Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 6608967, 19 pages
https://doi.org/10.1155/2023/6608967

Research Article

An Intelligent Fault Detection Framework for FW-UAV Based on Hybrid Deep Domain Adaptation Networks and the Hampel Filter

Yizong Zhang,[1] **Shaobo Li**,[1,2] **Qiuchen He**,[1] **Ansi Zhang**,[1,2] **Chuanjiang Li**,[1,3] **and Zihao Liao**[2]

[1]*School of Mechanical Engineering, Guizhou University, Guiyang 550025, China*
[2]*State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China*
[3]*Department of Mechanical Engineering, KU Leuven, Leuven 3001, Belgium*

Correspondence should be addressed to Shaobo Li; lishaobo@gzu.edu.cn

Received 16 December 2022; Revised 15 May 2023; Accepted 26 May 2023; Published 7 June 2023

Academic Editor: Fabio Caraffini

Copyright © 2023 Yizong Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fixed-wing unmanned aerial vehicles (FW-UAVs) play an essential role in many fields, but the faults of FW-UAV components lead to severe accidents frequently; so, there is a need to continuously explore more intelligent fault detection methods to improve the safety and reliability of FW-UAVs. Deep learning provides advanced solution ideas for future UAV fault detection, but the current lack of UAV monitoring data limits the advantages of deep learning in UAV fault detection, which are both a challenge and an opportunity. In this paper, we mainly consider the data availability of deep learning under various practical flight conditions of FW-UAVs and propose a fault detection framework based on hybrid deep domain adaptation BiLSTM networks and the Hampel filter (HDBNH), the main purpose of which is to learn the knowledge of acquired data for detecting FW-UAV faults in other unknown operating conditions. HDBNH consists of three modules: feature extractor, domain adaptor, and fault detector. The feature extractor is two BiLSTM networks constructed to extract the past and future state features from the time-series flight data. The discrepancy of feature distribution between different domains is effectively reduced in the domain adaptor by a hybrid adversarial and the maximum mean discrepancy (MMD) domain adaptation method. The fault detector consists of a fault classification module and a Hampel filter. According to the continuous and dynamic characteristics of FW-UAV state changes, the Hampel filter is used to detect and correct the predicted values of the fault classification module. Meanwhile, a new state sample preparation strategy is proposed to support the work of HDBNH better. Finally, the effectiveness of HDBNH is confirmed by conducting extensive experiments in real FW-UAV flight data.

1. Introduction

As one of the representatives of complex systems, unmanned aerial vehicles (UAVs) are widely used in various fields because of their low manufacturing cost, high mobility, and high efficiency. However, UAVs have more uncontrollable factors than manned aircraft, and there are more challenges and potential threats in the process of mission execution [1]. Researchers are continuously exploring more intelligent fault detection methods to reduce the failure of system components, improve the safety and reliability of UAV systems, and ensure that UAVs accomplish various complex tasks. Currently, the proposed methods mainly include model-based, knowledge-based, and deep learning-based methods.

The main idea of the model-based approach is to establish an accurate mathematical analytical model, compare the analytical model's theoretical value with the UAV's real state value, and judge the working state of the UAV system. The authors in reference [2] considered the use of a kinematic model and an adaptive extended Kalman filter (EKF) to detect UAV faults that minimize turbulent disturbances. However, errors associated with the linearization of the EKF may reduce the detection accuracy and may even lead to filter divergence. The adaptivity of the process noise covariance (R and Q) of the EKF to sensor/actuator faults is considered in [3] so that the estimation characteristics do not deteriorate. Compared to [3], the approach in [4] is more adaptive, with the KF embedded in the neural network used to weight the parameters of the neural network to update them to identify various faults in the UAV sensors and actuators. At present, model-based methods are the most widely researched and applied (especially various nonlinear observers and Kalman filters), which have certain superiority in real-time state analysis and real-time fault diagnosis, but establishing an accurate analytical model for complex UAVs is not easy to achieve, and there are often cases of tedious calculations and errors resulting in misdiagnosis or omission. The anti-interference capability also needs to be improved.

The knowledge-based methods consider the full application of the prior knowledge accumulated by the experts in practice to the fault detection of the UAVs, which is a process of simulating human logical thinking and reasoning. Knowledge-based fault trees [5], expert systems [6, 7], and fuzzy reasoning [8] methods have all evolved accordingly, but there has been a gradual decline in the related research and coverage. The UAV fault tree model is simulated by some methods (such as Monte Carlo), and the components with poor reliability and good reliability are found, which reduces the time cost of manual evaluation to a certain extent [9], but there are often various difficulties in obtaining the cause of the fault. The expert system uses the experience accumulated by domain experts to build a knowledge base and designs programs to simulate human experts' reasoning and decision-making process for fault diagnosis. However, it lacks effective self-learning and adaptive ability. The authors in reference [7] combine an expert system with the artificial neural network, which enhances its adaptive ability to a certain extent, promotes the development of this kind of method, and brings new challenges. The authors in reference [10] combine the fuzzy inference system (FIS) with a particle filter (PF)-estimated state residual to detect the abnormality of the UAV airborne navigation sensor, which improves the real-time performance of fault detection. However, the large amount of computation of PF and the low performance of FIS limits its ability to detect anomalies. In conclusion, although the knowledge-based method solves the problem of accurate modeling of the diagnosed system to some extent, it is faced with some problems, such as difficult knowledge reasoning, difficult knowledge acquisition, self-updating of related systems, and poor self-learning ability.

Deep learning, with strong nonlinear feature extraction ability, has yielded excellent results in many fields including fault diagnosis [11] and is increasingly considered for UAV faults. To give full play to the advantages of deep learning, many researchers try to collect available data through various methods, such as artificially destroying the blades of drones and collecting data in a safe area [12–14], obtaining fault data through simulation [15–17], and injecting faults into flying drones through software [18, 19]. Nevertheless, due to the multiple limitations of the UAVs themselves and the diversity and complexity of their mission environment, it still faces problems such as scarcity of fault samples, sample imbalance, and difficulty in obtaining samples from complex environments. Li et al. [20] proposed a Siamese hybrid neural network (SHNN) framework for UAV fault diagnosis in a limited sample space. However, the overall performance is still much worse than in other fields (e.g., bearing fault diagnosis). Yang et al. [21] used a sparse autoencoder to reconstruct the data to achieve the effect of data cleaning while preserving as much as possible the original fault knowledge of the data to improve the efficiency of diagnosis. Gao et al. [22] designed a transfer learning framework based on bidirectional long short-term memory (BiLSTM) networks using a multikernel MMD (MK-MMD) domain adaptation method to reduce the variability between two domains, applied to the case of insufficient samples in the target domain. Bondyra et al. [23] proposed a fault detection algorithm based on signal processing and machine learning to use the acceleration data of IMU sensors to accurately identify rotor faults. The abovementioned methods all use flight log data, and in order to explore other available data, the authors in reference [24, 25] considered using audio data to train a UAV fault detection model, but audio data are susceptible to interference and may be challenging to work in more complex situations. The deep learning-based approaches only need data to build fault detection models, which do not require the establishment of accurate mathematical models or rely on expert knowledge and is more intelligent than the previous two approaches and also cater to the trend of big data development for UAVs [26]. Deep learning provides an advanced solution for UAV fault detection in the future, but the lack of UAV monitoring data limits the advantages of deep learning in UAV fault detection technology, which is a challenge and an opportunity for UAV fault detection technology.

In summary, deep learning-based approaches in UAV fault detection have endless potential in the future but are currently facing problems such as the scarcity of fault samples and difficulties in obtaining fault samples from complex working environments. In this paper, we try to find new ways to solve the abovementioned problems to promote the continuous efforts and innovation of deep learning in FW-UAV fault detection. We consider that it is relatively convenient to obtain FW-UAV fault data in some specific environments (such as the test flight environment and the experimental environment), which contain the knowledge required for FW-UAV fault detection. If this knowledge can be used effectively, perhaps the data dilemma can be solved. Therefore, we propose an FW-UAV fault detection method based on hybrid deep domain adaptation BiLSTM networks and the Hampel filter (HDBNH), which combines the ideas and advantages of data-driven and model-based approaches to learn the knowledge of acquired data for detecting FW-UAV faults in an unknown working environment. Compared with the previous works, the main work and contributions of this paper are as follows:

(1) A state sample preparation strategy is proposed, which solves the problems of data complexity, redundancy, nonstandard, and frequency inconsistency, while the generated state samples better support the work of HDBNH.

(2) A novel BiLSTM network combining adversarial and MMD domain adaptation is proposed, effectively reducing the difference in feature distribution between the source and target domains and better-enabling knowledge transfer.

(3) According to the continuous and dynamic characteristics of FW-UAV states [27], the Hampel filter is proposed for detecting and correcting the predicted values of BiLSTM models to improve fault detection accuracy further.

The article continues as follows: Section 2 presents the related work and briefly discusses them. Section 3 presents the proposed HDBNH framework in detail. In Section 4, the real fault dataset and the state sample preparation strategy are presented. Section 5 conducts experiments and analyses them from different perspectives. The main conclusions are given in Section 6.

2. Related Work

In this section, we will briefly review and discuss some of the work related to the proposed methodology.

2.1. Unsupervised Domain Adaptation. Transfer learning is one of the cutting-edge directions in machine learning research today [28]. The core idea is to learn and accumulate knowledge and experience in the source domain and apply it effectively to the target domain, thus compensating for the lack of labeled data. However, achieving this goal requires that the distribution of features in the source and target domains be as similar as possible. For this reason, unsupervised domain adaptation (UDA) techniques have become a hot research topic in recent years, aiming to minimize the differences in feature distribution between different domains, as shown in Figure 1.

MMD is a measure of the distance between two probability distributions. The main idea is to map two probability distributions into a high-dimensional reproducing kernel Hilbert space (RKHS) and then calculate their distance in this space, as shown in the following equation:

$$\mathrm{MMD} = \left\| \frac{1}{m} \sum_{i=1}^{m} \varnothing\left(x_s^i\right) - \frac{1}{n} \sum_{j=1}^{m} \varnothing\left(x_t^j\right) \right\|_H^2, \qquad (1)$$

where $\| \cdot \|_H$ is the RKHS, $\varnothing(\bullet)$ is a mapping, and m and n are the number of samples in the source and target domains, respectively. The MMD approach has been widely studied given its ability to effectively solve the UDA problem, typically representing DDC [29], which uses MMD to align the features between the layers of two networks. Based on the DDC, the DAN proposed by [30] uses MK-MMD to achieve better performance. In the recent research work, the authors in [31–33] have used MMD directly to learn generic domain-invariant feature representations. Specifically, reference [34] employed MK-MMD at several higher layers with varying weights to achieve effective domain feature transfer of different faults, while the authors in reference [35] applied MMD to reduce distribution differences between training and test battery data, thereby enabling health assessment of lithium-ion batteries under different usage conditions, and the authors in reference [36] used MK-MMD to minimize differences in the marginal probability distributions of metastable features to eliminate each fault diagnosis task-specific distribution differences of high-level features in the discriminator. In addition, many scholars have worked on improving the performance of MMD methods. For example, a cross-domain active learning method based on Hellinger distance and MMD was proposed by the authors in [37]. Also, the discriminative heterogeneous MMD method (DMMD) proposed in [38] aims to minimize the variance of the domain probability distribution while retaining known discriminative information. In [39], an instance-weighted dynamic MMD (IDMMD) was proposed to dynamically estimate the effects of marginal and conditional distributions of bearing fault data and to adapt the target domain to the source domain.

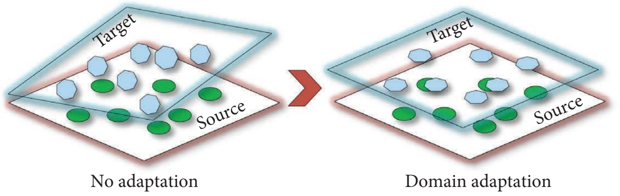


Figure 1: UDA schematic.

