

Article

A Fault Diagnosis Method for the Autonomous Underwater Vehicle via Meta-Self-Attention Multi-Scale CNN

Yimin Chen, Yazhou Wang, Yang Yu , Jiarun Wang and Jian Gao 

School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

* Correspondence: nwpuyuy@nwpu.edu.cn

Abstract: Autonomous underwater vehicles (AUVs) are an important equipment for ocean investigation. Actuator fault diagnosis is essential to ensure the sailing safety of AUVs. However, the lack of failure data for training due to unknown ocean environments and unpredictable failure occurrences is challenging for fault diagnosis. In this paper, a meta-self-attention multi-scale convolution neural network (MSAMS-CNN) is proposed for the actuator fault diagnosis of AUVs. Specifically, a two-dimensional spectrogram of the vibration signals obtained by a vibration sensor is used as the neural network's inputs. The diagnostic model is fitted by executing a subtask-based gradient optimization procedure to generate more general degradation knowledge. A self-attentive multi-scale feature extraction approach is used to utilize both global and local features for learning important parameters autonomously. In addition, a meta-learning method is utilized to train the diagnostic model without a large amount of labeled data, which enhances the generalization ability and allows for cross-task training. Experimental studies with real AUV data collected by vibration sensors are conducted to validate the effectiveness of the MSAMS-CNN. The results show that the proposed method can diagnose the rudder and thruster faults of AUVs in the cases of few-shot diagnosis.

Keywords: fault diagnosis; few-shot diagnosis; meta-learning; self-attention



Citation: Chen, Y.; Wang, Y.; Yu, Y.; Wang, J.; Gao, J. A Fault Diagnosis Method for the Autonomous Underwater Vehicle via Meta-Self-Attention Multi-Scale CNN. *J. Mar. Sci. Eng.* **2023**, *11*, 1121. <https://doi.org/10.3390/jmse11061121>

Academic Editor: Rafael Morales

Received: 25 April 2023

Revised: 16 May 2023

Accepted: 23 May 2023

Published: 26 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The oceans contain a great number of natural resources and play a vital role in human development, but ocean exploration remains challenging. As an efficient way of apprehending ocean exploration and utilization, AUVs have gained wide application in marine ecological research, mineral resource exploitation, scientific exploration, and the military. However, the complex marine environment poses significant challenges to AUV operations. The unknown ocean conditions may cause equipment malfunction, leading to mission failure or even equipment loss. Therefore, for the purpose of ensuring sailing safety, it is important to identify the failures and diagnose the actuator faults of the AUVs.

In recent years, various researchers have conducted a significant number of studies in the field of actuator fault diagnosis of AUVs. The health status of the actuator is diagnosed and identified by machine learning and deep learning methods. Fuzzy logic torque controllers with nonlinear friction compensation are used to improve the deterioration of trajectory tracking performance in rotating tandem elastic actuator systems caused by these nonlinear elements and to optimize trajectory tracking performance [1]. To diagnose thruster faults in AUVs under noisy environments, Sun et al. [2] proposed a thruster fault diagnosis method for AUVs based on deep neural networks and denoising autoencoder. Liu et al. [3] proposed a deep reinforcement learning-based controller in a simulation environment for studying the control performance of the vector thruster. The information that can be measured by the sensors inside the AUVs is used as an input parameter. Tsai et al. [4] proposed a fault diagnosis method for underwater thrusters based on deep convolutional neural networks, where the raw data were transformed from the time domain to the frequency domain by fast Fourier transform and used as the input to

the neural network. Chu et al. [5] proposed an underwater thruster fault diagnosis method based on the random forest regression and support vector machine, which mainly solves the problem of insufficient diagnostic accuracy due to sample imbalance. Zhu et al. [6] proposed a fault diagnosis method based on Bayesian networks to solve the problem of difficulty in locating specific faults in the thrusters of autonomous underwater vehicles. A deep belief network is introduced in the multi-sensor information fusion model to identify uncertain and unknown signal, continuously changing the failure modes of deep-sea manned submersible thrusters [7]. The FDI method proposed by Talebi et al. [8] is used to detect, isolate, and identify the severity of actuator failures in the presence of disturbances and fault uncertainty in the model. To improve the performances of thruster fault diagnosis, Tian et al. [9] proposed a possible fuzzy C-means algorithm for thruster fault classification. Omerdic et al. [10] proposed a novel thruster fault diagnosis and adaptation system for open-frame underwater vehicles. The FDS monitors the status of each thruster using a fault detection unit associated with each thruster, which is capable of detecting the internal and external fault states of the thruster. Raanan et al. [11] proposed a Gaussian particle filtering algorithm to estimate the failure model of AUVs. The Bayesian algorithm is used to implement the fault detection of AUV thrusters, and a nearest neighbor classifier is used to accomplish the fault diagnosis. Ji et al. [12] proposed a new sequential convolutional neural network (SeqCNN), where a model-free fault diagnosis method is based on deep learning algorithms. The proposed SeqCNN aims to extract global and local features from state data and classify the extracted information into different fault types. Yeo et al. [13] proposed a system that can monitor the health of AUV thrusters using a convolutional neural network (CNN). The acoustic signal of the thruster was used as input data to achieve the fault diagnosis of the thruster of the AUVs. Tsai et al. [14] proposed a modified CNN based on merging two signals to classify faults. Experimental results show that the proposed multi-signal approach achieves excellent thruster fault diagnosis results. Kim et al. [15] proposed a fault detection system in which new vibration data were generated using a generative adversarial network (GAN) and was applied to a long short-term memory neural network. In the fault detection experiments of the underwater thruster, the vibration characteristics of the vibration sensor data obtained from the experiments and the data generated by the GAN were compared and analyzed using the fast Fourier transform. To study the model-free trajectory tracking control problem of AUVs, Bingul et al. proposed a novel control structure based on the model-free control principle to ensure stable and accurate trajectory tracking of AUVs in complex underwater environments [16].

The rudder is a kind of servo mechanism which is widely used in the control system of aircraft, ships, missiles, navigators, etc. It receives control signals and drives the deflection of the rudder plate to control the navigator's attitude and trajectory. It is necessary to analyze the state, static, and dynamic parameters of the rudder system during operation to ensure the accurate control of the rudder system. Wang et al. [17] proposed a convolutional neural network combining a particle swarm optimization algorithm and a grayscale optimization algorithm to accomplish the automatic multi-fault classification of rudders. An adaptive sampling algorithm (ASCIN) considering information instances was proposed by Li et al. [18,19]. The optimal parameters of ASCIN were searched by the whale optimization algorithm to overcome the limitations of conventional detection devices. Zhou et al. [20] proposed a deep neural network-based health monitoring method. Autoencoders were used to reduce the feature dimensionality and combined with the SoftMax classifier for health monitoring. Ren et al. [21] proposed a fault diagnosis method based on a Back Propagation Neural Network (BPNN), which was optimized using the SODEBBO optimization algorithm to accomplish the automatic classification of multiple faults in rudder testing. Qin et al. [22] optimized the BPNN with particle swarm optimization hybrid fruit fly algorithm to improve the diagnostic performance. Chang et al. [23] proposed a random forest algorithm based on a shuffle frog-jumping algorithm to make the decision-making of the model more efficient and accurate. Xu et al. [24] proposed the combination of a nonlinear unknown input observer with adaptive thresholding, which helps to reduce the effect of

model uncertainty. Chang et al. [25] proposed a machine learning method-based optimized decision tree algorithm to solve the common decision difficulties caused by low-precision decisions and high voting competition in tree models. Jiang et al. [26] used finite impulse response and principal component analysis to solve the fault diagnosis problem of AUV actuators under winding failure and improve the overall reliability of underwater vehicles. Xuan et al. [27] proposed a fault diagnosis method based on convolutional neural networks and autoencoders to solve the problem of fault detection and isolation in uncoupled fault modes. Jia et al. [28] proposed an ensemble deep autoencoder based on an extreme learning machine, which has significant advantages in handling imbalanced data.

However, in previous studies, machine learning and data augmentation methods have been widely used for fault analysis and diagnosis. Traditional machine learning, which heavily relies on a priori knowledge, often struggles with selecting appropriate features and can result in poor performance. Data augmentation methods, on the other hand, primarily generate more similar samples to expand the dataset, and while GAN can also perform data augmentation, they inherently require a large amount of data to accurately learn the data distribution of the samples. In practical underwater equipment working environments, the equipment is often expensive, and simulating failures can incur substantial costs, leading to sparse data. Consequently, data augmentation methods also suffer from certain drawbacks.

A fault diagnostic method based on meta-deep learning is thus proposed in this paper to achieve fault identification in the cases of a few-shot diagnosis. Specifically, the mechanical equipment's vibration signals are first collected to create a two-dimensional spectrogram with time–frequency information. Next, a feature extraction model and classifier are built. This model is optimized using a subtask-based gradient descent optimization approach, in order to extract degraded information that can represent the fault status. Then, a multi-scale self-attentive convolutional neural network is proposed for feature extraction, which allows for the model to locate critical features and obtain various degradation indicators considering both global and local information. The contributions of this work are summarized as follows:

1. A diagnostic model that considers both global and local information in feature extraction is developed by introducing the multi-scale deep learning method into the self-attentive mechanism. The extracted features that are more suitable for characterizing the health status of the AUV actuators can be automatically obtained for fault diagnosis and classification.
2. The meta-learning approach is incorporated so that the diagnostic model can be trained iteratively with few-shot and converge rapidly. The meta-learning approach enables cross-task training and completes cross-device fault diagnostics by enhancing the generalization abilities without requiring a large amount of labeled data.
3. A meta-self-attentive multi-scale deep learning method for actuator fault diagnosis of AUVs is proposed to diagnose the rudder and thruster faults with a few-shot. Experimental studies demonstrate the effectiveness of the proposed method in actuator fault diagnosis of AUVs.

