On-Device Friendly Fault Diagnosis Framework Leveraging Periodicity-Guided Patch Representation of Bearing Signals

Anonymous

Abstract

Fault diagnosis of rotating machinery is critical for ensuring safety and reliability in modern manufacturing systems. While deep learning approaches have achieved remarkable progress in this domain, their deployment on edge and resource-constrained devices is hindered by high computational and memory demands. To address this challenge, we present an on-device fault diagnosis framework that explicitly leverages the periodic characteristics of bearing signals through a periodicity-guided patch representation. In our framework, bearing signals are decomposed into fixed-length patches aligned with their intrinsic periodicity, enabling effective temporal feature preservation while facilitating extensive parameter sharing within a lightweight Transformer architecture. This design drastically reduces the number of learnable parameters and computational overhead without degrading diagnostic accuracy. Experimental validation on both public and private datasets demonstrates that the proposed framework achieves superior fault classification performance with as few as 7,513 parameters, which is less than that of linear regression. The results highlight the practicality of the proposed approach for on-device deployment and its potential as a scalable and efficient solution for intelligent fault diagnosis in industrial environments.

Keywords: Rotating machinery, Fault diagnosis, On-device, Periodicity, Patch representation

1. Introduction

Fault diagnosis systems are fundamental to modern industrial operations, serving as critical enablers of operational safety, reliability, and cost efficiency in complex machinery environments (Lei et al., 2020; Zhao et al., 2022). Traditional approaches to fault diagnosis, ranging from linear models and signal processing techniques to statistical analysis, have provided important foundations for condition monitoring (Rauber et al., 2014; Jia et al., 2016; Zhuo and Ge, 2021). These methods were subsequently enhanced through machine learning, which enabled more robust pattern extraction and improved classification performance (Razavi-Far et al., 2017; Mao et al., 2018; Ma and Wu, 2018). More recently, deep learning has emerged as the dominant paradigm, offering superior accuracy by capturing nonlinear dependencies in high-dimensional bearing signals and achieving state-of-the-art performance across various diagnostic tasks (Zhang et al., 2017; Karim et al., 2017; Bai et al., 2018; Hou et al., 2023).

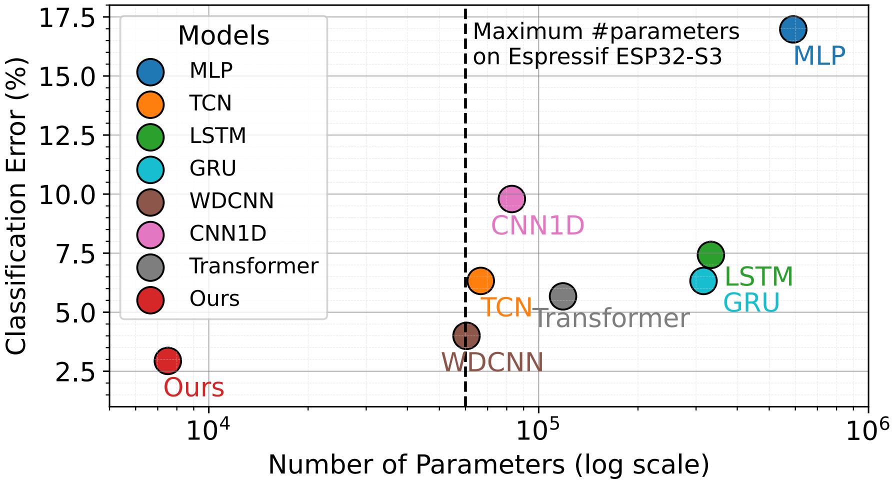


Figure 1: Classification error versus model size on CWRU. Lower is better for both axes. The proposed framework achieves the lowest classification error with a substantially smaller parameter, thereby facilitating practical deployment on resource-constrained devices.

Despite these advances, deep learning models generally incur significant computational and memory costs, which hinder their direct deployment in resource-constrained environments such as embedded systems and industrial edge devices (Cheng et al., 2024; Sangaiah et al., 2019; Liu et al., 2024). For example, the Espressif ESP32-S3, a representative IoT device with only 512 KB of SRAM, can accommodate at most 60,000 parameters. However, existing models for fault diagnosis require parameter counts that exceed this limit (Figure 1). Specifically, WDCNN (Zhang et al., 2017) requires 60,424 parameters. This highlights a significant gap between deep learning-based fault diagnosis systems and real-world manufacturing environments in terms of practical on-device deployment.

To address the increasing demand for deploying intelligent diagnostic systems on resource-constrained devices, we propose an **On-Device friendly Fault diagnosis framework (ODF)** that leverages the periodicity inherent in bearing signals.

Our approach is motivated by the observation that bearing vibration signals exhibit repetitive structures, which can be effectively exploited to design a more compact yet powerful model. Specifically, we segment the original bearing signals into multiple fixed-length patches and introduce a periodicity-guided patch representation that enables efficient feature sharing across layers. This design substantially reduces the number of trainable parameters while preserving the critical temporal patterns essential for accurate fault classification. Building on the Transformer architecture (Vaswani, 2017), each patch feature interacts with class-specific tokens, allowing the model to focus on discriminative fault signatures while maintaining minimal computational overhead. As illustrated in Figure 1, the proposed framework achieves a better trade-off between computational efficiency and diagnostic accuracy. This enables the active adaptation of fault diagnosis models to on-device environment. This work highlights the potential of combining domain knowledge with architectural optimization to enable practical and scalable fault diagnosis solutions.

The remainder of this paper is organized as follows: Section 2 provides background information on bearing fault diagnosis and recent modeling approaches. Section 3 introduces our motivation of a periodicity-guided structure and architecture and details its design principles. Section 4 demonstrates the effectiveness of the proposed framework through extensive experiments on both public datasets and simulated bearing systems. Finally, Section 5 concludes the paper and discusses potential directions for future research.