Adversarial domain adaptation (ADA) is an important branch of UDA, where the main idea is to approximate the distribution of the source and target domains by training a generative model. ADA methods usually employ an adversarial generative network (GAN) framework [40], where discriminators and generators can learn from each other through adversarial training, where the generators try to generate samples that match the target domain distribution in order to trick the discriminators; the DANN [41] is the most representative approach. In the early years, the authors in [42, 43] only focused on feature matching in some scenarios and did not focus on whether the matched features could improve performance. In recent years, many ADA improvements have been proposed by the researchers for specific tasks. For example, the authors in [44] improved its performance by designing a new framework and a new loss formulation; a novel domain adaptation scheme for adversarial entropy optimization (AEO) is introduced in [45]. The authors in reference [46] proposes a more suitable training and more generalized ADA method, using residual connectivity to share features and reconstruct adversarial losses.

MMD achieves impressive results, but this approach may not be effective for domain adaptation if there are large differences between the data in the source and target domains. Compared to MMD, ADA achieves better results in most cases, but it also has numerous limitations [47] as follows:

(1) ADA is a domain-adaptive method based on deep learning, which needs a lot of data to play a better effect. Therefore, in the limited data scenario, the domain-adaptive method based on MMD is more effective.

(2) In general, the optimization goal of a domain discriminator is to maximize the domain classification error in order to achieve domain adaptation. However, simply maximizing the domain classification error does not guarantee the desired domain adaptation effect, and there is a risk that the feature distributions of the source and target domains may be confused due to overoptimization. In other words, the domain classification accuracy reflects the distance of the feature distribution between the source and target domains, and the ideal domain classification accuracy is about 50%, as shown in Figure 2(a). However, it is possible that overoptimization leads to a domain classification accuracy of approximately 0%, as shown in Figure 2(b); the discriminator's recognition result is the exact opposite of the true one.

(3) The optimization processes of the generator and discriminator may interfere with each other, resulting in unstable training or difficulty in convergence. For example, Figure 2(c) shows the results of inadequate optimization.

For the effective adaptation of the different mission domains of the FW-UAV, we propose a novel hybrid domain adaptation method, the details of which are described in detail in Section 3.2.

2.2. Hampel Filter. The Hampel filter is a median absolute deviation-based filter that handles outliers and noise in time series. The filter identifies and replaces outliers by comparing the distance between each data point and its neighborhood data points one by one, resulting in smooth time-series data [48]. The Hampel filter is widely used in medical research for data processing and anomaly detection processes [49, 50]. In the field of intelligent manufacturing, the Hampel filter is mainly used in processing machine operation data [51], detecting manufacturing surface defects [52], and other related work; among them, the authors in [53] applied the Hampel filter to the study of intelligent fault diagnosis of wind turbines.

In contrast to general filters, the Hampel filter does not require the assumption that the data obey a Gaussian or some other specific probability distribution, and the median is an unbiased estimator that resists interference from extreme values and outliers, making the Hampel filter suitable for data filtering and preprocessing tasks in a wide range of anomalous scenarios.

3. HDBNH Framework

The initial stages of minor faults in the FW-UAV will not severely impact the FW-UAV, but failure to detect and address these minor faults promptly will lead to catastrophic accidents. Therefore, it is essential to test the functional components of FW-UAVs regularly or irregularly. The proposed HDBNH framework can serve for test flights, periodic inspections, and other work and can detect minor faults in the early stage of the FW-UAV, thus providing fault information to engineers and avoiding more significant losses. In this section, the working principle of HDBNH is described in detail. As shown in Figure 3, the HDBNH framework mainly consists of three parts: feature extractor, domain adaptor, and fault detector.

3.1. Feature Extractor. The flight data record detailed information about the FW-UAV flight process and imply much knowledge. The feature extractor is expected to extract the features of FW-UAV faults from the flight data and use them for subsequent fault detection. The feature extractor consists of two weight-sharing BiLSTM networks (F_1 and F_2) with 3 layers and 64 hidden units in each layer. The BiLSTM networks can handle both forward and backward time-series data and extract important features of the past and future. The process is as follows:

$$
\begin{aligned}
&\overrightarrow{i}_t = \sigma\left(\overrightarrow{W}_i\overrightarrow{x}_t + \overrightarrow{V}_i\overrightarrow{h}_{t-1} + \overrightarrow{b}_i\right)\\
&\overrightarrow{f}_t = \sigma\left(\overrightarrow{W}_f\overrightarrow{x}_t + \overrightarrow{V}_f\overrightarrow{h}_{t-1} + \overrightarrow{b}_f\right)\\
&\overrightarrow{o}_t = \sigma\left(\overrightarrow{W}_o\overrightarrow{x}_t + \overrightarrow{V}_o\overrightarrow{h}_{t-1} + \overrightarrow{b}_o\right)\\
&\overrightarrow{c}_t = \overrightarrow{f}_t \odot \overrightarrow{c}_{t-1} + \overrightarrow{i}_t \odot \tan h\left(\overrightarrow{W}_c\overrightarrow{x}_t + \overrightarrow{V}_c\overrightarrow{h}_{t-1} + \overrightarrow{b}_c\right)\\
&\overrightarrow{h}_t = \overrightarrow{o}_t \odot \tan h\ \left(\overrightarrow{c}_t\right)\\
&\overleftarrow{i}_t = \sigma\left(\overleftarrow{W}_i\overleftarrow{x}_t + \overleftarrow{V}_i\overleftarrow{h}_{t+1} + \overleftarrow{b}_i\right)\\
&\overleftarrow{f}_t = \sigma\left(\overleftarrow{W}_f\overleftarrow{x}_t + \overleftarrow{V}_f\overleftarrow{h}_{t+1} + \overleftarrow{b}_f\right)\\
&\overleftarrow{o}_t = \sigma\left(\overleftarrow{W}_o\overleftarrow{x}_t + \overleftarrow{V}_o\overleftarrow{h}_{t+1} + \overleftarrow{b}_o\right)\\
&\overleftarrow{c}_t = \overleftarrow{f}_t \odot \overleftarrow{c}_{t+1} + \overleftarrow{i}_t \odot \tan h\left(\overleftarrow{W}_c\overleftarrow{x}_t + \overleftarrow{V}_c\overleftarrow{h}_{t+1} + \overleftarrow{b}_c\right)\\
&\overleftarrow{h}_t = \overleftarrow{o}_t \odot \tan h\left(\overleftarrow{c}_t\right),
\end{aligned}
\tag{2}
$$

where W, V, and b are the parameters of the model; σ is the sigmoid function and t is the time point; and i_t, f_t, and o_t are the input, oblivion and output gates, respectively. $\odot$ is the element-wise product, and c_t is a memory cell. The output of BiLSTM is composed of $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ as follows:

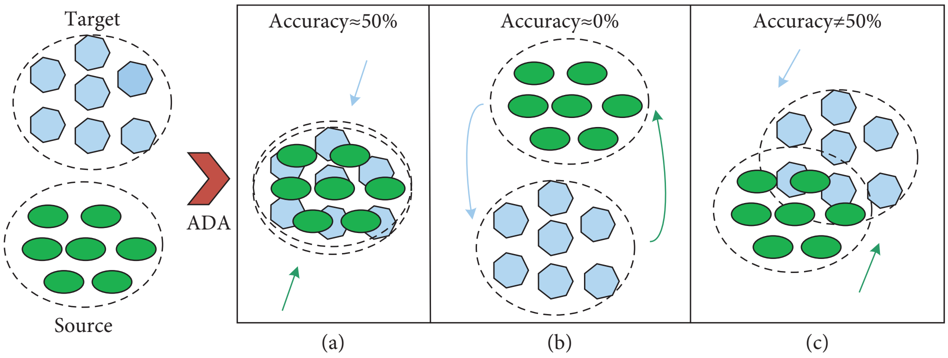


Figure 2: Possible problems with the ADA optimization process. The green and light blue arrows indicate that the source and target domain distributions migrate with the optimization process, respectively. (a) Ideal goal. (b) Overoptimization. (c) Under optimization.

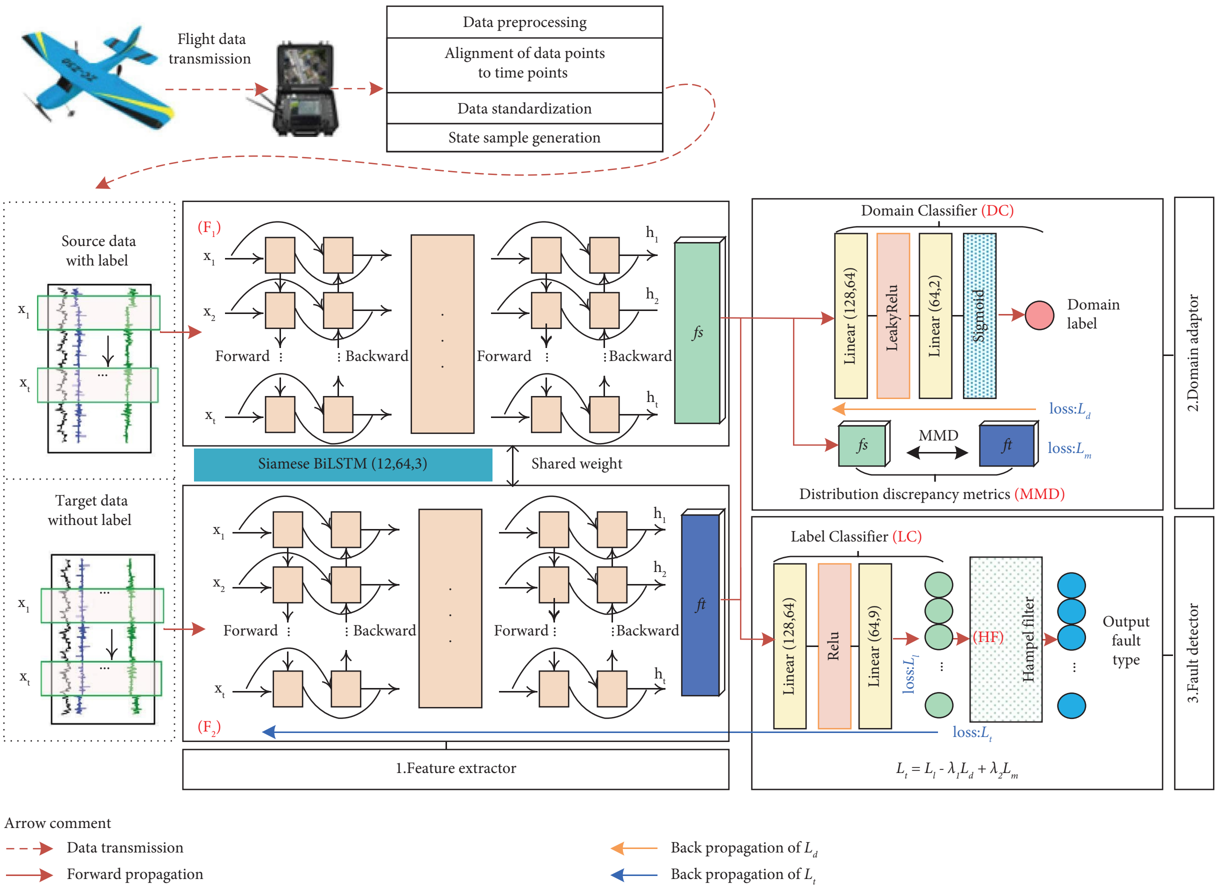


Figure 3: HDBNH framework.

$$h_t = \overrightarrow{h}_t \odot \overleftarrow{h}_t. \qquad (3)$$

3.2. Domain Adaptor. In the real world, FW-UAV mission environments are exceedingly complex and are influenced by multiple factors such as human operation, weather and the task load. Therefore, there exist significant nonlinear differences between data from different tasks, making effective domain adaptation exceptionally difficult. To mitigate these challenges and successfully perform fault detection tasks in the target domain, we propose a hybrid deep domain adaptation method that effectively reduces the feature distribution differences between the source and target domains. As shown in the upper right of Figure 3 (domain adaptor), this method includes an MMD module and a domain classifier (DC).

