frac{\sigma^*}{\gamma^*}\left(\left(\frac{qn}{N_t}\right)^{-\gamma^*} - 1\right) \tag{16}$$

where q is the risk coefficient and n is the number of data inputs. Based on the above equation, the adaptive threshold th, for subsequent input node parameter sequence points X_i, if $X_i > th$, it is considered that the point is in an abnormal state. If $Y_i = X_i - t$, then the exceeding part is updated in the algorithm by computing the updated γ^*, σ^*, and the threshold th. The name of this thresholding method is Streaming Peaks-Over-Threshold (SPOT).

4. Drift Distribution for Streaming Data Anomalies

However, when the missile system malfunctions, the distribution of the original flight state data changes. Therefore, the precision of the Pareto distribution parameters obtained using Maximum Likelihood Estimation will significantly decrease, leading to an inability to describe the extreme value distribution of the faulty data. To address this issue, the DSPOT method is introduced to complement the shortcomings of SPOT. Unlike SPOT, which models absolute value data, DSPOT models relative value data. DSPOT assumes that the local relative changes in the data generally follow the same distribution, allowing it to more sensitively detect abnormal changes in the data after a fault occurs, i.e., at time i, the input data changes from X_i to $X_i - M_i$ where M_i is the average of the data selected in the sliding window, $M_i = \sum_{k=1}^{d} X_{i-k}/d$,$d$ is the window length, and $X_{i-1},\ldots,X_{i-d}$ is the last d normal data points.

Based on the Drift Streaming Peaks-Over-Threshold principle, the algorithm takes as input a sequence of anomalous values to be detected and calculates the size of the adaptive threshold in real time. At a given moment, if the anomaly output from HBOS exceeds the adaptive threshold, it is determined that a system fault has occurred, and the current timestamp is recorded.

3.2.3. Overall Flow of the Detection Algorithm

The online fault-detection algorithm based on the HBOS method constructs a histogram probability density model by collecting online feature value data under normal operating conditions. The real-time multidimensional data to be detected is input into the histogram probability density model to obtain anomaly scores. Combined with the current

anomaly scores and adaptive threshold, the real-time fault-detection structure is obtained. The detection algorithm process is shown in Figure 4.

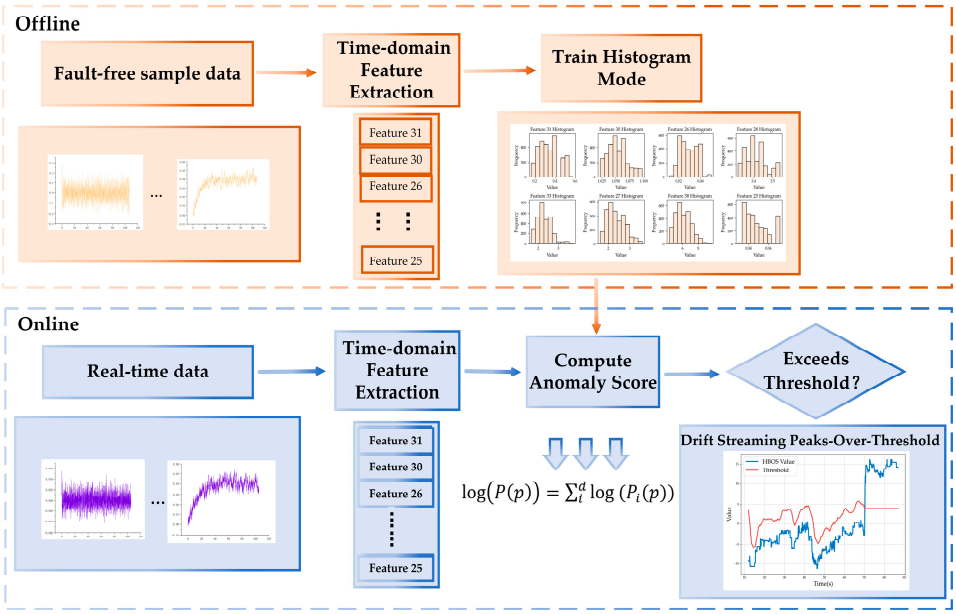

Figure 4. Overall flow of the detection algorithm.

4. Fault-Localization Algorithm Based on Symbolic Directed Graphs

Fault localization follows fault detection closely, requiring accuracy and real-time performance. This paper adopts a hierarchical approach to establish a hierarchical directed graph for fault-localization inference algorithms. Real-time monitoring of node parameters is conducted, and the first fault state of each node is obtained through adaptive thresholds, determining the effectiveness of fault propagation for each layer of nodes. Finally, a fault-localization inference algorithm based on the principle of compatible paths is proposed.

4.1. Hierarchical Symbolic Directed Graph

4.1.1. Symbolic Directed Graph

The symbolic directed graph (SDG) is a type of directed graph model used to represent and analyze the dynamic behavior of systems in a qualitative graphical manner. It is typically employed to describe target objects and reveal the interrelationships among the internal components of these objects. The SDG model offers convenience in modeling, provides a visual representation, and enables qualitative reasoning analysis without the need for precise mathematical models [44].

The symbolic directed graph is represented as $G = (V, E)$, where $V = (v_1, v_2, \ldots, v_n)$ represents the set of nodes; the status symbol of a node is $\psi = (i), i \in \{+, 0, -\}$, where " $+$ " indicates the node exceeding the normal level, "0" represents the node at the normal level, and " $-$ " indicates the node below the normal level; $E = (e_1, e_2, \ldots, e_k)$ represents the set of directed edges where $e_k = (v_i, v_j)$ indicates the directed edge from node v_i to node v_j, and the symbols of directed edges $\varphi = (e_k) \in \{+, -\}$, where " $+$ " denotes positive action and " $-$ " denotes negative action. Generally, dashed lines represent negative actions, while solid lines represent positive actions.

4.1.2. The Hierarchical Approach Based on Warshall

The symbolic directed graph contains information about several nodes, representing only the simplest connectivity between them. It is difficult to reflect the specific location of each node within the system and its interaction with the overall system from individual nodes. In short, the symbolic directed graph contains limited information that is insufficient

to provide comprehensive details for the effectiveness of fault propagation and cannot ensure the efficiency of fault-localization reasoning logic [45].

