Hindawi
Shock and Vibration
Volume 2023, Article ID 1639287, 21 pages
https://doi.org/10.1155/2023/1639287

Research Article

Multiscale Time-Frequency Sparse Transformer Based on Partly Interpretable Method for Bearing Fault Diagnosis

Shouquan Che,[1] **Jianfeng Lu**,[2] **Congwang Bao**,[1,3] **Caihong Zhang**,[1] **and Yongzhi Liu**[1]

[1]*College of Mining and Mechanical Engineering, Liupanshui Normal University, Liupanshui 553000, China*
[2]*College of Mechanical Engineering, Guizhou University, Guiyang 550025, China*
[3]*College of Mechanical Engineering, China University of Mining and Technology, Xuzhou 100083, China*

Correspondence should be addressed to Shouquan Che; chesq_njtu@163.com

Received 11 June 2023; Revised 6 July 2023; Accepted 15 July 2023; Published 4 August 2023

Academic Editor: Xingxing Jiang

Transformer model is being gradually studied and applied in bearing fault diagnosis tasks, which can overcome the feature extraction defects caused by long-term dependencies in convolution neural network (CNN) and recurrent neural network (RNN). To optimize the structure of existing transformer-like methods and improve the diagnostic accuracy, we proposed a novel method based on the multiscale time-frequency sparse transformer (MTFST) in this paper. First, a novel tokenizer based on shot-time Fourier transform (STFT) is designed, which processes the 1D format raw signals into 2D format discrete time-frequency sequences in the embedding space. Second, a sparse self-attention mechanism is designed to eliminate the feature mapping defect in naive self-attention mechanism. Then, the novel encoder-decoder structure is presented, the multiple encoders are employed to extract the hidden feature of different time-frequency sequences obtained by STFT with different window widths, and the decoder is used to remap the deep information and connect to the classifier for discriminating fault types. The proposed method is tested in the XJTU-SY bearing dataset and self-made experiment rig dataset, and the following work is conducted. The influences of hyperparameters on diagnosis accuracy and number of parameters are analysed in detail. The weights of the attention mechanism (AM) are visualized and analysed to study the interpretability, which explains the partly working pattern of the network. In the comparison test with other existing CNN, RNN, and transformer models, the diagnosis accuracy of different methods is statistically analysed, feature vectors are presented via the t-distributed stochastic neighbor embedding (t-SNE) method, and the proposed MTFST obtains the best accuracy and feature distribution form. The results demonstrate the effectiveness and superiority of the proposed method in bearing fault diagnosis.

1. Introduction

The rotating machinery plays a pivotal role in modern industrial systems, which are widely used in aerospace engineering, motor industry, manufacturing industry, and other important fields [1]. Bearing, as the core component of rotating machinery, its failure mechanism, especially the monitoring and identification of the faults, has become a research hotspot. The study of compact and effective online condition monitoring and fault diagnosis method is essential and necessary for the operation of complex mechanical systems [2, 3].

Generally, bearing faults diagnosis approaches consist of two categories: model-based [4] and data-driven methods [5]. Model-based methods established fault feature detection and classification model through a large amount of prior knowledge, but the diagnosis accuracy is not satisfactory under complex conditions. Data-driven methods aim to establish complex nonlinear projection relationships between the sensor data and fault types, and they are becoming more and more attractive with the development of big data and the various bearing fault diagnosis algorithms in machine learning (ML). Currently, the most common ML methods utilized in the bearing fault diagnosis include K-nearest neighbor (KNN), support vector machine (SVM) [6], multilayer perceptron (MLP) [7], hidden Markov model (HMM) [8], and variational mode decomposition (VMD) [9, 10]. However, the traditional ML methods can no longer meet the requirements due to its shallow feature extraction and presentation framework. Recently, deep learning (DL) has achieved great success in bearing fault diagnosis owing to its strong model-fitting ability and generalization ability.

On the other hand, deep learning network can conveniently stack and combine learning layers to handle the diagnosis under different equipment and work conditions. Commonly, deep learning approach includes auto-encoder (AE) [11], deep belief network (DBN) [12], convolutional neural network (CNN), and recurrent neural network (RNN). Among these methods, CNNs have attracted more researchers' attention because it is more suitable for processing periodic signals and have a stronger ability to learn features from mechanical vibration signals [13]. The CNN-based frameworks extract and connect local features of interspaces by sharing convolutional kernels in the deep layers, which guarantees the effectiveness of bearing fault diagnosis. Gao et al. [14] proposed a method based on parameter optimization maximum correlated kurtosis deconvolution (MCKD) and CNN for bearing fault diagnosis, and MCKD is used to filter and denoise the raw signals and then input the results to the CNN model for fault classification. Liu et al. [15] proposed a two-stage framework for rolling bearing fault severity recognition via data mining integrated with CNN, which introduced matrix profile (MP) to mine the impulse from the raw vibration signals and then conducted a CNN that combined with *softmax* regression for fault recognition. The current relevant works of CNN are carried out in the direction of model structure optimization and combination with traditional ML methods. Researchers attempt to learn more effective features with a more compact and effective structure to avoid problems such as gradient failure in the algorithms [16]. For instance, Wang et al. [17] proposed a squeeze-and-excitation-enabled CNN (SECNN) that can assign a certain weight to each channel and enforce the model focusing on the major features. Xu et al. [18] combined the variational mode decomposition (VMD) method and a deep CNN to develop a bearings fault classification network.

As an effective model in sequence data processing, the RNN network is widely used in bearing fault diagnosis. Researchers proposed the gated recurrent unit (GRU) and long short-term memory unit (LSTM) to solve the problems such as long-term dependencies and gradient vanishing in the vanilla RNN model. The improved RNN models achieve more attractive results than the baseline approach. An et al. [19] employed an RNN framework with LSTM by the idea of an infinitesimal method to realize the intelligent fault diagnosis under time-varying working conditions. Zhang et al. [20] proposed a method based on RNN with GRU and MLP to implement fault recognition, which achieves excellent diagnosis results and exhibits the robustness against the noise. Zhao et al. [21] proposed a complex deep learning model by combing CNN and LSTM, which is denoted as a bidirectional long short-term memory network (CBLSTM). CBLSTM adopted CNN to learn local features and then input the results into a bidirectional LSTM to extract global features. The emerging bearing diagnosis methods based on CNN and RNN continue to mature. However, there are still some inherent defects such as information loss, the receptive filed is too small, and the lack of long-term dependencies in CNN and RNN.

Recently, attention mechanism (AM) is introduced to solve the problems mentioned previously. AM can associate different positions or channel features of a sequence and pay more attention to the informative data, which is designed as a component combined with CNN or RNN and widely applied in various tasks such as natural language processing (NLP), computer vision (CV), and fault diagnosis [22]. AM enhanced the performance of the backbone of CNN or RNN but failed to completely avoid the shortcoming of these classical models. Furthermore, in 2017, Vaswani et al. [23] came up with a new architecture called a transformer, which abandons all the convolutional and recurrent modules and is based only on the attention mechanism and fully connected layers. Transformer attained the best performance in the task of machine translation at that time. The framework BERT based on a transformer proposed by Devlin et al. [24], which is developed to generate word vectors, achieved excellent results in NLP tasks. In the field of NLP, transformer broke new ground and almost entirely replaced RNN at present. In 2021, a pioneering framework based on transformer-named vision transformer (ViT) [25] employed in computer vision (CV) has achieved encouraging performance in image classification tasks. The test results indicate that ViT outperforms other state-of-the-art methods in condition of pretraining on a larger dataset. Meanwhile, ViT showed a strong data extensibility. Its performance continues to improve even as the data amount and model scale increase. Furthermore, the powerful parallelism in the computing of ViT means a greater advantage in large-scale data processing. A variety of modified models that diverge around ViT have been proposed and achieved excellent performance in CV tasks, such as CrossViT [26] and PVT [27]. Clearly, transformer model has been an important branch of deep learning besides CNN and RNN.

The outstanding performance in encoding and extracting hidden features of sequences, which makes the transformer neural network, has been a promising method in the field of bearing fault diagnosis where the vibration data are the main judgment input. Ding et al. [28] proposed a transformer framework named TFT for bearing fault diagnosis, which designed a tokenizer and encoder module to extract abstractions from the input time-frequency representations (TFRs) of vibration signals. BAFT [29] proposed by Jiao et al. developed a partly interpretable network based on transformer and a binary arborescent filter to classify the bearings faults effectively and visually presented the partly hidden features inside the model, which achieved a superior performance and excellent antinoise validity. Jin et al. [30] proposed a time-series transformer (TST) to recognize the bearing fault modes, which designed a sequence generation method that handles raw vibration signals in a 1D format time series segment. The series is then input into the encoder of transformer to learn the features. The test results show that TST has a better fault identification capability than traditional CNN and RNN models. Du et al. [31] proposed a transformer-like framework for fault diagnosis under complex conditions, which extracted the features from the high-dimensional raw signals with noise by a stacked denoising auto-encoders (SDAE) module and obtained the target features by the self-attention mechanism of transformer deep neural network.

The most common data-driven works for bearing fault diagnosis are conducted through the analysis of vibration signals. However, the data that are input into the deep learning model are preprocessed by different approaches in different frameworks. The preprocess methods can be roughly divided into three categories: (1) sampling raw signals or its simple processing results [32]. In time-series transformer (TST) [30], the input vibration time series is trimmed into several subsequences with the given length. Huang et al. [16] proposed a work that applied maximum pooling and average pooling layers to extract different scale information as the input of AM module in transformer; (2) preprocessing by the feature-based model [33]. Du et al. [31] proposed a work that established a stacked denoising auto-encoder (SDAE) module to generate low-dimensional features of input signals. Jiao et al. [29] proposed a framework that developed a binary arborescent filter to extract the statistical feature and then input the encoder module of transformer network; (3) preprocessing by domain transformation. The time-series signals are transformed into frequency representation (FR) [34, 35] or time-frequency representation (TFR) [36, 37]. In TCN [38], the FR that is transformed by a fast Fourier transform (FFT) module from vibration signals is input into transformer network. In TFT [28], the input signals are first processed to 2D TFR by synchrosqueezed wavelet transform (SWT) and then flattened and mapped as the tokenizer of transformer module. In general, the methods based on domain transformation have better performance.

As mentioned previously, the transformer-like approaches have achieved excellent performance in bearing fault diagnosis due to the powerful modelling and feature extraction ability of the self-attention mechanism in transformer. However, there are some limitations in the existing transformer-like bearing fault diagnosis models:

(1) Almost all methods only use part components of the transformer framework, which weakens the model's ability to sequence information

(2) Ignoring the interference of secondary information in self-attention weights can reduce the performance for fault diagnosis of transformer

The motivation of this paper is to develop a new transformer-based method that can extract more effective hidden representations for bearing fault diagnosis in a simple and generalized way. The proposed new end-to-end approach named multiscale time-frequency sparse transformer (MTFST), which established the diagnosis model between the TFRs and bearing fault types. MTFST achieves good results in evaluation; furthermore, its superiority over the other deep learning models is proved on the test datasets. The main contributions of this paper are summarized as follows:

(1) The STFT method is employed to obtain the multiscale TFRs of raw vibration signals by varying the window width, and the novel tokenizer based on the different scale TFRs is designed to present the discriminant feature in multilevel.

(2) A sparse self-attention mechanism (SSAM) is studied to focus on the primary information of self-attention, enabling the hidden features to be more discriminative.

(3) A novel encoder-decoder structure is developed to extract the hidden features and long-term dependence of multiscale TFRs. The proposed framework is more compatible with the vanilla transformer than the existing models and better at fault diagnosis. And the visualization analysis of model weights solves the problem that traditional deep learning fails to interpret to some extent.

The rest of the paper is arranged as follows. The theoretical foundations of transformer are introduced in the second section. Structural framework and algorithmic flow of the proposed MTFST are introduced in the third section. The fourth section includes the introduction of the dataset, experiment setting, and the ablation analysis of hyperparameters, and bearing fault diagnosis results under two datasets are also evaluated and analysed in this section. The conclusions of the paper and future research plan are given out in the last section.

2. Preliminaries

This section will introduce the basic structure and core components of the vanilla transformer.