First, the MMD module can measure the difference of feature distribution between the source and target domains as in the following equation. The MMD module reduces the

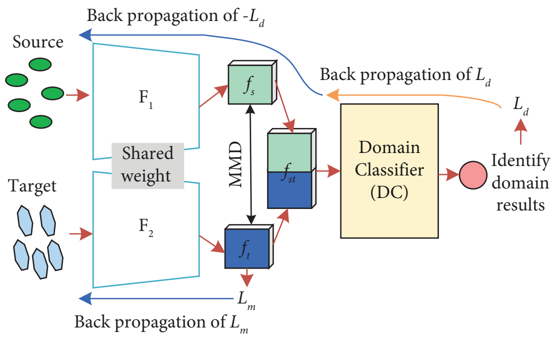


Figure 4: Working details of the domain adaptor.

difference in edge distribution between the source and target domain data by using L_m backpropagation and updating the F_1 and F_2 parameters, as shown in Figure 4.

$$L_m = \left\| \frac{1}{m} \sum_{i=1}^{m} f_s^i - \frac{1}{n} \sum_{j=1}^{n} f_t^j \right\|_H^2, \tag{4}$$

where $\| \cdot \|_H$ is the regenerated Hibelt space, and f_s^i and f_t^j are the ith source-domain feature and the ith target domain feature, respectively. m and n are the number of samples in the source and target domains, respectively.

Next, f_s and f_t are concatenated as shown in equation (5) and fed into the DC for domain classification. In this work, we set up a virtual domain label y_D to supervise the classification task and calculate the domain classification error by equation (6). The optimization process employs adversarial training to ensure that the feature extractor can extract similar features from both the source and target domains. If the DC cannot accurately recognize features from the source and target domains, domain adaptation is achieved. Therefore, our optimization goal is to maximize the domain classification error of the DC. To achieve this goal, the parameters of the F_1, F_2, and DC are optimized through different processes. The DC optimized through L_d backpropagation, while the F_1 and F_2 are optimized through $-L_d$ backpropagation, as shown in Figure 4.

$$f_{st} = \text{Concat}\left[f_s, f_t\right], \tag{5}$$

where Concat is the concatenation character.

$$L_d = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \cdot \log p\left(y_i\right) + \left(1 - y_i\right) \cdot \left(\log 1 - p\left(y_i\right)\right)\right], \tag{6}$$

where n is the batch size, y_i is the label 0 or 1 of the domain, and $p(y_i)$ is the predicted value of DC.

The hybrid deep domain adaptation method proposed in this study partially addresses the limitations of traditional MMD and ADA methods. MMD methods always provide domain adaptation based on distance measurement, while ADA can achieve consistency of data distribution through adversarial training. Combining both methods can compensate for their respective shortcomings, better achieve domain adaptation between source and target domains, and improve the performance of domain adaptation tasks, as shown in Figure 5 (these limitations are detailed in Section 2.1, and we will verify them through ablation experiments in Section 5.1.1).

3.3. Fault Detector.

The fault detector module is designed to accurately identify the state of the FW-UAV, which is our ultimate goal. Its structure is shown in Figure 3 bottom right and consists of a label classifier (LC) and a Hampel filter (HF).

The LC consists of a fully connected layer that allows the initial detection of faults in the FW-UAV. Its fault classification error is calculated via equation (7). The optimization objective of the LC is to minimize the fault classification error.

$$L_l = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} y_j * \log\left(p y_j\right), \tag{7}$$

where n is the batch size, k is the number of classes, y_j is the actual value, and $p(y_i)$ is the predicted value of LC.

We designed a HF to test and correct the predicted values of the deep learning model because the change process of the FW-UAV's state during flight is continuous and dynamic [27], which means that the transition process between different states takes a certain time. In other words, different states before and after the transformation should be maintained for a certain period and show different characteristics. If the minor faults detected by the deep learning model only occur in a moment and then return to its original state, it does not affect the original flight, which may be a misjudgment of the deep learning model, and we need to correct it to an original state through the HF. The main implementation process is shown in Figure 6. LC preliminarily detects the fault of FW-UAV and continuously outputs the results Y^l as shown in equation (8) and calculate the median m_i (equation (9)) and the median s_i (equation (10)) of the window length of Y^l within $2k + 1$. If $|y_i - m_i| \geq 3s_i$, return the new results Y^h with m_i instead of y_i as shown in equation (11). In this way, HF will verify and correct the detected results of each LC in turn.

$$Y^l = \left[y_1, y_2, \ldots, y_i \ldots y_{n-1}, y_n\right], \tag{8}$$

$$m_i = \text{median}\left(y_{i-k}, \ldots, y_i, \ldots, y_{i+k}\right), \tag{9}$$

$$s_i = \text{median}\left(\left|y_{i-k} - m_i\right|, \ldots, \left|y_i - m_i\right|, \ldots, \left|y_{i+k} - m_i\right|\right), \tag{10}$$

$$Y^h = \left[y_1, y_2, \ldots, m_i \ldots y_{n-1}, y_n\right], \tag{11}$$

where y_i is the fault detection result at time point i, median is the median calculator, and $| \cdot |$ is the absolute value.

It is worth noting that unlike methods such as the model-based KF, the HF can detect and correct only based on the output values of the deep learning model without the need to build any mathematical model, which is the biggest advantage of the HF.

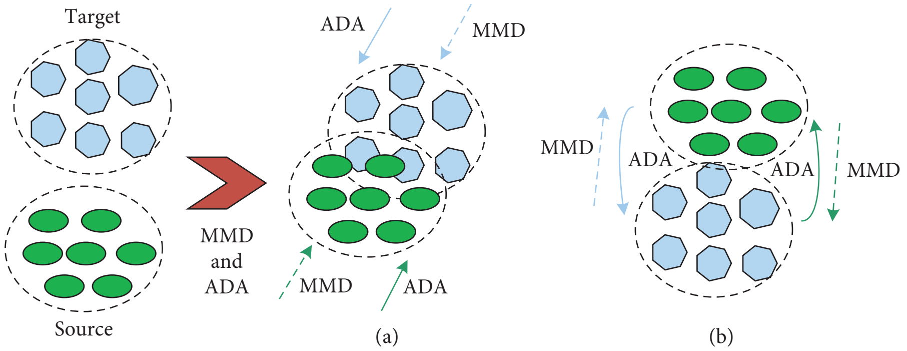


Figure 5: (a) MMD and ADA promote domain adaptation to make the optimization result as close to the ideal result as possible and solve the underoptimization problem. (b) MMD can mitigate the overoptimization of ADA. The green and light blue arrows indicate that the source and target domain distributions migrate with the optimization process, respectively. The dotted and solid lines represent MMD and ADA, respectively.

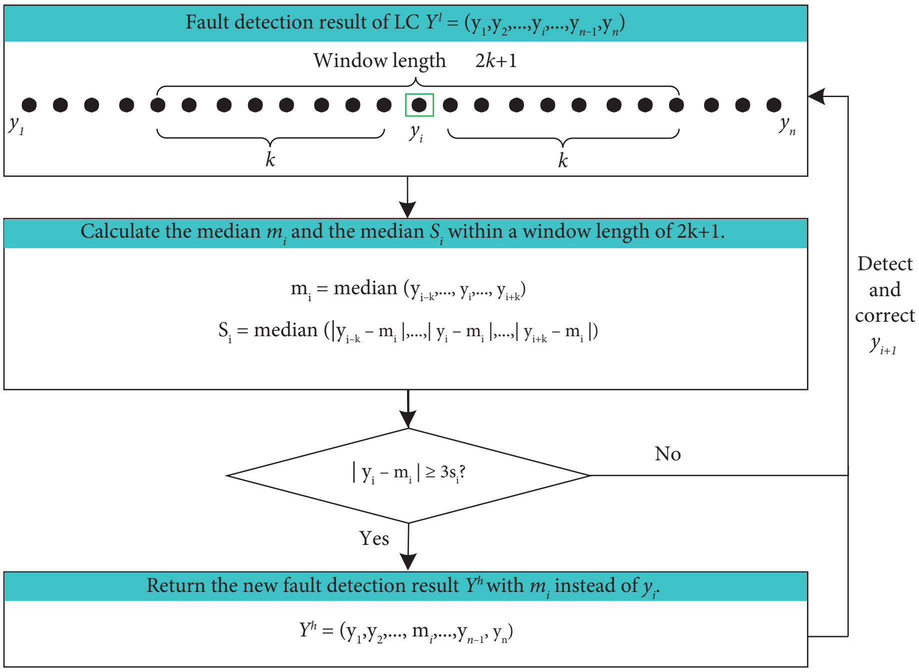

Fault detection result of LC $Y^l = (y_1, y_2, \ldots, y_i, \ldots, y_{n-1}, y_n)$
Window length 2k+1
y_1
y_i
y_n
k
k
Calculate the median m_i and the median S_i within a window length of 2k+1.
m_i = median $(y_{i-k}, \ldots, y_i, \ldots, y_{i+k})$
S_i = median $(|y_{i-k} - m_i|, \ldots, |y_i - m_i|, \ldots, |y_{i+k} - m_i|)$
$|y_i - m_i| \geq 3s_i$?
No
Yes
Return the new fault detection result Y^h with m_i instead of y_i.
$Y^h = (y_1, y_2, \ldots, m_i, \ldots, y_{n-1}, y_n)$
Detect
and
correct
y_{i+1}

Figure 6: HF in turn detects and corrects the results of LC.

3.4. HDBNH Framework Workflow. In summary, the entire HDBNH framework has 3 optimization objectives:

(1) Minimize the fault classification error in LC.

(2) Maximize the domain classification error in DC. If the domain classifier cannot accurately classify the features between the source and target domains, it means that the effect of domain adaptation is achieved.

(3) Minimize the MMD between the source and target domain features.

The total optimization objective can be written as follows:

$$L_t = L_l - \lambda_1 L_d + \lambda_2 L_m, \tag{12}$$

where λ_1 and λ_2 determine the intensity of the domain training and increase from 0 to 1 with training using formula $2/e^{-10*E/\max E+1} - 1$, and E is the training Epoch. The workflow of the HDBNH framework around the above-mentioned objectives is shown in Algorithm 1.

4. Data Processing

4.1. Real Datasets. Real flight data (https://github.com/mrtbrnz/fault_detection/tree/master/data) used in this work are provided by [18]. The flight experiment system and FW-UAV specification are shown in Figure 7. The ground control station (GCS) can set up autonomous flight missions and manually inject faults during flight. In case of severe faults, the FW-UAV can be controlled manually with an RC-transmitter. The X-Bee radio modem is used for telemetry

```
Datasets processing: $D_S\{x_s, y_s\} \longrightarrow D_{\text{train}}\{x_s, y_s\}, D_{val}\{x_s, y_s\}$ and $D_t\{x_t\} \longrightarrow D_{test}\{x_t\}$; set the virtual domain label $y_D = [0, 1]$
Training:
    Input: $D_{\text{train}}\{x_s, y_s\}, D_{\text{val}}\{x_s, y_s\}, D_{\text{test}}\{x_t\}, y_D$
    For $i$ in train epochs
        $f_s, f_t = F_1(x_s), \mathrm{F}_2(x_t)$
        Backpropagation $L_d(f_s f_t), y_D,$
        Optimizer Adam (DC. parameters)
        $y_{pre} = \mathrm{LC}\ (f_s)$
        Calculate $L_l(y_{\text{pre}} y_s), L_m(f_s f_t)$
        Backpropagation total $L_{\text{t}} = L_l - \lambda_1 L_d + \lambda_2 L_m$
        Optimizer Adam ($F_1$. parameters, $F_2$. parameters, LC. parameter)
        Save the best model on $D_{\text{val}}$
    End for
Testing:
    Input: $D_{\text{test}}$
    $f_t = F_2(x_t)$
    $y^l = LC(f_t)$
    $y^h = \mathrm{HF}\ (y^l)$
    Output: $y^h$
```

Algorithm 1: Training and testing of the HDBNH framework.