The background information in Section 2 serves as the foundation for the remainder of this essay. Our suggested few-shot identification method is thoroughly explained in Section 3. The AUVs dataset is demonstrated in Section 4 along with a thorough experimental validation that contrasts it with other diagnosis techniques. A summary of this paper and a prognosis for further investigation are provided in Section 5.

2. Background

This section consists of three subsections that introduce the fundamental concepts required to build the model. The first subsection explains the concept of small sample learning and the objective function for model optimization. The second subsection covers the method of meta-learning and the dataset division. Lastly, the third subsection discusses the self-attentive mechanism concept and how to apply it.

2.1. Few-Shot Learning

Few-shot learning attempts to overcome the problem of insufficient data, whereas traditional machine learning and deep learning call for a high number of training samples and will significantly worsen recognition performance in the event of insufficient training samples. Nonetheless, in this instance, few-shot learning also produces good recognition performance. From a learning task \mathcal{T} [27], a small-scale training set $D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^N$, and a test set $D_{\text{test}} = \{x_i\}_{i=1}^M$, define $P(x, y)$ is the joint probability distribution of x and y , \hat{h} is the optimal hypothesis from x to y , while less-sample learning can be obtained \hat{h} by learning on D_{train} and testing on D_{test} . To approximate \hat{h} , it determines a hypothetical space H consisting of a hypothesis $h(\cdot; \theta)$, which θ represents all the parameters used by h . It can be thought of as an optimization process that finds H to obtain θ , holding the optimal $h^* \in H$ in the training set.

The objective function, which is written as Equation (1), is the function that is typically maximized or minimized to determine the corresponding parameter weights in traditional machine learning and deep learning problems. Equation (2) gives a concrete representation of the loss function.

$$\min_{\theta} \sum_{(x_i, y_i) \in D_{\text{train}}} \text{Loss}(h(x_i; \theta), y_i) \quad (1)$$

$$\text{Loss}(h(x_i; \theta), y_i) = - \sum_{x^{(j)}, y^{(j)} \in D} f_{\theta}(x^{(j)}) \log(y^{(j)}) \quad (2)$$

where θ is defined as the parameter of the less-shot learning model.

The sample size N is much smaller in few-shot learning, leading to more pronounced deviations between target values. The resulting model h_N is likely to be overfitted, so it is crucial to investigate appropriate methods for few-sample learning.

2.2. Meta-Learning

A large amount of data from a certain scenario is generally utilized to train a model in machine learning, and the model needs to be retrained when the scenario changes. However, this is not in line with the law of human learning knowledge, as humans and even infants have the inherent ability to rapidly learn new concepts with a small number of samples and accurately summarize them in unknown situations.

The intelligent recognition models currently established require large-scale data, but the cost of obtaining these data is generally relatively high, resulting in unsatisfactory recognition results. These basic recognition models are all focused on a specific task, and when encountering cross-task situations, they need to acquire a large amount of data in another task domain to retrain the network. This is in stark contrast to humans mastering knowledge and migrating to uncharted territory. However, meta-learning [29] makes it possible to discover prototypes in a given domain under specific tasks and optimize the network by extracting general knowledge of different subtasks. The method can be seen as a solution for few-shot learning, mining more valuable parameters and enhancing its generalization ability among other differently distributed data [27].

The training unit is the task in meta-learning, which is generally divided into a training task and a testing task. Many subtasks need to be prepared to learn the model, aiming to learn better hyperparameters that are required to fit the new tasks. A learning task $T = \{y, f\}$ consists of two parts: the label space y and the prediction function f . The prediction function is trained on a given feature vector and label pair $\{x_i, y_i\}$, $x_i \in x, y_i \in y$.

2.3. Self-Attentive Mechanism

Attention is a complex cognitive function that is indispensable to humans, which refers to the ability to choose some information while ignoring another. The ability of the human brain to select, consciously or unconsciously, small portions of useful information to focus on from this large amount of input and to ignore other information is known as

attention. When using neural networks to process a large amount of input information, you can also learn from the attention mechanism of the human brain and select only some key information inputs for processing to improve the efficiency of neural networks. It is also possible to improve the efficiency of the neural network by drawing on the attention mechanism of the human brain when a neural network is used to process a large amount of input information.

Let us use $X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{D \times N}$ to represent N groups of input information, where the D dimensional vector $x_n \in \mathbb{R}^D$, $n \in [1, N]$ represents a set of input information. To save computing resources and reduce the amount of calculation, it is not necessary to input all the information into the neural network, but only to select some task-related information from X . The calculation of the attention mechanism can be divided into two steps: one is to calculate the attention distribution on all input information, and the other is to calculate the weighted average of the input information according to the attention distribution. To select task-related information from input vector $[x_1, x_2, \dots, x_N]$, a task-related representation called query vector needs to be introduced, and the correlation between each input vector and query vector is calculated through a scoring function.

In order to improve the feature extraction ability of the model, the self-attentive mechanism often adopts the Query-Key-Value (Query-KV) model. Given the query vector $q_n \in Q$, the output vector h_n can be obtained by Equation (4).

$$\alpha_{n,j} = \text{soft max}(s(k_j, q_n)) \quad (3)$$

$$h_n = \text{att}((K, V), q_n) = \sum_{j=1}^N \alpha_{n,j} v_j \quad (4)$$

where $n, j \in [1, N]$ is the position of the sequence of output and input vectors, $\alpha_{n,j}$ denotes the weight of the first output concern to the first input, $s(\cdot)$ denotes the scoring function, k_j denotes the j th element of the value vector, and v_j denotes the j th element of the value vector.

Generally, the scoring function is calculated using the method of scaling the dot product, with the formula is $s(h, q) = \frac{h^T q}{\sqrt{D}}$, and the output sequence can be calculated as

$$H = \text{soft max}(\frac{K^T Q}{\sqrt{D_k}}) V \quad (5)$$

where K, V, Q are the key, value, and query matrix, respectively. $\sqrt{D_k}$ is a normalization constant to ensure gradient stability.

3. Proposed Method

This section consists of three subsections that introduce the proposed method and the steps involved in building the network model. The first subsection describes the overall framework and implementation steps for the fault diagnosis of AUV actuators. The second subsection explains the self-attentive multiscale feature extraction method. Lastly, the third subsection explains the meta-learning optimization method in the case of a few-shot.

3.1. Research Steps of Intelligent Fault Diagnosis

This section proposes a fault diagnosis method based on meta-learning, which is mainly applied to few-shot learning. Figure 1 represents the overall framework of few-shot fault diagnosis consisting of six main steps.

First, the vibration data of the equipment movement are collected as the original signal. The signal is normalized to avoid the presence of singular values that affect the convergence speed of the network.

Then, the short-time Fourier transform is performed on the processed data to obtain a two-dimensional feature map that contains time-domain information and frequency-domain information, which is convenient for the subsequent feature extraction.

Third, in contrast to the traditional deep learning fitting model, the training and testing datasets are redefined to perform model fitting in a subtask-based learning approach. The task-based training approach is to divide the data into two parts: the training task and the testing task. The training dataset is usually a dataset from other related domains, which is used to assist learning. In this part of the data, the support set serves as the learning data for each subtask, while the query set is used for cross-subtask training to learn the meta-knowledge of the model, aiming to obtain more sensitive parameter states of the established meta-multi-scale feature extraction model. The division between the support set and the query set is different in the test dataset. The former is to fine-tune the meta-learning framework that is already fitted by the training task dataset, while the latter is to complete a small number of predictions and verify the feasibility of the established model.

Fourth, a meta-learning model based on self-attentive multi-scale feature extraction is proposed for the fault diagnosis problem of AUVs based on the industrial status of sparse fault data, which can be used to achieve fault identification and health prediction for AUVs.

Fifth, the model is trained and fine-tuned using the data segment in the third step and working on finding the optimal parameters.

Finally, the already fitted meta-learning network is used for fault diagnosis of the autonomous underwater vehicle.

