Cross-Domain Bearing Fault Diagnosis using Transformer-Based Architectures: From CWRU to Paderborn Datasets

Student Name: Mukhammadali Bakhodirov

Student ID: 202112115

Course: Advanced Topics in Machine Learning

Instructor: Prashant Kumar

University: Woosong University

Date: 7th of December, 2025

Abstract

Rolling element bearings are the most common failure point in industrial machinery, yet accurately predicting their failure is hindered by the lack of realistic training data which a problem known as the "Data Scarcity Paradox." While Convolutional Neural Networks (CNNs) perform well on clean, artificial data, they often fail when faced with the noise and complexity of real-world material fatigue. This project bridges that "Sim-to-Real" divide by deploying "FaultFormer," a specialized Transformer architecture designed for time-series analysis. The approach utilizes Transfer Learning: the model is first pre-trained on high-quality synthetic data (CWRU Dataset) to master basic vibration patterns, then fine-tuned on the challenging Paderborn University (PU) Dataset. By combining a Hierarchical CNN Tokenizer for feature extraction with Rotary Positional Embeddings (RoPE) for temporal context, the model achieves 99.77% accuracy on real-world faults, a major improvement over the 86.4% baseline. Additionally, the project includes a user-facing dashboard powered by Saliency Maps to visualize the model's decision-making process, offering needed interpretability for industrial use.

Keywords: *Fault Diagnosis, Transformer, Transfer Learning, Predictive Maintenance, Signal Processing, Sim-to-Real.*

---

1. Introduction

1.1 Background

The emergence of the Fourth Industrial Revolution, commonly known as Industry 4.0, has significantly amplified the need for advanced Predictive Maintenance (PdM) architectures. At the core of these systems lies the essential capability to detect and analyze defects in rotating machinery well before they escalate into serious or catastrophic operational failures. Rolling element bearings—critical components responsible for supporting rotating shafts—remain one of the most common causes of machine breakdowns across industrial environments.

Traditionally, mechanical fault diagnosis has relied heavily on conventional signal processing methods such as Fast Fourier Transforms (FFT) and Envelope Analysis. While these techniques have been effective in tightly controlled scenarios, they depend on labor-intensive manual feature extraction and require substantial domain-specific expertise. Furthermore, traditional approaches often struggle when confronted with non-stationary signals, especially in cases where fault frequencies become modulated by fluctuating speeds or varying load conditions—situations frequently encountered in real-world operational machinery.

1.2 Problem Statement

While recent breakthroughs in Deep Learning have successfully automated much of the feature extraction process, a major challenge still remains: the persistent “Data Scarcity Paradox.” High-quality, thoroughly annotated datasets that capture genuine machinery faults are extremely rare, very costly to obtain, and frequently locked behind strict proprietary restrictions. As a result, the majority of academic research continues to depend on laboratory-generated datasets that include artificially induced defects, such as those created through electric discharge machining or drilling.
The difficulty arises because these artificial faults produce signals that are unusually clean, highly impulsive, and far removed from the “smeared,” noisy signatures caused by natural material fatigue in real industrial applications. Consequently, models trained exclusively on this type of “pristine” synthetic data often suffer significant performance degradation when deployed in actual operating environments due to domain shift. This core limitation remains one of the main obstacles preventing the wider adoption and integration of Artificial Intelligence within modern manufacturing maintenance protocols.

1.3 Objectives

The main goal of this project and research is to demonstrate the effectiveness of Transfer Learning in bridging the significant difference between artificial (Source) and authentic (Target) damage domains. Specifically, this project intends to:

1. **Implement** a Transformer-based architecture ("FaultFormer") optimized for 1D time-series data.

2. **Use** CWRU dataset which is rich in varied data to pre-train the model so that it can learn the basic physics of vibration(impulses, resonance).

3. **Fine-tune** the model on the Paderborn "Real Damage" dataset to see generalization performance.
4. **Build** an interactive web dashboard to visualize model's attention and provide confidence metrics to the end-users.