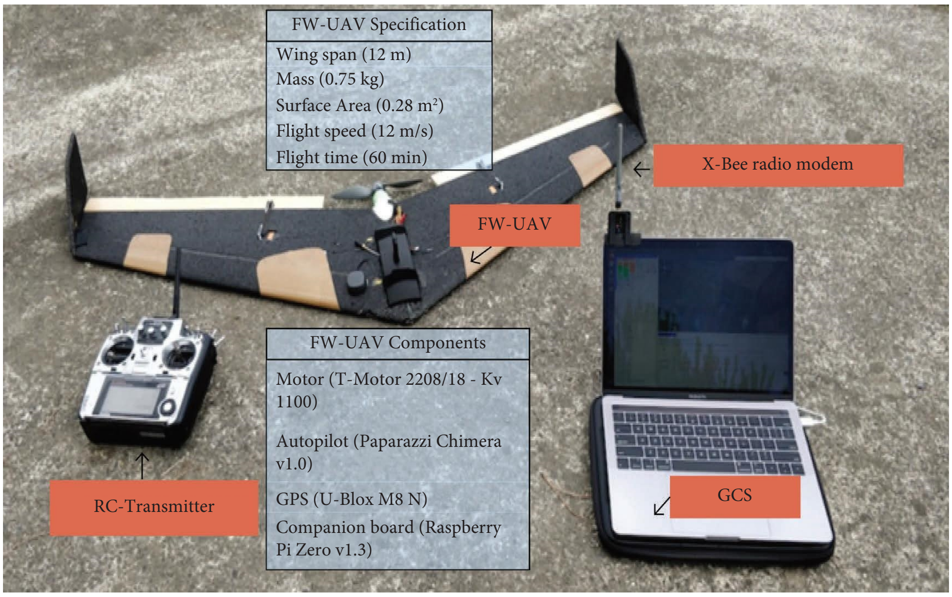


Figure 7: Flight experiment system and FW-UAV specification.

and datalink communication, and the GCS receives the flight data from the FW-UAV via the X-Bee.

The experimental system mainly simulates the situation where faults occur in the aerodynamic control surfaces, and the fault model is defined as follows [18]:

$$u_{\text{app}} = \mathrm{d}u_{\text{com}} + e, \tag{13}$$

where u_{app} is the control deflection of the final application, u_{com} is the desired control deflection of the ground control, d is the efficiency loss of control surfaces, the value of d can be set to simulate the degree of control surface failure, and e is the deflection error.

According to equation (13), the two aerodynamic control surfaces of the FW-UAV injected into the fault through the GCS can then be rewritten as follows:

$$\begin{bmatrix} u_{\text{app}_1} \\ u_{\text{app}_2} \end{bmatrix} = \begin{bmatrix} d_1, 0 \\ 0, d_2 \end{bmatrix} \begin{bmatrix} u_{\text{com}_1} \\ u_{\text{com}_2} \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \tag{14}$$

where 1 represents the right-wing control surface and 2 represents the left-wing control surface.

Finally, faults were injected into the FW-UAV being flown through the GCS, and flight data were recorded for different dates in July 2020. In this work, experiments were conducted mainly using the flight data of the 12th, 13th, 21st, and 23rd, where two states were simulated in the data of 12th and 13th, i.e., normal and $d_1 = 0.3$ (30% efficiency of the right-wing control surface), and nine states (normal, $d_1 = 0.3$, and $d_2 = 0.3$, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9) were simulated on 21st and 23rd.

4.2. State Sample Preparation Strategy. In order to obtain samples suitable for the HDBNH framework and fully utilize the performance of HDBNH, we performed a series of processing on the data.

First of all, flight data record more than 50 variables. To save computing resources and ensure computing speed, we need to select a small number of related variables to support our work. We refer to the choices in [13, 16, 18] and select v, ψ, θ, ϕ, a_x, a_y, a_z, ω_x, ω_y, ω_z, u_1, and u_2, 12 variables, as shown in Figure 8. v is the airspeed, ψ is the yaw angle, θ is the pitch angle, ϕ is the roll angle, a_{xyz} is the linear acceleration in three directions, ω_{xyz} is the angular rate in three directions, and u_1 and u_2 are the control commands of the autopilot. These 12 variables are assembled to form the state vector X:

$$X = \left[v\psi\theta\phi a_x a_y a_z \omega_x \omega_y \omega_z u_1 u_2\right]. \tag{15}$$

Second, the data of each variable are collected by different sensors, which means that the sampling frequency is also inconsistent, where the sampling frequency of a_{xyz} and ω_{xyz} is 50 Hz and the sampling rate of the rest is 20 Hz, resulting in the data points and time points not aligned. To solve this problem, we use a linear interpolation method to make the data frequency of all variables become 20 Hz and then align the time points by moving the time axis, as shown in Figure 9. In addition, each variable has different units and value sizes, and all variables were standardized to eliminate the possibility of dominance by one variable.

Finally, based on the t time point, the data of the previous 20 time points (1 s) are chosen as a state sample X_t (equation (16)) at moment t. Similarly, slide one time points (0.05 s) to obtain the state sample X_{t+1}, as shown in Figure 10. Visualizing X_t as shown in Figure 11 provides a more intuitive understanding of the sample states at moment t.

$$X_t = \begin{bmatrix} v_t \cdots a_{\mathrm{zt}} \cdots \omega_{\mathrm{zt}} \cdots u_{\mathrm{2t}} \\ v_{t-1} \cdots a_{\mathrm{zt}-1} \cdots \omega_{\mathrm{zt}-1} \cdots u_{\mathrm{2t}-1} \\ \cdots\cdots \\ v_{t-19} \cdots a_{\mathrm{zt}-19} \cdots \omega_{\mathrm{zt}-19} \cdots u_{\mathrm{2t}-19} \end{bmatrix}. \tag{16}$$

Through the abovementioned series of data processing, we get four state sample sets as shown in Table 1. In fact, such a state sample preparation strategy gives a better performance of the HF, which is further described in Section 5.1.2.

5. Experiments and Results

In this section, we will experiment with different perspectives to verify the performance of the HDBNH framework. The GCS computer configuration of the experiment is as follows: an Ubuntu 18.04 operating system, an Intel (R) Xeon (R) Silver 4210R CPU @ 2.40 GHz, a GeForce RTX 2080 Ti GPU with CUDA 11.6, and Torch 1.4.0. The training epoch was 25, the batch size was 64, and the network parameters were updated using an Adam optimizer with an initial learning rate of 0.01.

5.1. Ablation Study. In this section, we mainly analyze the performance of domain adaptors and the HF through ablation experiments.

5.1.1. Ablation Study of the Domain Adaptor. To verify the effect of hybrid deep domain adaptation of the HDBNH framework, the relevant modules of the domain adaptor were decomposed and ablation experiments were performed according to the control variable principle (as shown in Table 2). Each experiment was repeated five times and averaged. The experimental results are shown in the Table 3, where the mutual transfer between A and B is binary classification tasks, and the mutual transfer between C and D is 9 classification tasks.

From Table 3, it can be seen that HDBNH has obvious advantages, which shows that our proposed hybrid deep domain adaptation method can learn domain-invariant features well to improve fault detection accuracy. FDLH and FMLH are generally better than FLH, but the improvement is limited and even negative optimization occurs on individual tasks, such as FMLH is lower than FLH in $A \longrightarrow B$ and FDLH is lower than FLH in $C \longrightarrow D$. The main reason for this result is the large differences in the distribution of features in different domains of the FW-UAV and the inherent limitations of the MMD domain adaptation and ADA optimization process (as described in Section 2.1), which do not allow for effective domain adaptation. In contrast, HDBNH's hybrid depth domain adaptation approach allows for effective domain adaptation between the source and target domains in the more complex FW-UAV domain adaptation task.

As an example of experiment $C \longrightarrow D$, the source-domain features and target domain features extracted by F_1 and F_2 are visualized by T-SNE to better observe the effect of domain adaptation. As shown in Figure 12, HDBNH shows the best domain adaptation, where the feature distribution spaces of the same faults are matched in the source and target domains; at the same time, the differences in the feature distributions of different faults are more obvious. This indicates that HDBNH is better able to identify multiple types of faults. The matching of the feature distributions of the source and target domains of FLH, FDLH, and FMLH are relatively less effective. We also found more severe class-level alignment confusion for FLH, FDLH, and FMLH, such as matching the feature distributions of fault 4 in the source domain and fault 5 in the target domain, which is an ineffective or negatively acting domain adaptation phenomenon. This validates the associated domain adaptation problem as described in Section 2.1.

5.1.2. Performance Analysis of the HF. The flight data are highly time series and reflect the FW-UAV's ever-changing state. If a fault detected by LC occurs for

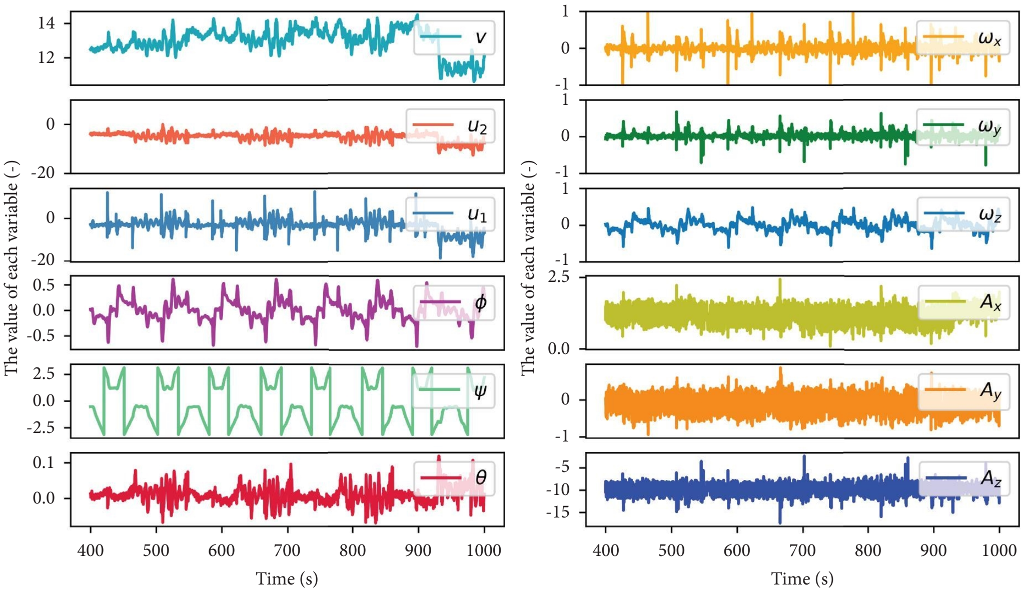


Figure 8: 12 variables at 400–1000 seconds.