By employing a hierarchical strategy algorithm to process nodes in the symbolic directed graph, nodes are divided into several layers, making the overall system structure clearer and indirectly improving the efficiency of fault-localization reasoning. In this hierarchy, the first layer consists of source nodes, typically representing the nodes where faults originate. The higher the level of a node in the hierarchical directed graph, the less it is affected by the source nodes and the later it is affected. The steps of the hierarchical strategy are as follows:

- The adjacency matrix of the directed graph obtained based on the symbol-directed graph model is represented by the following formula.

$$A_{ij} = \begin{cases} 1, \text{If there is an edge from vertex i to vertex } j \\ 0, \text{Otherwise} \end{cases} \tag{17}$$

- Based on the reachable matrix A_{ij}, the Warshall algorithm is utilized to compute the reachability matrix of the directed graph, which is defined as follows:

$$R_{ij} = \begin{cases} 1, \text{There exists a path from node i to node } j \\ 0, \text{ There is no path from node i to node } j \end{cases} \tag{18}$$

- Based on the reachable matrix R_{ij}, determine whether the directed graph satisfies the hierarchical conditions, i.e., whether there exists a permutation matrix T such that $T^T AT$ is a block triangular matrix. If the hierarchical conditions are satisfied, based on the intersection of the reachable set and the antecedent set, the node can be categorized as a first-layer node. By traversing and computing all nodes, the set of first-layer nodes is obtained. After removing the nodes categorized as first-layer nodes, repeat the process iteratively until the directed graph is completely layered. The reachable set is defined as the set of nodes corresponding to the elements with a value of 1 in the i-th row of the reachable matrix. The antecedent set is defined as the set of nodes corresponding to the elements with a value of 1 in the j-th column of the reachable matrix.

4.2. Bilateral Drift Streaming Peaks-Over-Threshold

Symbolic directed graph nodes can abstractly represent subcomponents in an object system or parameters within a system. As indicated, the states of these nodes can be categorized as "+1", "−1", or "0" [46,47]. These three states are determined based on a comparison between the current state of the node and the historical normal operational states. In other words, leveraging historical prior knowledge, the current state of the node is assessed to be either within normal operational bounds, beyond normal bounds or below normal bounds. Consequently, corresponding upper and lower threshold values need to be set, enabling online anomaly detection for node parameters.

The threshold generation principle of the Bilateral Drift Streaming Peaks-Over-Threshold method is similar to that of the DSPOT method. The upper threshold is derived from the extreme value distribution of the data through Maximum Likelihood Estimation, while the lower threshold is derived from the distribution of the minimum values in the data through Maximum Likelihood Estimation. If a node parameter's input sequence point is above or below the respective threshold, it is deemed as an outlier point, denoted as an upper outlier point X_i^{fu} or lower outlier point X_i^{fd} accordingly. Let the time of fault occurrence be t_f, and t be a time point after the fault occurrence. During the time interval $t - t_f$, if the node parameter does not exist or falls within the normal range, it is considered that the node is not affected by the fault at time t, and the node's state at time t is considered to be "0". Conversely, if the node parameter exists and is an outlier point during the time interval $t - t_f$, it is considered that the node is affected by the fault at time t. Moreover, if

the number of upper outlier points X_i^{fu} exceeds the number of lower outlier points X_i^{fd} within this time interval:

$$A = \left\{X_i^{fu}\right\}, B = \left\{X_i^{fd}\right\}, \text{ card } A > \text{card } B \tag{19}$$

it is considered that the node's state at time t is "1"; otherwise, it is considered to be "-1".

4.3. Propagation Effectiveness and Fault Inference Localization Strategy

In an SDG model, if $\psi(v_i) \neq 0$, it indicates that node v_i has been affected by a fault, making it an effective node; otherwise, it is an ineffective node [44]. The principle of compatible pathways can be utilized to assess propagation effectiveness. The fundamental logic for determining propagation effectiveness is to assess whether there exist effective propagation parent nodes within the set of nodes beyond the second layer in a hierarchical directed graph.

The principle of compatible pathways can be utilized to assess propagation effectiveness. The fundamental logic for determining propagation effectiveness is to evaluate whether there exist effective propagation parent nodes within the set of nodes beyond the second layer in a hierarchical directed graph. Let node i belong to the set of nodes beyond the second layer. If its initial fault state is ψ_i, and if there exists a connected parent node j with an initial fault state ψ_j, where the connection line state between the parent and child nodes is φ_{ij}, then:

$$\psi_i * \varphi_{ij} * \psi_j = 1 \tag{20}$$

Satisfying Equation (14) indicates that the child node i is an effective node, and the node j is an effective propagation parent node of the child node i. The connection branch between the parent and child nodes is a compatible branch, indicating that the fault propagates to the child node through this branch.

Once propagation effectiveness is ensured, the basic logic for tracing and locating faults involves inferring the fault location based on the initial fault states of the second-layer node set. When there are multiple connection pathways between the second-layer nodes and the first-layer fault source nodes, and the states of these connection pathways can be either -1 or 1, some of these pathways may become compatible pathways for propagating faults when faults occur in the first-layer nodes. Based on the initial fault states of the second-layer nodes, if there exists a change in the first-layer nodes such that all pathways pointing towards the second-layer nodes become compatible pathways, then the node in question is considered a fault source node.

$$\begin{gathered} K = \left\{\psi_1^i \middle| \psi_1^i \text{ represents the states of all nodes in the first layer}\right\} \\ F = \left\{\psi_2^i \middle| \psi_2^i \text{ represents the states of all nodes in the sec ond layer}\right\} \\ U = \left\{i \middle| \psi_1^i \in K, \forall \psi_2^j \in F,\ \psi_i * \varphi_{ij} * \psi_j = 1\right\} \end{gathered} \tag{21}$$

where U represents the final set of located fault source nodes.

5. Experiment and Verification

5.1. Missile Flight Semi-Physical Simulation Platform

This paper focuses on the analysis of control surface and lifting surface faults of missile systems during cruising, considering the influence of stochastic environments and errors inherent to the aircraft. A semi-physical real-time simulation platform for missile flight, as shown in Figure 5, is developed. The platform consists of three main components:

- Real-time simulation computer: Deployed to simulate missile aerodynamic parameters, dynamic kinematics, navigation, guidance control algorithms, and injection of lifting surface faults.
- Three-degree-of-freedom platform: Used to respond to the real-time simulation computer's output of missile flight attitude. Simultaneously, it utilizes onboard inertial

sensors to collect attitude information and transmit it back to the simulation engine, forming a closed-loop control.

- Servo loading platform: Responds to the simulation engine's output of servo deflection angle information. Simultaneously, it transmits the encoder-collected servo surface angle back to the simulation computer. Additionally, this platform is equipped with clutches and load-loading platforms to simulate typical faults in control surfaces.