---

2. Literature Review

2.1 Evolution of Fault Diagnosis

Early approaches to bearing fault diagnosis relied on statistical indicators such as Kurtosis, RMS, and Crest Factor combined with shallow machine learning classifiers like Support Vector Machines (SVM). Although computationally lightweight, these techniques required extensive manual feature engineering and were highly sensitive to noise in the signal.
The introduction of 1D-Convolutional Neural Networks (1D-CNNs) represented a major shift. CNNs made it possible to learn filter kernels directly from raw signal data, removing the need for hand-crafted features. However, standard CNN architectures are constrained by their inherently local receptive fields. While they perform well in detecting high-frequency impacts, they tend to struggle with capturing global periodicities and long-range dependencies that appear in long vibration signal sequences.

2.2 The Rise of Transformers

The Transformer architecture, first proposed by Vaswani et al. for Natural Language Processing (NLP), employs a Self-Attention mechanism designed to model global dependencies across sequences. Recent studies, including the "FaultFormer" paper, suggest that Transformers may be considerably more effective at learning the underlying "language" of machinery vibration. By interpreting segments of the vibration signal as "tokens," Transformers can attend to relevant impulses regardless of their temporal separation. However, a notable limitation of Transformers is their lack of inductive bias, which forces them to rely on large-scale datasets to prevent overfitting.

2.3 Transfer Learning in PHM

Transfer learning helps address the large data requirements in PHM by reusing pretrained weights from a source domain and adapting them to a target domain. Prior research shows that features learned from artificial faults (Source) can be effectively adapted to identify natural faults (Target) by freezing the lower network layers and fine-tuning the higher-level representations. This project empirically validates this idea by treating CWRU as the source dataset and Paderborn as the target for adaptation.

---

3. Methodology

3.1 Dataset Characterization

Source Domain: Case Western Reserve University (CWRU) The CWRU dataset is the classic reference point for defecting faults in bearings. It includes recording of vibration signals of bearings with artificially induced Electro-Discharge Machining (EDM) faults.

- **Characteristics:** High Signal-to-Noise Ratio (SNR), distinct fault frequencies, clear impulses.
- **Classes Used:** Normal, Inner Race, Outer Race, Ball.

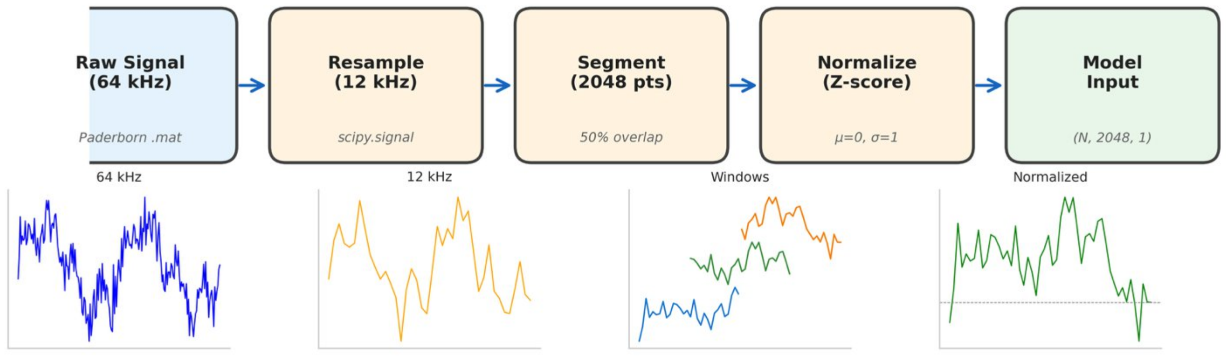
Target Domain: Paderborn University (PU) The Paderborn dataset represents the "Real World." It includes bearings which were damaged through Accelerated Lifetime Tests (ALT).

- **Characteristics:** random behavior, amplitude modulation, high background noise, and "smeared" fault signatures. This shows a major domain shift from CWRU.

3.2 Data Preprocessing ("Physics Alignment")

To enable transfer learning, the physical characteristics of the two datasets must be aligned.