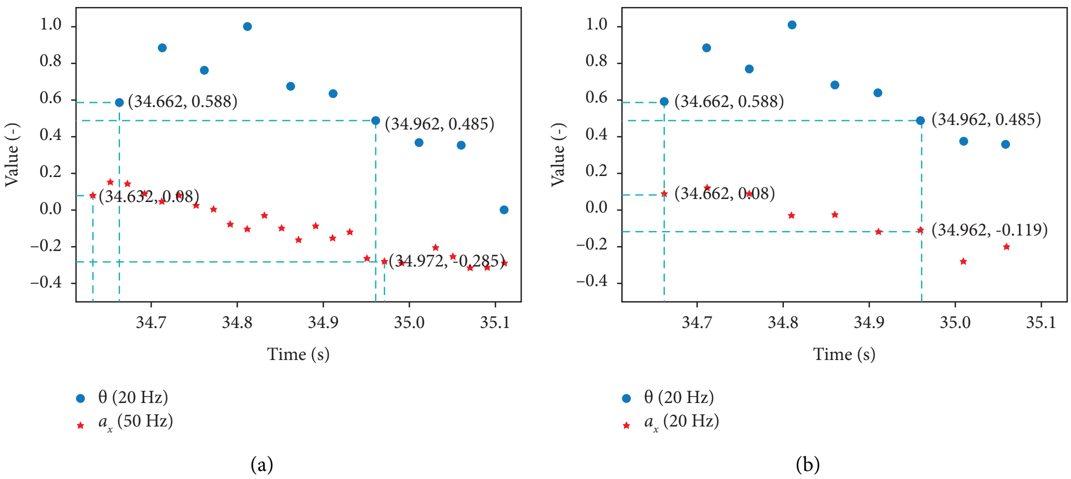


Figure 9: Data point and time point alignment processing: (a) original and (b) after alignment.

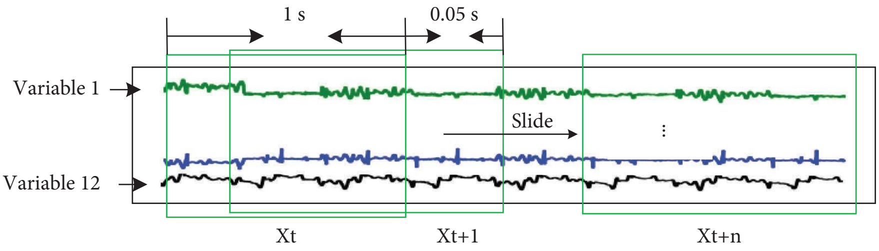


Figure 10: State sample preparation strategy.

a short period and then subsequently returns to its original state, this is likely to be an error judgment of LC. Therefore, the HF module is added for judging and correcting the results of LC. This is one of the main innovations in this work. In this study, the window length $(2k + 1)$ for H is set to 21, corresponding to a time length of 1 second. But in fact, the window length can be set according to different situations. Experiments without and with the HF participation (denoted by FDML and HDBNH, respectively) were conducted separately to test the validity of the HF. Each experiment was repeated five times, and the results are shown in Figure 13.

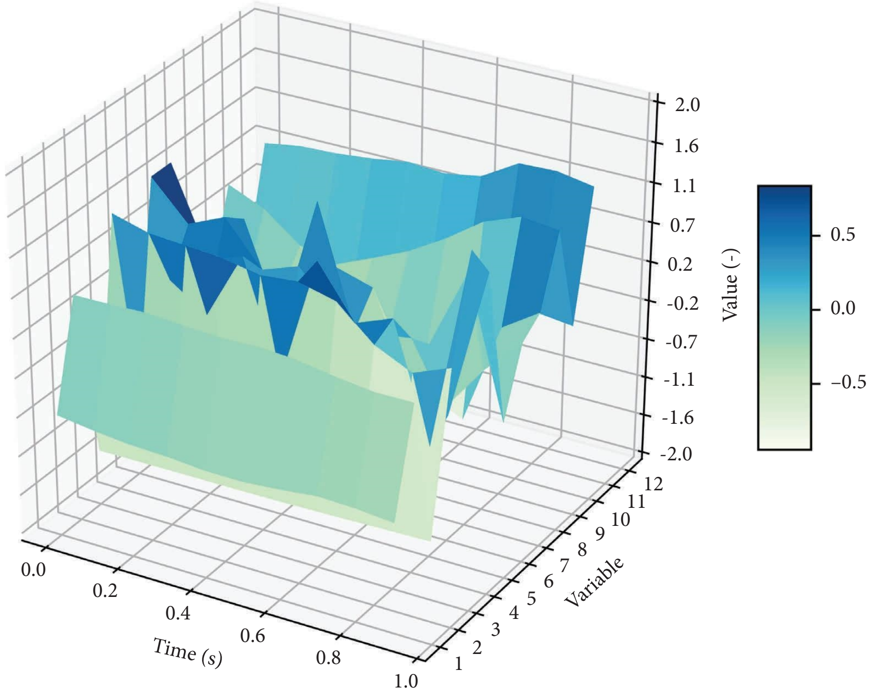


Figure 11: Visualizing X_t.

Table 1: Details of the state sample sets.

Sample sets	Dates	Wind speeds	Fault labels	Number of samples
A	12	<2.0 m/s	Normal ⟶ 0, $d_1 = 0.3$ ⟶ 1	17992
B	13	8.0 m/s	Normal ⟶ 0, $d_1 = 0.3$ ⟶ 1	17987
C	21	2.5 m/s	Normal ⟶ 0, $d_1 = 0.3$ ⟶ 1, $d_2 = 0.9 \sim 0.3$ ⟶ 2 ~ 8	44011
D	23	5.0 m/s	Normal ⟶ 0, $d_1 = 0.3$ ⟶ 1, $d_2 = 0.9 \sim 0.3$ ⟶ 2 ~ 8	44007

Table 2: Detailed rules of ablation experiments for verifying the domain adaptor.

Participation modules (see Figure 3)	Remarks	Names
F_1, F_2, LC, HF	No domain adaptation method	FLH (baseline)
F_1, F_2, DC, LC, HF	ADA method, similar to DANN [41]	FDLH
F_1, F_2, MMD, LC, HF	MMD domain adaptation, similar to DDC [29]	FMLH
F_1, F_2, DC, MMD, LC, HF	Hybrid MMD and ADA	HDBNH

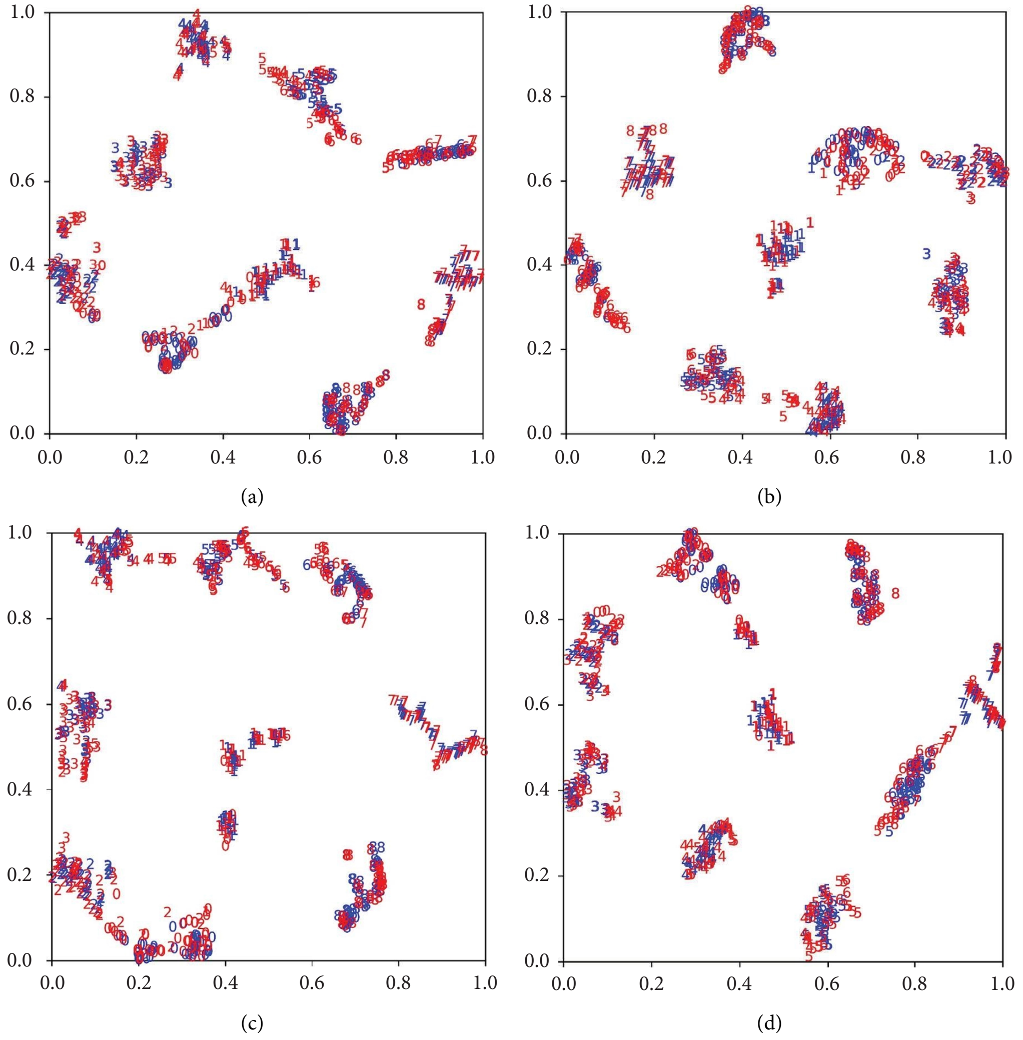
As can be seen from Figure 13, the HF substantially improves the accuracy of fault detection, especially in the transfer task of $C \longrightarrow D$, which is 10.51% higher than FDML without the HF. Figure 14 shows the real fault labels, output of LC, and output of the HF over time and the embedded figure shows the partial details. As shown in Figure 14, the HF effectively corrects the false predictions of LC, such as the misprediction around 231.4 s. However, there are also many mispredictions that are not corrected, such as between 230.5 s and 231 s. This is due to the multiple false predictions of LC in a short period of time, which prevent the HF from working. The prerequisite for the HF to function better is that LC has a good performance; so, we choose a BiLSTM network to extract past and future features from the time-series flight data, while adding DC and MMD for domain adaptation to improve the performance of LC.

k is the most crucial parameter of the HF. Therefore, we did the experiment of gradually increasing k from 0 to 30 to observe the effect of k on the results, and the experimental results are shown in Figure 15. It can be seen from Figure 15 that when k increases from 0 to 10, the accuracy of all fault detection tasks also increases significantly, but as k continues to increase, the improvement of accuracy is not obvious and tends to be stable. Because the HF detects outliers by referring to values in the window, the number of values in the window is too small to provide an accurate reference. In the state sample preparation strategy (Section 4.2), the sampling frequency of all variables is first changed to 20 Hz by interpolation, and a state sample is taken by sliding each time point so that 21 state samples of the FW-UAV are obtained in 1 second (except for the very first second); in

Table 3: Experimental results.

Tasks	FLH (baseline)	FDLH	FMLH	HDBNH
$A \longrightarrow B$	85.26 ± 2.71	**86.27 ± 1.48**	85.07 ± 1.62	86.03 ± 2.88
$B \longrightarrow A$	94.05 ± 0.28	96.94 ± 0.48	97.43 ± 0.10	**97.44 ± 0.14**
$C \longrightarrow D$	86.90 ± 0.72	86.75 ± 0.97	89.40 ± 1.06	**89.54 ± 0.87**
$D \longrightarrow C$	87.53 ± 1.52	92.17 ± 1.21	91.48 ± 1.54	**93.26 ± 1.31**
Average	88.44 ± 1.31	90.53 ± 1.04	90.85 ± 1.08	**91.57 ± 1.30**

Note: Bolded are optimal.



(a)

(b)

(c)

(d)

Figure 12: T-SNE visualization of the distribution of source-domain features and target domain features. Red indicates source-domain features, blue indicates target domain features, and the numbers 0–8 are labels for each class of faults as in Table 1. (a) FLH. (b) FDLH. (c) FMLH. (d) HDBNH.

other words, LC detects the state of the FW-UAV 21 times in 1 second. Such high-frequency detection allows the HF to work better.