2. Related Work

2.1. Fault Diagnosis System

A fault diagnosis system plays a critical role in industrial operations by ensuring reduced maintenance costs and high productivity (Lei et al., 2020). Such systems are primarily designed to detect and identify faults in machinery and equipment. Among these, bearing fault diagnosis systems are particularly important, as bearings are fundamental components within industrial engineering systems (Lanham, 2002). Typically, bearings consist of multiple components, such as balls and races, which can have various fault conditions.

In order to detect fault conditions in bearing systems, researchers commonly analyze the vibration signals of bearing components. Let a vibration signal be defined as $\boldsymbol{x} = [x_1, x_2, \cdots, x_k] \in \mathbb{R}^k$. The goal of a bearing fault diagnosis system is to differentiate between vibration signals $\boldsymbol{x}$ under normal conditions and those under fault conditions. Specifically, the objective is to estimate $P(y|\boldsymbol{x})$, where $y \in \mathbb{Y}$ represents the condition of the bearing system, and $\mathbb{Y}$ denotes a set of normal/fault states of the bearing system. Typically, $y = 0$ corresponds to the normal state, while $y \neq 0$ indicates fault conditions. For instance, $y = 1$ may represent ball faults.

Therefore, we aim to generate a function $f(\boldsymbol{x})$ that approximates the true conditional distribution $P(y|\boldsymbol{x})$. In general, $f(\boldsymbol{x})$ is a vector that provides the probability of $P(y|\boldsymbol{x})$ for each y, yielding a prediction $\hat{y}$ becomes as follows:

$$\hat{y} = \arg\max_{y \in \mathbb{Y}} [f(\boldsymbol{x})]_y, \tag{1}$$

where $[f(\boldsymbol{x})]_y$ denotes the y-th component of the output of $f(\boldsymbol{x})$.

Traditional fault diagnosis techniques built $f(\boldsymbol{x})$ based on statistical methods and model-based approaches. These methods include vibration analysis, signal processing, and physical modeling (Yang et al., 2005; Paliwal et al., 2014). However, they often struggle to handle the increasing complexity of modern industrial equipment and the volume of generated data.

To address this problem, researchers have adopted machine learning techniques that automatically extract features from raw input data without manual intervention (Zhang et al., 2020). Models such as XGBoost have shown high performance in fault diagnosis tasks. More recently, using deep learning models, researchers even achieved superior performance on the same tasks. Temporal convolutional network (TCN) (Zheng et al., 2021), fully convolutional networks with long short-term memory (LSTM-FCN) (Li et al., 2022), deep convolutional neural network with wide first-layer kernel (WDCNN) (Zhang et al., 2017), and a variant of one-dimensional CNN (CNN1D) (Chen et al., 2020) represent significant advancements in fault diagnosis systems using deep learning architectures.

Despite their success, these models require significant computational resources and memory, limiting their applicability in resource-constrained environments such as Internet of Things (IoT) devices, which are commonly adopted in real-world manufacturing applications. As the rising cost of detection machines reduces the advantages of fault diagnosis systems, achieving a balance between cost and benefit becomes a critical research area (Cheng et al., 2024).

2.2. Recent Advances in Time Series Deep Learning

Since deep learning models have emerged as a powerful tool for various domains, researchers have developed domain-specific versions for modeling time series data. Specifically, Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) are widely known that shows decent performance in time series forecasting and prediction, including fault diagnosis (Elsayed et al., 2018).

In recent years, Transformer models (Vaswani, 2017) have revolutionized time series modeling by introducing a self-attention mechanism capable of capturing global dependencies in data. Let us define a multivariate time series task. A input time series can be denoted as $\mathbf{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]^T \in \mathbb{R}^{m \times k}$. Especially, we call k is the input sequence length. First, Transformer embeds the input into a higher-dimensional space and augments it with positional encodings $\boldsymbol{p}$ to incorporate sequential order:

$$\boldsymbol{z}_0 = \text{Embedding}(\mathbf{x}) + \boldsymbol{p}, \tag{2}$$

where $\boldsymbol{z}_0$ is an embedding feature and each element of $\boldsymbol{z}_0$ has a predefined embedding dimension of d.

Given the initial embedding feature $\boldsymbol{z}_0$, Transformer adopts multiple layers of self-attentions. For each layer l, the self-attention mechanism computes the attention score based on query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) matrices. For instance, if we adopt linear transformations,

$$\mathbf{Q} = \boldsymbol{z}_l \mathbf{W}^Q, \quad \mathbf{K} = \boldsymbol{z}_l \mathbf{W}^K, \quad \mathbf{V} = \boldsymbol{z}_l \mathbf{W}^V, \tag{3}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$ are learnable parameters. The attention output is then computed as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}. \tag{4}$$

Given this attention output, Transformer processes the output through residual connections and layer normalization:

$$\boldsymbol{z}_{l+1} = \text{LayerNorm}(\boldsymbol{z}_l + \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})). \tag{5}$$

At last, it is often followed by a residual layer as follows:

$$\boldsymbol{z}_{l+1} = \text{LayerNorm}(\boldsymbol{z}_{l+1} + \text{FFN}(\boldsymbol{z}_{l+1})), \tag{6}$$

where FFN is a feed forward network.

Through the attention mechanism, Transformer processes entire sequences, significantly improving representation quality and performance on time series data. Indeed, the current state-of-the-art performance is achieved by variants of these Transformer models (Nie et al., 2023; Kim et al., 2024a). While several related works have explored the application of Transformers in fault diagnosis systems (Vu et al., 2024; Hou et al., 2023), in this work, we address the limitation of Transformer in terms of computational efficiency in real-world applications by manipulating the Transformer architecture to reduce its parameter count without compromising performance.