2.1. Transformer. The transformer framework was proposed by Vaswani et al. [23] to optimize the traditional patterns of Seq2Seq. The novel structure is entirely based on the attention mechanism to draw global dependencies between the input sequence and output results, which solves the problems of difficulty to model the global relationships between local information in traditional convolution operation. Furthermore, this model increased parallel efficiency and reduced computing consumption. The overall architecture of the transformer is shown in Figure 1, which consists of encoder and decoder modules, and those two components are stacked by multiple basic transformer blocks. The basic transformer blocks include multihead self-attention mechanism, position-wise feed-forward network, layer normalization module, and residual connector. The embedding layers before the encoder and decoder convert the one-hot tokens into a new tensor, and the tensor is added to a sinusoidal position encoding. The encoder in the transformer receives the input sequences, and the decoder remaps the output of encoder to obtain the results.

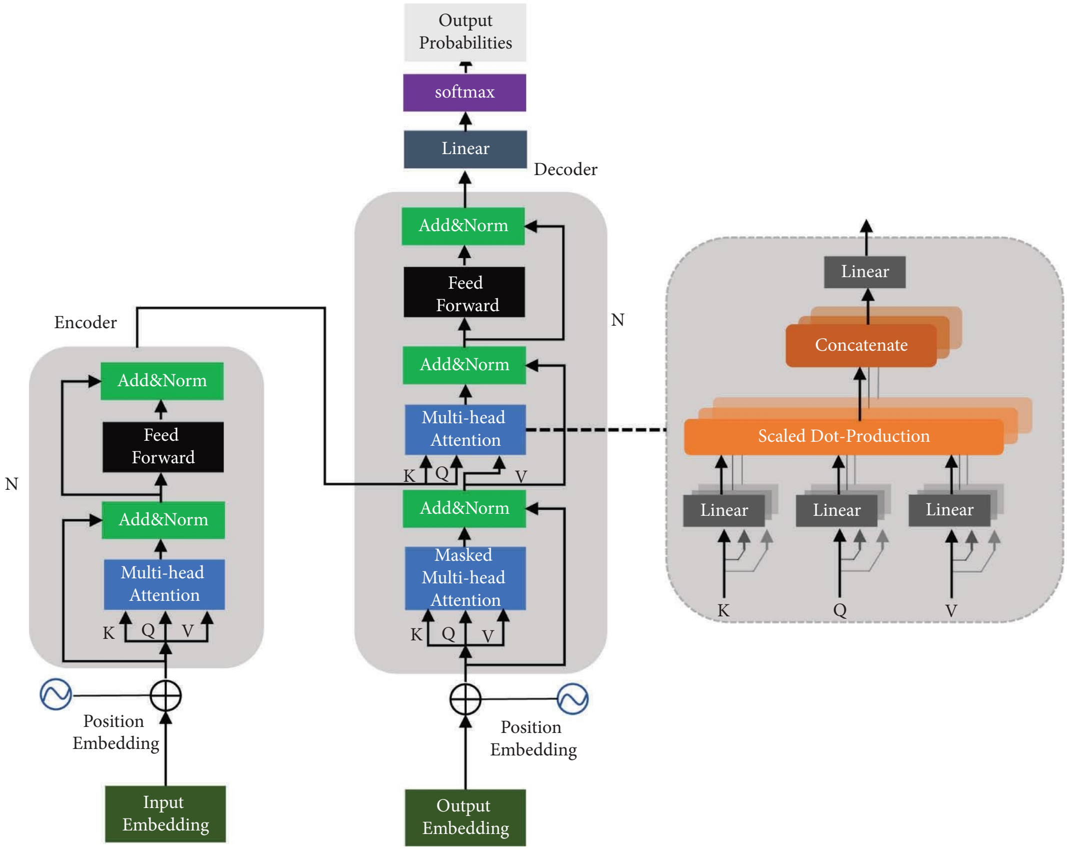


FIGURE 1: Architecture of the vanilla transformer.

In encoder, N blocks are stacked, and those blocks are taking same structure but different parameters. The block consists of two layers, and the first layer includes a multihead attention module and a residual network. The second layer includes a positionwise fully connected feed-forward network and a residual network. The output of each layer can be presented as follows:

$$X_o = \mathrm{LayerNorm}\left(X_i + F_{(A/FC)}(X_i)\right), \qquad (1)$$

where X_o and X_i denote the output and input of each layer, respectively. $F_{(A/FC)}$ denotes multihead self-attention or positionwise forward network. *LayerNorm* is the layer normalization.

There are three layers in decoder, and they consist of the basic transformer blocks presented before. The first layer is masked multihead attention that is used to extract the hidden feature of the input sequence with AM and attached mask coding, which can prevent label leakage in the Seq2Seq task [29]. The second layer is employed to map the output from the first layer of decoder and encoder. The positionwise forward network and residual operation are used in the third layer to extract the local and global deep information. Finally, the output of the decoder inputs into a linear layer and a *softmax* activation function to obtain the probabilities.

2.2. Multihead Self-Attention Mechanism. The multihead self-attention mechanism (MSA) is built based on self-attention mechanism, which is the core component of transformer model and employed to gather the information from input sequence and learn the hidden features. Self-attention mechanism can be regarded as a method that maps the different weight information of input sequence, which obtains the output from a query (Q), a set of key (K), and a value (V) vector. As shown in equation (2), the output of self-attention mechanism is a weighted sum of V, and the weight matrix is related to the dot product of K and V.

It can be seen that the self-attention extracts information in V based on the similarity between K and Q.

$$A_s(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \qquad (2)$$

where d_k denotes the scaled factor, it is the dimension of Q and K, and *softmax* denotes an activation function.

In order to obtain different subspaces of hidden information rather than only one nonlinear transformation result, the multihead self-attention is proposed to concatenate and map the input tokens' different projections that are parallel computed by multiple independent self-attention mechanisms. The calculation process is shown as follows:

$$F_A(X) = cocat\left(\text{head}_1, \ldots, \text{head}_n\right)W^O,$$
$$\text{where}\,\text{head}_i = A_s\left(XW_i^q, XW_i^k, XW_i^v\right), \tag{3}$$

where n denotes the number of heads that is the number of self-attention module. $W_i^q \in \mathbb{R}^{d_{\text{model}} * d_k}$, $W_i^k \in \mathbb{R}^{d_{\text{model}} * d_k}$, and $W_i^v \in \mathbb{R}^{d_{\text{model}} * d_v}$ denote the weight matrix of Q, K, and V in ith self-attention module, respectively, and $d_k = d_v = (d_{\text{model}}/n)$. X denotes the embeddings. *cocat* designed as a concatenate function. $W^O \in \mathbb{R}^{n*d_k * d_{\text{model}}}$ denotes the weight matrix of linear projection on concatenated multi-head.

2.3. Positionwise Forward Network. The feed-forward network is a fully connected layer, and it includes two linear transformations and a *ReLU* activation, which is expressed as follows:

$$F_{FC} = ReLU\left(0, xw_1 + b_1\right)w_2 + b_2, \tag{4}$$

where $w_1 \in \mathbb{R}^{d_{\text{model}} * d_{ff}}$, $b_1 \in \mathbb{R}^{d_{ff}}$, $w_2 \in \mathbb{R}^{d_{ff} * d_{\text{model}}}$, and $b_2 \in \mathbb{R}^{d_{\text{model}}}$ denote the weights and bias of two linear transformations, respectively.

2.4. The Transformer-Like Methods for Bearing Fault Diagnosis. The vanilla transformer employed an encoder-decoder structure to solve the Seq2Seq tasks as mentioned above. The encoder in the framework receives the embedding information for feature learning. The decoder is employed to generate a new sequence through the encoder output and the last layer's result of the decoder itself. Generally, transformers divided into three categories include (1) encoder-decoder (e.g., for Seq2Seq), encoder only (e.g., for classification), and decoder only (e.g., for language modelling) [28]. Existing transformer-like approaches usually adopt the encoder-only model for the fault diagnosis tasks, such as the TST [30], TFT [28], and BAFT [29] mentioned previously, the series are embedded to token sequence with class information and then input to the encoder, and the hidden features with category information are mapped by the classifier to obtain the fault types. In PRT [32], the framework adopts an enhanced encoder network that includes an embedding patch encoder and a class information encoder to learn the hidden features and dependencies. In the framework proposed by Du et al. [31], which remapped the forward eigenmatrix by a position transformation to obtain a backward eigenmatrix, two feature matrixes are input into the pair attention-mechanism neural networks to better learn the essential characteristics of the fault data. And the network is closer to the vanilla transformer in structure, but the different attention-based neural modules lack feature interaction. In essence, for this network, the encoder that is used to extract the features of the backward eigenmatrix can be considered as a learning enhancement module of another one.

3. Multiscale Time-Frequency Transformer

In this section, the proposed multiscale time-frequency transformer (MTFST) will be introduced in detail. The core components include tokenizer, encoder, decoder, and classifier.

3.1. Tokenizer

3.1.1. Raw Signal Preprocessing. Vibration signals that are sampled from sensors are 1D time series; in our work, the input raw signals will be processed to 2D format TFRs. Thus, a specific tokenizer based on shot-time Fourier transform (STFT) was designed. STFT is a domain transform method based on the windowed Fourier transform algorithm, which assumed that the signals to be processed are stationary for short intervals in the analysis window. By moving the window function along the time axis, STFT analyzes the signal segments to obtain the local spectrum [39]. STFT is defined as follows:

$$STFT(t, f) = \int_{-\infty}^{+\infty} x(\tau)h(\tau - t)e^{-j2\pi f\tau}\mathrm{d}\tau. \tag{5}$$

Given a certain TFR $x \in \mathbb{R}^{N_t * N_f}$, where N_t and N_f denote the length along the time and frequency axis, representing it as a patch sequences $[x_f^1, x_f^2, \ldots, x_f^{N_t}]$, where the subsequence $x_f^i \in \mathbb{R}^{N_f}$. For consistency of subsequent operations, the token embeddings are obtained by projecting the TFR sequences to another $x \in \mathbb{R}^{d_m * N_f}$ by a linear transformation. The process is expressed as follows:

$$x' = W_{emb}x, \tag{6}$$

where $x' \in \mathbb{R}^{d_{\text{model}} * N_f}$, which represents the learnable linear mapping of TFR along the time axis, and $W_{emb} \in \mathbb{R}^{d_{\text{model}} * N_t}$.

Finally, the TFRs are discretely represented as temporal sequences of the instantaneous frequency spectrum. As mentioned above, processing such a sequence is the strength of a transformer-like structure [28].

3.1.2. Position Encoding. The vanilla transformer framework designed position encoding to represent the relative or absolute position information of the embedding sequence.

There are two methods including 1D and 2D format position encoding [30], and the results in reference [25] show that the two methods have no significant performance gaps. In our proposed work, a kind of sinusoid encoding method was adopted only to mark the location information of the sequence, which is expressed as follows:

$$E_{pos}(pos, 2i) = \sin\left(\frac{pos}{1000^{(2i/N)}}\right),$$
$$E_{pos}(pos, 2i+1) = \cos\left(\frac{pos}{1000^{(2i/N)}}\right), \tag{7}$$

where *pos* denotes the position of the patch among the sequence. N and i denote the dimension of the position vector and the current dimension, respectively.

Finally, the tokens sequence is defined as

$$x_{seq} = W_{emb}\left[x_f^1, x_f^2, \ldots, x_f^{N_t}\right] + E_{pos}, \tag{8}$$

where $E_{pos} \in \mathbb{R}^{d_{\text{model}} * N_t}$.

On the other hand, there are two main ways to represent the deep features extracted from the tokens sequence, including obtaining the information of the last transformer layer or learning the feature by adding a class token into the sequence [30]. The comparison results in reference [40] show that the class tokens are nonessential. Therefore, the class tokens are abandoned, and another method of expressing characteristics is employed in our work.