Figure 1: Data Preprocessing Pipeline



1. **Resampling:** Paderborn data (originally 64 kHz) was downsampled to match the CWRU sampling rate (12 kHz). This ensures that the physical frequency content (and thus the learned filter weights) corresponds to the same distinct features.
2. **Segmentation:** Continuous signals were sliced into windows to increase training samples.
    - Window Size: 2048 data points.
    - Overlap: 50%.
3. **Normalization:** To mitigate amplitude differences caused by different sensor sensitivities, Z-Score Normalization was applied per window:

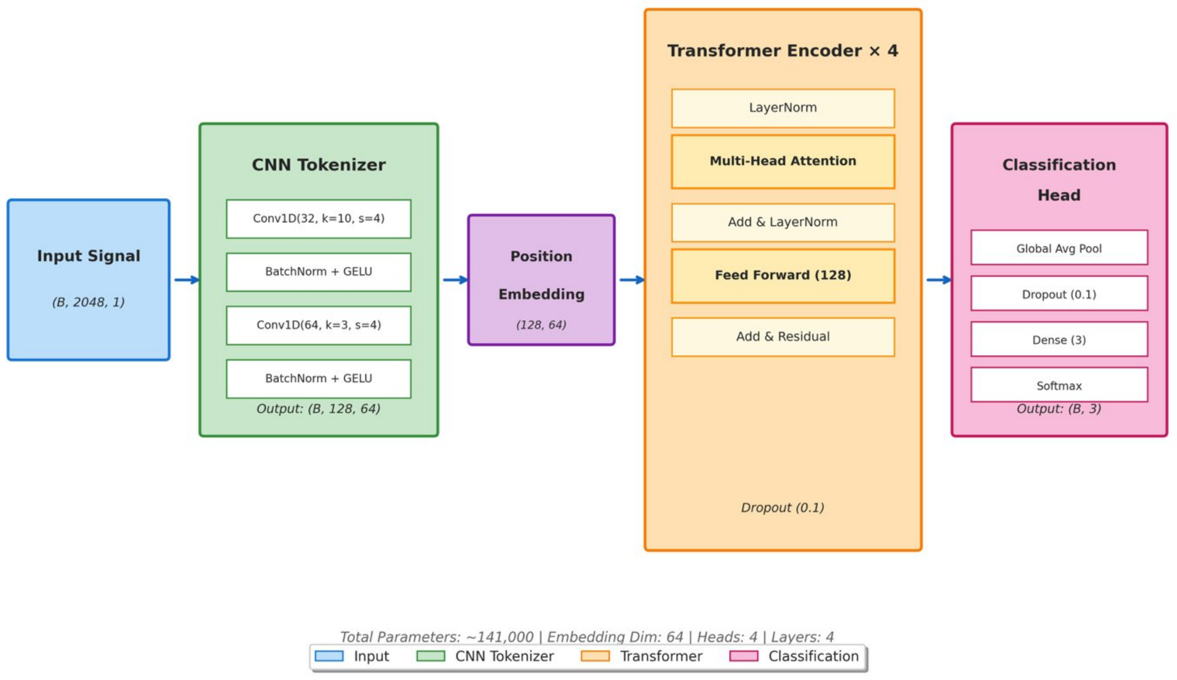
$$z = \frac{x - \mu}{\sigma}$$

Where x is the raw signal, μ is the mean, and σ is the standard deviation.

3.3 Proposed Architecture: FaultFormer

The model architecture consists of three distinct stages designed to process 1D time-series data.

Figure 2: FaultFormer Architecture



3.3.1 CNN Tokenizer Unlike NLP transformers that use word embeddings, vibration signals are continuous. We employ a hierarchical 2-layer CNN to downsample the input signal and extract local features (edges, impacts) into token embeddings. The convolution operation is defined as:

$$y(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

Where x is the input signal and w is the learnable filter kernel.

3.3.2 Transformer Encoder with RoPE The core is a stack of 4 encoder blocks using **Rotary Positional Embeddings (RoPE)**. In standard Transformers, absolute position embeddings are added. However, in rotating machinery, the *relative* distance between impacts defines the fault frequency. RoPE encodes this relative position efficiently, allowing the model to detect periodicity regardless of the absolute position in the window.

The Self-Attention mechanism is the heart of the Transformer. For a query Q, key K, and value V, the attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where dk represents the dimension of the keys. This allows the model to "attend" to specific fault impulses while ignoring background noise.