5.2. Compared to Other Methods. In this section, we use FDML and HDBNH to compare with some current mainstream methods, including SVM, CNN, SHNN, DDC, MMDA, DANN, CNN_FT, and BiLSTM_FT; each method is introduced as shown in Table 4. Four experiments with transfer tasks were set up for each method, and again, each experiment was repeated five times, and the experimental results are shown in Figure 16. It is worth noting that we try to set the details of each experiment to the optimal case to ensure fairness of comparison, including data format, hyperparameters, and training process based on the characteristics of each method.

From Figure 16, we can see that HDBNH has the best performance in all transfer tasks with an average accuracy of 91.57%, which is much higher than other methods and the

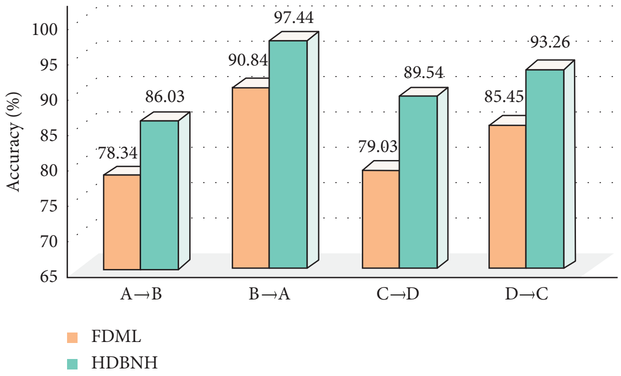


Figure 13: Comparison of experimental results with and without the HF participation.

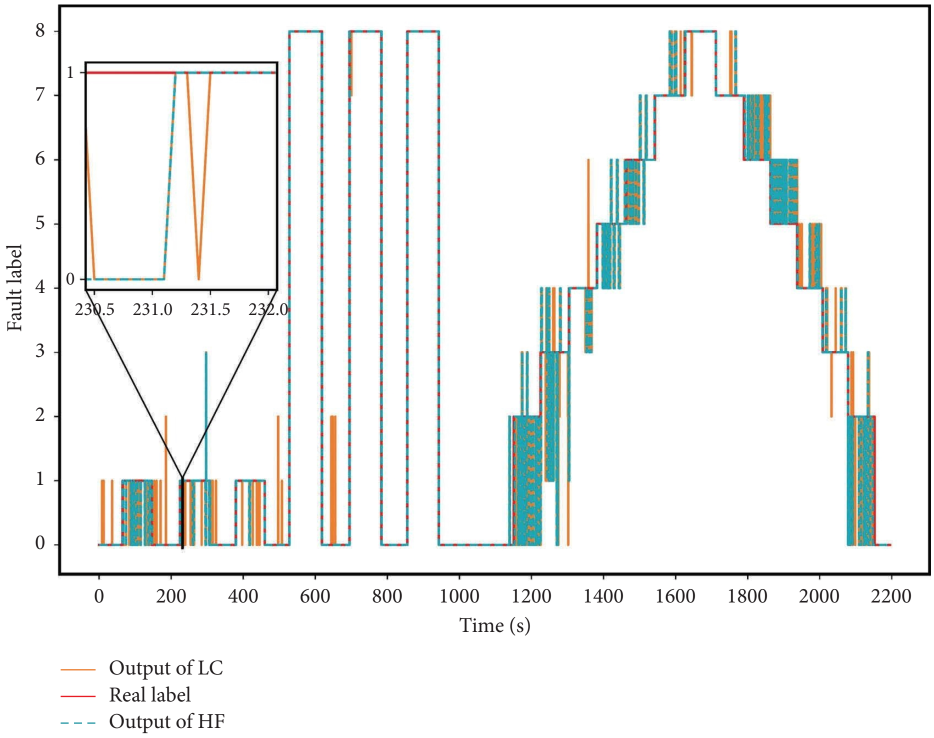


Figure 14: Real fault labels, output of LC, and output of the HF over time.

highest accuracy of 97.44% in $B \longrightarrow A$. FDML outperforms other methods in $B \longrightarrow A$, $C \longrightarrow D$, and $D \longrightarrow C$ except HDBNH, and the average value is second only to HDBNH. This indicates that our proposed framework has excellent performance in fault detection of FW-UAV and can perform most transfer tasks well. BiLSTM_FT and CNN_FT also achieved good fault detection accuracy, with average accuracies of 83.11% and 80.58%, respectively. However, both methods are fine-tuned using a small number of labeled target samples and are supervised domain adaptation methods, while the other domain adaptation methods are unsupervised, which is not strictly fair. To our surprise, DDC and DANN have the worst performance, even lower than SVM, CNN, and SHNN without using any domain adaptation method. DDC and DANN do not work as intended in our experiments and make the performance worse instead, due to the inability to effectively align the fault features in our flight data. In fact, MMD or adversarial-based domain adaptation methods focus on domain-level adaptation and ignore class-level adaptation, leading to class confusion adaptation and making domain-invariant features poorly learned, which is consistent with the results of 4.1.1. In addition, as most of the comparison methods use a different dataset in the original papers than our dataset, the results will be different. However, the SVM and SHNN original papers use the same dataset as the one in this paper, and the results are both different and similar. The SVM agrees with our results in the original paper in the $B \longrightarrow A$ task, but in the other tasks, our results are better than the original paper. The reasons for the inconsistent results include differences in data processing, variable selection, and the details of method reproduction. The results of SHNN are generally consistent

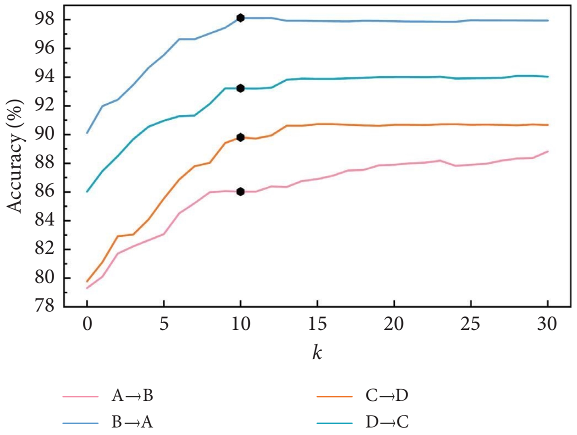


Figure 15: Curves of experimental results with increasing k.

Table 4: Information of various methods.

Methods	Domain adaptation methods	Details
SVM [18]	None	Traditional machine learning method: support vector machine
CNN	None	Ordinary convolution neural network
SHNN [20]	None	Few-shot learning method based on hybrid CNN and LSTM Siamese network
DDC [29]	MMD	An adaptive MMD criterion metric is added to the previous layer of the classifier
MMDA [54]	MMD	Multilayer MMD domain adaptation
DANN [41]	Adversarial	Domain adaptation based on adversarial
CNN_FT [55]	Fine tuning	Fine-tune the full connection layer with the labeled target sample
BiLSTM_FT [56]	Fine tuning	Fine-tuning method for residual life prediction
FDML (ours)	Hybrid adversarial and MMD	HDBNH framework removes the HF
HDBNH (ours)	Hybrid adversarial and MMD	Based on hybrid domain-adaptive BiLSTM networks and the HF

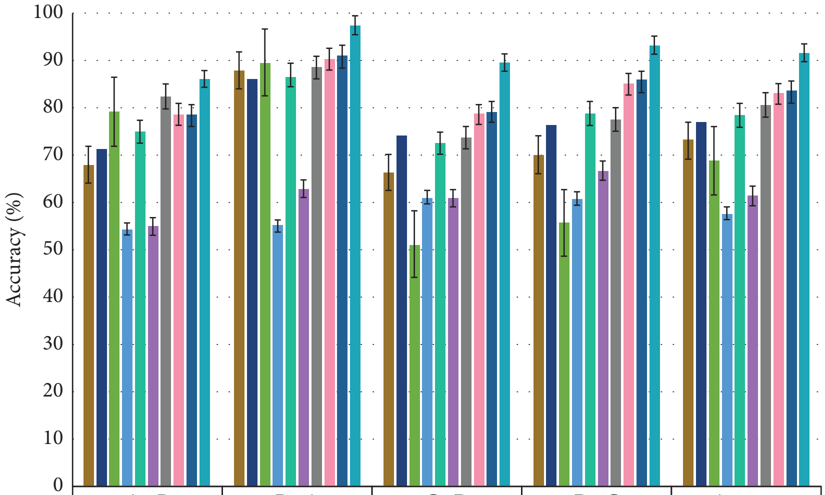
with the original paper; SHNN is a few-shot learning method based on hybrid CNN and LSTM Siamese network, which will be more advantageous in the case of limited samples.

5.3. The Influence of Wind Speed. The FW-UAVs are affected by many factors (such as wind, payload, and icing) during the execution of the mission. It is impossible to obtain data from all mission environments for training fault detection models. HDBNH is a transfer learning idea that is expected to learn from certain mission environment data and be used to detect the states of FW-UAVs in other mission environments. However, in transfer learning, the knowledge learned by the model in the source domain is crucial. In this section, we will explore the effect of model learning knowledge under different wind speeds. Figure 17 shows the experimental results of each model trained in the source domain with different wind speeds and detecting faults in the target domain. The datasets A, B, C, and D in the experiments were obtained from the missions on the 12th, 13th, 21st, and 23rd days with wind speeds <2 m/s, 8 m/s, 2.5 m/s, and 5 m/s, respectively.

In Figure 17, the results for each model B (8 m/s) $\longrightarrow A$ (<2 m/s) are better than A (<2 m/s) $\longrightarrow B$ (8 m/s), and the results for each model D (5 m/s) $\longrightarrow C$ (2.5 m/s) are better than C (2.5 m/s) $\longrightarrow D$ (5 m/s). To analyze the reason, the flight traces of different dates were printed as shown in Figure 18. In fact, the ground control station sets an "8" shaped autonomous flight path, but it can be seen from the figure that the greater the wind speed, the greater the deviation of the track. It may be because as wind speed increases and working conditions become more complex, state features become more obvious; as a result, the model is easier to extract and apply fault features in more complex working conditions. At the same time, we also found that the difference between the results of $B \longrightarrow A$ and $A \longrightarrow B$ is greater than that of $C \longrightarrow D$ and $D \longrightarrow C$ because the wind speed difference between A and B is greater.

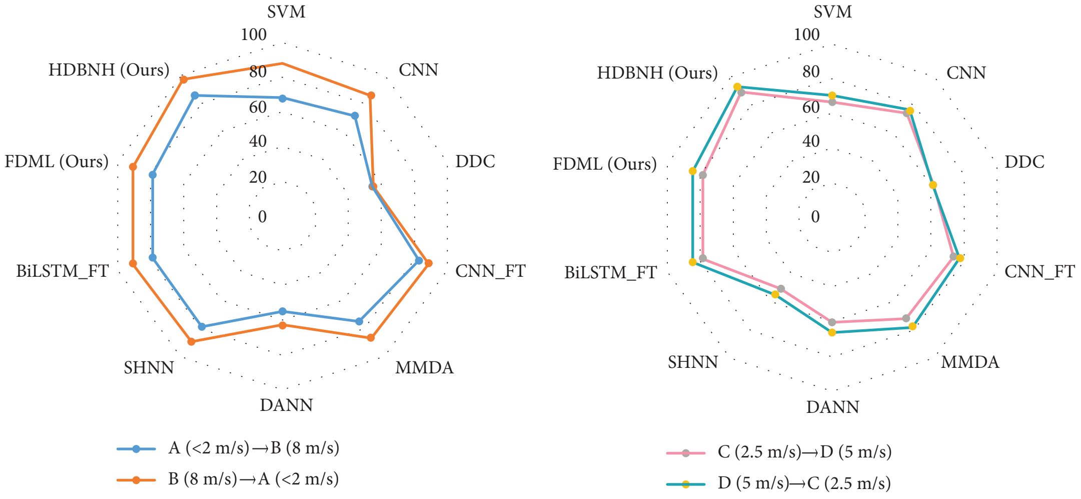
Although the abovementioned results show that the wind speed is greater, the model learning effect is better, but we think there will be an optimal wind speed, which needs to get more data and verify in the future.