3.2. Sparse Self-Attention Mechanism. Sparse self-attention mechanism (SSAM) is designed to eliminate the reduction of feature discrimination caused by focusing on the secondary information. Inspired by the authors in reference [41], in the SSAM, each attention feature of TFRs is determined by the top P input information that is most similar to it, which differs from the naive SAM that calculates features by all input information. As shown in the middle of Figure 2, the similarities of input K and Q are calculated first, and the indices matrix of P largest elements are selected to mask the *softmax* results. The calculation examples are shown in right in Figure 2. The sparse self-attention is defined as follows to replace equation (2).

$$A_s(Q, K, V) = \text{mask}\left(\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V\right). \tag{9}$$

And the sparse ratio r_s is defined as follows:

$$r_s = \frac{P}{W * H}, \tag{10}$$

where W and H denote the size of the attention weight matrix.

3.3. Encoder and Decoder. The proposed MTFST employed three encoders to extract the multiple perspective deep features from the multiscale TFR embeddings, and the encoders share the same structure as the one in vanilla transformer described in Section 2 but are different in parameters. Encoder consists of the basic blocks that include a multihead self-attention module, feed-forward layer, and normalization layer with residual connector.

The decoder is used to extract the dependencies and fuse the corresponding information from the outputs of the different encoders. The structure of the proposed decoder is different from the vanilla transformer and similar to the encoder. Note that the decoder takes the output of different encoders as the input.

3.4. Training of MTFST. As a general deep learning scheme, the proposed MTFST framework adopts labelled fault datasets for supervisory training, and an error back-propagation (BP) algorithm is employed to minimize the loss. For a given training dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, which contains n samples, the loss function is defined as follows:

$$\mathcal{J}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{C-E}(\widehat{y}_i, y_i), \tag{11}$$

where $\widehat{y}_i$ and y_i denote the prediction output and the ground truth of sample x_i, respectively. θ denotes the trainable parameters of the network, and $\mathcal{L}_{C-E}$ presents the cross-entropy loss function.

Additionally, the Adam optimizer [42] is employed to train MTFST, which adopts an adaptive and exponential smoothing gradient strategy to accelerate the loss convergence. As similar with the vanilla transformer, the dropout training manner [43] is employed in the network, which randomly masks the connections of some neurons to reduce overfitting. Algorithm 1 shows the training step of MTFST, and the architecture is shown in Figure 3.

4. Case Study and Analysis

In this section, two case studies are implemented to verify and analyze the effectiveness of the proposed MTFST in rolling bearing fault diagnosis. The data collected from XJTU-SY open dataset and self-made mine motor traction dataset are employed for testing and comparison with other state-of-the-art methods. All the validation experiments are conducted on a computer with a Intel 10700F CPU, a NVIDIA RTX 3080 GPU with 32GB RAM. Besides, Ubuntu 18.06, Python 3.6, TensorFlow 2.6, and CUDA 11.02 are adopted for the whole network construction.

4.1. Case 1: XJTU-SY Bearing Dataset

4.1.1. Dataset Description and Experiment Settings. XJTU-SY bearing datasets are provided by Xi'an Jiaotong University (XJTU) and the Changxing Sumyoung Technology Co., Ltd. (SY). The datasets contain complete run-to-failure data of 15 rolling element bearings that were acquired by conducting many accelerated degradation experiments [44]. The testbed of rolling element bearings is shown in Figure 4, and the vibration signals collected by 5 bearings under three operating conditions include (1) 2100 rpm (35 Hz) and load rating of 12 kN; (2) 2250 rpm (37.5 Hz) and 11 kN; (3) 2400 rpm (40 Hz) and 10 kN. The sampling frequency is set to 25.6 kHz, and a total of 32768 points are recoded for each sampling. In our work, the recoded data of Bearing 2_1, Bearing 2_5, Bearing 3_3, Bearing 3_4, Bearing 1_4, Bearing 2_3, Bearing 3_1, Bearing 3_4, and Bearing 3_2 are employed in the experiment, which includes four fault types: inner race (IR), cage, out race (OR), and inner race, ball, cage and out race (IBCO). In addition, the batch size is set to 40, and the Hanning window is used for STFT.

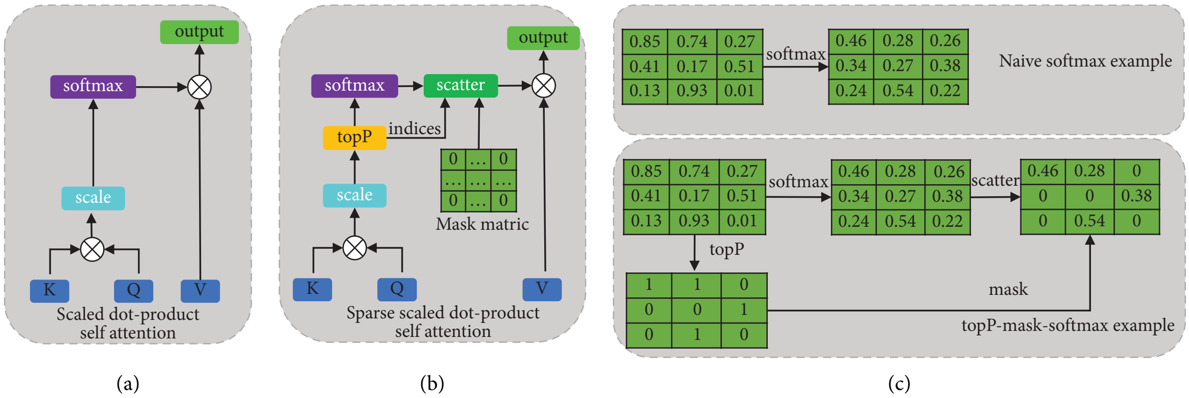


Figure 2: Self-attention mechanism. (a) The scaled dot-product self-attention used in MSA, (b) the sparse scaled dot-product self-attention used in MTFST, and (c) the examples of different attention mechanisms.

4.1.2. Ablation Study. In this section, we will discuss the influences of the hyperparameters settings on the diagnosis performance of the proposed model. The hyperparameters contain three STFT window widths w_1, w_2, w_3 for obtaining multiscale TFRs, token embedding dimension d_{model}, and also indicate self-nonlinear transformation dimension of self-attention module, hidden layer dimension d_{ff} of feed-forward network, number of attention head n, block number of encoder N_e, block number of decoder N_d, dropout rate r_d, and sparse ration of sparse self-attentionr_s. Each model with different parameters trained for 5 runs and the test performance is displayed in Table 1, where the baseline row denotes the model used in the following experiments. It can be seen from the table that the STFT window width can significantly affect the performance of MTFST. Excessive emphasis on the precise scale of the TFRs in time or frequency will reduce the performance of the model. It is worth noting that the input order of the TFRs with different window scale can greatly affect the model's performance.

The test shows that the TFR features corresponding to small window width have better effectiveness as the input of Q in decoder. Dimension d_{model} and d_{ff} can obviously affect the number of parameters in the network. There is a consistent trend that a number of attention head and block number of encoder and decoder have a strong impact on the model performance, which means too small number leads to learning insufficient features, while overfitting is occurred in a too large value. The appropriate setting value of sparse ratio r_s can avoid the interference of secondary features and effectively improve model performance, while overfitting has occurred in a too small r_s. The corresponding accuracy statistics under different hyperparameters are shown in Figure 5.

4.1.3. Diagnosis Results Based on MTFST. In this section, the model established based on the hyperparameters selected in the previous section is trained and tested on the XJTU-SY bearing dataset. The dataset is divided randomly to 80% train data and 20% test data. In each batch, the data with different fault labels used for training and testing are evenly distributed but randomly shuffled, and the data in the fault datasets with small samples were repeatedly used during training. The proposed MTFST is trained 30 epochs to learn a robust diagnosis model, and the training process is repeated 10 runs under the same condition to eliminate the effects of the random initialization.

The variation of loss and accuracy under the XJTU-SY dataset during the training process is shown in Figure 6. It can be seen from the boxplot results of the training set that some of the wobbles occurred in both loss function value and diagnosis accuracy in the early training stages, while the performance improves significantly after 5 epochs and becomes stable after 20 epochs. It indicates the good convergence of the model under the strategy of gradient back-propagation. With the iteration of training, the performance in the test set is improving. Although some fluctuations still occurred in the accuracy, there is no great gap between the top accuracy and minimum accuracy, and the high and stable average accuracy presents the effectiveness of MTFST. These analysis results indicate that the proposed MTFST has strong and robust model fitting ability and generalization. To further analyze the model performance, the fault diagnosis of the confusion matrix of top accuracy and minimum accuracy is presented in Figure 7, which is sorted from the results of 10 rounds of repetitive training process. The rows denote the ground truth of the samples, and the columns represent the predicted fault labels of the MTFST.

4.1.4. Visualization of Network. In this section, first, the attention weights are visualized to attain a further understanding of how MTFST works. Instead of the class token in conventional transformer architecture, the deep hidden features extracted by the attention mechanism are mapped directly to the diagnosis results in MTFST. Thus, the attention weights could reflect the relationships of the deep TFRs patches in each attention mechanism-based layer; furthermore, these relationships can represent which features are considered valid and which are redundant. The attention weights, i.e., the results of $softmax((QK^T)/\sqrt{d_k})$, are calculated and concatenated in the multihead self-attention network, and the weight matrixes are averaged to show the attention level. We list the partly weights

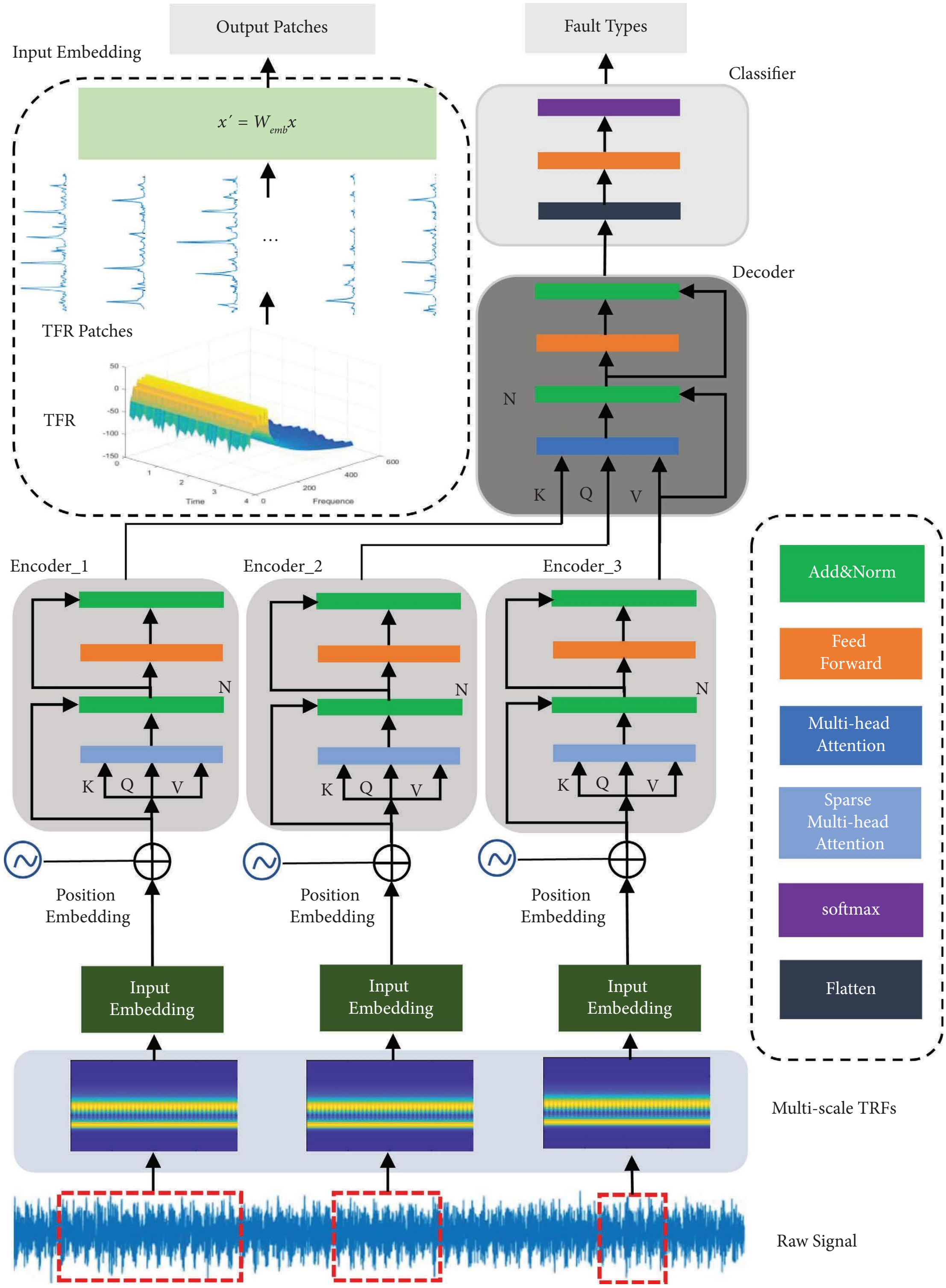


Figure 3: The structure of the proposed MTFST.