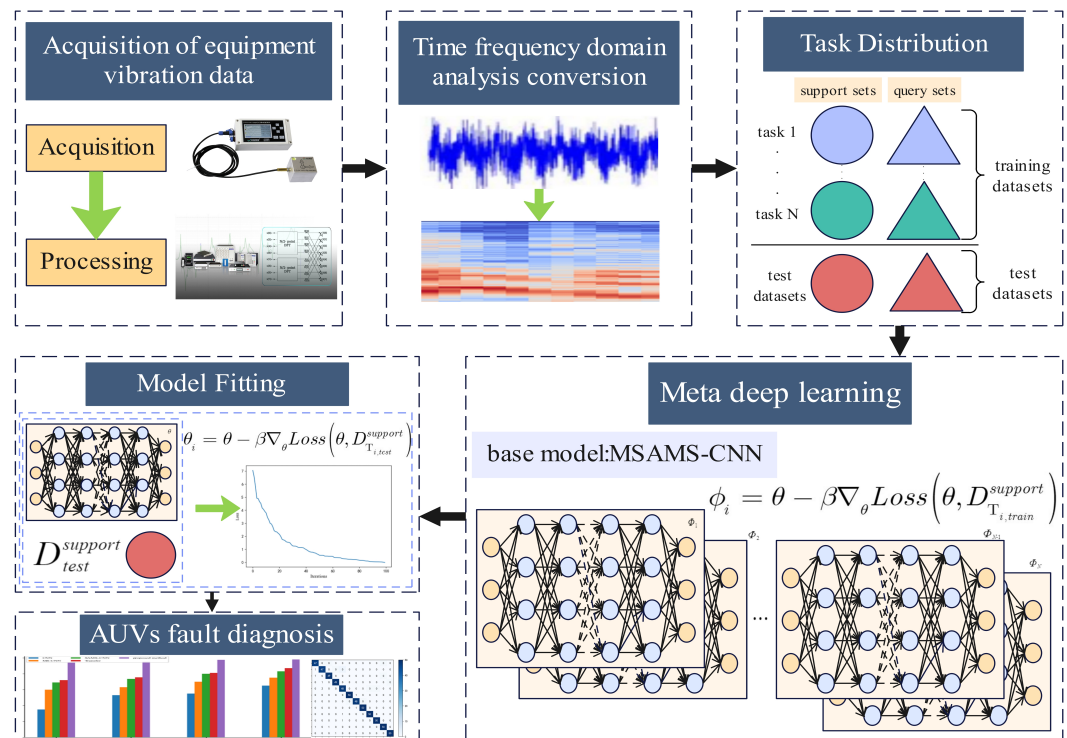


Figure 1. Framework for few-shot fault diagnosis.

3.2. Fault Diagnosis Model with Self-Attentive Multi-Scale Feature Extraction

In recent years, CNNs have been widely used in the field of fault detection of mechanical equipment, which usually consists of convolutional layers, batch normalization layers, and pooling layers, and has significant capabilities in extracting features. Although CNNs can be found to perform well in this field, in some studies [30,31], the performance of this method needs to be improved in the case of few-shot learning. This section will comprehensively use the knowledge of multi-scale kernel and self-attention mechanism to propose a self-attention multi-scale feature extraction model for the fault diagnosis of AUVs with a few-shot.

The proposed model contains three branches, i.e., three different convolution kernel sizes. Different convolution kernels are performed on the input data to obtain different types of features, and self-attention modules are added after the features are extracted

by each convolution kernel. Multi-scale can take into account both global and local features, and the attention mechanism enables the model to learn important information autonomously and reduce the number of operations. To connect these three branches, a concatenate layer is used directly to connect the three branches. However, the output dimensions of different convolutional kernels are inconsistent, and pooling and batch normalization layers need to be added before the concatenation layer to ensure dimensionality consistency. Overall, the general architecture of the proposed model is shown in Figure 2, and the detailed steps are as follows:

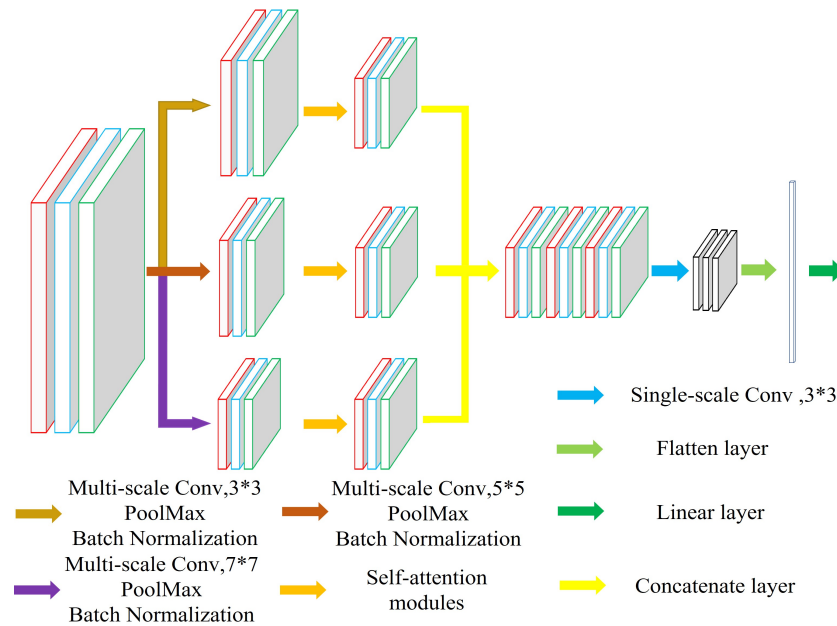


Figure 2. Self-attention multi-scale feature extraction model.

The original data are first subjected to a short-time Fourier transform. The basic idea is to first multiply a function and a window function, then perform a one-dimensional Fourier transform, and obtain a series of spectral information by sliding the window function. These results are successively stitched together to obtain a two-dimensional time–frequency map. The basic operation formula is as follows:

$$X(t, \omega) = \int_{-\infty}^{+\infty} x(t)h(t - \omega)e^{-j\omega t}dt \quad (6)$$

where $x(t)$ is the time domain signal, $h(t - \omega)$ is the window function, and X is used to represent the result of the short-time Fourier transform.

Input the spectrogram X to the multi-scale convolutional layer for feature extraction, and the convolution calculation formula is as follows:

$$Z_1(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} x_{i,j} \cdot k_{(u-i,v-j)}^1 \quad (7)$$

$$Z_2(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} x_{i,j} \cdot k_{(u-i,v-j)}^2 \quad (8)$$

$$Z_3(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} x_{i,j} \cdot k_{(u-i,v-j)}^3 \quad (9)$$

among them, k^1, k^2, k^3 represent three different convolution kernels, and Z_1, Z_2, Z_3 represent the results calculated by different convolution kernels.

The results of the multi-scale convolution are down-sampled and normalized. The features are activated in the batch normalization layer using the ReLU activation function. The specific calculation formula is as follows:

$$(\widehat{Z}_1, \widehat{Z}_2, \widehat{Z}_3) = \max pool(Z_1, Z_2, Z_3) \quad (10)$$

$$(A_1, A_2, A_3) = \text{ReLU}(\widehat{Z}_1, \widehat{Z}_2, \widehat{Z}_3) \quad (11)$$

where $\widehat{Z}_1, \widehat{Z}_2, \widehat{Z}_3$ denote the result of the pooling operation and A_1, A_2, A_3 denote the output after normalization, which is dimensionally consistent throughout the processing.

To further extract features, the model is equipped with the ability to extract important features autonomously through the self-attentive module, using A_1, A_2, A_3 features as input. Then, the input sequence is $A = [a_1, a_2, \dots, a_N] \in \mathbb{R}^{D \times N}$ and the output sequence is $H = [h_1, h_2, \dots, h_N] \in \mathbb{R}^{D_v \times N}$. The specific calculation process is as follows:

Each input a_i is mapped to three different spaces. Figure 3 represents the transformation process to obtain the query vector $q_i \in \mathbb{R}^{D_k}$, the key vector $k_i \in \mathbb{R}^{D_k}$, and the value vector $v_i \in \mathbb{R}^{D_v}$. For the whole input sequence A , the linear mapping process can be abbreviated as Equations (12)–(14):

$$Q = W_q A \in \mathbb{R}^{D_k \times N} \quad (12)$$

$$K = W_k A \in \mathbb{R}^{D_k \times N} \quad (13)$$

$$V = W_v A \in \mathbb{R}^{D_v \times N} \quad (14)$$

where $W_q \in \mathbb{R}^{D_k \times D_x}$, $W_k \in \mathbb{R}^{D_k \times D_x}$, $W_v \in \mathbb{R}^{D_v \times D_x}$ are the parameter matrices of the linear transformation, $Q = [q_1, q_2, \dots, q_N]$, $K = [k_1, k_2, \dots, k_N]$, $V = [v_1, v_2, \dots, v_N]$ denote the matrices composed of query vector, key vector, and value vector, respectively.

For the query vector $q_n \in Q$, the output vector h_n can be obtained by using the key-value pair attention mechanism of Equation (4).

Attention H_1, H_2, H_3 can be obtained by Equations (3)–(5) and (11)–(13).

Use a concatenate layer to merge the features of the three branches and turn the channel into a single-channel feature through point convolution, as follows:

$$H = \text{concat}(H_1, H_2, H_3) \quad (15)$$

The output features are further fed into the Softmax classifier for state identification and fault classification after the above operations. For clarity, the detailed parameters of the convolution part are given in Table 1.

Table 1. Parameters of the convolution part.

Convolution Category		Convolution Kernel (Channel \times Width \times Height)	Strides
Multi-scale Conv	Conv1	(3,3,3)	(1,1)
	Conv2	(3,5,5)	(1,1)
	Conv3	(3,7,7)	(1,1)
Signal-scale Conv	Conv4	(3,3,3)	(1,1)

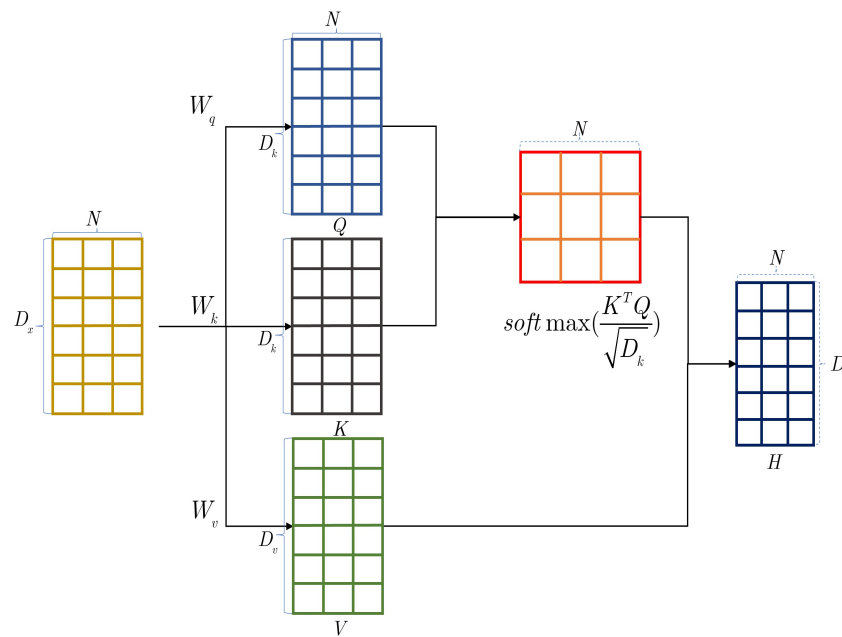


Figure 3. The calculation process of self-attention model.