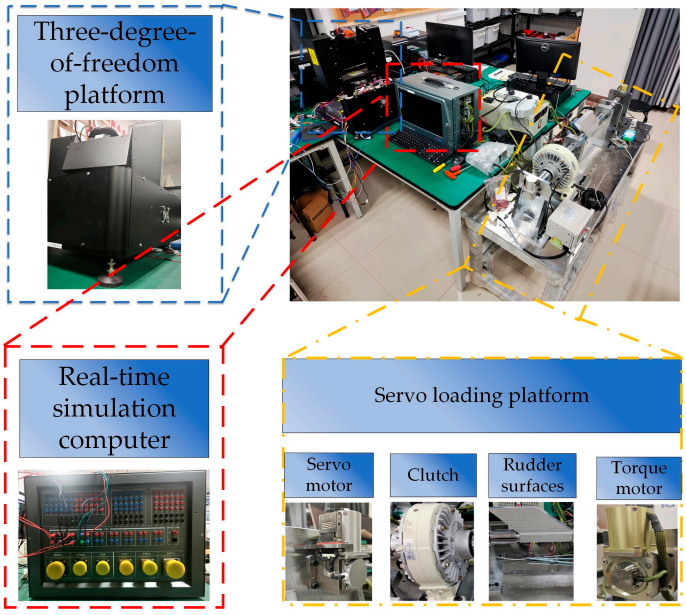

Figure 5. Missile semi-physical real-time simulation platform.

By engaging and disengaging the clutch, the connection between the motor shaft and the rudder surface shaft is simulated. The torque generated by the torque motor is used to simulate normal aerodynamic drag of the rudder surface as well as a stuck fault. The digital simulation of rudder surface damage and lifting surface damage is performed within the real-time simulation system.

Set the aerodynamic parameter random deviation to $\pm 10\%$ and mass and moment of inertia random deviation to $\pm 5\%$. By introducing random perturbations to the missile parameters, the flight states after each trim process are intentionally offset, simulating the uncertainties in missile flight. This random offset in the flight state captures the inherent unpredictability and variability in actual missile flight conditions. Initial attitude angles are all set to 0°, cruise speed is set to 240 m/s, cruise altitude is set to 1 km, and random wind speed is set to ± 5 m/s. Since the rudder loose fault corresponds to the complete loss of rudder effect, the rudder loose fault does not exist as the fault scope. Experimentally set the fault types and occurrence times as shown in Table 1:

Table 1. Fault Information Settings.

Fault Type	Fault Time (s)	Fault Scope
Rudder Stuck	70	$[-30^\circ, 30^\circ]$
Rudder Damage	70	[5%, 100%]
Rudder Loose	70	----
Lifting Surface Damage	70	[5%, 100%]

5.2. Evaluation Metrics

Fault-detection metrics mainly include fault-detection rate (FDR), detection accuracy, detection time, and false-alarm rate (FAR). TP (true positive) represents the number of positive samples that are correctly detected as positive. FP (false positive) represents the

number of negative samples that are incorrectly detected as positive. FN (false negative) represents the number of positive samples that are incorrectly detected as negative. TN (true negative) represents the number of negative samples that are correctly detected as negative [48].

Fault-Detection Rate reflects the sensitivity of the model and its ability to detect fault samples. A value closer to 1 indicates higher accuracy and stronger detection capability of the model. Correctly detected samples are defined as samples successfully detecting faults within 0.5 s after the fault occurrence. Otherwise, they are considered undetected faults, classified as missed samples. Detection time T_d reflects the speed of the model in detecting faults. T represents the moment when the fault is detected compared to the actual time of fault occurrence t_f. A smaller value indicates faster detection speed and better real-time performance of the model. False-Alarm Rate FAR reflects the proportion of incorrectly detected faults by the model. A value closer to 0 indicates better detection performance of the model.

Fault localization mainly focuses on localization accuracy and localization time. Their calculation formulas are shown in Equation (13):

$$\begin{gathered} P = \frac{\text{Number of correct locations}}{\text{Total number of test samples}} * 100\% \\ T_l = \widetilde{T} - t_f - T_d \end{gathered} \tag{22}$$

The accuracy P reflects the effectiveness of the localization algorithm, which is actually related to the total number of experiments. The closer the accuracy is to 100%, the better the performance of the localization algorithm. Localization time T_l is defined as the difference between the time $\widetilde{T}$ when the fault is located and the time t_f when the fault occurred, minus the time consumed for fault detection. A smaller value indicates better real-time performance of the localization algorithm.

5.3. Fault-Detection Algorithm Validation

Based on the missile semi-physical simulation platform, the online fault-detection algorithm was tested, and multiple experiments were conducted for each type of fault. Eight features were selected as detection inputs, which best represent the fault characteristics. Using fault-free flight data samples, histogram fitting was performed for each feature dimension, as shown in Figure 6. The number of bins was set to 10. The features in the figure represent the eight time-domain features mentioned in Section 3 of this paper's Feature Selection chapter.

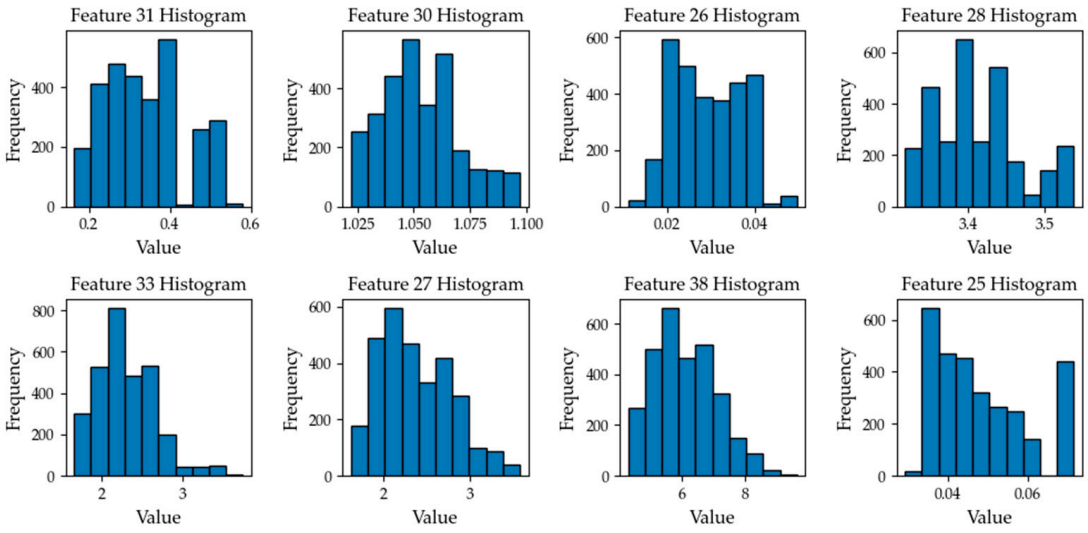

Figure 6. Histogram of missile states time-domain feature data.