representation in encoders and decoders of different fault labels in Figure 8. As seen from all the first layers of encoders, there is sparse attention in certain areas and little attention weights between patches. However, the network gradually assigns more attention weights to patches with significant characteristics layer by layer. In the last layer, there are strong weights between different patches and the attention focus on fixed regional deep features. Furthermore, the encoders with different scale TFRs input work into distinctive areas to grasp the complementary information. There are the same trends regardless of the different labels that the tokens around 40 in encoder1, tokens between 1 and

```
Input: Three multiscale TFRs $\mathscr{X}_{s1} = \{x_i, y_i\}_{i=1}^{n}$, $\mathscr{X}_{s2} = \{x_j, y_j\}_{j=1}^{n}$, $\mathscr{X}_{s3} = \{x_k, y_k\}_{k=1}^{n}$ where $x_i \in \mathbb{R}^{N_t^i * N_f^i}$, $x_j \in \mathbb{R}^{N_t^j * N_f^j}$, $x_k \in \mathbb{R}^{N_t^k * N_f^k}$, and $y_i = y_j = y_k$, which denote the fault types.
Set training batch $N_b$, training epoch $max\_epoch$, token embedding dimension $d_m$, self-attention weight matrix size $d_{\text{model}}$, number of head $n$, positionwise forward network weight matrix size $d_{ff}$, block number of encoder $N_e$, block number of decoder $N_d$, and number of fault types $N_f$.
Initialize trainable parameters $\{W, b\}$ of MSTFT
for $epoch$ in 1, 2, ..., $max\_epoch$ do
  for $step$ in 1, 2, ..., $max\_step$ do
    //Tokenizer
    for each $x_i$ in $\{x_i\}_{i=1}^{N_b}$, $x_j$ in $\{x_j\}_{j=1}^{N_b}$ and $x_k$ in $\{x_k\}_{k=1}^{N_b}$ do
      Reshape $x_i$, $x_j$,$x_k$ to $x_i' = W_{emb}^{i} x_i$, $x_j' = W_{emb}^{j} x_j$ and $x_k' = W_{emb}^{k} x_k$ then slice into patches sequence $[x_i^{',1}, x_i^{',2}, \ldots, x_i^{',N_t}]$, $[x_j^{',1}, x_j^{',2}, \ldots, x_j^{',N_t}]$, $[x_k^{',1}, x_k^{',2}, \ldots, x_k^{',N_t}]$;
      Add position encoding, obtain $x_{seq}^{i} = x_i' + E_{pos}^{i}$, $x_{seq}^{j} = x_j' + E_{pos}^{j}$, $x_{seq}^{k} = x_k' + E_{pos}^{k}$;
    end Stack batches, obtain sequences $X_0^i$, $X_0^j$, $X_0^k$.
    //Encoders
    forblock in 1, 2, ..., $N_e$ do
      $X_{\text{block}}^{i,t} = LayerNorm(X_{\text{block}-1}^{i} + F_A(X_{\text{block}-1}^{i}))$,
      $X_{\text{block}}^{i} = LayerNorm(X_{\text{block}}^{i,t} + F_{FC}(X_{\text{block}}^{i,t}))$;
      $X_{\text{block}}^{j,t} = LayerNorm(X_{\text{block}-1}^{j} + F_A(X_{\text{block}-1}^{j}))$,
      $X_{\text{block}}^{j} = LayerNorm(X_{\text{block}}^{j,t} + F_{FC}(X_{\text{block}}^{j,t}))$;
      $X_{\text{block}}^{k,t} = LayerNorm(X_{\text{block}-1}^{k} + F_A(X_{\text{block}-1}^{k}))$,
      $X_{\text{block}}^{k} = LayerNorm(X_{\text{block}}^{k,t} + F_{FC}(X_{\text{block}}^{k,t}))$.
    end
    //Decoder
    for$block$ in 0, 1, 2, ..., $N_d$ do
      If (block = = 0)
        $X_{\text{block}}^{d,t} = LayerNorm(X_{N_e}^{k} + F_A(X_{N_e}^{i}, X_{N_e}^{j}, X_{N_e}^{k}))$,
        $X_{\text{block}}^{d} = LayerNorm(X_{\text{block}}^{i,t} + F_{FC}(X_{\text{block}}^{i,t}))$;
      else
        $X_{\text{block}}^{d,t} = LayerNorm(X_{N_e}^{k} + F_A(X_{\text{block}}^{d}))$,
        $X_{\text{block}}^{d} = LayerNorm(X_{\text{block}}^{i,t} + F_{FC}(X_{\text{block}}^{i,t}))$;
    end
    //Classifier
    Obtain feature matrix $X_{N_d}^{d} \in \mathbb{R}^{N_b * d_m * d_{\text{model}}}$;
    $flatten(X_{N_d}^{d}) \longrightarrow X_{N_d}^{df} \in \mathbb{R}^{N_b * (d_m * d_{\text{model}})}$;
    $\hat{y} = CLA = sofmax(ReLU(0, X_{N_d}^{df} w_1' + b_1') w_2' + b_2')$;
    Batch loss $\mathscr{J} = (1/N_b) \sum_{p=1}^{N_b} \mathscr{L}_{C-E}(\hat{y}_p, y_p)$;
    Calculate gradients $(\partial \mathscr{J} / \partial W)$, $(\partial \mathscr{J} / \partial b)$;
    Update parameters $W \leftarrow W - \eta(\partial \mathscr{J} / \partial W)$, $b \longleftarrow b - \eta(\partial \mathscr{J} / \partial b)$;
  end
end
Output: Weights and biases $\{W, b\}$
```

Algorithm 1: Training of MTFST.

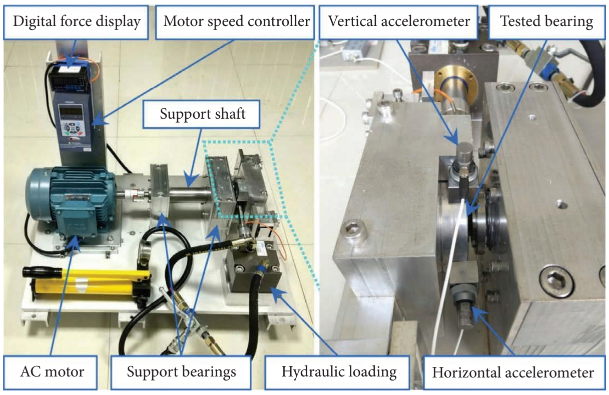


Figure 4: Testbed of XJTU-SY datasets.

Table 1: The ablation study of different hyperparameters.

Label		w_1	w_2	w_3	d_{model}	d_{ff}	n	N_e	N_d	r_d	r_s	Parameters number (M)	Average accuracy (%)
Baseline		512	256	128	64	128	8	6	6	0.1	0.5	1.53	99.34
A	A1	256	128	64								1.57	94.23
	A2	1024	512	256								1.53	92.15
	A3	128	256	512								2.07	88.16
B	B1				128	128						4.87	99.46
	B2				64	64						1.27	97.13
	B3				128	64						4.51	98.74
C	C1						16					1.53	95.77
	C2						4					1.53	83.12
D	D1							8				1.68	99.42
	D2							10				1.90	99.17
	D3							4				1.23	94.14
E	E1								8			1.60	98.34
	E2								10			1.68	94.12
	E3								4			1.45	97.76
F	F1									0.01		1.53	92.72
	F2									0.3		1.53	96.22
G	G1										0.3	1.53	97.32
	G2										0.7	1.53	98.67
	G3										1	1.53	98.46

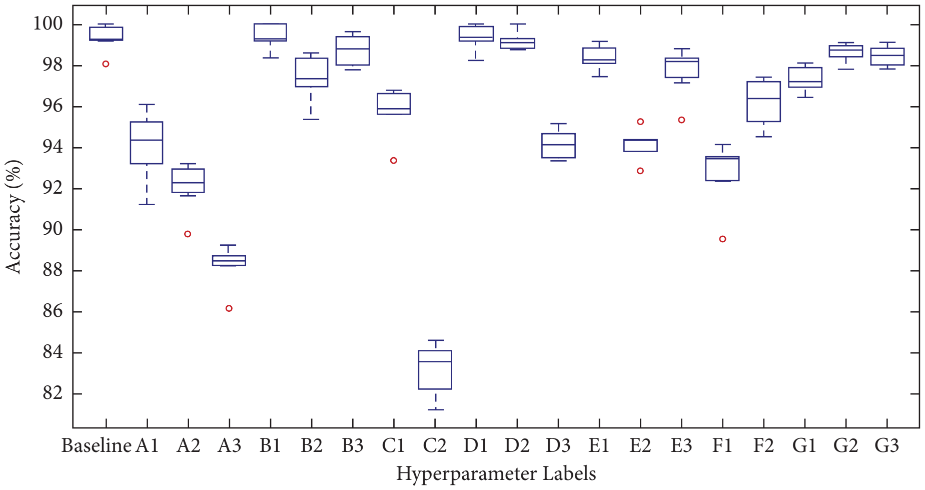
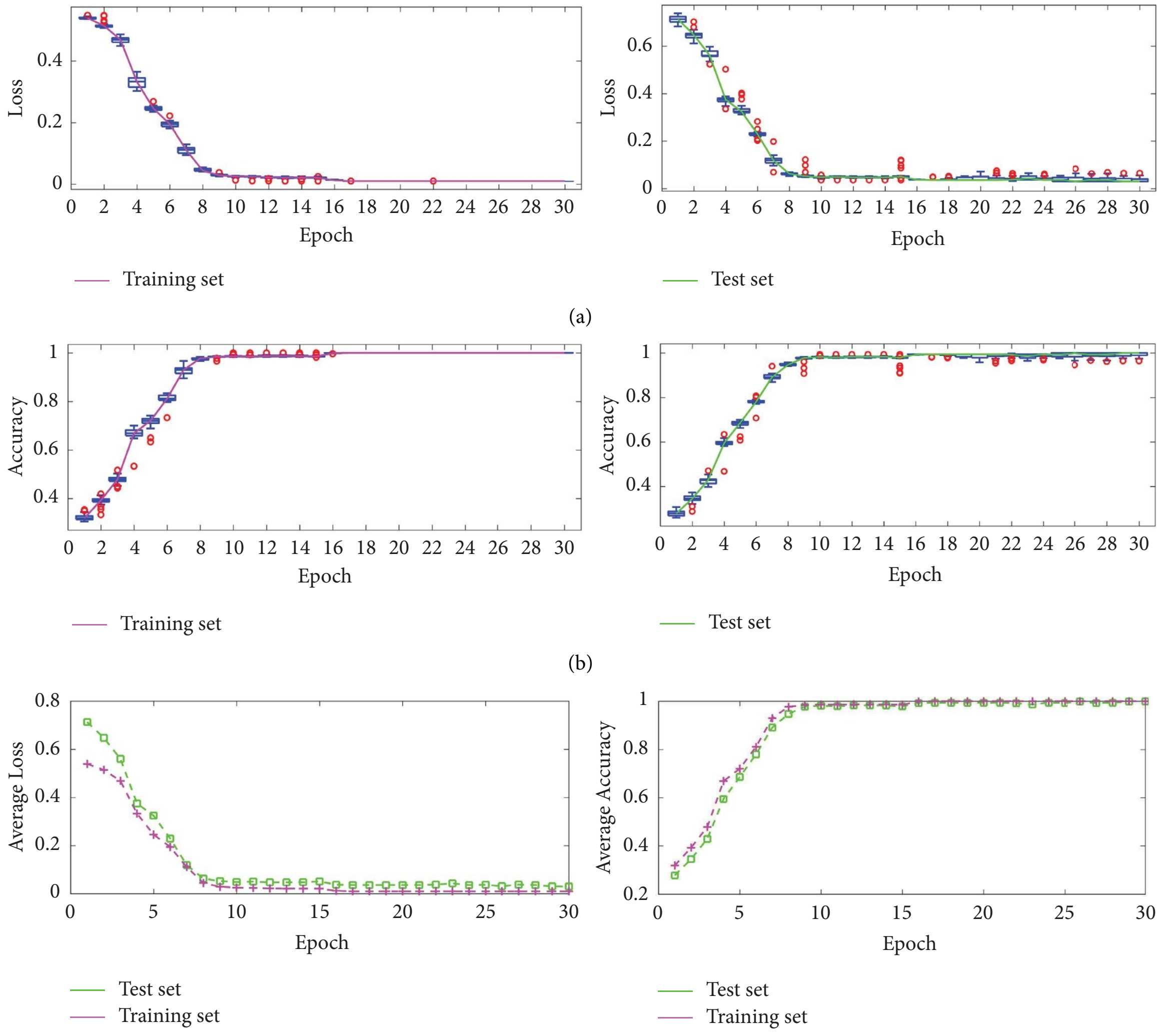


Figure 5: The accuracy boxplot of different hyperparameters.