3.3. Meta-Self-Attentive Multi-Scale CNN

The commonly used methods are meta-learning and transfer learning in cross-domain problems. From the left side of Figure 4, it can be seen that meta-learning summarizes N loss values from N tasks, and then the total loss can be expressed as $loss = \sum_{n=1}^N loss^n(\hat{\theta}^n)$. This method takes into account the optimization process of each task. Although the obtained optimal parameter θ^* cannot perform well in every task, it can have good generalization capability based on that parameter. A good recognition performance can be obtained by fine-tuning the model with the support set of test samples for each task. The model is trained mainly by obtaining the optimal parameter through the minimum loss function $loss(\theta^*) = \min\{loss^1(\hat{\theta}^1), loss^2(\hat{\theta}^2), \dots, loss^N(\hat{\theta}^N)\}$ of the N learning tasks. From the right side of Figure 4, it can be seen that the calculation θ^* is only the optimal parameter $\hat{\theta}^1$ for task 1. This will cause the optimization of other tasks to fall into local optimum.

The proposed self-attention multi-scale feature extraction model is described in Section 3.2, which will be trained and learned, since the failure data of autonomous underwater vehicles are very sparse in real-world environments. To cope with the few-shot learning problem, we use a meta-learning strategy to optimize the model. As shown in Figure 1, step 4 aims to generate a meta-deep learning model for generalization to the actual industrial field to implement underwater vehicle actuator fault diagnosis. We need to fit the model through a small number of datasets after establishing the fault diagnosis model, aiming to find the optimal parameters and ensure better performance in the fault diagnosis of underwater vehicles. Thus, the proposed model is trained with the training dataset delineated in step 2 in Figure 1. The training dataset can also be referred to as the source dataset and the test dataset becomes the target dataset; the whole training process is shown in Figure 5.

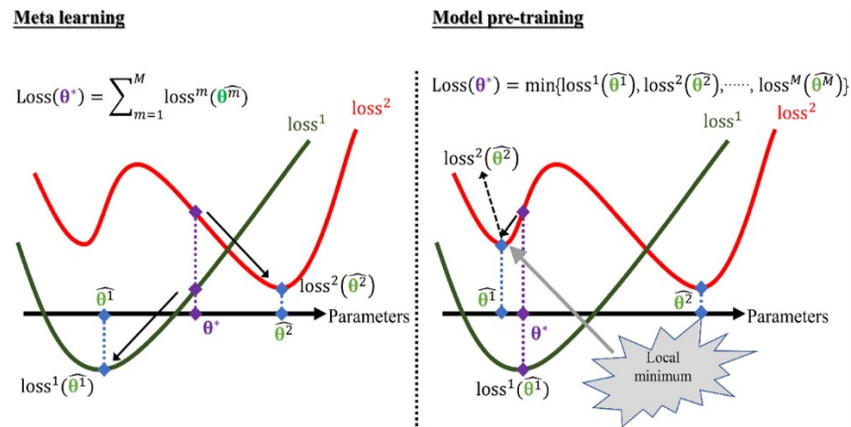


Figure 4. Difference between meta-learning and model pre-training.

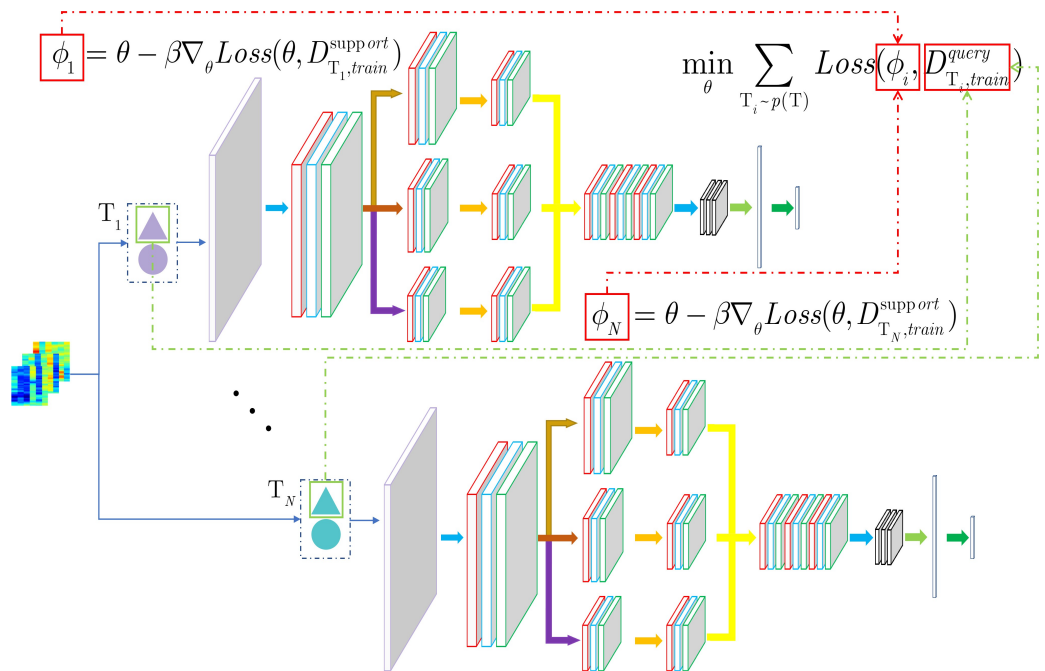


Figure 5. The training process of MSAMS-CNN.

According to the learning method in Section 2.1, it is necessary to optimize the network by minimizing the objective function, which is expressed as Equation (1).

Expected risk is the global concept that is used to measure the loss of the model concerning the joint distribution $P(x, y)$. Since $P(x, y)$ is unknown, the problem can be transformed into minimizing the expected loss on the training set. The problem then translates into optimizing the model by averaging the losses on the training set, as in Equation (16):

$$R(h) = R_{emp}(h) = \frac{1}{N} \sum_{n=1}^N L(y, h(x; \theta)) \quad (16)$$

where R_{emp} is called empirical risk, and this training process is called empirical risk minimization.

In summary, the total error [30] can be decomposed into Equation (19).

$$\varepsilon_{app}(H) = E[R(h^*) - \arg \min_f R(h)] \quad (17)$$

$$\varepsilon_{est}(H, N) = E[R_N(h) - R(h^*)] \quad (18)$$

$$E[R_N(\hat{h})] = \varepsilon_{app}(H) + \varepsilon_{est}(H, N) \quad (19)$$

where the approximation error $\varepsilon_{app}(H)$ measures the difference between the predicted and actual values of the few-shot learning model in H . The estimation error $\varepsilon_{est}(H, N)$ indicates the difference between the estimated and actual coefficients of the model.

The rules for dividing the dataset are shown in Figure 6. We randomly select N subtasks $T = \{T_1, T_2, \dots, T_N\}$ from the training set, and the data are divided into the support set and query set in each task $T_i, i \in [1, N]$, represented as $D = \{D^{Support}, D^{Query}\}$. Use the support set to complete the iterative optimization of each subtask, and obtain the best parameter θ^* by Equation (20), in which i represents the i th task.

$$\theta^{*(i)}(\phi) = \arg \min Loss^{task}(\theta, \phi, D_{train}^{Support(i)}) \quad (20)$$

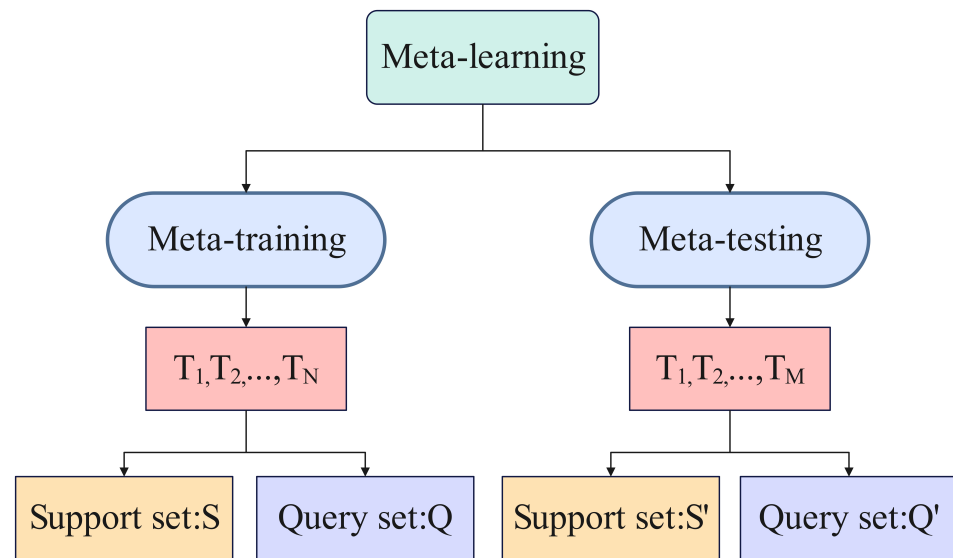


Figure 6. Meta-learning data segmentation.