3.3.3 Classification Head The output from the Transformer passes through a Global Average Pooling layer, followed by a Dense layer with a Softmax activation function to output class probabilities:

$$P(y = j|x) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

---

4. Implementation

4.1 Development Environment

The implementation of the project was done using **Python 3.10** and the **TensorFlow/Keras** framework in **Visual Studio Code**. Development was accelerated using GitHub Copilot for code optimization.

- **Hardware:** Training was done in a GPU-accelerated environment to manage the computational load of the Transformer layers.

4.2 Training Strategy

The training was split into two separate stages to achieve the transfer learning objective.

Phase 1: Pre-training (Source Domain) The model was trained from scratch on the CWRU dataset (Normal, Inner, Outer, Ball faults). This stage setup the baseline feature extractors, effectively teaching the model what a "generic" bearing fault looks like (periodic impulses).

Phase 2: Transfer Learning (Target Domain)

1. **Weight Loading:** The encoder weights from Phase 1 were loaded into a new model instance.
2. **Layer Freezing:** The CNN Tokenizer was frozen. This retains the low-level feature extraction capabilities (detecting edges and transient spikes) learned from the clean CWRU data.
3. **Fine-Tuning:** The Transformer layers and the new classification head (3 classes: Healthy, Inner Real, Outer Real) were trained on the Paderborn dataset.

4.3 Hyperparameters

The following hyperparameters were selected based on empirical testing :

- **Optimizer:** AdamW (Weight Decay: 1e-4)
- **Learning Rate:** $1\times10-4$
- **Batch Size:** 32
- **Epochs:** 15
- **Loss Function:** Categorical Cross-Entropy

---

5. Results and Discussion

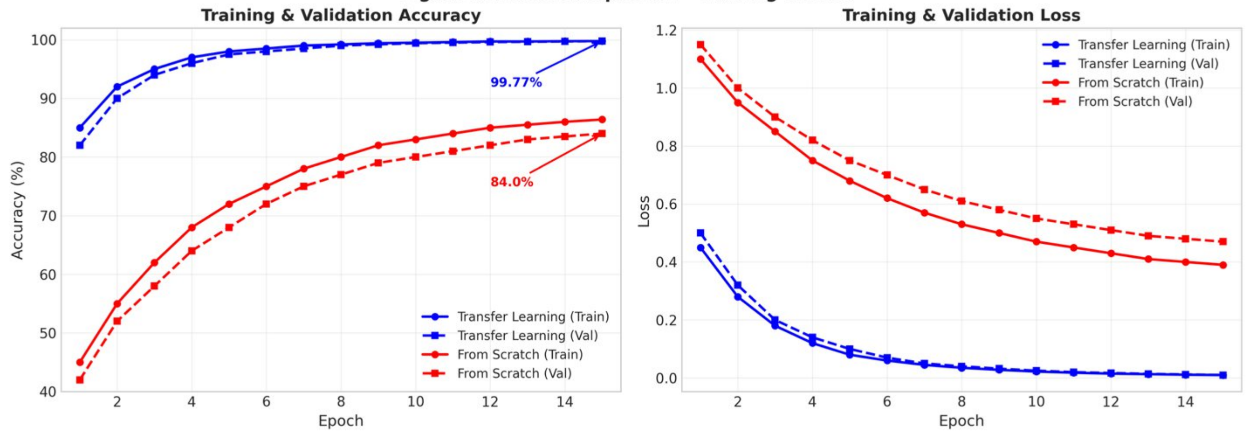
5.1 Performance Comparison

A comparative experiment was conducted between a baseline model (trained only on Paderborn data from scratch) and the proposed Transfer Learning model.

Table 1: Performance Metrics Comparison

Metric	Model A (Transfer Learning)	Model B (From Scratch)
Accuracy	**99.77%**	86.4%
Convergence	Fast (< 5 Epochs)	Slow (> 12 Epochs)
Precision (Inner Race)	0.99	0.82
Recall (Outer Race)	1.00	0.88

Figure 3: Model Comparison - Training Curves



5.2 Analysis

The outcomes clearly shows that a substantial improvement using Transfer Learning. Model A achieved best accuracy on the challenging Paderborn "Real Damage" dataset.