22 in encoder2, and tokens between 35 and 80 in encoder3 are the most active. This removes the suspicion that the multiple encoders in the network would generate redundant features. In the decoder module, a multilayer attention mechanism is employed to remap the deep features and connect the classifier. As shown in Figure 8, the tokens between 1 and 15 are the most active in the first layer, and the strong weights can fuse the output information of different encoders and focus on the relationships between the feature patches corresponding to the different window widths while the active attention tokens between 35 and 60 in the last layer can extract the distinct components on which classification decisions are made. It should be noted that, since it is a compound fault type, the larger span of the salient tokens is presented in the decoder attention map of IBCO than others, which is similar to the human reasoning logic.

Second, the distribution form of the feature vectors in the embedding space also presents the working pattern of MTFST. In Figure 9, the feature vectors extracted from encoders and decoders are visualized via t-SNE, which nonlinearized high-dimensional features to two-dimensional vectors to visualize the clustering degree of fault types. It can be seen that the visualization results of raw signals lack clear boundaries for fault type identification, resulting in classification failure. Figure 9(b) presents the results of TFRs possessing linear separability, but there is

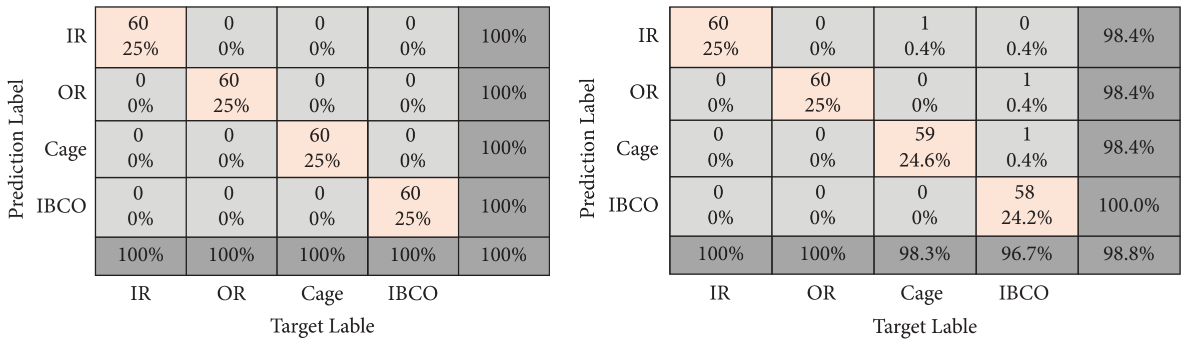


(a)

(b)

(c)

Figure 6: Loss function value and accuracy of training process under XJTU-SY dataset: (a) boxplot of loss function value, (b) boxplot of accuracy, and (c) average accuracy and loss throughout the training process.


Prediction Label
Target Lable

	IR	OR	Cage	IBCO	
IR	60 25%	0 0%	0 0%	0 0%	100%
OR	0 0%	60 25%	0 0%	0 0%	100%
Cage	0 0%	0 0%	60 25%	0 0%	100%
IBCO	0 0%	0 0%	0 0%	60 25%	100%
	100%	100%	100%	100%	100%

(a)

	IR	OR	Cage	IBCO	
IR	60 25%	0 0%	1 0.4%	0 0.4%	98.4%
OR	0 0%	60 25%	0 0%	1 0.4%	98.4%
Cage	0 0%	0 0%	59 24.6%	1 0.4%	98.4%
IBCO	0 0%	0 0%	0 0%	58 24.2%	100.0%
	100%	100%	98.3%	96.7%	98.8%

(b)

Figure 7: The confusion matrix of the MTFST under XJTU-SY dataset: (a) top accuracy (100%) and (b) minimum accuracy (98.8%).

a large number of overlapping areas in features; hence, it is hard to make classification with high accuracy. The results of encoders in MTFST are shown in Figures 9(c)–9(e), and we can observe that there are obvious decision boundaries between different fault types in the features generated by multiscale TFR encoders, which presents the effectiveness of encoders in coding discriminative class tokens. Nevertheless, many tokens in the encoders are still inevitably misclassified,

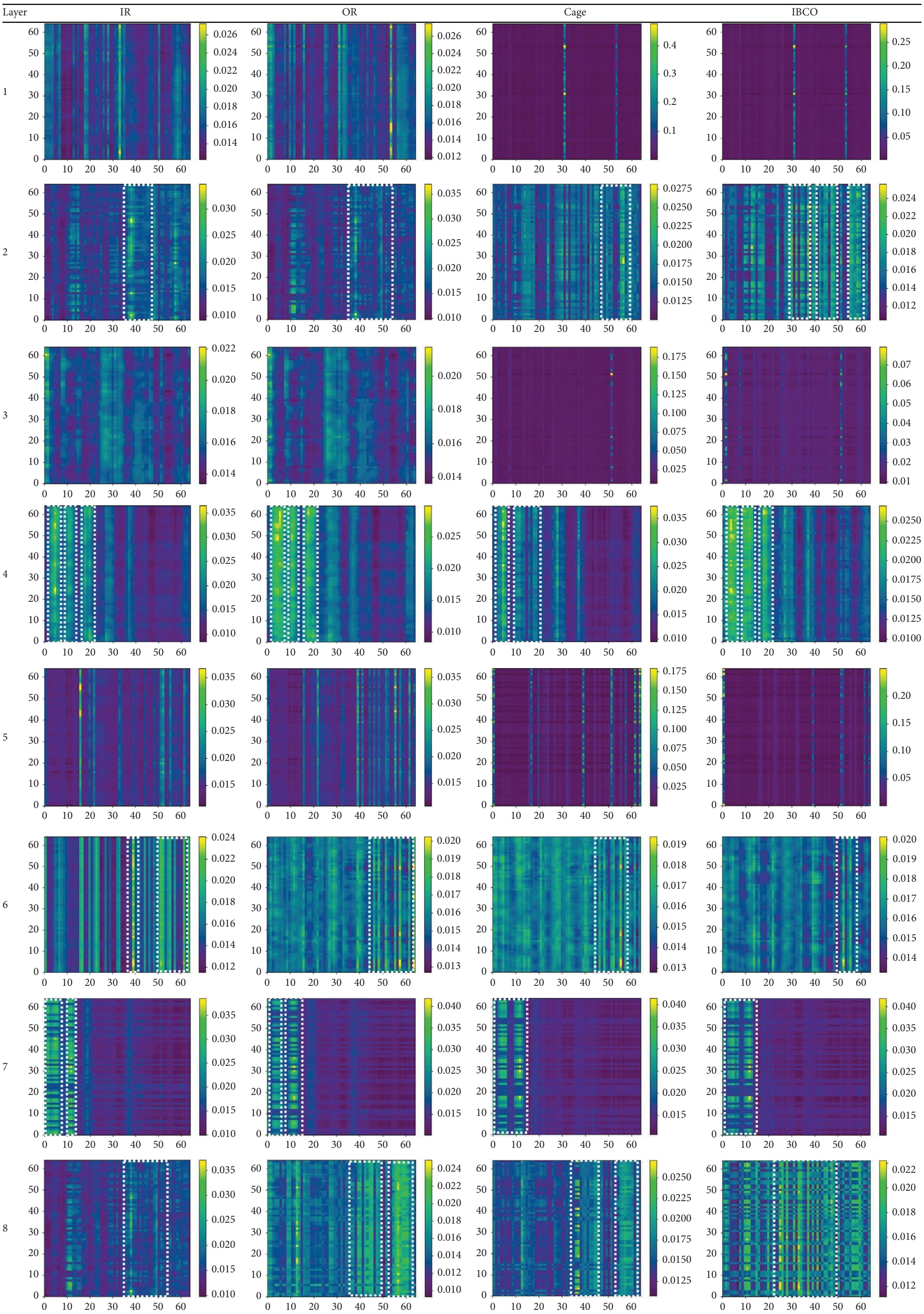
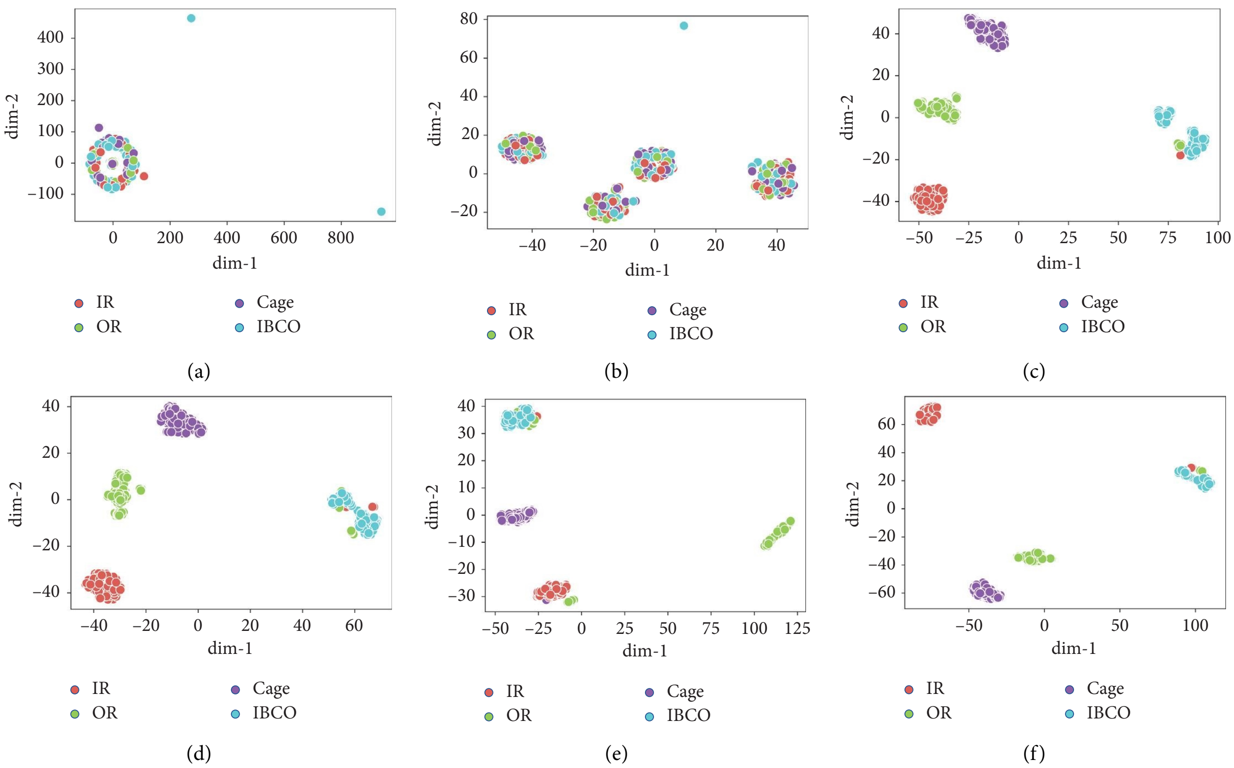


Figure 8: The attention weights of each fault label. From top to bottom, odd rows present the weights map of the first attention layer in encoder1, encoder2, encoder3, and decoder, respectively, and even rows correspond to the last layer. And the most active patches are in the dashed white box.