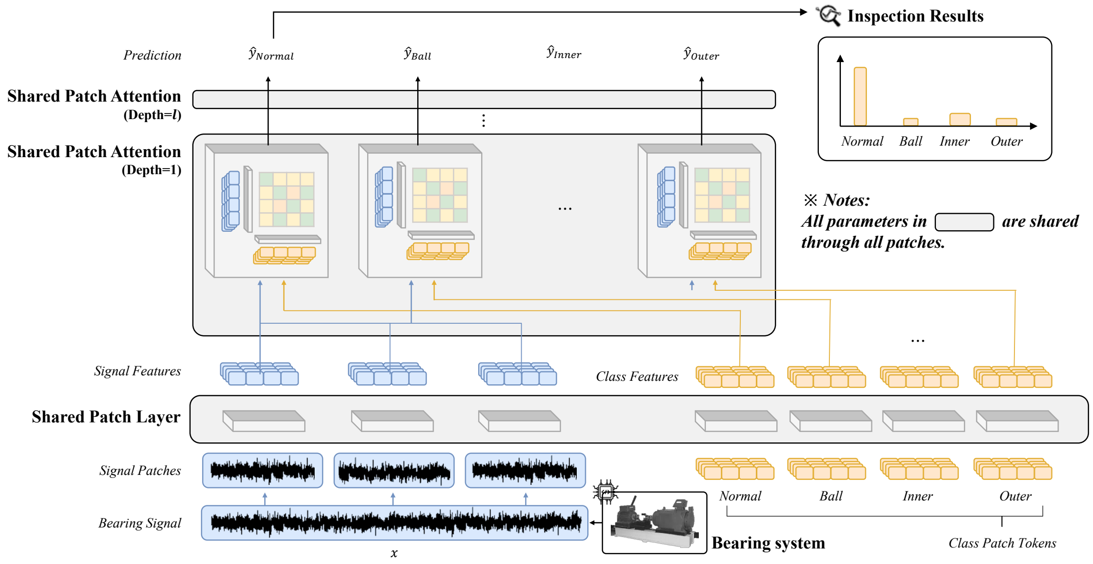
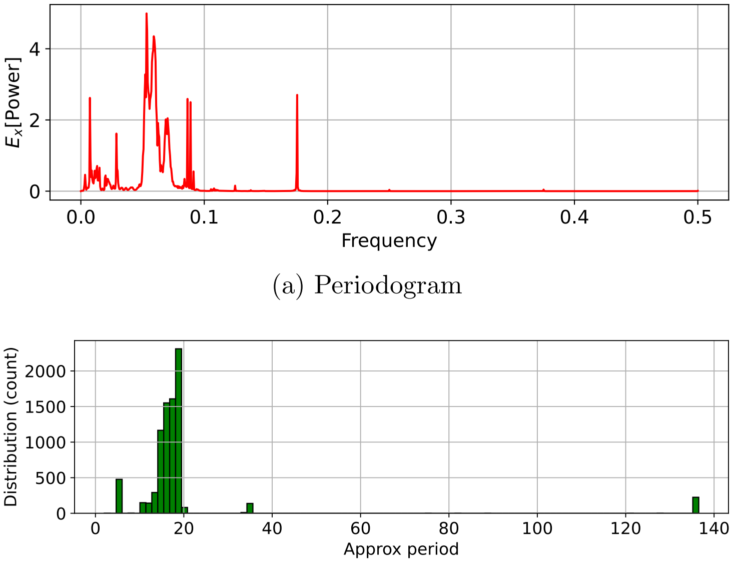


Figure 2: Illustration of the proposed architecture. The proposed model consists of two main components: (1) shared patch layer and (2) shared patch attention. Following the patching process, all parameters are aggressively shared to extract class-wise information based on class patch tokens. This results in a highly efficient model with only 7,513 parameters, fewer than the 8,196 parameters of a simple logistic regression model.

3. Methodology

3.1. Periodicity of Bearing Signals and Patches

In this section, we propose an ultra-lightweight architecture for bearing fault diagnosis. The core idea of the proposed method leverages the repetitive characteristics of bearing signals. Specifically, the original bearing signal is divided into multiple patches, and all parameters are aggressively shared after the patching process. The sharing mechanism consists of two main components: (1) shared patch layer and (2) shared patch attention. Figure 2 presents the illustration of the proposed method. The shared patch layer embeds each signal patch and class patch token into a unified embedding space to extract signal and class features. Building on this, the shared patch attention mechanism performs the attention operation using the signal and class features. Using each class feature as an input query, the shared patch attention becomes a class-specific diagnosis component. This aggressive parameter-sharing structure enables a significant reduction in the number of parameters. To the best of our knowledge, our model achieves the lowest parameter count among fault diagnosis models, with only 7,513 parameters (even fewer than 8,196 that of a simple logistic



(a) Periodogram

(b) Distribution of approx period

Figure 3: Ensuring validity of using patches. We plot (a) the expected periodogram of CWRU signals and (b) their distribution of the approximated period (i.e., 1 / peak frequency). Considering that the time dimension of each sample $k = 2048$, using patches can be effective in capturing their characteristics.

regression) while maintaining high diagnostic performance. In each subsection, we present the detail algorithm of the shared patch layer and shared patch attention.

The shared patch layer is designed to process input signals efficiently by dividing them into small patches instead of using the entire signal directly. This method can significantly reduce computational complexity and memory requirements, facilitating the development of a lighter architecture. Specifically, assuming the dimension of the original signal is k and the hidden layer dimension is h, the minimum number of parameters required in a traditional embedding layer is $k \times h$. In contrast, adopting smaller patches and a shared embedding approach reduces the parameters to kh/p using a single shared patch layer. Notably, the use of patches has recently been proven effective in time series forecasting (Nie et al., 2023), and we extend this concept to fault diagnosis systems.