5.4. Other Analysis. Experiments show that HDBNH has excellent robustness and generalization, can learn knowledge from one working condition well, and effectively detect the state of FW-UAV in other unknown working conditions. However, HDBNH has some limitations in real-time online detection, which is more suitable for offline detection, such as detecting FW-UAVs that have completed tasks, offline analysis of test flights, and regular or irregular offline detection. Because HDBNH has the process of domain adaptation training, it needs the participation of target domain samples; this means that if real-time fault detection is performed, it is necessary to obtain flight data from the FW-UAV in real-time for onsite training. However, the energy reserve of the FW-UAV is limited. Figure 19 shows



	A→B	B→A	C→D	D→C	Average
■ SVM	67.92	87.98	66.52	70.01	73.11
■ CNN	71.29	86.22	74.18	76.46	77.04
■ SHNN	79.18	89.59	51.21	55.68	68.92
■ DDC	54.45	55.16	61.19	60.78	57.90
■ MMDA	74.96	86.82	72.68	78.86	78.33
■ DANN	55.04	62.91	60.94	66.84	61.43
■ CNN_FT	82.51	88.51	73.73	77.58	80.58
■ BiLSTM_FT	78.55	90.34	78.61	84.93	83.11
■ FDML (Ours)	78.34	90.84	79.03	85.45	83.42
■ HDBNH (Ours)	86.03	97.44	89.54	93.26	91.57

Figure 16: Experimental results of different methods in different tasks.



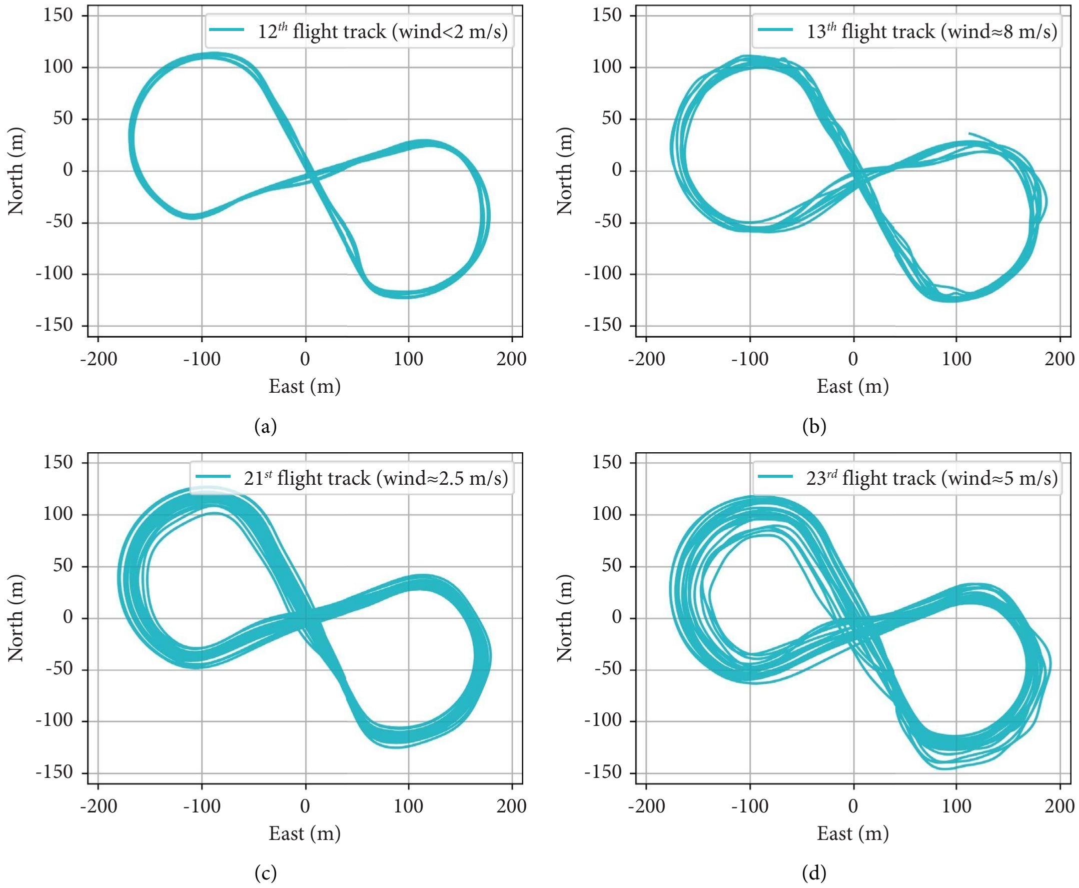
(a)

(b)

Figure 17: Results of model learning at different wind speeds: (a) binary classification task and (b) 9 classification task.

the training elapsed time of HDBNH, which takes about 400 s to achieve optimal training. Such a situation may be improved when the GCS computer configuration is upgraded. In addition, the detection process of the HF module will also generate additional delays. In this work, the additional delay caused by the HF is 0.5 s, which is half the length of the time window.

Meanwhile, it can be seen from Figure 19(a) that the change trend of each loss is consistent with the optimization objectives of HDBNH (see Section 3.4), where L_l, L_m, and L_t



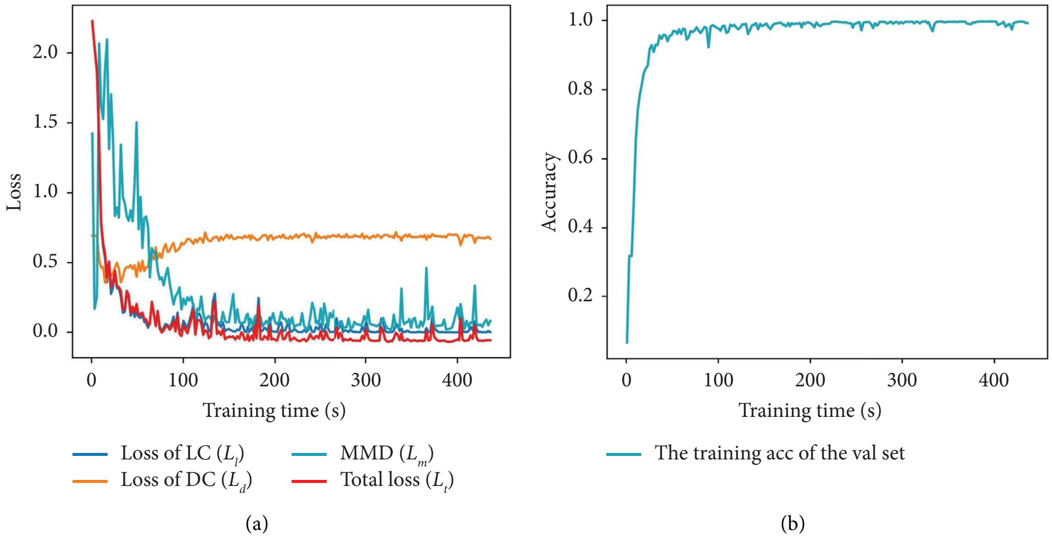
(a)

(b)

(c)

(d)

Figure 18: Flight tracks for different dates.



(a)

(b)

Figure 19: Relationship between HDBNH training time and the training effect (task $C \longrightarrow D$, the batch size is 64). (a) Loss and (b) training accuracy of the validation set.

gradually converge near 0, and L_d increases gently with training. It further proves the effectiveness of the designed HDBNH framework.

6. Conclusion

Deep learning provides advanced solution ideas for future UAV fault detection, but the current lack of UAV monitoring data limits the advantages of deep learning for UAV fault detection, which are a challenge and an opportunity. In this paper, we mainly consider the data availability of FW-UAVs under a variety of actual flight conditions and propose an FW-UAV fault detection framework based on hybrid deep domain adaptation BiLSTM networks and the HF (HDBNH), the main purpose of which is to learn the knowledge of acquired data for detecting FW-UAV faults in other unknown operating conditions. Also, a state sample preparation strategy is proposed for the HDBNH framework, which solves the problems of data complexity, redundancy, nonstandard, and frequency inconsistency, while the generated state samples better support the work of HDBNH.

Extensive experiments have been done in real flight data to verify the effectiveness of the hybrid deep domain adaptation method, the effectiveness of the HF module, and high fit between the state sample preparation strategy and the HDBNH framework. Compared with some current mainstream methods, HDBNH has better performance. It can learn detection knowledge from acquired flight data and effectively use the learned knowledge to detect faults in other unknown conditions. The effect of wind speed is also explored in this work, and it is believed that the higher the wind speed, the more complex the working conditions are and the more pronounced the state features will be. Therefore, it is also easier for the model to extract the state features from the flight data at larger wind speeds. Finally, the limitations of HDBNH are discussed, and it is pointed out that HDBNH is more suitable for offline detection.

HDBNH provides a new solution for FW-UAV fault detection. However, there is still a lot of work to be done, such as improving the capability of real-time detection and collecting more flight data for more experiments.

Data Availability

The data that support the findings of this study are available from the first author (Y. Z.) or the corresponding author (S. L.) upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Y. Z. performed conceptualization, methodology, visualization, data curation, original draft writing, reviewing and editing, and formal analysis. S. L. performed conceptualization, methodology, and funding acquisition. Q. H. performed conceptualization, methodology, data curation, reviewing and editing, and supervised the study. A. Z. performed reviewing and editing and supervised the study. C. L. and Z. L. performed reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

The authors would like to thank for the computing support of the State Key Laboratory of Public Big Data, Guizhou University. This work was supported by the National Natural Science Foundation of China (no. 52275480), the Guizhou Provincial Science and Technology Projects (no. ZK[2023] 059), the Talent Training Base Project of Colleges and Universities in Guizhou Province (no. [2020]005), the Guizhou Provincial Department of Education Youth Science and Technology Talent Growth Project (no. QJH KY [2022] 142), and the Scientific Research Project of Introducing Talents of Guizhou University (no. GDRJH (2021)74).

References

[1] Z. Yu, Y. Zhang, B. Jiang, J. Fu, and Y. Jin, "A review on fault-tolerant cooperative control of multiple unmanned aerial vehicles," *Chinese Journal of Aeronautics*, vol. 35, pp. 1–18, 2022.

[2] P. Lu, E.-J. van Kampen, C. de Visser, and Q. Chu, "Nonlinear aircraft sensor fault reconstruction in the presence of disturbances validated by real flight data," *Control Engineering Practice*, vol. 49, pp. 112–128, 2016.

[3] C. Hajiyev and H. E. Soken, "Robust Adaptive Kalman Filter for estimation of UAV dynamics in the presence of sensor/actuator faults," *Aerospace Science and Technology*, vol. 28, no. 1, pp. 376–383, 2013.

[4] A. Abbaspour, P. Aboutalebi, K. K. Yen, and A. Sargolzaei, "Neural adaptive observer-based sensor and actuator fault detection in nonlinear systems: application in UAV," *ISA Transactions*, vol. 67, pp. 317–329, 2017.

[5] G. P. Kladis, J. T. Economou, K. Knowles, A. Tsourdos, and B. A. White, "Digraph matrix reliability analysis for fault assessment for A UAV platform application. A fault-tree analysis approach," in *Proceedings of the 2008 IEEE Vehicle Power and Propulsion Conference*, pp. 1–6, Hei Longjiang, China, September 2008.

[6] Q. L. Zhang, Y. Liu, and P. L. Sun, "fault diagnosis expert system for electric power system of the large-scale UAVs based on virtual instrument," *Advanced Materials Research*, vol. 663, pp. 572–579, 2013.