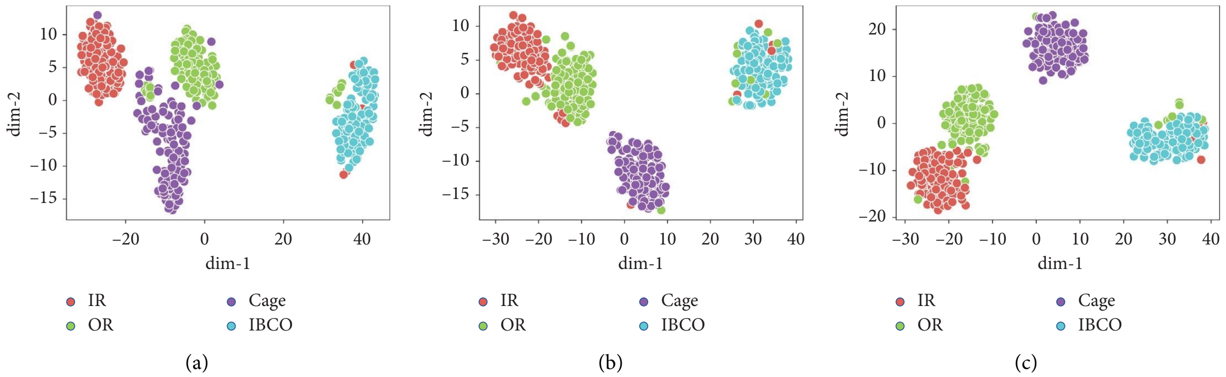
(a) (b) (c)

(d) (e) (f)

Figure 9: Visualization of the feature vector in different modules via t-SNE: (a) raw signals, (b) TFT, (c–e) 1st–3rd encoders, and (f) decoder.

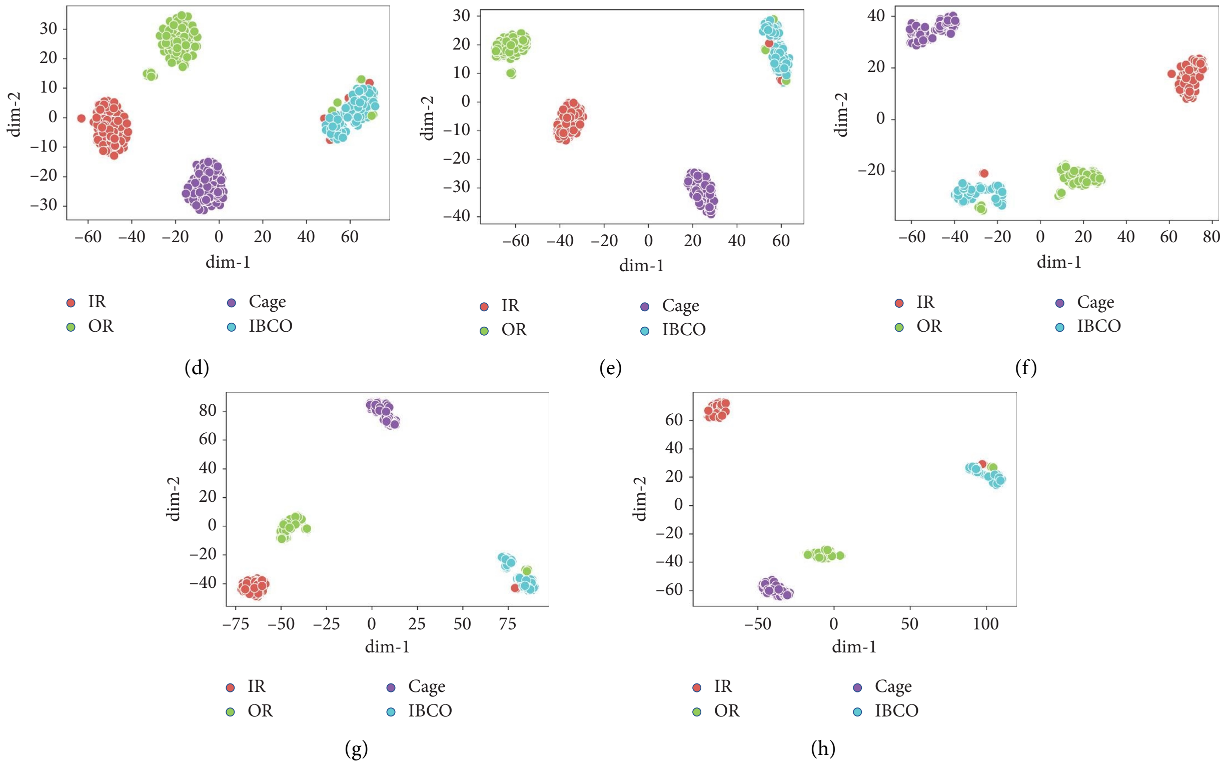
Table 2: The performance of different methods under the XJTU-SY dataset.

Method	Signal processing	Average accuracy (%)
CNN	Raw vibration signal	90.42
CNN-LSTM	Raw vibration signal	88.79
Bi-LSTM	TFR (SWT)	89.47
WRN-16-2	TFR (STFT)	90.53
TST	Raw vibration signal	93.37
TFT	TFR (SWT)	96.43
BAFT	Raw vibration signal	97.29
MTFST (proposed)	TFR (STFT)	99.34



(a) (b) (c)

Figure 10: Continued.



(d)

(e)

(f)

(g)

(h)

Figure 10: Visualization of the learned features in different methods via t-SNE: (a) CNN, (b) CNN-LSTM, (c) Bi-LSTM, (d) WRN-16-2, (e) TST, (f) TFT, (g) BAFT, and (h) MTFST (proposed).

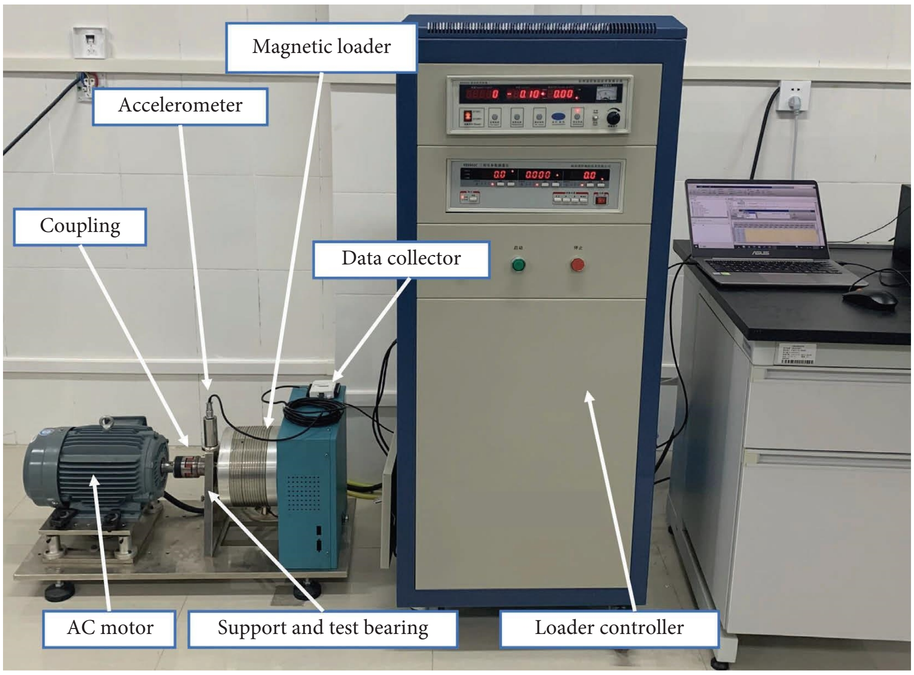


Figure 11: Uniaxial rolling bearing experimental platform.

Table 3: Description of self-made experimental rig dataset.

Defect mode	Fault type
Single defect	Inner race (IR) Out race (OR) Cage Ball
Compound defect	Out race and ball (OB) Inner race, ball, and cage (IBC)

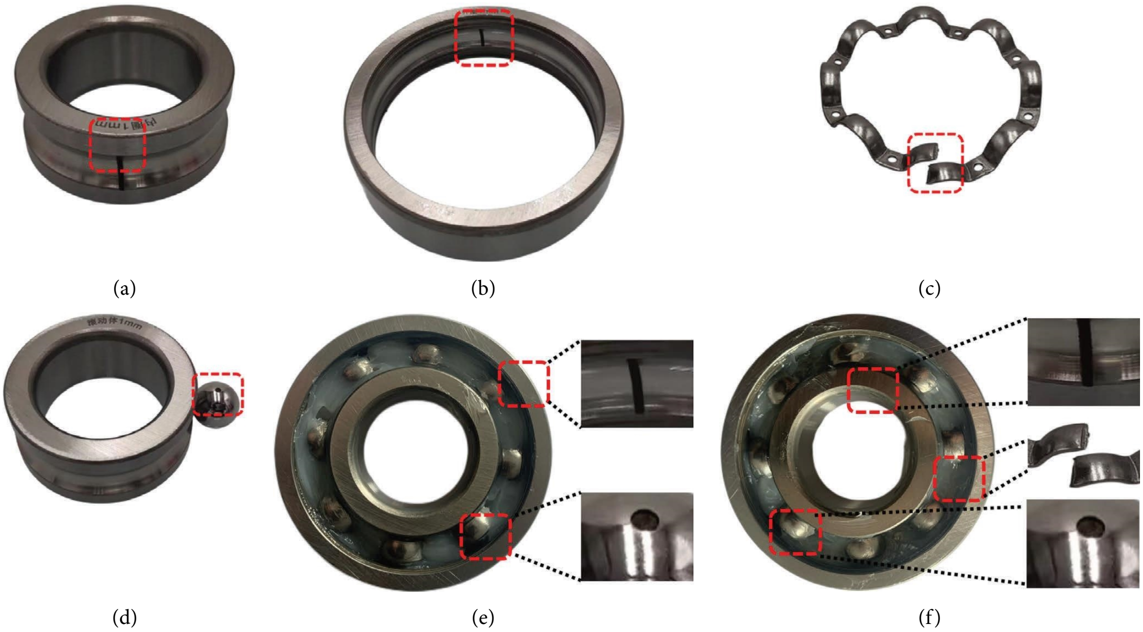

(a) (b) (c) (d) (e) (f)

Figure 12: Different defect types in the experiment rig: (a) inner race (IR), (b) out race (OR), (c) cage, (d) ball, (e) out race and ball (OB), and (f) inner race, ball and cage (IBC).

Table 4: The hyperparameters of MTFST.

Parameter name	w_1	w_2	w_3	d_{model}	d_{ff}	n	N_e	N_d	r_d	r_s	Batch size	Epoch
Value	1024	512	256	64	128	8	6	6	0.1	0.5	50	40

and the distribution discreteness in the same fault type is needed to be improved. The class tokens in decoder, which is the final output of MTFST, are shown in Figure 9(f). It is obvious that the distribution has well-defined interclass boundaries and compact intraclass distance, which illustrates that the decoder fuses the information of each encoder and further improves the ability in extracting and expressing hidden features of MTFST.

4.1.5. Comparison with Other Methods. In this section, the proposed network is compared with other deep learning methods to demonstrate the MTFST's effectiveness further. Among these methods, raw signals and TFR-based networks are adopted, including CNN [45], CNN-LSTM [46], Bi-LSTM [47], and WRN-16-2 [48]. Furthermore, the up-to-date transformer-like methods TST [30], TFT [28], and BAFT [29] are employed for the comparison. The parameters of the above methods are set as in the original papers.