Based on Equation (15). The risk coefficient q for our thresholds is set to 10^{-3}, and the length of the time-window d is set to 11. Based on the fitted histogram models and adaptive thresholds, real-time fault-detection results can be obtained. The detection results for some fault scenarios are depicted in Figure 7.

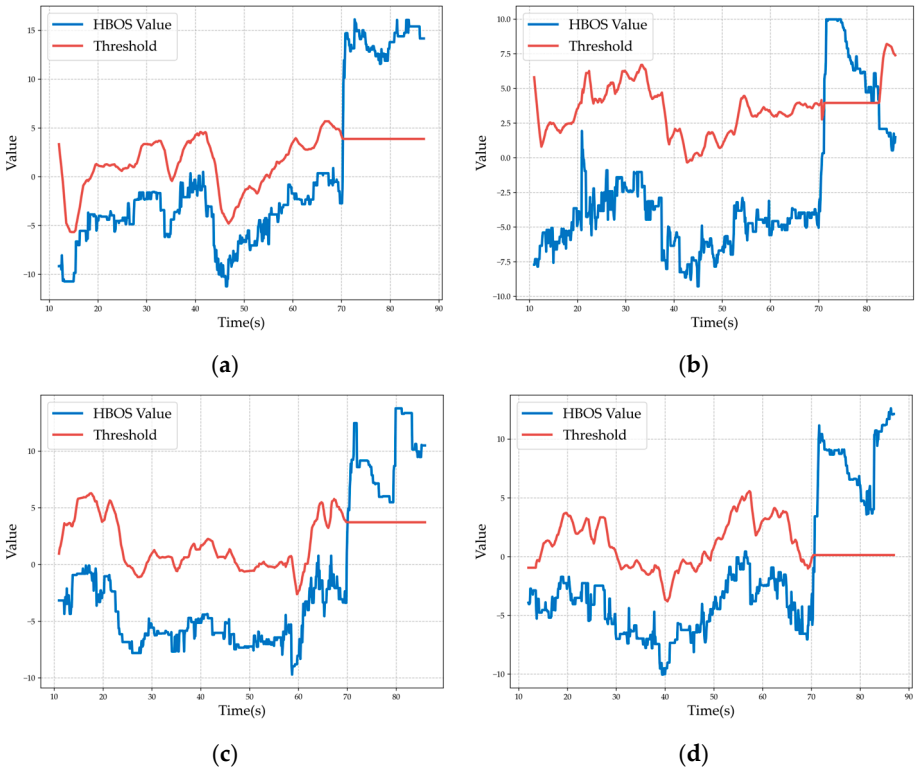

Figure 7. Fault-detection results: HBOS value and adaptive thresholds. (**a**) is the detection effect when rudder is stuck. (**b**) is the detection effect when rudder is loose. (**c**) is the detection effect when rudder is damaged. (**d**) is the detection effect under lifting surface damage.

It can be seen from the figure that the HBOS value does not exceed the threshold when no fault occurs (the time is less than 70 s), and the HBOS value quickly exceeds the adaptive threshold when the fault occurs.

To validate the performance of the algorithm, a large number of test samples are used to test the accuracy, false positive rate, detection time, and other metrics of the fault-detection algorithm. These test samples consist of data from the missile semi-physical simulation platform, including physical parameters and deviation information, to simulate the uncertainties present in real missile flights. The detection results statistics are summarized in Table 2.

Table 2. Statistical results of fault-detection effect.

Fault Type	Total Samples	Average Detection Rate	Average False Negative Rate	Average Detection Time
Rudder Stuck	400 groups	99.4%	0.23%	0.07 s
Rudder Loose	320 groups	98.9%	0.17%	0.11 s
Rudder Damage	240 groups	99.2%	0.31%	0.08 s
Lifting Surface Damage	160 groups	99.7%	0.21%	0.06 s

In the four types of fault scenarios, the online fault-detection algorithm demonstrates excellent performance with a relatively short average detection time and a high fault-detection rate. In many simulation experiments, there were only a few cases of false

negatives and false positives. False negatives occurred when significant fault characteristic values were unable to capture the subtle changes caused by minor faults, especially when the fault changes almost do not affect the normal flight of the missile, making it challenging for the algorithm to detect. False positives, on the other hand, happened due to fluctuations exceeding detection thresholds in some parameters affected by random conditions before the fault occurred or due to outliers in some flight parameters, leading to misjudgments by the detection algorithm.

To further validate the superiority of the algorithm, appropriate comparative experiments are set up using the same data samples. The comparison includes anomaly detection algorithms such as HBOS, KNN, LOF, and CBLOF [49]. A variable, denoted as *k*, represents the number of bins for the dynamic histogram in HBOS and the number of neighbors in the other algorithms. The Area Under the Curve (AUC) values [50] for each algorithm are then calculated and depicted in Figure 8.

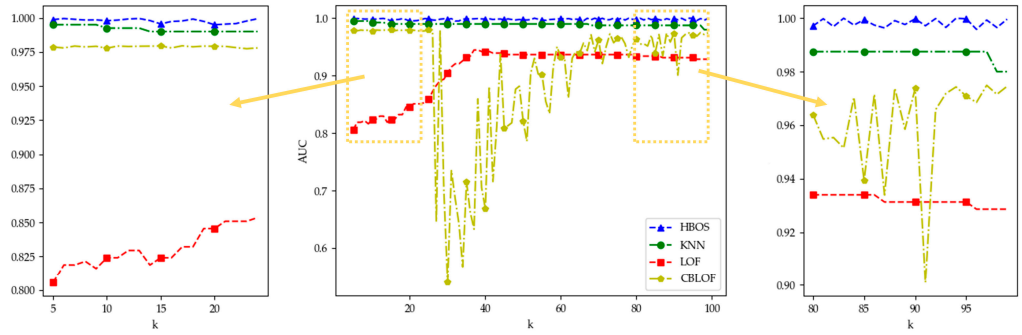

Figure 8. Comparison of AUC values of the four algorithms. k is the number of nearest neighbors, and in HBOS, the number of bins.

To better demonstrate the results of the comparative experiments, the AUC values of each algorithm will be presented in the format of Table 3, with data points taken at intervals of k = 15 from the graph.

Table 3. Comparison of AUC values of the four algorithms.