To ensure the validity of using small patches instead of the original signal in the domain of fault diagnosis, we conducted an experiment utilizing the periodogram, a spectral analysis tool that reveals the frequency components of signals. Figure 3 summarizes the periodogram results derived from the training dataset of the CWRU dataset (Smith and Randall, 2015). As shown in the periodogram (Figure 3a), the frequency components are primarily concentrated below 0.2, indicating that the original signals have repetitive characteristics. Moreover, this demonstrates that small patches are effective in capturing these characteristics. Furthermore, as shown in Figure 3b, the distribution of the approximate period is less than 140. Considering the original bearing signal length of 2048, these findings further support the feasibility of leveraging small patches for fault diagnosis applications.

Given the input bearing signal $\mathbf{x} = [x_1, x_2, \cdots, x_k] \in \mathbb{R}^k$, the signal is divided into patches by splitting it with a patch length of p. We verified that various patch lengths work effectively; unless specified otherwise, we use $p = 128$, based on the observation that the approximated period is primarily under 140 (Figure 3). This process generates k/p signal patches, where the i-th patch is denoted as $\boldsymbol{x}^{(i)} = [x_{1+ip}, \cdots, x_{p+ip}] \in \mathbb{R}^p$. The shared patch layer embeds each signal patch into signal features using shared weights. This patch-based strategy ensures a compact architecture, making it particularly suitable for resource-constrained environments while maintaining high diagnostic performance.

3.2. Periodicity-Guided Patch Representation

The fundamental structure of the Transformer (Vaswani, 2017) consists of multiple layers of self-attention mechanisms after the embedding layer. Self-attention transforms extracted features into more complex representations by utilizing query, key, and value matrices. However, as highlighted in prior studies (Zeng et al., 2023; Kim et al., 2024a), self-attention exhibits limitations in preserving temporal information and incurs significant computational overhead due to the calculation of all possible attention outputs between features. We highlight that this leads to a time complexity of $\mathcal{O}(k^2/p^2)$, that is, quadratically proportional to k. To address these issues, the proposed shared patch attention leverages a cross-attention mechanism rather than self-attention mechanism.

Cross-attention mechanism uses the external feature as the query, whereas key and value are calculated from the original input feature. For the external feature $\boldsymbol{q}_l$ for l-th layer, the cross-attention mechanism can be denoted as follows:

$$\mathbf{Q} = \boldsymbol{q}_l \mathbf{W}^Q, \quad \mathbf{K} = \boldsymbol{z}_l \mathbf{W}^K, \quad \mathbf{V} = \boldsymbol{z}_l \mathbf{W}^V, \tag{7}$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \tag{8}$$

$$\boldsymbol{q}_{l+1} = \text{LayerNorm}(\boldsymbol{q}_l + \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})). \tag{9}$$

The key difference in the self-attention mechanism, as defined in Equations 3 to 5, lies in the dimension of the attention matrix, which depends on the size of $\boldsymbol{q}_l$. Specifically, the time complexity reduces to $\mathcal{O}(kn/p)$, where n represents the number of classes (i.e., $|\mathbb{Y}| = n$). Since n is significantly smaller than k/p in fault diagnosis tasks, this results in a substantial computational advantage compared to the original time complexity of self-attention, which was $\mathcal{O}(k^2/p^2)$.

To generate fault condition predictions $\hat{y}$ for each signal feature, we establish class patch tokens for each condition. These tokens are then embedded into class features using the same shared patch layer that was utilized to extract signal features. Finally, the class features are employed to extract class-wise information from the signal features through shared patch attention. Since the number of conditions n is generally smaller than the dimension of signal features d, our model achieves a significantly reduced time complexity for each layer.

Algorithm 1 Proposed Method

```
Input: Input bearing signal $\boldsymbol{x}$, patch length $p$, embedding dimension $d$, number of layers $L$, number of target conditions $n$
Output: Fault condition prediction $\hat{\boldsymbol{y}} \in \mathbb{R}^n$
Transform bearing signal into patch features
$\boldsymbol{x}^{(i)} \leftarrow [x_{1+ip}, \cdots, x_{p+ip}] \in \mathbb{R}^p$ for $i = 1, ..., \lfloor |\boldsymbol{x}|/p \rfloor$
$\boldsymbol{z}_0 \leftarrow \text{Embedding}_{\text{SharedPatchLayer}}(\boldsymbol{x}^{(i)})$
Assign class patch tokens and extract features as queries
$[\boldsymbol{q}]_{ij} \sim \mathcal{N}$ for $i = 1, ..., n$ and $j = 1, ..., p$
$\boldsymbol{q}_0 \leftarrow \text{Embedding}_{\text{SharedPatchLayer}}(\boldsymbol{q})$
for $l = 0$ to $L - 1$ do
    $\mathbf{Q}_l \leftarrow \boldsymbol{q}_l \mathbf{W}_l^Q, \quad \mathbf{K}_l \leftarrow \boldsymbol{z}_l \mathbf{W}_l^K, \quad \mathbf{V}_l \leftarrow \boldsymbol{z}_l \mathbf{W}_l^V$
    $\boldsymbol{q}_{l+1} \leftarrow \text{LayerNorm}(\boldsymbol{q}_l + \text{Attention}(\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l))$
    $\boldsymbol{q}_{l+1} \leftarrow \text{LayerNorm}(\boldsymbol{q}_{l+1} + \text{FFN}(\boldsymbol{q}_{l+1}))$
end for
$\hat{\boldsymbol{y}} \leftarrow \text{Embedding}_{\text{Proj}}(\boldsymbol{q}_L)$
return $\hat{\boldsymbol{y}}$
```

3.3. On-device Friendly Fault Diagnosis

Given the multiple layers of shared patch attention, the detailed algorithm of our proposed method is presented in Algorithm 1. The class patch tokens are initialized using the normal distribution $\mathcal{N}$ and are updated via gradient descent, similar to other model parameters. This process enables the model to effectively learn class-specific information, allowing it to determine whether the input signal patches contain faults characteristics.