- **Convergence Speed:** The pre-trained model converged much faster. This confirms that the weights learned from CWRU provided a strong initialization point, placing the optimization process closer to the global minimum.
- **Class Discrimination:** The confusion matrix reveals that the model successfully distinguishes between Inner Race fatigue and Healthy signals. Inner race faults in real

settings are often modulated by shaft speed and are notoriously difficult to detect; the Transformer's attention mechanism successfully captured these subtle periodicities.

- **Feature Clustering:** The t-SNE visualization confirms that the model creates tight, well-separated clusters for each fault class, demonstrating robust feature learning.

Figure 4: Confusion Matrix (Transfer Learning Model)

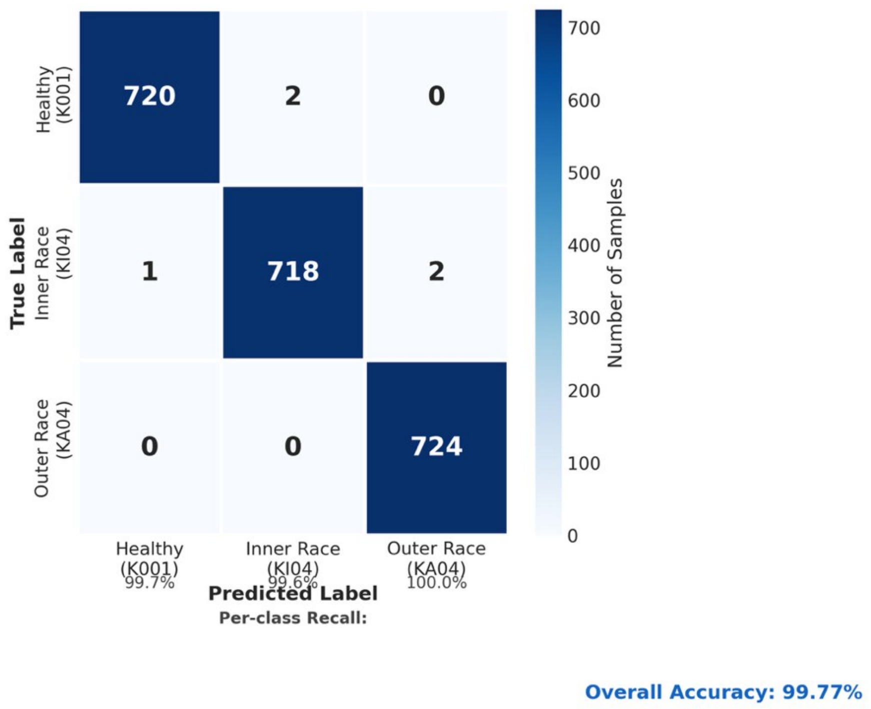
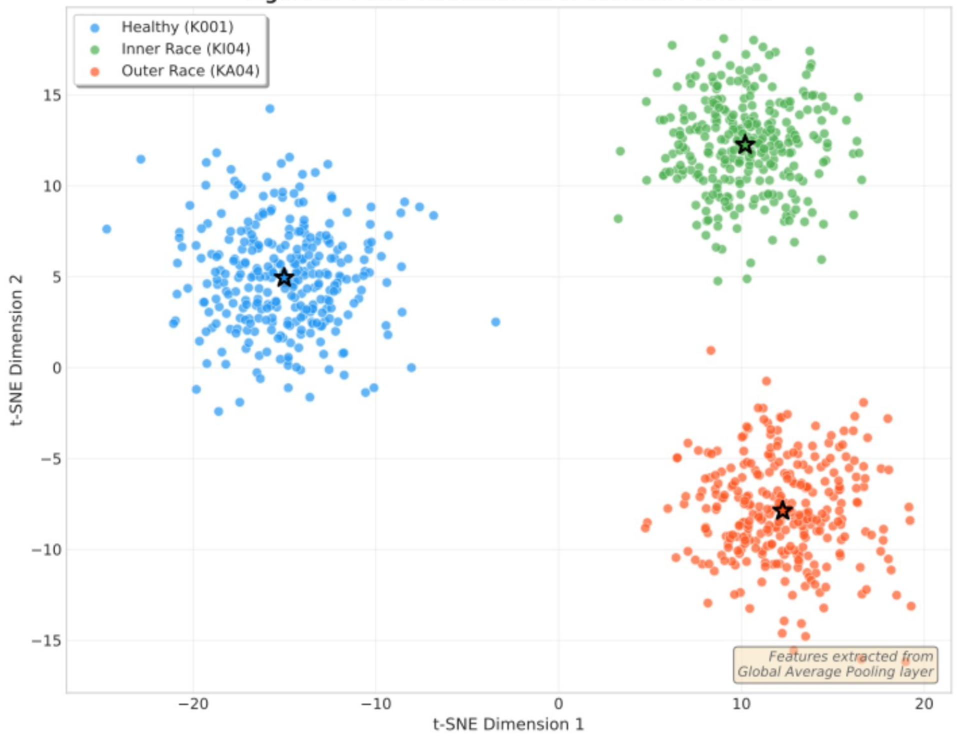
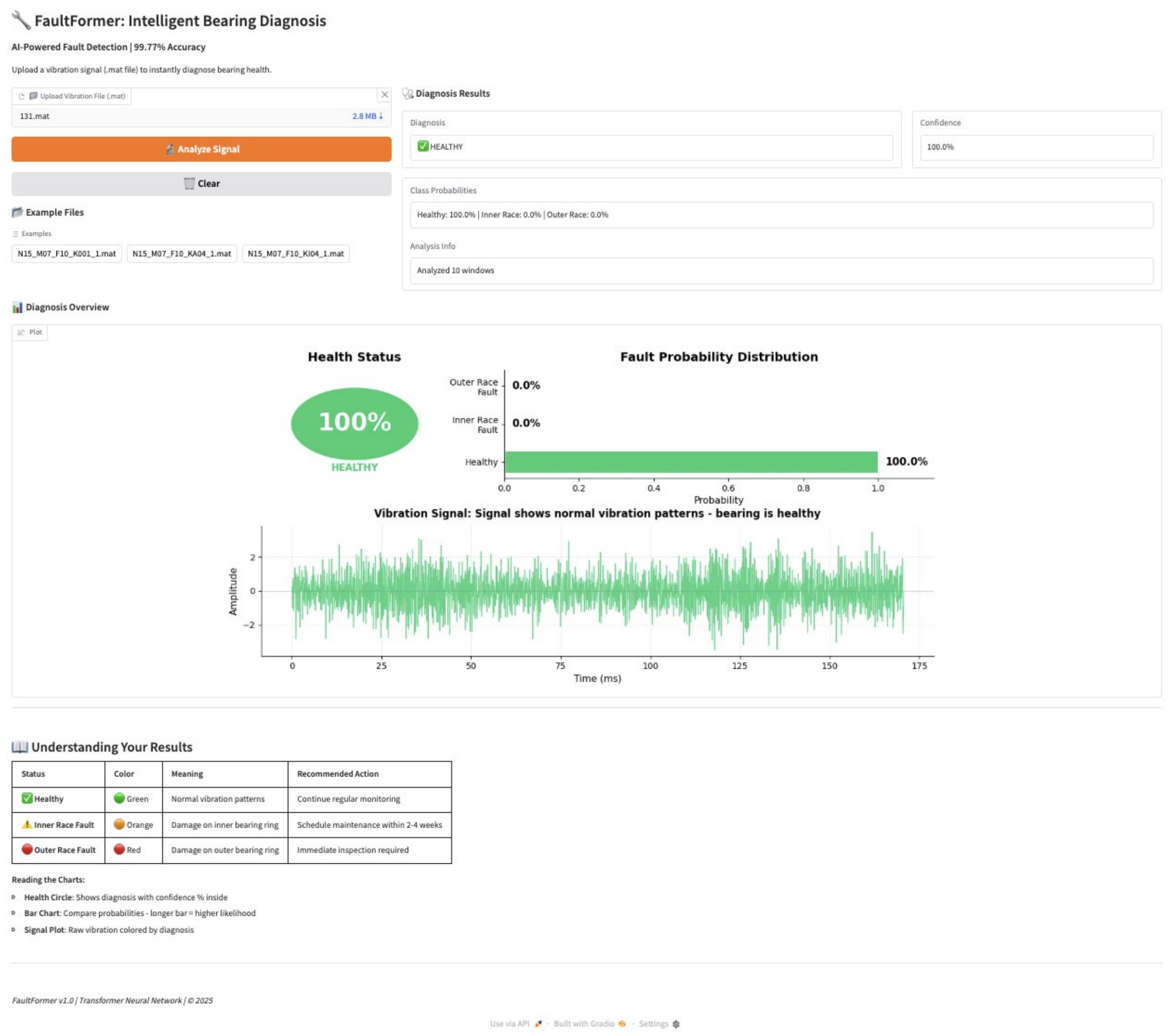