Using the query set of the training dataset to learn the meta-knowledge, the current optimal parameters ϕ^* are obtained by Equation (21). Since we are addressing a multi-classification problem, the cross-entropy loss function is used. Equation (22) is the cross-entropy loss function:

$$\phi^* = \arg \min_{\phi} \sum_{i=1}^N Loss^{meta}(\theta^{*(i)}(\phi), D_{train}^{query(i)}) \quad (21)$$

$$Loss(\theta^*, D^{train}) = - \sum_{x^{(j)}, y^{(j)} \in D^{train}} f_{\phi_i}(x^{(j)}) \log(y^{(j)}) \quad (22)$$

where $f_{\phi_i}(x^{(j)})$ represents the trained model, $y^{(j)}$ represents the sample label, and ϕ_i represents the model parameters updated by the i th task.

The performance needs to be tested under the test dataset after the model is fitted under the training dataset. In general, the training data and the test data are from different tasks. Since using the trained model directly on the test set does not work very well,

the model parameters are first fine-tuned using the support set of the test dataset. Finally, the model performance is tested using the query set of the test dataset.

$$\theta^* = \arg \min_{\theta} (\theta, \phi^*, D_{test}^{Support}) \quad (23)$$

For convenience, the above optimization process is organized into a pseudo-code, as described in Algorithm 1.

Algorithm 1 meta-learning strategy

```

1: while not done do
2:   repeat
3:     random sample task  $T_i \sim p(T)$  in total task
4:     for each  $T_i$  in  $D^{train}$  do
5:       gain  $D_{T_i}^{support}$  by sampling m input-output pairs from  $D^{train}$ ;
6:       evaluate models using cross-entropy loss;
7:       update parameters  $\theta$  with gradient optimization of Equation (20);
8:     end for
9:     gain  $D_{T_i}^{query}$  by sampling n input-output pairs from  $D^{train}$ ;
10:    update parameters  $\phi$  using  $D_{T_i}^{query}$  in Equation (21);
11:    for each  $T_i$  in  $D^{test}$  do
12:      gain  $D_{T_i}^{support}$  by sampling m input-output pairs from  $D^{test}$ ;
13:      Fine-tuning self-attentive multi-scale model parameters  $\hat{\theta}$  using  $D_{T_i}^{support}$  from  $D^{test}$ ;
14:    end for
15:    gain  $D_{T_i}^{query}$  by sampling n input-output pairs from  $D^{test}$ ;
16:    test model performance using  $D_{T_i}^{query}$  from  $D^{test}$ ;
17:  until Accomplish a few-shot fault diagnosis.

```

4. Experimental Result

4.1. Dataset Introduction

This section uses the vibration data of the AUVs of Northwestern Polytechnical University to verify the proposed actuator fault diagnosis method. Therefore, a high-precision vibration data acquisition experimental platform is established for data acquisition, as shown in Figure 7. The data are divided into rudder vibration data and thruster vibration data, and the sampling frequency is 25,600 Hz. These data are collected in the presence of other device movements, and all data contain noise. Both data types are intended for model training and testing.

The rudder and thruster do not have status feedback in the actual working environment, resulting in an unknown factor regarding whether they operate according to the specified command during operation. Therefore, we need to use the vibration signals to classify the commands of the rudder and thruster, and based on the classification results we can conclude whether the actuator is faulty. The command distribution of the actuator is shown in Table 2. There are 20 most common commands for thrusters and 12 for rudders, where the minus signs all indicate reverse rotation. To make full use of the vibration data of the actuator, the data are enhanced and resampled in the original data file, using a sliding window with a window size of T and a step size of l . T moves from the start time of the file to the end in step l , and the training and test samples are built. As shown in Figure 8, this article sets T to 2048 and l to 1356.

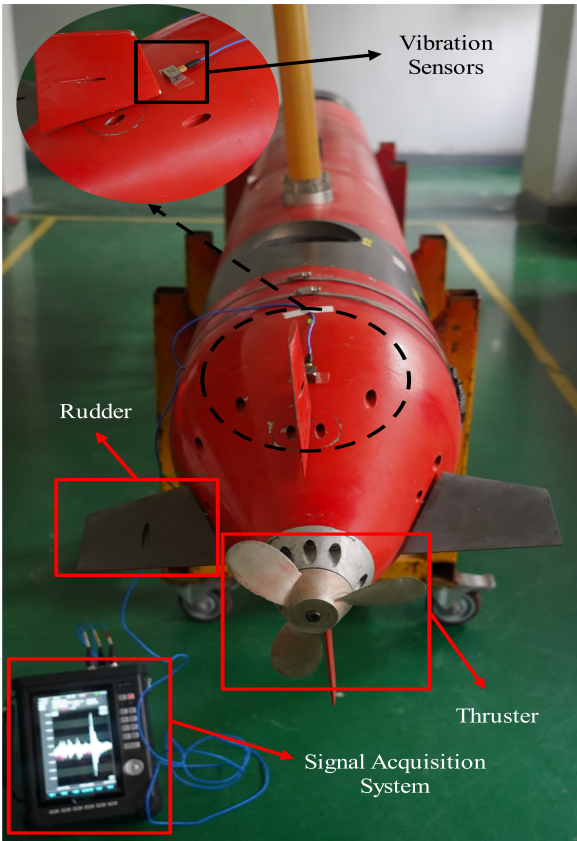


Figure 7. Physical view of data acquisition experiment platform.

Table 2. Distribution of instructions for implementing agencies.

Implementing Agency	Command Distribution	Data Name
Thruster (v)	−1	Thruster 1
	−0.9	Thruster 2
	−0.8	Thruster 3
	−0.7	Thruster 4
	−0.6	Thruster 5
	−0.5	Thruster 6
	−0.4	Thruster 7
	−0.3	Thruster 8
	−0.2	Thruster 9
	−0.1	Thruster 10
	0.1	Thruster 11
	0.2	Thruster 12
	0.3	Thruster 13
	0.4	Thruster 14
	0.5	Thruster 15
	0.6	Thruster 16
	0.7	Thruster 17
	0.8	Thruster 18
	0.9	Thruster 19
	1	Thruster 20
Rudder (°)	0→30→0	Rudder 1
	0→25→0	Rudder 2
	0→20→0	Rudder 3
	0→15→0	Rudder 4
	0→10→0	Rudder 5
	0→5→0	Rudder 6
	0→−5→0	Rudder 7
	0→−10→0	Rudder 8
	0→−15→0	Rudder 9
	0→−20→0	Rudder 10
	0→−25→0	Rudder 11
	0→−30→0	Rudder 12

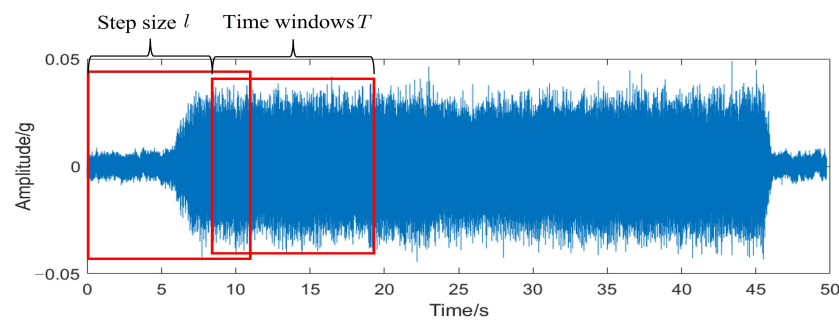


Figure 8. Schematic diagram of sample segmentation.

The typical instructions for the rudder and thruster of the AUV are shown in Table 2 together with the accompanying data files. With the use of these data, we have to produce a meta-training set and meta-test set with various instructions representing various categories, each of which contains some of the samples. The experiment is divided into two cases: Case1 has the thruster as the source domain and the rudder as the target domain; Case 2 has the rudder as the source domain and the thruster as the target domain. The specific dataset partitioning is shown in Table 3.

Table 3. Divide the data into meta-training sets and meta-test sets.

Datasets		Data File	Number of Samples per Class
Case 1	Meta-training	Thruster	50
	Meta-test	Rudder	50
Case 2	Meta-training	Rudder	50
	Meta-test	Thruster	50

Figure 9 shows part of the time domain waveform of the rudder, which was collected in the operating mode where the rudder and other mechanical devices are operating together. It shows that the vibration signal of the rudder contains background noise, which will be more challenging for the proposed fault diagnosis method. Figure 10 shows part of the time domain waveform of the thruster, where the thruster and the other mechanical devices are also in a commotion mode of operation and the vibration signal contains background noise.