Algorithms	k = 15	k = 30	k = 45	k = 60	k = 75	k = 90
HBOS (AUC)	**0.99744**	**0.99835**	**0.99770**	**0.99830**	**0.99808**	**0.99996**
KNN (AUC)	0.99000	0.99000	0.99000	0.99000	0.98750	0.98750
LOF (AUC)	0.81887	0.89337	0.93930	0.93662	0.93662	0.93126
CBLOF (AUC)	0.97929	0.70004	0.93759	0.95030	0.95720	0.95848

From the comparative experiments, it can be observed that due to the influence of uncertainty factors in the data, conventional anomaly detection algorithms struggle to maintain satisfactory classification performance. However, the proposed detection algorithm based on HBOS and adaptive thresholds leverages the advantages of HBOS in handling long-tail distribution data and uncertainty data. Additionally, the adaptive thresholding capability for real-time adjustment of data distribution helps maintain high classification performance in uncertain data.

To ensure the real-time performance of the algorithm, the computation time of each algorithm is recorded and depicted in Figure 9.

Experimental results reveal that on the same computational platform processing identical data, the HBOS algorithm exhibits the shortest computation time, thus ensuring the highest real-time performance among other algorithms. This fulfills the real-time requirements for missile flight scenarios.

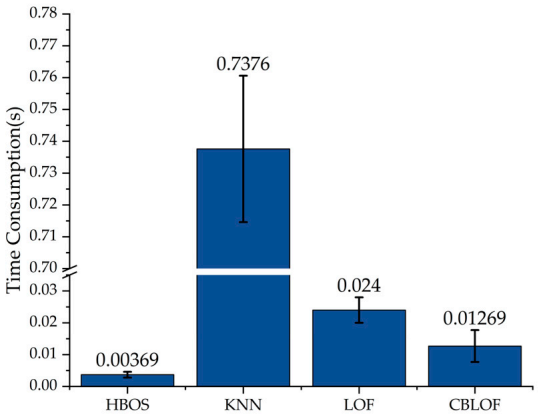

Figure 9. Comparison of average detection time of four algorithms.

5.4. Fault-Localization Algorithm Validation

According to the overall topology and component composition of the aircraft, combined with the collection of fault types in flight information, the selected main components and their corresponding flight parameters are shown in Table 4.

Table 4. Node Names and Symbols.

Node	Name	Symbol	Node	Name	Symbol
1	Angle Of Attack	α	11	Pitch Angle	θ
2	Sideslip Angle	β	12	Yaw Angle	ϕ
3	*X*-axis Acceleration	Ax	13	Rudder 1	D1
4	*Y*-axis Acceleration	Ay	14	Rudder 2	D2
5	*Z*-axis Acceleration	Az	15	Rudder 3	D3
6	*X*-axis Angular Velocity	p	16	Rudder 4	D4
7	*Y*-axis Angular Velocity	q	17	Flight Altitude	H
8	*Z*-axis Angular Velocity	r	18	Thrust Command	TC
9	Velocity	V	19	Left Lifting Surface	SL
10	Roll Angle	γ	20	Right Lifting Surface	SR

According to all the indicated flight parameters, abstract them into directed graph nodes; based on empirical knowledge and principles of aircraft kinematics and dynamics, determine the connections and influence relationships between each node to obtain the symbolic directed graph model of the aircraft, as shown in Figure 10.

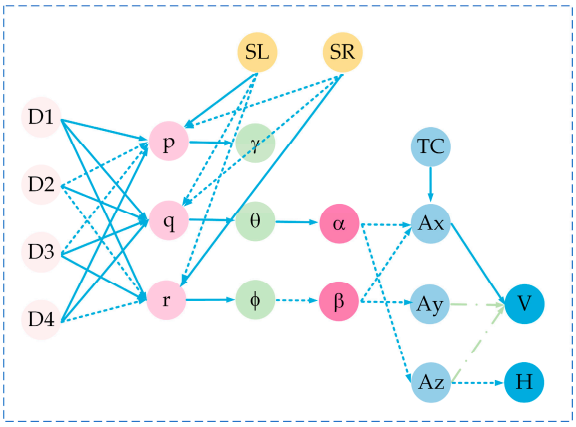

Figure 10. The symbolic directed graph model of the aircraft.

As well as solid and dashed lines representing increase and decrease actions, respectively, there exists a special type of green dash-dot line. Its significance lies in that regardless of any changes in the parent node, it will always exert an increase effect on the child node.

Based on this model, we establish an adjacency matrix and conduct layering determination, meaning there exists a permutation matrix T, such that $T^T AT$ becomes a block triangular matrix. After determining the layering of this directed graph, we utilize the Warshall algorithm to compute the reachability matrix. Following the layering strategy, the directed graph is partitioned into six layers, as shown in Figure 11.

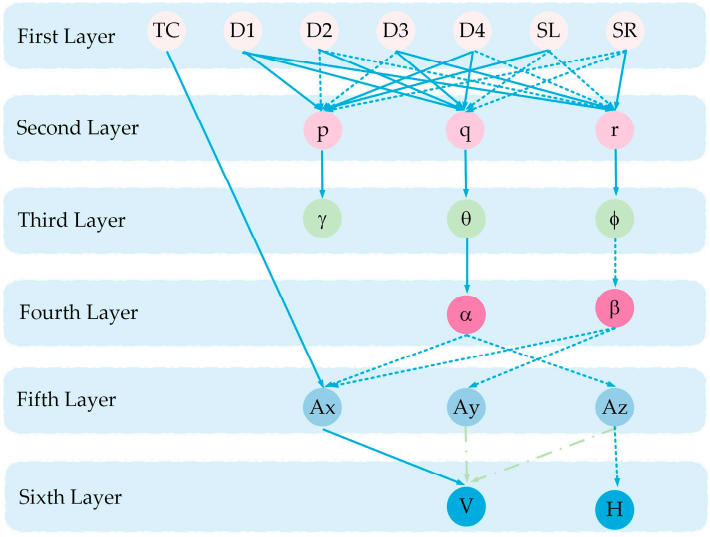

Figure 11. The layering results of the directed graph.

Using the aircraft simulation platform, experiments on online fault-localization algorithms were conducted, with multiple trials for each fault type. The same number of simulated fault groups as in fault detection were maintained. The final results are shown in Table 5.

Table 5. Simulation results of fault-localization algorithms.