[7] L. Xiao, D. Sun, Y. Liu, and Y. Huang, "A combined method based on expert system and BP neural network for UAV systems fault diagnosis," in *Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence*, pp. 3–6, Las Vegas, USA, December 2010.

[8] Z. Yu, Y. Zhang, B. Jiang, J. Fu, Y. Jin, and T. Chai, "Composite adaptive disturbance observer-based decentralized fractional-order fault-tolerant control of networked UAVs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 799–813, 2022.

[9] Q. Wang, J. Mao, and H. Wei, "Reliability analysis of multi-rotor UAV based on fault tree and Monte Carlo simulation," in *Advances in Mechanical Design*, J. Tan, F. Gao, and C. Xiang, Eds., Springer, Singapore, pp. 1525–1534, 2018.

[10] J. Bu, R. Sun, H. Bai et al., "Integrated method for the UAV navigation sensor anomaly detection," *IET Radar, Sonar & Navigation*, vol. 11, no. 5, pp. 847–853, 2017.

[11] C. Li, S. Li, H. Wang, F. Gu, and A. D. Ball, "Attention-based deep meta-transfer learning for few-shot fine-grained fault diagnosis," *Knowledge-Based Systems*, vol. 264, Article ID 110345, 2023.

[12] P. Yang, C. Wen, H. Geng, and P. Liu, "Intelligent Fault diagnosis method for blade damage of quad-rotor UAV based on stacked pruning sparse denoising autoencoder and convolutional neural network," *Machines*, vol. 9, no. 12, p. 360, 2021.

[13] P. Yang, H. Geng, C. Wen, and P. Liu, "An intelligent quadrotor fault diagnosis method based on novel deep residual shrinkage network," *Drones*, vol. 5, no. 4, p. 133, 2021.

[14] G. Iannace, G. Ciaburro, and A. Trematerra, "fault diagnosis for UAV blades using artificial neural network," *Robotics*, vol. 8, no. 3, p. 59, 2019.

[15] E. Baskaya, M. Bronz, and D. Delahaye, "Fault detection & diagnosis for small UAVs via machine learning," in *Proceedings of the 2017 IEEE/AIAA 36th Digital Avionics Systems Conference*, pp. 1–6, DASC), St. Petersburg, Fl, USA, September 2017.

[16] D. Guo, M. Zhong, H. Ji, Y. Liu, and R. Yang, "A hybrid feature model and deep learning based fault diagnosis for unmanned aerial vehicle sensors," *Neurocomputing*, vol. 319, pp. 155–163, 2018.

[17] Y. Chen, C. Zhang, Q. Zhang, and X. Hu, "UAV fault detection based on GA-BP neural network," in *Proceedings of the 2017 32nd Youth Academic Annual Conference of Chinese Association of Automation*, pp. 806–811, Hefei, China, May 2017.

[18] M. Bronz, E. Baskaya, D. Delahaye, and S. Puechmore, "realtime fault detection on small fixed-wing UAVs using machine learning," in *Proceedings of the 2020 AIAA/IEEE 39th Digital Avionics Systems Conference*, pp. 1–10, DASC), San Antonio, Texas, USA, October 2020.

[19] A. Keipour, M. Mousaei, and S. Scherer, "ALFA: a dataset for UAV fault and anomaly detection," *The International Journal of Robotics Research*, vol. 40, no. 2-3, pp. 515–520, 2021.

[20] C. Li, S. Li, A. Zhang et al., "A Siamese hybrid neural network framework for few-shot fault diagnosis of fixed-wing unmanned aerial vehicles," *Journal of Computational Design and Engineering*, vol. 9, no. 4, pp. 1511–1524, 2022.

[21] J. Yang, Y. Guo, and W. Zhao, "Aircraft actuator fault diagnosis using deep learning based sparse representation and TSM," in *Proceedings of the 2019 IEEE Aerospace Conference*, pp. 1–9, Big Sky, Ma, USA, March 2019.

[22] T. Gao, W. Sheng, Y. Yin, and X. Du, "A transfer learning based unmanned aerial vehicle MEMS inertial sensors fault diagnosis method," *Journal of Physics: Conference Series*, vol. 1852, no. 4, Article ID 042084, 2021.

[23] A. Bondyra, P. Gasior, S. Gardecki, and A. Kasiński, "Fault diagnosis and condition monitoring of UAV rotor using signal processing," in *Proceedings of the 2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications*, pp. 233–238, Poznan, Poland, September 2017.

[24] A. Altinors, F. Yol, and O. Yaman, "A sound based method for fault detection with statistical feature extraction in UAV motors," *Applied Acoustics*, vol. 183, Article ID 108325, 2021.

[25] W. Liu, Z. Chen, and M. Zheng, "An audio-based fault diagnosis method for quadrotors using convolutional neural network and transfer learning," in *Proceedings of the 2020 American Control Conference*, pp. 1367–1372, ACC, Philadelphia, PA, USA, July 2020.

[26] J. Miao, J. Wang, H. Zhang, and Q. Miao, "Overview of research progress of fault diagnosis technology for UAV," *Journal of Instrumentation*, vol. 41, pp. 56–69, 2020.

[27] S. Liang, S. Zhang, Y. Huang, X. Zheng, J. Cheng, and S. Wu, "Data-driven fault diagnosis of FW-UAVs with consideration of multiple operation conditions," *ISA Transactions*, vol. 126, pp. 472–485, 2022.

[28] B. Zhao, X. Zhang, Z. Zhan, and S. Pang, "Deep multi-scale convolutional transfer learning network: a novel method for intelligent fault diagnosis of rolling bearings under variable working conditions and domains," *Neurocomputing*, vol. 407, pp. 24–38, 2020.

[29] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: maximizing for domain invariance," 2014, https://arxiv.org/abs/1412.3474.

[30] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning*, F. Bach and D. Blei, Eds., pp. 97–105, Jmlr-Journal Machine Learning Research, San Diego, Ca, USA, 2015, https://www.webofscience.com/wos/alldb/full-record/WOS:000684115800011.

[31] G. Xiang, W. Chen, Y. Peng, Y. Wang, and C. Qu, "Deep transfer learning based on convolutional neural networks for intelligent fault diagnosis of spacecraft," in *Proceedings of the 2020 CHINESE AUTOMATION CONGRESS (CAC 2020)*, pp. 5522–5526, Shanghai, China, November 2020.

[32] L. Wan, Y. Li, K. Chen, K. Gong, and C. Li, "A novel deep convolution multi-adversarial domain adaptation model for rolling bearing fault diagnosis," *Measurement*, vol. 191, Article ID 110752, 2022.

[33] K. Wang, W. Zhao, A. Xu, P. Zeng, and S. Yang, "One-dimensional multi-scale domain adaptive network for bearing-fault diagnosis under varying working conditions," *Sensors*, vol. 20, no. 21, p. 6039, 2020.

[34] N. Lu, H. Hu, T. Yin, Y. Lei, and S. Wang, "Transfer relation network for fault diagnosis of rotating machinery with small data," *IEEE Transactions on Cybernetics*, vol. 52, no. 11, pp. 11927–11941, 2022.

[35] G. Ma, S. Xu, T. Yang et al., "A transfer learning-based method for personalized state of health estimation of lithium-ion batteries," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2022.

[36] B. Zhao, X. Zhang, Z. Zhan, and Q. Wu, "Deep multi-scale separable convolutional network with triple attention mechanism: a novel multi-task domain adaptation method for intelligent fault diagnosis," *Expert Systems with Applications*, vol. 182, Article ID 115087, 2021.

[37] F. Sun, R. Sun, and J. Yan, "Cross-domain active learning for electronic nose drift compensation," *Micromachines*, vol. 13, no. 8, p. 1260, 2022.

[38] L. Li and M. Cai, "Cross-species data classification by domain adaptation via discriminative heterogeneous Maximum mean discrepancy," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 1, pp. 312–324, 2021.

[39] Y. Liao, R. Huang, J. Li, Z. Chen, and W. Li, "Dynamic distribution adaptation based transfer network for cross domain bearing fault diagnosis," *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, p. 52, 2021.

[40] X. Pei, S. Su, L. Jiang, C. Chu, L. Gong, and Y. Yuan, "Research on rolling bearing fault diagnosis method based on generative adversarial and transfer learning," *Processes*, vol. 10, no. 8, p. 1443, 2022.

[41] Y. Ganin, E. Ustinova, H. Ajakan et al., “Domain-adversarial training of neural networks,” in *Domain Adaptation in Computer Vision Applications*, G. Csurka, Ed., Springer International Publishing, Berlin, Germany, pp. 189–209, 2017.

[42] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” 2015, https://arxiv.org/abs/1409.7495.

[43] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proceedings of the 30TH IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2017)*, pp. 2962–2971, IEEE Comp Soc, CVF, Honolulu, Hawaii, July 2017.

[44] A. Chadha and Y. Andreopoulos, “Improved techniques for adversarial discriminative domain adaptation,” *IEEE Transactions on Image Processing*, vol. 29, pp. 2622–2637, 2020.

[45] A. Ma, J. Li, K. Lu, L. Zhu, and H. T. Shen, “Adversarial Entropy optimization for unsupervised domain adaptation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6263–6274, 2022.

[46] G. Cai, Y. Wang, L. He, and M. Zhou, “Unsupervised domain adaptation with adversarial residual transform networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 3073–3086, 2020.

[47] J. Jiang, Y. Shu, J. Wang, and M. Long, “Transferability in deep learning: a survey,” 2022, https://arxiv.org/abs/2201.05867.

[48] R. K. Pearson, Y. Neuvo, J. Astola, and M. Gabbouj, “Generalized Hampel filters,” *EURASIP Journal on Applied Signal Processing*, vol. 2016, no. 1, p. 87, 2016.

[49] O. Salem, Y. Liu, and A. Mehaoua, “A lightweight anomaly detection framework for medical wireless sensor networks,” in *Proceedings of the 2013 Ieee Wireless Communications and Networking Conference (Wcnc)*, pp. 4358–4363, Ieee, New York, NY, USA, May 2013, https://www.webofscience.com/wos/alldb/full-record/WOS:000326048104077.

[50] S. Bhowmik, B. Jelfs, S. P. Arjunan, and D. K. Kumar, “Outlier removal in facial surface electromyography through Hampel filtering technique,” in *Proceedings of the 2017 IEEE Life Sciences Conference (LSC)*, pp. 258–261, Sydney, Australia, December 2017.

[51] K. Liu, P. Yan, Z. Ye, D. He, Q. Shi, and J. Li, “An application of multistep long-short term memory algorithm to predict the performance of capacitor voltage transformer,” in *Proceedings of the 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing*, pp. 236–239, Chengdu China, December 2019.

[52] S. Pomberger, M. Stoschka, and M. Leitner, “Cast surface texture characterisation via areal roughness,” *Precision Engineering*, vol. 60, pp. 465–481, 2019.

[53] P. Marti-Puig, A. Blanco-M, J. J. Cárdenas, J. Cusidó, and J. Solé-Casals, “Effects of the pre-processing algorithms in fault diagnosis of wind turbines,” *Environmental Modelling & Software*, vol. 110, pp. 119–128, 2018.

[54] D. Xiao, Y. Huang, L. Zhao, C. Qin, H. Shi, and C. Liu, “Domain adaptive motor fault diagnosis using deep transfer learning,” *IEEE Access*, vol. 7, pp. 80937–80949, 2019.

[55] F. Li, J. Chen, J. Pan, and T. Pan, “Cross-domain learning in rotating machinery fault diagnosis under various operating conditions based on parameter transfer,” *Measurement Science and Technology*, vol. 31, no. 8, Article ID 085104, 2020.

[56] A. Zhang, H. Wang, S. Li et al., “Transfer learning with deep recurrent neural networks for remaining useful life estimation,” *Applied Sciences*, vol. 8, no. 12, p. 2416, 2018.