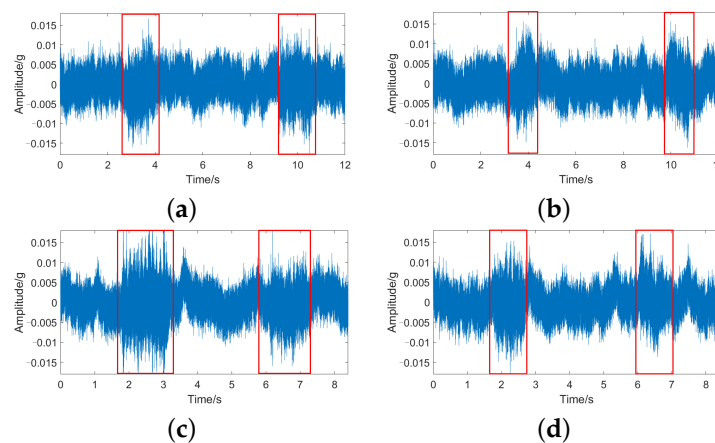


Figure 9. Time domain waveform of rudder. (a) $0^\circ \rightarrow 30^\circ \rightarrow 0^\circ$ Rudder signal. (b) $0^\circ \rightarrow 20^\circ \rightarrow 0^\circ$ Rudder signal. (c) $0^\circ \rightarrow -30^\circ \rightarrow 0^\circ$ Rudder signal. (d) $0^\circ \rightarrow -20^\circ \rightarrow 0^\circ$ Rudder signal.

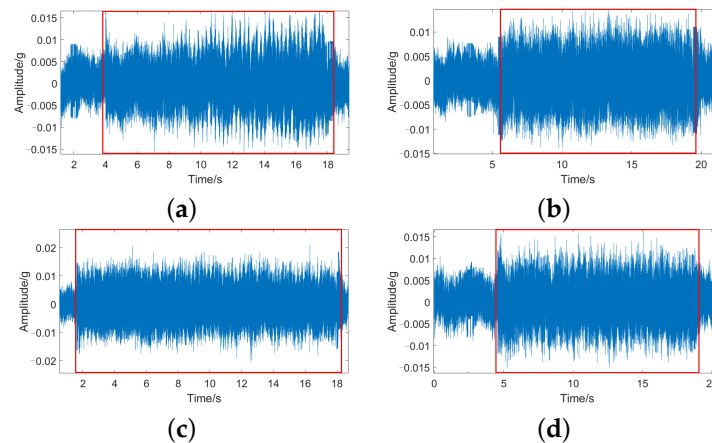


Figure 10. Time domain waveform of rudder. (a) 0.5 vs. Thruster signal. (b) -0.5 vs. Thruster signal. (c) 0.8 vs. Thruster signal. (d) -0.8 vs. Thruster signal.

4.2. Benchmark Algorithms

To verify the effectiveness of the proposed actuator fault diagnosis method, four ablation experiments were set up. Both the ablation experiments and the proposed model use the actuator vibration data of the AUVs and are identical except for the different model structures. The following four ablation experiments are presented:

CNN: First, a CNN is used to implement fault detection, and the model consists of two convolutional layers, two pooling layers, two normalization layers, and two fully connected layers. The convolution kernel size of both convolutional layers is 3×3 .

Multi-scale CNN: The second ablation experiment uses a multi-scale convolutional neural network as a model, consisting of two convolutional layers, two pooling layers, two normalization layers, and two fully connected layers. However, the first convolutional layer no longer uses a single-scale convolution kernel but uses a multi-scale convolution kernel with a convolution kernel size of 3×3 , 5×5 , and 7×7 , respectively. The convolution kernel size of the second convolutional layer is 3×3 .

Self-attention Multi-scale CNN: Based on the above model, the self-attention mechanism is introduced after the first convolutional layer to further extract more valuable features.

Transfer Network: Based on the above model, the network training is performed using the source domain data, waiting for the model to converge, and then directly using the model for fault diagnosis in the target domain.

Recurrent Neural Networks (RNNs): An RNN classical network with an input layer, hidden layer, and output layer set up to perform fault diagnosis and identification using vibration data from underwater vehicles.

Long Short-Term Memory (LSTM): Based on the RNN network, it changes the way data are transmitted internally by adding input and output gates as well as forgetting gates. This network is used to perform fault diagnosis on the actuators of the underwater vehicle.

4.3. Experimental Results

The previous section introduces the acquisition of datasets, naming of data files, data enhancement operations, and partitioning of meta-training datasets and meta-test datasets using the experimental platform. Then, the meta-training dataset is used to iteratively train the network to converge the model, and the computer software and hardware information used are shown in Table 4. The model is fine-tuned using a support set of meta-test data to fit the parameters to the optimal parameters used for the current task. The 100 diagnostic experiments were performed on the query set, mainly to exclude the chance to ensure the generality of the proposed network. The experimental results will be described in detail in Sections 4.3.1 and 4.3.2.

Table 4. Main parameters of software and hardware for this study.

CPU	GPU	Memory	Operating System	Deep Learning Architecture
Intel i5-7300HQ	GTX3090	24 GB	Ubuntu 18.04 LTS	Torch 1.11

4.3.1. Case I

The initial learning rate was discussed and validated by pre-experiments in the diagnostic experiments for Case 1, and the validation results are displayed in Figure 11. Combining the aforementioned findings, 4×10^{-3} and 3×10^{-3} were chosen as the initial meta-learning rate and initial base learning rate, respectively. The proposed model converged after 1000 iterations on the training dataset. The model was then fine-tuned 100 times on the support set of the test dataset, and the loss variation curves of the training and testing processes are shown in Figure 12. A trained model was obtained and tested using the query set of the test set after the fine-tuning of training and testing. From the test results in Figure 13, it can be seen that there are 600 test samples in total, and the number of correctly classified samples is 588, with an accuracy rate of 98%. Among them, the classification accuracy of the six commands 5° , -10° , -15° , -20° , -25° , and -30° was 100%, which produced positive results.

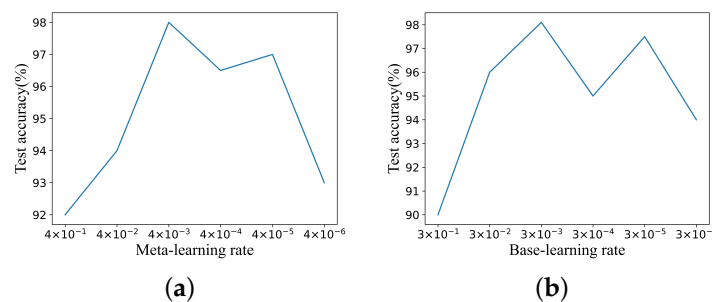


Figure 11. Case 1 learning rate discussion validates experimental results. (a) Meta-learning rate discussion. (b) Base-learning rate discussion.

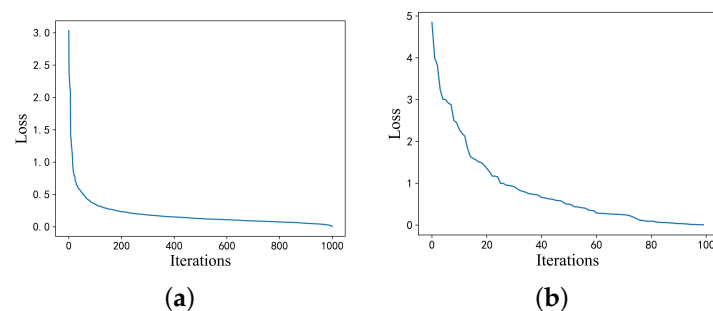


Figure 12. Case 1 loss attenuation curve. (a) Iterative training loss attenuation curve. (b) Iterative test loss attenuation curve.

To demonstrate the superiority of the proposed method, six ablation experiments were conducted to classify and identify the vibration data of the underwater vehicle, with the thruster data being the source domain and the rudder data being the target domain. The specific results are shown in Figure 14, which indicates that the proposed method achieves a significant advantage over the other methods. In addressing time-series data, our proposed method outperforms RNN and LSTM, while the recognition ability of the other six methods improves as the number of samples per class increases, but they are still significantly lower than the recognition ability of our proposed method.

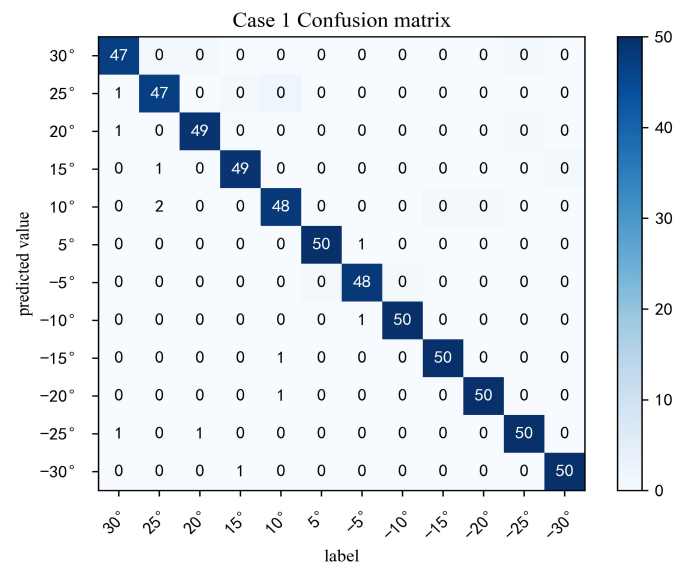


Figure 13. Test confusion matrix for Case 1.