Fault Type	Total Samples	Fault-Localization Accuracy	Average Fault-Localization Time
Rudder Stuck	400 groups	98.50%	0.324 s
Rudder Loose	320 groups	98.12%	0.319 s
Rudder Damage	240 groups	97.91%	0.321 s
Lifting Surface Damage	160 groups	98.75%	0.318 s

In a large number of simulation experiment samples, a few cases of localization failure occurred, which can be classified into cases of propagation ineffectiveness and mislocalization. The propagation ineffectiveness error cases are mainly due to certain deviations in the node parameter data and the failure to capture the first fault characteristic state of the corresponding parameter within the length of 0.3 s of collected data, while the mislocalization cases are mainly caused by occasional effects of random conditions and deviations, resulting in erroneous judgments of the first fault characteristic state of some nodes by the adaptive threshold, leading to final fault-localization errors. In summary, despite the existence of a few cases of localization failure, the localization accuracy of the aforementioned simulation experiments remains at an excellent level.

Taking the example of the fault of the second rudder stuck at $-14°$, after detecting the fault, the real-time fault-localization algorithm was run. The missile flight status data (taking the angle of attack, sideslip angle, Y-axis acceleration, and roll angle as examples) and adaptive thresholds after the fault are shown in Figure 12.

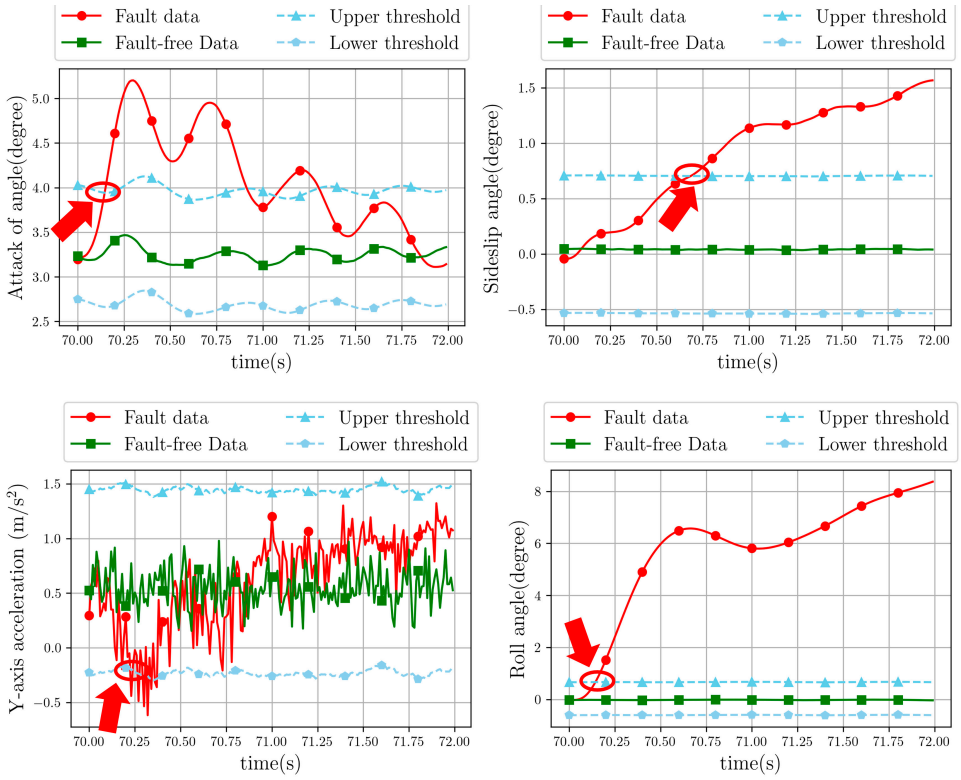

Figure 12. Calculation of node status based on bilateral adaptive threshold.

The state of the node is obtained by marking with red circles and arrows whether the flight state exceeds the upper or lower bounds of the threshold. When the flight status exceeds the upper threshold, the first fault characteristic state of the node is "1"; when it falls below the lower threshold, the node state is "−1". If it does not exceed the threshold, the node state is "0". The first fault characteristic state of all nodes was obtained, as shown in Table 6.

Table 6. Stuck fault Node status in a flight condition.

ID	1	2	3	4	5	6	7
NodeSymbol	p	q	r	γ	θ	ϕ	α
Node Status	1	−1	1	1	−1	1	−1
ID	8	9	10	11	12	13	
NodeSymbol	β	Ax	Ay	Az	V	H	
Node Status	−1	1	1	1	0	0	

At this point, the fault's influence has not propagated to the sixth layer yet, and neither the speed nor the altitude exceeds the adaptive threshold. Therefore, the state of the nodes is "0". Thus, it is only necessary to validate the propagation validity of the nodes from the second layer to the fifth layer. Based on Equation (14), it is determined that the data satisfies propagation validity, where effective fault sub-nodes located at or above the second layer all have effective propagation parent nodes.

According to the second-layer effective fault nodes and the localization strategy, the final localization result D2 is output. When the first fault feature state of D2 is "−1", it can become an effective propagation parent node for all nodes in the second layer, indicating correct localization with a localization time T_l of 0.315 s.

In some localization cases, there are situations where the localization result contains two nodes in the fault source set. In such cases, precise localization can be achieved

based on the deflection status of the rudder nodes in the localization result. The specific verification results are shown in Table 7.

Table 7. Locating the fault source multi-node scenario.

Fault Type	Number of Experiments	Localization Result	Rudder Inferred State	Rudder Actual State
Stuck third rudder	20	D3/SL	1	1
Stuck fourth rudder	20	D4/SR	−1	−1
Left lifting surface damaged	20	D3/SL	1	−1
Right lifting surface damaged	20	D4/SR	1	−1

Based on Table 6, it can be inferred that when there are multiple fault source nodes in the preliminary localization results, if the corresponding rudder deflection states can uniquely locate the fault source node, then when the inferred rudder state matches the actual rudder deflection state, it indicates a rudder fault; when the inferred rudder state is opposite to the actual rudder deflection state, it indicates the exclusion of a rudder fault and implies a lifting surface fault. In the aforementioned simulation experiments, the accuracy of this method reached over 97%, demonstrating the precision of online fault localization based on this approach.

6. Conclusions

This paper proposes fault detection and fault-localization methods specifically designed for typical faults in missile rudder surfaces and lifting surfaces. We analyze the unique characteristics of the research object and identify the limitations of existing fault detection and localization methods when applied to the research object in this paper.

For fault detection, we propose a fault-detection algorithm based on HBOS and adaptive thresholding. After optimal feature selection, the algorithm fits a dynamic histogram model to normal data features. The algorithm then identifies outliers from the input data based on whether their values exceed the adaptive threshold, which indicates the occurrence of a system fault.