Layer	**Number of Parameters**
Class patch tokens	96
Shared patch layer	3,096
Cross-attention (Attention weights)	1,848
Cross-attention (Residual weights)	2,448
Projection layer	25

Table 1: Number of model parameters for each layer

In Table 1, we summarized the number of model parameters per layer for a single layer of cross-attention with a patch length of $p = 128$ and an embedding dimension of $d = 24$. Notably, only 3,096 parameters are allocated to the shared patch layer, and 1,848 parameters to the cross-attention layer. In contrast, utilizing a vanilla Transformer for this task would require 66,048 parameters per attention layer, which is approximately 15 times more than the total parameters of our cross-attention layer (4,292). We emphasize that the shared patch attention mechanism, which shares all parameters across both signal features and class features, significantly contributes to this lightweight and efficient structure as illustrated in Figure 2.

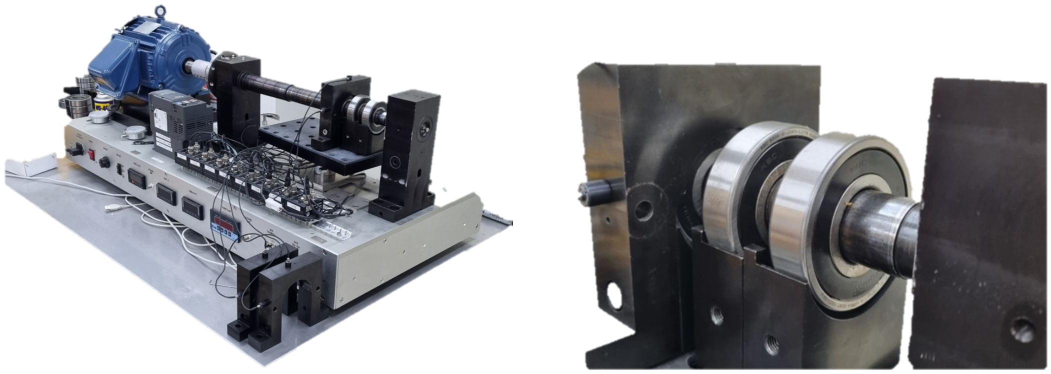
4. Experiments

In this section, we present the experimental settings and results to evaluate the performance and computational efficiency of our proposed method in comparison to other models. In Section 4.1, we describe the experimental setup and the baseline models with hyperparameters used in our experiments. Additionally, we present a new dataset designed to simulate real-world manufacturing applications with multiple components. In Section 4.2, we compare the performance of our proposed model against baseline architectures, highlighting its adaptability to resource-constrained environments, such as IoT devices. Finally, in Section 4.3, we provide an ablation study of our proposed method and suggest practical use cases for the model.

4.1. Experimental Settings

4.1.1. Datasets

As datasets, we utilized the CWRU bearing dataset, a widely recognized benchmark in the fault diagnosis domain. The vibration signals in the CWRU dataset were recorded at sampling rates of 12kHz and 48kHz under four motor load conditions. The dataset includes three different fault conditions: inner race, outer race, and ball faults. Each with fault diameters of 0.007, 0.014, and 0.021 inches, respectively. For



(a) Simulated bearing system

(b) Bearings used to generate the dataset

Figure 4: Simulated bearing fault diagnosis system for gathering a new dataset to mimic real-world applications. We used three bearings and make three different fault states.

our study, we used the signals collected from the drive-end accelerometer at a 48kHz sampling rate. Fault states with different fault diameters and motor loads as equivalent, resulting in four different states $\mathbb{Y}$: one normal state ($y = 0$) and three fault states ($y = 1$ for ball fault; 2 for inner fault; 3 for outer fault). The CWRU dataset contains a total of 8,168 samples. Following prior work (Lee et al., 2025; Kim et al., 2024b), we first split the original dataset into a 7:3 ratio, creating the training and testing datasets. Then, 20% of the training dataset is used as the validation dataset. This results in 4,573 training samples, 1,144 validation samples, and 2,451 testing samples. We fixed the random seed to 42 to ensure reproducibility.

To conduct practical evaluations of our model, we developed a new bearing fault diagnosis dataset using a realistic bearing fault diagnosis system. As illustrated in Figure 4, the setup replicates a manufacturing system with three distinct bearings. Using this system, we collected bearing signals under various conditions. The bearings have inner and outer diameters of 35mm and 80mm, respectively. The system operated at a rotational speed of 900 RPM with a vertical load of 200 kgf, and vibration signals were recorded at a sampling rate of 10 kHz.

The dataset assumes four conditions: normal, ball fault, inner race fault, and outer race fault, which are commonly considered in previous studies. Each class has the same number of samples. To ease comparison, we use the same labeling set as in the CWRU dataset. It is important to note that only signals from the first bearing of the rotating machinery were used. For simplicity, we refer to this dataset as the SB (Simulated Bearing) dataset. The dataset comprises a total of 3,512 samples, which were divided using the same splitting method as in the CWRU dataset. This resulted in 1,996 training samples, 492 validation samples, and 1,056 testing samples.

For both datasets, the bearing signal was normalized to a range of -1 to 1.