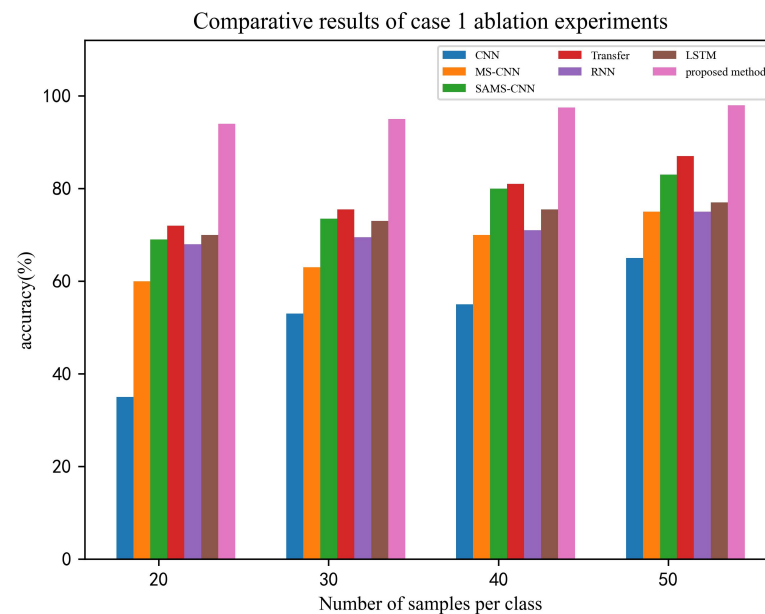


Figure 14. Results of ablation experiment in case 1.

4.3.2. Case II

The initial learning rate is discussed and investigated and experimentally validated in the diagnostic experiment of Case 2. According to the results in Figure 15, 4×10^{-4} , and 2×10^{-3} were chosen as the initial meta-learning rate and initial base learning rate, respectively. The loss variation curves of the training process and testing process of the model are shown in Figure 16. A trained network is obtained after training and testing fine-tuning, which is tested using the query set of the test set. From the test results of Figure 17, it can be seen that there are 1000 test samples in total, and the number of correctly classified samples is 979, with an accuracy rate of 97.9%. Among them, the classification accuracy of the eight commands -0.5 v, -0.3 v, -0.1 v, 0.4 v, 0.5 v, 0.6 v, 0.7 v, and 0.8 vs. is 100%, which also produced positive results.

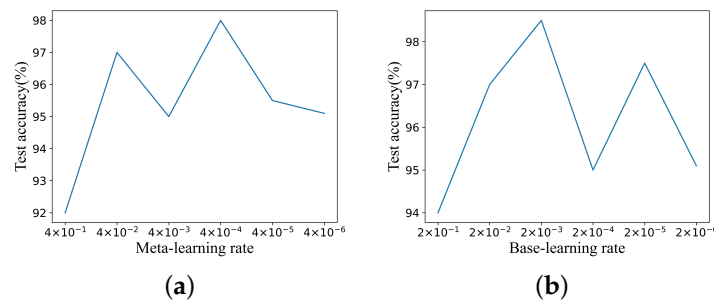


Figure 15. Case 2 learning rate discussion validates experimental results. (a) Meta-learning rate discussion. (b) Base-learning rate discussion.

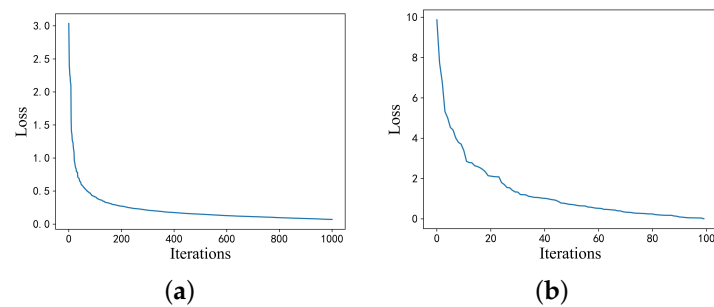


Figure 16. Case 2 loss attenuation curve. (a) Iterative training loss attenuation curve. (b) Iterative test loss attenuation curve.

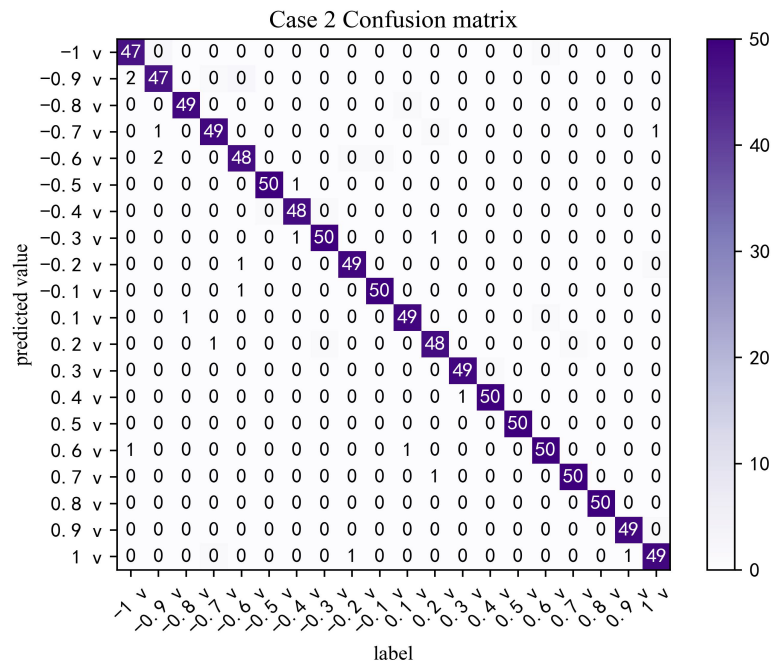


Figure 17. Test confusion matrix for Case 2.

The proposed method's superiority is demonstrated through six ablation experiments, which aim to classify and identify the vibration data of the AUV, with the source domain being the rudder data and the target being the thruster data. Comparative results of the seven experiments are illustrated in Figure 18, where the proposed method outperforms other methods with a significant advantage. Moreover, the proposed method exhibits better performance in addressing the time-series problem, outperforming RNN and LSTM. Although the recognition ability of the other six methods improves as the number of samples per class increases, they are still significantly lower than the proposed method's

recognition ability. Notably, the proposed method displays less sensitivity to the number of samples, suggesting that it has a distinct advantage in the few-shot problem.

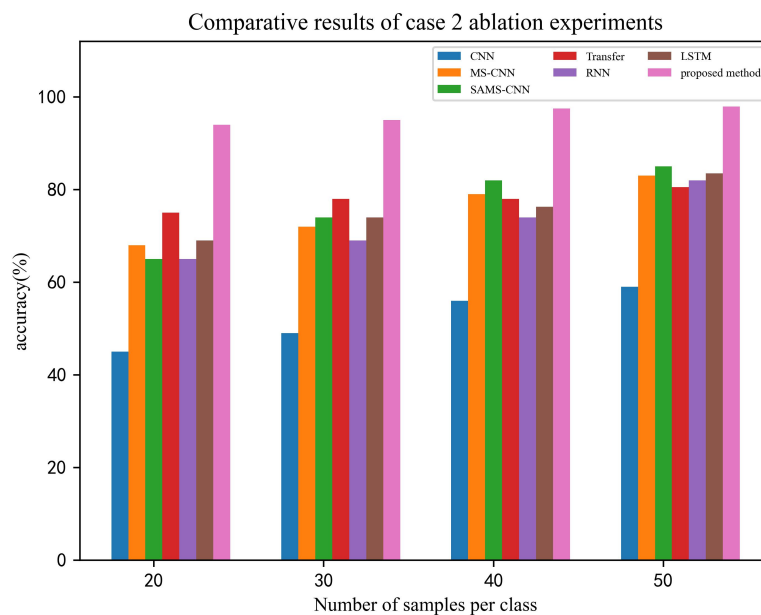


Figure 18. Results of ablation experiment in case 2.

4.4. Discussion of Model Convergence Speed

To demonstrate the rapid convergence property of the proposed method, we compared it with the transfer network, RNN, and LSTM architectures set up in Section 4.2. All variables, except for the network model, were kept consistent and convergence time was utilized as the performance metric for comparison. We conducted the experiments on a computer with the same hardware configuration and used the thruster data as the source domain data and the rudder data as the target domain data. The specific results are shown in Figure 19.

Based on the experimental results, it is evident that the proposed method requires the shortest amount of time for each training round, while the LSTM model takes the longest time. This can be attributed to the fact that the proposed method is a shallow model with fewer training parameters, which allows it to learn the optimal parameters more rapidly. Moreover, as highlighted in Section 4.3, the proposed method exhibits fast convergence and superior recognition performance.

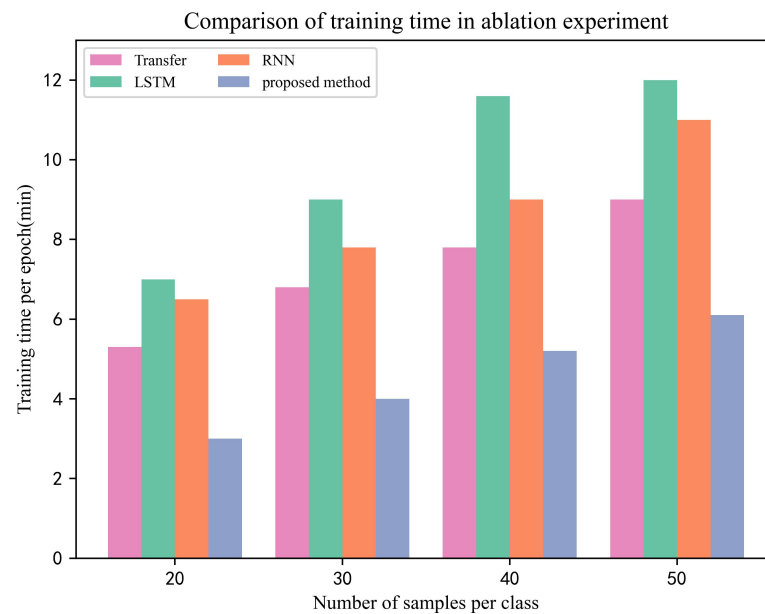


Figure 19. Comparison of convergence speed of different models.