The dataset is randomly divided into training set and test set, and the train/test ratio is set to 0.8/0.2. The diagnosis results of different methods are listed in Table 2. In general, TFR-based methods obtain better performance than vibration signals-based methods.

The proposed MTFST achieves the best average accuracy with 99.34%. In addition, t-SNE is also used to investigate the effects of fault feature extraction and the representation ability of different models. The tests are closest to the average accuracy of each network as an example. As shown in Figure 10, the hidden features extracted by MTFST possess the best intraclass compactness and interclass separability. The results denote that MTFST achieves prime fault diagnosis performance in the XJTU-SY dataset.

Table 5: The diagnosis results from different methods.

Working conditions (Nm)	Defect mode	Label	Average accuracy of different methods (%)							
			CNN	CNN-LSTM	Bi-LSTM	WRN-16-2	TST	TFT	BAFT	MTFST (proposed)
2	Single defect	A1	86.7	88.3	89.2	90.0	87.5	89.2	90.8	**92.5**
	Single defect + compound defect	A2	82.5	84.2	88.3	85.0	85.0	87.5	86.7	**90.8**
5	Single defect	B1	90.0	85.8	87.5	90.0	90.0	89.2	90.8	**93.3**
	Single defect + compound defect	B2	82.5	79.2	85.0	88.3	89.2	89.2	**90.8**	90.0
10	Single defect	C1	89.7	90.5	89.6	87.7	90.1	90.5	90.4	**92.5**
	Single defect + compound defect	C2	83.3	87.5	84.2	87.5	85.8	88.3	89.2	**90.8**

The bold values indicate the optimal diagnosis accuracy.

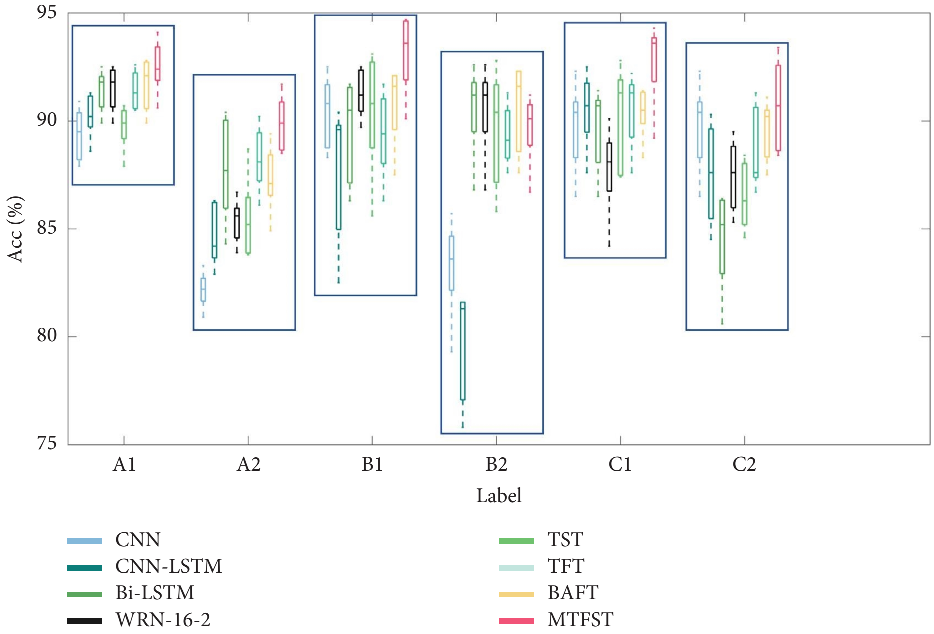
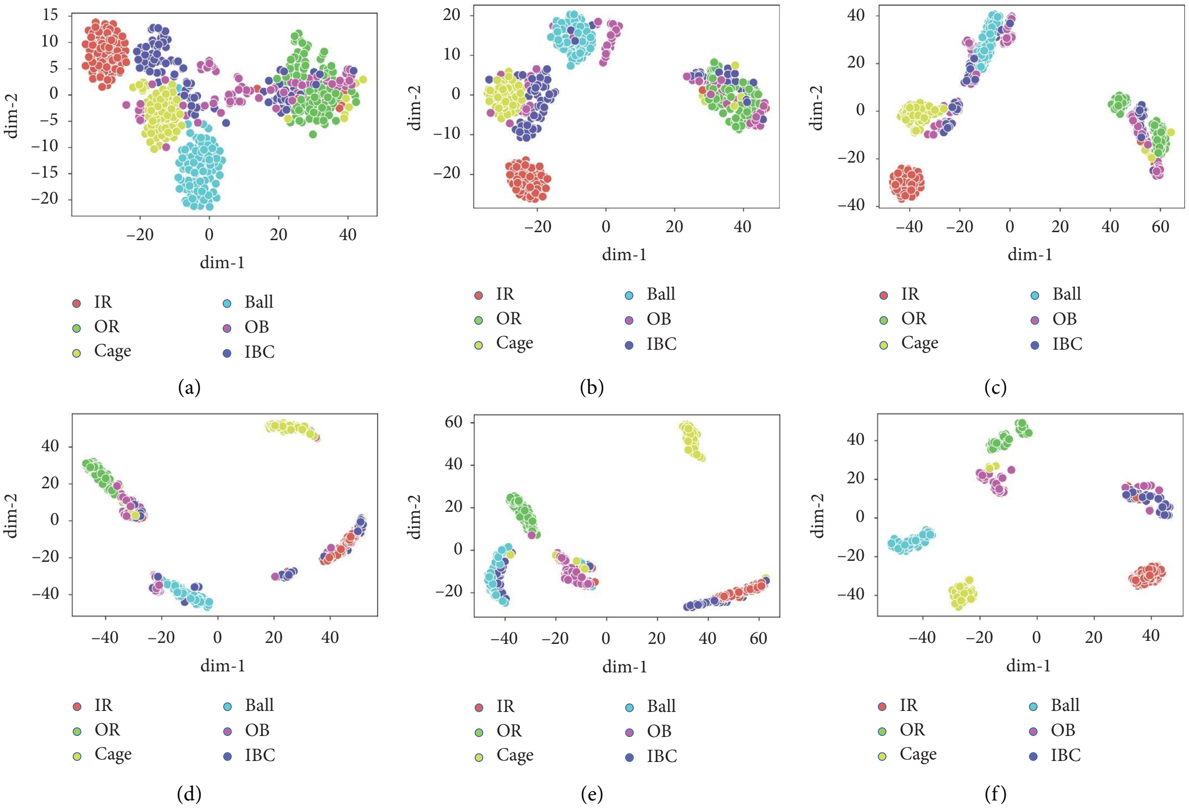


Figure 13: The boxplot of accuracy in different models.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 14: Continued.

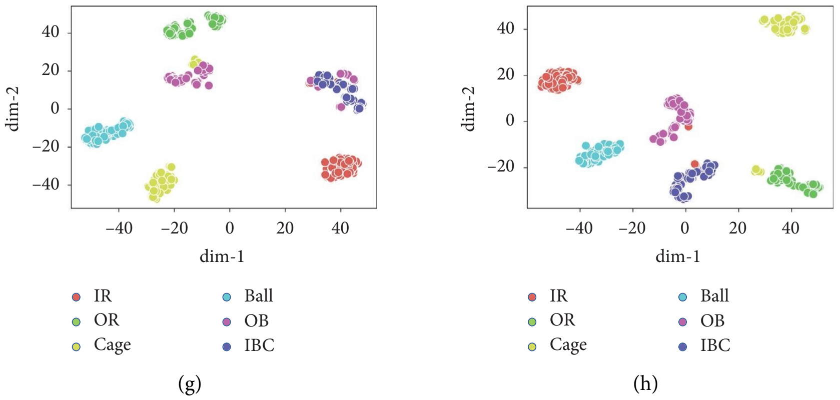
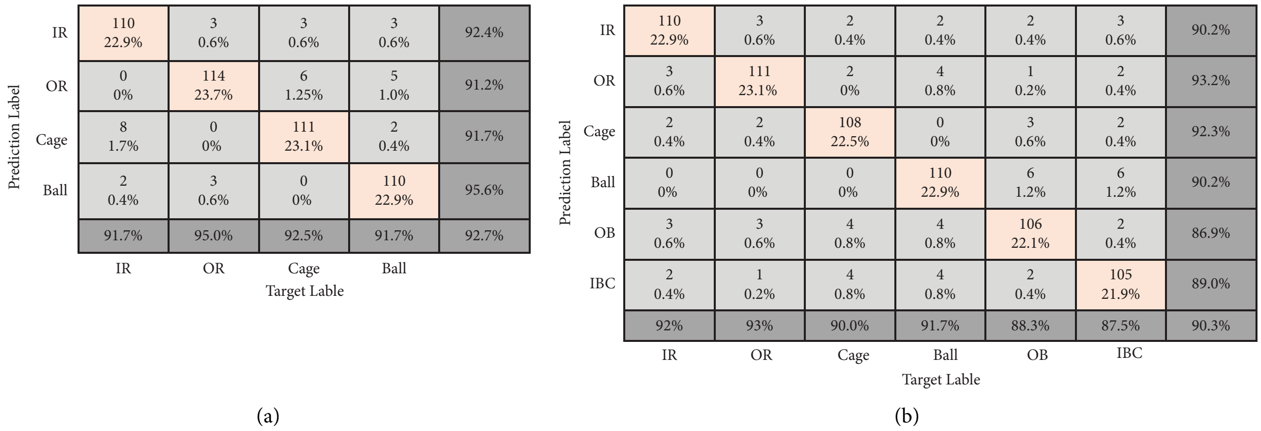


Figure 14: Visualization of the learned features in different methods via t-SNE: (a) CNN, (b) CNN-LSTM, (c) Bi-LSTM, (d) WRN-16-2, (e) TST, (f) TFT, (g) BAFT, and (h) MTFST (proposed).



Prediction Label	IR	OR	Cage	Ball	
IR	110 22.9%	3 0.6%	3 0.6%	3 0.6%	92.4%
OR	0 0%	114 23.7%	6 1.25%	5 1.0%	91.2%
Cage	8 1.7%	0 0%	111 23.1%	2 0.4%	91.7%
Ball	2 0.4%	3 0.6%	0 0%	110 22.9%	95.6%
	91.7%	95.0%	92.5%	91.7%	92.7%

Target Lable

(a)

Prediction Label	IR	OR	Cage	Ball	OB	IBC	
IR	110 22.9%	3 0.6%	2 0.4%	2 0.4%	2 0.4%	3 0.6%	90.2%
OR	3 0.6%	111 23.1%	2 0%	4 0.8%	1 0.2%	2 0.4%	93.2%
Cage	2 0.4%	2 0.4%	108 22.5%	0 0%	3 0.6%	2 0.4%	92.3%
Ball	0 0%	0 0%	0 0%	110 22.9%	6 1.2%	6 1.2%	90.2%
OB	3 0.6%	3 0.6%	4 0.8%	4 0.8%	106 22.1%	2 0.4%	86.9%
IBC	2 0.4%	1 0.2%	4 0.8%	4 0.8%	2 0.4%	105 21.9%	89.0%
	92%	93%	90.0%	91.7%	88.3%	87.5%	90.3%

Target Lable

(b)

Figure 15: Classification confusion matrix of self-made experiment rig under 10 Nm: (a) single defect and (b) compound defect.

4.2. Case 2: Self-Made Experimental Rig. A proprietary uniaxial rolling gear test rig is used to simulate the different working conditions in our experimentation, which contains a motor, coupling, test bearing, adjustable magnetic loader, acceleration sensor, and data acquisition system as shown in Figure 11. In the test, a three-phase asynchronous motor commonly used in mine water pump and small hoist is employed. The rated power of the motor is 2.2 kw, and working speed is 1430 r/min. The magnetic loader is controlled by an NX6000 dynamometer, and working load torques are set to 2 Nm, 5 Nm, and 10 Nm. A set of deep-groove ball bearings (SKF-4306) with different defects are used for vibration monitoring as listed in Table 3. As shown in Figure 12, there are 6 bearing states containing single defect and compound defect, to simulate the faults frequently occurring in mining machine. For these defects, they are created by a linear cutting machine, and inner race, out race, and ball are defected with a same size of 1 mm width and 0.5 mm depth, and cage is cut radially. The vibration signals under different working conditions are acquired by a piezoelectric accelerometer CYQ9250 and amplified by a data collector NI USB-6009 with a sample frequency of 10 kHz and 20 minutes duration. Finally, the signals are randomly split into train and test sets with a sample size of 500 and 120 under per condition, and each sample contains 20,000 points.