To address the fault-localization problem, we propose a hierarchical symbolic directed graph model that captures the interdependencies among missile flight states and fault propagation mechanisms. Using bilateral adaptive thresholding, we determine the states of each node and infer the location of the fault by reasoning upward through the layers of nodes.

The proposed algorithms effectively address the fault detection and fault-localization issues related to control surfaces and lift surfaces in practical missile scenarios. Through testing on a semi-physical platform, the following conclusions are drawn:

1. For the fault-detection algorithm, multiple sets of rudder surface and lifting surface data with typical faults are collected under random perturbations of aerodynamic parameters, mass, and moment of inertia. The average accuracy of the proposed fault-detection algorithm is no less than 98.9%, with an average false-alarm rate of no more than 0.31% and an average detection time of no more than 0.11 s. When comparing the AUC values and computation time of the proposed fault-detection algorithm with typical anomaly detection algorithms, the proposed algorithm shows the highest accuracy and the shortest computation time, making it better suited for missile fault-detection scenarios.
2. For the fault-localization algorithm, multiple sets of rudder surface and lifting surface data with typical faults are collected under random perturbations of aerodynamic parameters, mass, and moment of inertia. In a few specific node state scenarios, there may be two possible fault locations during the upward reasoning process of the constructed layered directed graph. However, combining the actual position of the control surface enables accurate fault localization without compromising the

localization accuracy. Therefore, the fault-localization algorithm achieves an accuracy of no less than 97.91% and an average localization time of no more than 0.324 s.

The work in this paper can be applied to the analysis of other complex aircraft fault types, suitable for scenarios with few fault data samples and complex flight environments. It contributes to simplifying the research on fault detection and localization algorithms for complex systems, providing a foundation and support for system health management and fault diagnosis.

Author Contributions: Conceptualization, H.H., Y.C., and B.J.; methodology, H.H. and Y.C.; validation, H.H. and Y.C.; formal analysis, H.H., W.L. and Y.C.; investigation, H.H. and K.G.; data curation, H.H. and W.L.; writing—original draft preparation, H.H. and Y.C.; writing—review and editing, H.H. and B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (No. 2023YFB3307102) and the National Natural Science Foundation Integration Project (No. U22B6001).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: The authors of this paper would like to thank the FDD team at Nanjing University of Aeronautics and Astronautics for their support of this research. We also thank the editors for their rigorous and efficient work and the reviewers for their helpful suggestions.

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

References

1. Betts, R.K. Cruise missiles: Technology, strategy, politics. *Wash. Q.* **1981**, *4*, 66–80. [CrossRef]
2. Siouris, G.M. Cruise missiles. In *Missile Guidance and Control Systems*; Springer: New York, NY, USA, 2004; pp. 521–587.
3. Rong, H.; Tian, J.; Zhao, T. Temporal uncertainty analysis of human errors based on interrelationships among multiple factors: A case of Minuteman III missile accident. *Appl. Ergon.* **2016**, *52*, 196–206. [CrossRef]
4. Liu, Z.; Xiao, M.; Zhu, H. Acquisition of missile fault diagnosis knowledge based on incomplete information of flow graph. *Earth Environ. Sci.* **2021**, *632*, 032055. [CrossRef]
5. Chen, J.; Yao, B.; Lu, Q. A safety dynamic evaluation method for missile mission based on multi-layered safety control structure model. *Reliab. Eng. Syst. Saf.* **2024**, *241*, 109678. [CrossRef]
6. Chen, H.; Jiang, B. A review of fault detection and diagnosis for the traction system in high-speed trains. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 450–465. [CrossRef]
7. Abid, A.; Khan, M.T.; Iqbal, J. A review on fault detection and diagnosis techniques: Basics and beyond. *Artif. Intell. Rev.* **2021**, *54*, 3639–3664. [CrossRef]
8. Mellit, A.; Tina, G.M.; Kalogirou, S.A. Fault detection and diagnosis methods for photovoltaic systems: A review. *Renew. Sustain. Energy Rev.* **2018**, *91*, 1–17. [CrossRef]
9. Gururajapathy, S.S.; Mokhlis, H.; Illias, H.A. Fault location and detection techniques in power distribution systems with distributed generation: A review. *Renew. Sustain. Energy Rev.* **2017**, *74*, 949–958. [CrossRef]
10. Gao, Z.; Cecati, C.; Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3757–3767. [CrossRef]
11. Zhang, K.; Jiang, B.; Yan, X.G. Incipient fault detection for traction motors of high-speed railways using an interval sliding mode observer. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 2703–2714. [CrossRef]
12. Dai, X.; Gao, Z. From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis. *IEEE Trans. Ind. Inform.* **2013**, *9*, 2226–2238. [CrossRef]
13. Li, W.; Yang, W.; Jin, G. Clustering federated learning for bearing fault diagnosis in aerospace applications with a self-attention mechanism. *Aerospace* **2022**, *9*, 516. [CrossRef]
14. Huang, P.; Wang, T.; Ding, L. Comparative Analysis of Real-Time Fault Detection Methods Based on Certain Artificial Intelligent Algorithms for a Hydrogen–Oxygen Rocket Engine. *Aerospace* **2022**, *9*, 582. [CrossRef]
15. Memarzadeh, M.; Matthews, B.; Templin, T. Multiclass Anomaly Detection in Flight Data Using Semi-Supervised Explainable Deep Learning Model. *J. Aerosp. Inf. Syst.* **2022**, *19*, 83–97. [CrossRef]
16. Ince, T.; Kiranyaz, S.; Eren, L. Real-time motor fault detection by 1-D convolutional neural networks. *IEEE Trans. Ind. Electron.* **2016**, *63*, 7067–7075. [CrossRef]
17. Guo, K.; Wang, N.; Liu, D. Uncertainty-aware LSTM based dynamic flight fault detection for UAV actuator. *IEEE Trans. Instrum. Meas.* **2022**, *72*, 1–13. [CrossRef]