4.1.2. Baselines and Hyperparameters

We adopt a range of methods from both machine learning and deep learning as baselines for comparison. For traditional machine learning approaches, we use (1) Logistic Regression (LR) and (2) XGBoost. For deep learning approaches, following prior studies (Zhang et al., 2020; Zheng et al., 2021; Li et al., 2022; Zhang et al., 2017; Chen et al., 2020), we include seven different methods: (3) Multi-Layer Perceptron (MLP), (4) Temporal Convolutional Network (TCN) (Zheng et al., 2021), (5) Fully Convolutional Network combined with Long Short-Term Memory (LSTM-FCN) (Li et al., 2022), (6) Gated Recurrent Unit combined with Fully Convolutional Network (GRU-FCN), (7) Deep Convolutional Neural Network with a wide first-layer kernel (WDCNN) (Zhang et al., 2017), (8) a variant of one-dimensional Convolutional Neural Networks (CNN1D) (Chen et al., 2020), and (9) Transformer. In total, we trained and evaluated 10 different models, including our proposed method.

For hyperparameters, we primarily use the default scikit-learn settings for multi-label classification in machine learning approaches. For logistic regression, we employ the lbfgs solver with a maximum of $3,000$ iterations and balanced class weights. For XGBoost, the model is configured with 100 estimators and a maximum depth of 4. For deep learning methods, we adopt the same architecture as described in prior studies. To utilize the vanilla Transformer, we reshape the input to mimic a multi-dimensional time series format with a sequence length of 128, which demonstrates improved performance compared to the standard vanilla implementation. For our model, we adopt patch $p = 128$ and embedding dimension $d = 24$ with a single cross-attention layer $L = 1$ with 8 multi-heads. Dropout is set to 0.1. Training settings are unified across models, utilizing the Adam optimizer with a learning rate of 1e-3 and an automatic learning rate scheduler (Defazio et al., 2024). We observe that 300 epochs are sufficient to achieve optimal performance for each model. During training, we apply an early stopping mechanism with a patience check and select the best-performing model based on validation data.

4.1.3. Evaluation Metric

To evaluate the performance of the proposed method, we utilized standard classification metrics, including accuracy (Acc.), recall (Rec.), precision (Pre.), and F1 score (F1). Accuracy measures the ratio of correctly predicted samples to the total number of samples, providing an overall measure of performance. Recall calculates the proportion of actual positive samples (i.e., normal samples) correctly identified by the model. Precision evaluates the proportion of predicted positive samples that

Table 2: Performance comparison of various models on the CWRU dataset. The proposed method achieves the highest performance while utilizing the fewest parameters.

Model Info.		Performance			
Model	#Params	Acc.	Rec.	Pre.	F1
LR	8,196	0.3219	0.3108	0.3120	0.3110
XGBoost	9,200	0.7952	0.8198	0.8316	0.8237
MLP	591,364	0.8303	0.8573	0.8591	0.8581
TCN	66,754	0.9368	0.9466	0.9486	0.9475
LSTM-FCN	332,804	0.9257	0.9386	0.9391	0.9387
GRU-FCN	316,036	0.9368	0.9477	0.9484	0.9478
WDCNN	60,424	0.9600	0.9666	0.9670	0.9668
CNN1D	82,924	0.9021	0.9186	0.9195	0.9191
Transformer	118,532	0.9433	0.9530	0.9530	0.9530
Ours	**7,513**	**0.9706**	**0.9755**	**0.9755**	**0.9755**

are actually positive. The F1 score, defined as the harmonic mean of precision and recall, offers a balanced assessment.

4.2. Performance and Resource Comparison

We first compared the performance of all methods alongside their number of parameters (#Params). Tables 2 and 3 summarize the results on the CWRU and SB datasets, respectively. As shown in the tables, the proposed model (Ours) consistently outperforms all baseline methods, achieving the highest accuracy, recall, precision, and F1 score on both datasets.

However, our model requires approximately 2% of the parameters used by LSTM-FCN, 6% of the parameters used by the Transformer, and 12.5% of the parameters required by WDCNN. Despite having significantly fewer parameters, the proposed method achieves superior performance, indicating that it efficiently extracts meaningful features for fault diagnosis without relying on large model capacities.

Moreover, the results on the SB dataset further validate the robustness and generalizability of our method. Similar to the CWRU dataset, the proposed model demonstrates the highest performance across all metrics, verifying its capability to handle diverse datasets and conditions. These findings highlight that our approach achieves state-of-the-art performance while maintaining a compact and computationally efficient design.

Table 3: Performance comparison of various models on the SB dataset. The proposed method achieves the highest performance while utilizing the fewest parameters.

Model Info.		Performance			
Model	#Params	Acc.	Rec.	Pre.	F1
LR	8,196	0.2846	0.2882	0.2905	0.2840
XGBoost	9,200	0.6404	0.6447	0.6450	0.6433
MLP	591,364	0.7837	0.7878	0.7979	0.7889
TCN	66,754	0.9877	0.9874	0.9874	0.9874
LSTM-FCN	332,804	0.9668	0.9672	0.9666	0.9668
GRU-FCN	316,036	0.9668	0.9667	0.9662	0.9663
WDCNN	60,424	0.9877	0.9875	0.9873	0.9873
CNN1D	82,924	0.9677	0.9678	0.9670	0.9672
Transformer	118,532	0.9649	0.9647	0.9639	0.9639
Ours	**7,513**	**0.9915**	**0.9913**	**0.9913**	**0.9913**

Table 4: Comparison of FLOPs (in kFLOPs) for different batch sizes. Our model achieves the lowest computational cost.