The proposed MTFST and other deep networks are tested in the self-made experiment dataset to validate the effectiveness and superior performance of our method. The hyperparameters of MTFST are shown in Table 4. Again, the experiment of each model is conducted for 5 runs to exclude the effect of the randomness of the data. The average accuracy of the test set under different defect types and working conditions is listed in Table 5. The accuracy boxplots in different methods are shown in Figure 13, which tailed the results of 5 runs. It can be seen that the accuracy in single defect diagnosis is generally higher than that of compound defect types. In general, the proposed MTFST obtains better performance than other comparative groups. As shown in Figure 14, the t-SNE of extracted features is used to further illustrate the performance of MTFST. Note that, for each model, the test under working conditions of 10 Nm and closest to the average accuracy is used as examples. From the results, it can be observed that MTFST is better at learning to distinguish the hidden characteristics of fault types. The confusion matrixes are shown in Figure 15 and further present the diagnosis results of MTFST in the test example.

5. Conclusions

In this paper, a novel transformer-like bearing fault diagnosis network that processes the TFRs of raw vibration signal is established. The XJTU-SY and self-made experiment rig datasets are used to verify the effectiveness, and the diagnosis results of some existing networks based on CNN, RNN, and transformer are analysed in the experiment as the comparison groups of the proposed MTFST. The main conclusions are as follows:

(1) The novel tokenizer based on TFRs that obtained by different window widths STFT is designed, which can code the multiscale complementary TFR information to grasp more discriminative features.

(2) The designed sparse self-attention mechanism (SSAM) can effectively eliminate the interference of secondary information and obtain a better performance than naive self-attention mechanism.

(3) The proposed MTFST discards the recurrence structure and convolutional operations and focuses on the multihead attention mechanism, which improves diagnostic performance and has partial interpretability. Furthermore, the encoder-decoder framework of MTFST is closer to the vanilla transformer and better in extracting hidden features than existing transformer-like algorithms.

Experiment results indicate that MTFST can effectively detect rolling bearings faults, which extends the kind of diagnosis methodology based on transformer. Future research will focus on the following aspects to ensure the further improvement. First, CNN models and transformer are integrated to enhance model performance by adding the small-field features. Second, the adaptive STFT window widths and sparse ratio of SSAM can be studied to improve the generalization. Third, the method can be tested on different rotating machines and application scenarios, such as gearbox and remaining useful life (RUL) estimation.

Data Availability

The data supporting the current study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors would like to thank the Youth Program of the Education Foundation of Guizhou Province, China (No. QianJiaoHeKYZi[2020]125).

References

[1] X. Zhao, M. Jia, J. Bin, T. Wang, and Z. Liu, "Multiple-order graphical deep extreme learning machine for unsupervised fault diagnosis of rolling bearing," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2011.

[2] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: a review," *Mechanical Systems and Signal Processing*, vol. 108, pp. 33–47, 2018.

[3] Q. Song, X. Jiang, G. Du, J. Liu, and Z. Zhu, "Smart multichannel mode extraction for enhanced bearing fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 189, Article ID 110107, 2023.

[4] P. Frank, S. X. Ding, and T. Marcu, "Model-based fault diagnosis in technical processes," *Transactions of the Institute of Measurement and Control*, vol. 22, no. 1, pp. 57–101, 2000.

[5] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques–Part I: fault Diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[6] Z. Liu, W. Guo, J. Hu, and W. Ma, "A hybrid intelligent multifault detection method for rotating machinery based on RSGWPT, KPCA and twin SVM," *ISA Transactions*, vol. 66, pp. 249–261, 2017.

[7] H. Ren, X. Zhu, and J. Wang, "An effective model fusion method for bearing fault diagnosis," in *Proceedings of the 2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Ottawa, ON, Canada, May 2022.

[8] H. T. Zhou, J. Chen, G. M. Dong, H. C. Wang, and H. D. Yuan, "Bearing Fault recognition method based on neighbourhood component analysis and coupled hidden Markov model," *Mechanical Systems and Signal Processing*, vol. 66-67, pp. 568–581, 2016.

[9] X. X. Jiang, Q. Y. Song, and H. E. Wang, "Central frequency mode decomposition and its applications to the fault diagnosis of rotating machines," *Mechanism and Machine Theory*, vol. 174, Article ID 104919, 2022.

[10] X.-X. Jiang, J. Wang, and C.-Q. Shen, "An adaptive and efficient variational mode decomposition and its application for bearing fault diagnosis," *Structural Health Monitoring*, vol. 20, no. 5, pp. 2708–2725, 2021.

[11] W. Mao, W. Feng, Y. Liu, and D. Zhang, "A new deep autoencoder method with fusing discriminant information for bearing fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 150, Article ID 107233, 2021.

[12] H. Shao, H. Jiang, H. Zhang, and W. Duan, "Rolling bearing fault feature learning using improved convolutional deep Belief network with compressed sensing," *Mechanical Systems and Signal Processing*, vol. 100, pp. 743–765, 2018.

[13] F. Jia, Y. Lei, N. Lu, and S. Xing, "Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization," *Mechanical Systems and Signal Processing*, vol. 110, pp. 349–367, 2018.

[14] S. Gao, S. Shi, and Y. Zhang, "Rolling bearing compound fault diagnosis based on parameter optimization MCKD and convolutional neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, Article ID 3508108, 2022.

[15] D. Liu, L. Cui, W. Cheng, and D. Zhao, "Rolling bearing fault severity recognition via data mining integrated with

convolutional neural network," *IEEE Sensor journal*, vol. 22, no. 6, pp. 5678–5777, 2022.

[16] Y. Huang, A. Liao, D. Hu, and W. Shi, "Multi-scale convolutional network with channel attention mechanism for rolling bearing fault diagnosis," *Measurement*, vol. 203, Article ID 111935, 2022.

[17] H. Wang, J. Xu, R. Yan, and R. Gao, "A new intelligent bearing fault diagnosis method using SDP representation and SE-CNN," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 2377–2389, 2020.

[18] Z. Xu, C. Li, and Y. Yang, "Fault diagnosis of rolling bearing of wind turbines based on the variational mode decomposition and deep convolutional neural networks," *Applied Soft Computing*, vol. 95, Article ID 106515, 2022.

[19] Z. An, S. Li, J. Wang, and X. Jiang, "A novel bearing intelligent fault diagnosis framework under time-varying working conditions using recurrent neural network," *ISA Transactions*, vol. 100, pp. 155–170, 2020.

[20] Y. Zhang, T. Zhou, X. Huang, and L. Cao, "Fault diagnosis of rotating machinery based on recurrent neural networks," *Measurement*, vol. 171, Article ID 108774, 2021.

[21] R. Zhao, R. Q. Yan, J. J. Wang, and K. Z. Mao, "Learning to monitor machine health with convolutional Bi-directional LSTM networks," *Sensors*, vol. 17, p. 273, 2017.

[22] X. Li, W. Zhang, and Q. Ding, "Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism," *Signal Processing*, vol. 161, pp. 136–154, 2019.

[23] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the 31st Conference and Workshop on Neural Information Processing Systems (NIPS)*, Red Hook, New York, NY, USA, 2017.

[24] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, June 2019.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: transformers for image recognition at scale," in *Proceedings of International Conference on Learning Representations (ICLR)*, January 2021.

[26] C. Chen, Q. Fan, and R. C. Panda, "Cross-attention multi-scale vision transformer for image classification," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, October 2021.

[27] W. Wang, E. Xie, X. Li et al., "Pyramid vision transformer: a versatile backbone for dense prediction without convolutions," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Montreal, Canada, October 2021.

[28] Y. Ding, M. Jia, Q. Miao, and Y. Cao, "A novel time–frequency transformer based on self–attention mechanism and its application in fault diagnosis of rolling bearings," *Mechanical Systems and Signal Processing*, vol. 168, Article ID 108616, 2022.

[29] Z. Jiao, L. Pan, W. Fan, and Z. Xu, "Partly interpretable transformer through binary arborescent filter for intelligent bearing fault diagnosis," *Measurement*, vol. 203, Article ID 111950, 2022.

[30] Y. Jin, L. Hou, and Y. Chen, "A time series transformer based method for the rotating machinery fault diagnosis," *Neurocomputing*, vol. 494, pp. 379–395, 2022.

[31] X. Du, L. Jia, and I. Haq, "Fault diagnosis based on SPBO-SDAE and transformer neural network for rotating machinery," *Measurement*, vol. 188, Article ID 110545, 2022.

[32] S. Zhu, B. Liao, Y. Hua, and C. Zhang, "A transformer model with enhanced feature learning and its application in rotating machinery diagnosis," *ISA Transactions*, vol. 133, pp. 1–12, 2022.

[33] H. Wu, J. Li, Q. Zhang, J. Tao, and Z. Meng, "Intelligent Fault diagnosis of rolling bearings under varying operating conditions based on domain-adversarial neural network and attention mechanism," *ISA Transactions*, vol. 130, pp. 477–489, 2022.

[34] H. Zhou, P. Cheng, S. Shao, and Y. Zhao, "Intelligent bearing fault diagnosis method based on A domain aligned clustering network," *Measurement Science and Technology*, vol. 34, Article ID 44001, 2023.

[35] M. Hakim, A. A. B. Omran, J. I. Inayat-Hussain et al., "Bearing Fault diagnosis using lightweight and robust one-dimensional convolution neural network in the frequency domain," *Sensors*, vol. 22, p. 5793, 2020.

[36] W. Fu, X. Jiang, B. Li, C. Tan, B. Chen, and X. Chen, "Rolling bearing fault diagnosis based on 2D time-frequency images and data augmentation technique," *Measurement Science and Technology*, vol. 34, Article ID 45005, 2023.

[37] L. Jia, T.-W. Chow, and Y. Yuan, "GTFE-net: a gramian time frequency enhancement CNN for bearing fault diagnosis," *Engineering Applications of Artificial Intelligence*, vol. 119, Article ID 105794, 2023.

[38] X. Pei, X. Zheng, and J. Wu, "Rotating machinery fault diagnosis through a transformer convolution network subjected to transfer learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, Article ID 2515611, 2021.

[39] A. Zhao, K. Subramani, and P. Smaragdis, "Optimizing short-time fourier transform parameters via gradient descent," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Canada, June 2021.

[40] X. Chen, S. Xie, and K. M. He, "An empirical study of training self-supervised vision transformers," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2021.

[41] Z. H. Fu, Z. H. Fu, Q. J. Liu, W. R. Cai, and Y. H. Wang, "Sparse TT: visual tracking with sparse transformers," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, May 2022.

[42] D.-P. Kingma and B. A. Jinny, "A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representation, (ICLR)*, San Diego, CA, USA, 2015.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. D. Salakhutdinov, "A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[44] B. Wang, Y.-G. Lei, N.-P. Li, and N.-B. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Transactions on Reliability*, vol. 69, pp. 1–12, 2018.

[45] W. Huang, J. Cheng, Y. Yang, and G. Guo, "An improved deep convolutional neural network with multi-scale

information for bearing fault diagnosis," *Neurocomputing*, vol. 359, pp. 77–92, 2019.

[46] R. Yang, S.-K. Singh, M. Tavakkoli et al., "CNN-LSTM deep learning architecture for computer vision-based modal frequency detection," *Mechanical Systems and Signal Processing*, vol. 144, Article ID 106885, 2020.

[47] Z. Zhao, T. Li, J. Wu et al., "Deep learning algorithms for rotating machinery intelligent diagnosis: an open source benchmark study," *ISA Transactions*, vol. 107, pp. 224–55, 2020.

[48] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," 2016, https://arxiv.org/abs/1605.07146.