5. Conclusions

In this paper, we propose the MSAMS-CNN fault diagnosis method, which can accomplish the fault diagnosis task of autonomous underwater vehicle actuators. Both self-attentive mechanisms and multi-scale approaches are used in extracting vibration data features to ensure that the extracted features can characterize the actuator fault states. Additionally, we utilize a meta-learning training and testing strategy to fit and evaluate the proposed model, effectively overcoming the issue of a limited AUV actuator dataset. Furthermore, the convergence speed of our method is faster in comparison to other approaches. The trained model demonstrates favorable generalization ability and is proficient in accomplishing cross-target fault diagnosis tasks. Experimental results show that the method outperforms the set up ablation experimental model in terms of diagnostic accuracy. In the future, the network needs to be further investigated and improved to enhance the model fitting speed and accuracy and to reduce the dependence on the computer hardware configuration.

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

Funding: This work is supported in part by the National Key Research and Development Program: 2021YFC2803000, in part by the Fundamental Research Funds for the Central Universities under Grant of 3102021HHZY030007, in part by the National Natural Science Foundation of China under Grant of 51979228 and 52102469 and in part by the National Basic Scientific Research Program under Grant of JCKY2019207A019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon reasonable request from the corresponding author.

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

Abbreviations

MSAMS-CNN	Meta-self-attention multi-scale convolutional neural network
CNN	Convolutional neural network
AUVs	Autonomous underwater vehicles
DNN	Deep neural network
DAE	Denosing autoencoder

References

1. Fotuhi, M.J.; Bingul, Z. Fuzzy torque trajectory control of a rotary series elastic actuator with nonlinear friction compensation. *ISA Trans.* **2021**, *115*, 206–217. [[CrossRef](#)] [[PubMed](#)]
2. Sun, Y.; Wang, Z.; Zhang, G. Fault diagnosis method of autonomous underwater vehicle based on deep learning. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *470*, 012035. [[CrossRef](#)]
3. Liu, T.; Hu, Y.; Xu, H. Deep reinforcement learning for vectored thruster autonomous underwater vehicle control. *Complexity* **2021**, *2021*, 1–25. [[CrossRef](#)]
4. Tsai, C.M.; Wang, C.S.; Chung, Y.J.; Sun, Y.D.; Perng, J.W. Multi-Sensor Fault Diagnosis of Underwater Thruster Propeller Based on Deep Learning. *Sensors* **2021**, *21*, 7187. [[CrossRef](#)]
5. Chu, Z.; Li, Z.; Gu, Z.; Chen, Y.; Zhang, M. The title of the cited article. *Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ.* **2022**, *2022*, 14750902221095423.
6. Zhu, D.; Jiang, Y. Thruster fault diagnosis in autonomous underwater vehicle based on Bayesian network. In Proceedings of the 2021 International Conference on Electrical, Communication, and Computer Engineering, Kuala Lumpur, Malaysia, 12–13 June 2021; pp. 1–5.
7. Zhu, D.; Cheng, X.; Yang, L.; Chen, Y.; Yang, S.X. Information fusion fault diagnosis method for deep-sea human occupied vehicle thruster based on deep belief network. *IEEE Trans. Cybern.* **2021**, *52*, 9414–9427. [[CrossRef](#)]
8. Talebi, H.A.; Khorasani, K. A neural network-based multiplicative actuator fault detection and isolation of nonlinear systems. *IEEE Trans. Control. Syst. Technol.* **2012**, *21*, 842–851. [[CrossRef](#)]
9. Tian, Q.; Wang, T.; Liu, B.; Ran, G. Thruster fault diagnostics and fault tolerant control for the autonomous underwater vehicle with ocean currents. *Machines* **2022**, *10*, 582. [[CrossRef](#)]
10. Omerdic, E.; Roberts, G. Thruster fault diagnosis and accommodation for open-frame underwater vehicles. *Control. Eng. Pract.* **2004**, *12*, 1575–1598. [[CrossRef](#)]
11. Raanan, B.Y.; Bellingham, J.; Zhang, Y.; Kemp, M.; Kieft, B.; Singh, H.; Girdhar, Y. Detection of unanticipated faults for autonomous underwater vehicles using online topic models. *J. Field Robot.* **2018**, *35*, 705–716. [[CrossRef](#)]
12. Ji, D.; Yao, X.; Li, S.; Tang, Y.; Tian, Y. Model-free fault diagnosis for autonomous underwater vehicles using sequence convolutional neural network. *Ocean Eng.* **2021**, *232*, 108874. [[CrossRef](#)]
13. Yeo, S.J.; Choi, W.S.; Hong, S.Y.; Song, J.H. Enhanced convolutional neural network for in Situ AUV thruster health monitoring using acoustic signals. *Sensors* **2022**, *22*, 7073. [[CrossRef](#)] [[PubMed](#)]
14. Tsai, C.M.; Wang, C.S.; Chung, Y.J.; Sun, Y.D.; Perng, J.W. Multi-sensor Fusion Time-Frequency Analysis of Thruster Blade Fault Diagnosis Based on Deep Learning. *IEEE Sens. J.* **2022**, *22*, 19761–19771. [[CrossRef](#)]
15. Kim, M.; Cho, H.; Choo, K.B.; Huang, J.; Jung, D.W.; Park, J.H.; Lee, J.H.; Jeong, S.K.; Ji, D.H.; Choi, H.S. Design of Underwater Thruster Fault Detection Model Based on Vibration Sensor Data: Generative Adversarial Network-based Fault Data Expansion Approach for Data Imbalance. *Sens. Mater.* **2022**, *34*, 3213–3227. [[CrossRef](#)]
16. Bingul, Z.; Gul, K. Intelligent-PID with PD Feedforward Trajectory Tracking Control of an Autonomous Underwater Vehicle. *Machines* **2023**, *11*, 300. [[CrossRef](#)]
17. Wang, W.; Yang, R.; Guo, C.; Qin, H. CNN-based hybrid optimization for anomaly detection of rudder system. *IEEE Access* **2021**, *9*, 121845–121858. [[CrossRef](#)]
18. Li, L.; Yang, R.; Guo, C.; Ge, S.; Chang, B. The data learning and anomaly detection based on the rudder system testing facility. *Measurement* **2020**, *152*, 107324. [[CrossRef](#)]
19. Li, L.; Yang, R.; Guo, C.; Ge, S.; Chang, B. A novel application of intelligent algorithms in fault detection of rudder system. *IEEE Access* **2019**, *7*, 170658–170667. [[CrossRef](#)]
20. Zhou, Y.; Sun, M.; Hao, M.; Chen, Z. Rudder Health Monitoring and Data Visualization Based on Feature Extraction. In Proceedings of the 2021 IEEE 10th Data Driven Control and Learning Systems Conference, Suzhou, China, 14–16 May 2021; pp. 625–630.
21. Ren, H.; Guo, C.; Yang, R.; Wang, S. Fault diagnosis of electric rudder based on self-organizing differential hybrid biogeography algorithm optimized neural net-work. *Measurement* **2023**, *208*, 112355. [[CrossRef](#)]
22. Qin, H.; Yang, R.; Guo, C.; Wang, W. Fault diagnosis of electric rudder system using PSOFOA-BP neural network. *Measurement* **2021**, *186*, 110058. [[CrossRef](#)]
23. Chang, B.; Yang, R.; Guo, C.; Ge, S.; Li, L. A new application of optimized random forest algorithms in intelligent fault location of rudders. *IEEE Access* **2019**, *7*, 94276–94283. [[CrossRef](#)]

24. Xu, Q.N.; Zhou, H.; Yu, F.; Wei, X.Q.; Yang, H.Y. Effective model based fault detection scheme for rudder servo system. *J. Cent. South Univ.* **2014**, *21*, 4172–4183. [[CrossRef](#)]
25. Chang, B.; Yang, R.; Guo, C.; Ge, S.; Li, L. Performance evaluation and prediction of rudders based on machine learning technology. *Proc. Inst. Mech. Eng. Part J. Aerosp. Eng.* **2019**, *233*, 5746–5757. [[CrossRef](#)]
26. Jiang, Y.; He, B.; Lv, P.; Guo, J.; Wan, J.; Feng, C.; Yu, F. Actuator fault diagnosis in autonomous underwater vehicle based on principal component analysis. In Proceedings of the 2019 IEEE Underwater Technology, Kaohsiung, Taiwan, 16–19 April 2019; pp. 1–5.
27. Wang, X.; Sun, H.; Lan, X. Fault diagnosis research of UUV thruster based on sliding window and convolutional neural network. In Proceedings of the 2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications, Dalian, China, 20–21 August 2022; pp. 1–7.
28. Jia, Z.; Liu, Z.; Cai, Y. A novel fault diagnosis method for aircraft actuator based on ensemble model. *Measurement* **2021**, *176*, 109235. [[CrossRef](#)]
29. Ding, P.; Jia, M.; Zhao, X. Meta deep learning based rotating machinery health prognostics toward few-shot prognostics. *Appl. Soft Comput.* **2021**, *104*, 107211. [[CrossRef](#)]
30. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.* **2020**, *53*, 1–34. [[CrossRef](#)]
31. Hutter, F.; Kotthoff, L.; Vanschoren, J. *Automated Machine Learning: Methods, Systems, Challenges*; Springer Nature: Berlin, Germany, 2019.

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