Model	Batch-size (B)					
	$B = 1$	$B = 2$	$B = 4$	$B = 8$	$B = 16$	$B = 32$
MLP	590.8	1,181.7	2,363.4	4,726.8	9,453.6	18,907.1
TCN	148,172.0	296,344.0	592,688.0	1,185,376.0	2,370,752.0	4,741,504.0
LSTM-FCN	537,968.1	1,075,936.3	2,151,872.5	4,303,745.0	8,607,490.0	17,214,980.1
GRU-FCN	537,968.1	1,075,936.3	2,151,872.5	4,303,745.0	8,607,490.0	17,214,980.1
WDCNN	180.4	360.8	721.6	1,443.2	2,886.4	5,772.8
CNN1D	75,656.2	151,312.4	302,624.9	605,249.8	1,210,499.6	2,420,999.2
Transformer	12,968.4	25,936.9	51,873.8	103,747.6	207,495.2	414,990.3
Ours	**153.9**	**307.8**	**615.6**	**1,231.1**	**2,462.2**	**4,924.4**

Table 4 compares the computational cost of different models in terms of FLOPs (in kFLOPs) for varying batch sizes (B). The results demonstrate that our proposed model achieves the lowest computational cost across all batch sizes. For example, at $B = 32$, our model requires only 4,924.4 kFLOPs, which is approximately 3.8 times less than the WDCNN and over 3,500 times less than the LSTM-FCN. This substantial reduction in computational overhead highlights the efficiency of our method, making it highly suitable for real-time and resource-constrained applications.

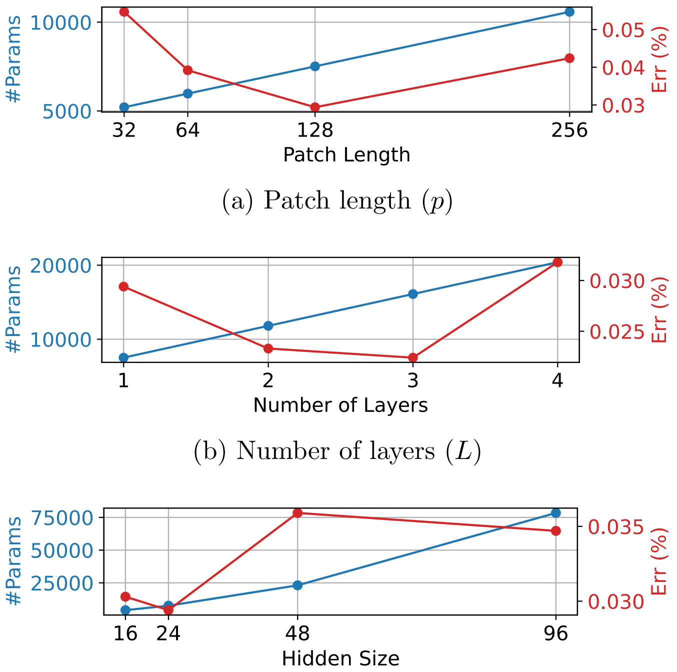
Table 5: Model Compatibility with Practical Devices

Device	LR	XGBoost	MLP	TCN	Ours
Raspberry Pi 4 Series	✓	✓	✓	✓	✓
Intel Xeon W Series	✓	✓	✓	✓	✓
Qualcomm Snapdragon 845	✓	✓	✓	✓	✓
Texas Instruments TDA4VM	✓	✓	✓	✓	✓
Google Coral Dev Board	✓	✓	✗	✓	✓
ALIF DevKit-E7 Gen2	✓	✓	✗	✗	✓
Arduino Nicla Vision	✓	✗	✗	✗	✓
STMicro STM32H7	✓	✗	✗	✗	✓
Espressif ESP32-S3	✓	✗	✗	✗	✓

Lastly, in Table 5, we provide a summary of model compatibility with practical devices in real-world systems. We assume 20% of the total memory is reserved for the operating system and frameworks, with an additional 20% allocated for activation memory. Based on these assumptions, we calculate the maximum number of parameters each model can support to generate predictions for a single example. Since LSTM-FCN, GRU-FCN, WDCNN, CNN1D, and Transformer exhibit the same trends as TCN, their results are summarized collectively under TCN. Among all the models, only LR and our method are compatible with all devices. However, given the poor predictive performance of LR, we can strongly recommend using our method for practical deployments, as it offers both superior performance and broader device compatibility.

4.3. Ablation Study

To evaluate the impact of key components of our model on the performance and computational efficiency, we conducted an ablation study by varying patch length, the number of layers, and hidden size, as summarized in Figure 5. The results show that a patch length p of 128 achieves the best balance between performance and parameter count. This can be further analyzed for various datasets including other fault diagnosis systems. For the number of layers L, the best performance is achieved for $L = 3$, but increasing L does not always improve the performance. To adjust model size without sacrificing accuracy, we recommend changing the number of layers, as these configurations maintain high performance while minimizing computational costs.



(a) Patch length (p)

(b) Number of layers (L)

(c) Hidden size (h)

Figure 5: Ablation study on the proposed model. We conduct the experiment with varying patch length, number of layer, and hidden size.

5. Conclusion

In this paper, we proposed a lightweight transformer-based fault diagnosis model leveraging aggressive patch layer sharing. Through extensive experiments, we demonstrated that our model outperforms baseline methods for diverse measures with significantly fewer parameters. These results emphasize the suitability of our approach for real-world, resource-constrained environments, such as IoT devices in industrial systems. As future work, we will explore extending this framework to other domains and further optimizing the trade-offs between accuracy and efficiency.

References

Bai, S., Kolter, J.Z., Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 .

Chen, C.C., Liu, Z., Yang, G., Wu, C.C., Ye, Q., 2020. An improved fault diagnosis using 1d-convolutional neural network model. Electronics 10, 59.