18. Cui, H.; Yan, J.; Zhang, P. Research on Fault Detection Technology of Air-to-Air Missile Based on Data Mining. In Proceedings of the 5th China Aeronautical Science and Technology Conference, Jiaxing, China, 2 November 2021; pp. 788–795.
19. Hosking, J.R.; Wallis, J.R.; Wood, E.F. Estimation of the generalized extreme-value distribution by the method of probability-weighted moments. *Technometrics* **1985**, *27*, 251–261. [CrossRef]
20. Zhu, J.; Lubkeman, D.L.; Girgis, A.A. Automated fault location and diagnosis on electric power distribution feeders. *IEEE Trans. Power Deliv.* **1997**, *12*, 801–809.
21. Yadav, A.; Dash, Y. An overview of transmission line protection by artificial neural network: Fault detection, fault classification, fault location, and fault direction discrimination. *Adv. Artif. Neural Syst.* **2014**, *2014*, 230382. [CrossRef]
22. He, F.; Hu, X.; Zhang, D. Ground fault location research based on multidimensional scaling and density clustering algorithm. *J. Phys. Conf. Ser.* **2022**, *2306*, 012009. [CrossRef]
23. Xu, Y.; Xu, B.; Huang, Y. Grounding fault location of large turbo generator based on stator winding distribution. *J. Phys. Conf. Ser.* **2022**, *2351*, 012030. [CrossRef]
24. Zhang, R.; Liu, L. Distribution network regionalized fault location based on an improved manta ray foraging optimization algorithm. *Electronics* **2022**, *11*, 2342. [CrossRef]
25. Zhang, Z.; Lei, Y.; Mao, X. A study of effectiveness of deep learning in locating real faults. *Inf. Softw. Technol.* **2021**, *131*, 106486. [CrossRef]
26. Dashtdar, M.; Dashti, R.; Shaker, H.R. Distribution network fault section identification and fault location using artificial neural network. In Proceedings of the 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE), Istanbul, Turkey, 3–5 May 2018; pp. 273–278.
27. Reche, E.A.; de Sousa, J.V.; Coury, D.V. Data mining-based method to reduce multiple estimation for fault location in radial distribution systems. *IEEE Trans. Smart Grid* **2018**, *10*, 3612–3619. [CrossRef]
28. Thukaram, D.; Khincha, H.P.; Vijaynarasimha, H.P. Artificial neural network and support vector machine approach for locating faults in radial distribution systems. *IEEE Trans. Power Deliv.* **2005**, *20*, 710–721. [CrossRef]
29. Ai, S.; Song, J.; Cai, G. A real-time fault diagnosis method for hypersonic air vehicle with sensor fault based on the auto temporal convolutional network. *Aerosp. Sci. Technol.* **2021**, *119*, 107220. [CrossRef]
30. Jin, L.; Du, M.; Ma, J. Deep Learning Based Active Fault-Tolerant Control for Missile Actuators. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 5760–5766.
31. Bajpai, G.; Chang, B.C.; Kwatny, H.G. Design of fault-tolerant systems for actuator failures in nonlinear systems. In Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301), Anchorage, AK, USA, 8–10 May 2002; Volume 5, pp. 3618–3623.
32. Pashilkar, A.A.; Sundararajan, N.; Saratchandran, P.A. fault-tolerant neural aided controller for aircraft auto-landing. *Aerosp. Sci. Technol.* **2006**, *10*, 49–61. [CrossRef]
33. Boskovic, J.D.; Mehra, R.K. Stable multiple model adaptive flight control for accommodation of a large class of control effector failures. In Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251), San Diego, CA, USA, 2–4 June 1999; Volume 3, pp. 1920–1924.
34. Zhong, L. Contribution to Fault Tolerant Flight Control under Actuator Failures. Doctoral Dissertation, INSA De Toulouse, Toulouse, France, 2014.
35. Goldstein, M.; Dengel, A. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. In Proceedings of the 35th German Conference on Artificial Intelligence, Saarbrücken, Germany, 24–27 September 2012; Volume 1, pp. 59–63.
36. Jain, P.H.; Bhosle, S.P. Analysis of vibration signals caused by ball bearing defects using time-domain statistical indicators. *Int. J. Adv. Technol. Eng. Explor.* **2022**, *9*, 700.
37. Nguyen, T.P.; Khlaief, A.; Medjaher, K. Analysis and comparison of multiple features for fault detection and prognostic in ball bearings. In Proceedings of the Fourth European Conference of the Prognostics and Health Management Society, Utrecht, The Netherlands, 3–6 July 2018; pp. 1–9.
38. Shukla, S.; Yadav, R.N.; Sharma, J. Analysis of statistical features for fault detection in ball bearing. In Proceedings of the 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India, 10–12 December 2015; pp. 1–7.
39. Rényi, A. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*; University of California Press: Berkeley, CA, USA, 1961; Volume 4, pp. 547–562.
40. Cui, H.; Zhang, L.; Kang, R. Research on fault diagnosis for reciprocating compressor valve using information entropy and SVM method. *J. Loss Prev. Process Ind.* **2009**, *22*, 864–867. [CrossRef]
41. Zeng, Z.; Jin, G.; Xu, C. Spacecraft telemetry anomaly detection based on parametric causality and double-criteria drift streaming peaks over threshold. *Appl. Sci.* **2022**, *12*, 1803. [CrossRef]
42. Beirlant, J.; Goegebeur, Y.; Segers, J. *Statistics of Extremes: Theory and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
43. Grimshaw, S.D. Computing maximum likelihood estimates for the generalized Pareto distribution. *Technometrics* **1993**, *35*, 185–191. [CrossRef]
44. Guohua, W.; Jiaxin, W.; Diping, Y. Research on Fault Diagnosis Based on Hierarchical Signed Directed Graph for Nuclear Power Plants. *Int. Conf. Nucl. Engineering. Am. Soc. Mech. Eng.* **2020**, *83778*, V002T08A024.

45. Aini, A.; Salehipour, A. Speeding up the Floyd–Warshall algorithm for the cycled shortest path problem. *Appl. Math. Lett.* **2012**, *25*, 1–5. [CrossRef]
46. Liu, Y.K.; Wu, G.H.; Xie, C.L. A fault diagnosis method based on signed directed graph and matrix for nuclear power plants. *Nucl. Eng. Des.* **2016**, *297*, 166–174. [CrossRef]
47. He, B.; Chen, T.; Yang, X. Root cause analysis in multivariate statistical process monitoring: Integrating reconstruction-based multivariate contribution analysis with fuzzy-signed directed graphs. *Comput. Chem. Eng.* **2014**, *64*, 167–177. [CrossRef]
48. Cao, X.; Yang, R.; Guo, C. Conditional Enhanced Variational Autoencoder-Heterogeneous Graph Attention Neural Network: A Novel Fault Diagnosis Method for Electric Rudders Based on Heterogeneous Information. *Sensors* **2024**, *24*, 272. [CrossRef] [PubMed]
49. Teng, M. Anomaly detection on time series. *2010 IEEE Int. Conf. Prog. Inform. Comput.* **2010**, *1*, 603–608.
50. Huang, J.; Ling, C.X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 299–310. [CrossRef]