Cheng, S., Song, J., Zhou, M., Wei, X., Pu, H., Luo, J., Jia, W., 2024. Ef-detr: A lightweight transformer-based object detector with an encoder-free neck. IEEE Transactions on Industrial Informatics .

Defazio, A., Yang, X.A., Khaled, A., Mishchenko, K., Mehta, H., Cutkosky, A., 2024. The road less scheduled, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems. URL: https://openreview.net/forum?id=0XeNkkENuI.

Elsayed, N., Maida, A.S., Bayoumi, M., 2018. Deep gated recurrent and convolutional network hybrid model for univariate time series classification. arXiv preprint arXiv:1812.07683 .

Hou, Y., Wang, J., Chen, Z., Ma, J., Li, T., 2023. Diagnosisformer: An efficient rolling bearing fault diagnosis method based on improved transformer. Engineering Applications of Artificial Intelligence 124, 106507.

Jia, F., Lei, Y., Lin, J., Zhou, X., Lu, N., 2016. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. Mechanical systems and signal processing 72, 303–315.

Karim, F., Majumdar, S., Darabi, H., Chen, S., 2017. Lstm fully convolutional networks for time series classification. IEEE access 6, 1662–1669.

Kim, D., Park, J., Lee, J., Kim, H., 2024a. Are self-attentions effective for time series forecasting?, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems. URL: https://openreview.net/forum?id=iN43sJoib7.

Kim, H., Lee, S., Lee, J., Lee, W., Son, Y., 2024b. Evaluating practical adversarial robustness of fault diagnosis systems via spectrogram-aware ensemble method. Engineering Applications of Artificial Intelligence 130, 107980.

Lanham, C., 2002. Understanding the tests that are recommended for electric motor predictive maintenance. Baker Instrument Company .

Lee, S., Kim, H., Lee, W., Son, Y., 2025. Black-box adversarial examples via frequency distortion against fault diagnosis systems. Applied Soft Computing , 112828.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., Nandi, A.K., 2020. Applications of machine learning to machine fault diagnosis: A review and roadmap. Mechanical Systems and Signal Processing 138, 106587.

Li, Y., Zou, W., Jiang, L., 2022. Fault diagnosis of rotating machinery based on combination of wasserstein generative adversarial networks and long short term memory fully convolutional network. Measurement 191, 110826.

Liu, W., Xu, X., Qi, L., Zhou, X., Yan, H., Xia, X., Dou, W., 2024. Digital twin-assisted edge service caching for consumer electronics manufacturing. IEEE Transactions on Consumer Electronics .

Ma, Y., Wu, X., 2018. Elastic net representation-based projections for bearing fault classification, in: 2018 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE. pp. 1–7.

Mao, W., Tian, S., Liang, X., He, J., 2018. Online bearing fault diagnosis using support vector machine and stacked auto-encoder, in: 2018 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE. pp. 1–7.

Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J., 2023. A time series is worth 64 words: Long-term forecasting with transformers, in: The Eleventh International Conference on Learning Representations. URL: https://openreview.net/forum?id=Jbdc0vTOcol.

Paliwal, D., Choudhur, A., Govandhan, T., 2014. Identification of faults through wavelet transform vis-à-vis fast fourier transform of noisy vibration signals emanated from defective rolling element bearings. Frontiers of Mechanical Engineering 9, 130–141.

Rauber, T.W., de Assis Boldt, F., Varejao, F.M., 2014. Heterogeneous feature models and feature selection applied to bearing fault diagnosis. IEEE Transactions on Industrial Electronics 62, 637–646.

Razavi-Far, R., Farajzadeh-Zanjani, M., Saif, M., 2017. An integrated class-imbalanced learning scheme for diagnosing bearing defects in induction motors. IEEE Transactions on Industrial Informatics 13, 2758–2769.

Sangaiah, A.K., Medhane, D.V., Han, T., Hossain, M.S., Muhammad, G., 2019. Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics. IEEE Transactions on Industrial Informatics 15, 4189–4196.

Smith, W.A., Randall, R.B., 2015. Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. Mechanical systems and signal processing 64, 100–131.

Vaswani, A., 2017. Attention is all you need. Advances in Neural Information Processing Systems .

Vu, M.H., Nguyen, V.Q., Tran, T.T., Pham, V.T., Lo, M.T., 2024. Few-shot bearing fault diagnosis via ensembling transformer-based model with mahalanobis distance metric learning from multiscale features. IEEE Transactions on Instrumentation and Measurement .

Yang, H., Mathew, J., Ma, L., 2005. Fault diagnosis of rolling element bearings using basis pursuit. Mechanical Systems and Signal Processing 19, 341–356.

Zeng, A., Chen, M., Zhang, L., Xu, Q., 2023. Are transformers effective for time series forecasting?, in: Proceedings of the AAAI conference on artificial intelligence, pp. 11121–11128.

Zhang, S., Zhang, S., Wang, B., Habetler, T.G., 2020. Deep learning algorithms for bearing fault diagnostics—a comprehensive review. IEEE Access 8, 29857–29881.

Zhang, W., Peng, G., Li, C., Chen, Y., Zhang, Z., 2017. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. Sensors 17, 425.

Zhao, Z., Liu, P.X., Gao, J., 2022. Model-based fault diagnosis methods for systems with stochastic process–a survey. Neurocomputing 513, 137–152.

Zheng, H., Wu, Z., Duan, S., Chen, Y., 2021. Research on fault diagnosis method of rolling bearing based on tcn, in: 2021 12th International Conference on Mechanical and Aerospace Engineering (ICMAE), IEEE. pp. 489–493.

Zhuo, Y., Ge, Z., 2021. Data guardian: A data protection scheme for industrial monitoring systems. IEEE Transactions on Industrial Informatics 18, 2550–2559.