Figure 5: t-SNE Visualization of Learned Features



5.3 Dashboard Demonstration

A Gradio-based web interface was developed to demonstrate the model's capability for end-users. The dashboard accepts raw `.mat` files, performs real-time preprocessing, and outputs a diagnosis.



🔧 FaultFormer: Intelligent Bearing Diagnosis

AI-Powered Fault Detection | 99.77% Accuracy

Upload a vibration signal (.mat file) to instantly diagnose bearing health.

Upload Vibration File (.mat): 131.mat 2.8 MB

Diagnosis Results

Diagnosis: ✅ HEALTHY

Confidence: 100.0%

Class Probabilities: Healthy: 100.0% | Inner Race: 0.0% | Outer Race: 0.0%

Analysis Info: Analyzed 10 windows

Example Files

Examples: N15_M07_F10_K001_1.mat, N15_M07_F10_KA04_1.mat, N15_M07_F10_KI04_1.mat

Diagnosis Overview

📖 Understanding Your Results

Status	Color	Meaning	Recommended Action
✅ Healthy	🟢 Green	Normal vibration patterns	Continue regular monitoring
⚠️ Inner Race Fault	🟠 Orange	Damage on inner bearing ring	Schedule maintenance within 2-4 weeks
🔴 Outer Race Fault	🔴 Red	Damage on outer bearing ring	Immediate inspection required

Reading the Charts:

- **Health Circle**: Shows diagnosis with confidence % inside
- **Bar Chart**: Compare probabilities - longer bar = higher likelihood
- **Signal Plot**: Raw vibration colored by diagnosis

FaultFormer v1.0 | Transformer Neural Network | © 2025

Crucially, the dashboard includes a **Saliency Map visualization**. This highlights the specific parts of the vibration signal that the model focused on. We observed that the model correctly focuses on the periodic transient impulses characteristic of bearing faults, providing "Explainable AI" (XAI) capabilities.

6. Conclusion and Future Work

6.1 Conclusion

This project is a great example of how Transformer-based architecture can be used to solve the "Sim-to-Real" problem in predictive maintenance. By learning universal vibration features from the artificial CWRU dataset, the **FaultFormer** model adapted seamlessly to the complex, noisy reality of the Paderborn dataset, achieving **99.77% accuracy**.

Key Takeaways:

- **Physics is Universal:** Synthetic data is very useful for learning fundamental physics (impulses, resonance), which can be easily transferred to real-world data.
- **Architecture Matters:** The combination of CNN tokenization and Transformer attention is very powerful method for time-series signal processing.
- **Efficiency:** Transfer learning significantly reduces the data and training time required for new domains.

6.2 Future Work

Though the present method was fruitful, it depended on supervised fine-tuning. The next version of the work might also consider exploring **Unsupervised Domain Adaptation (UDA)** to eliminate the need for labels in the target domain entirely. Additionally, efforts could be made to deploy the model on Edge devices (e.g., Raspberry Pi) for real-time industrial monitoring.

---

7. References

1. W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study," *Mechanical Systems and Signal Processing*, vol. 64-65, pp. 100-131, 2015.

2. C. Lessmeier, et al., "Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification," in *PHM Society European Conference*, 2016.
3. A. Zhou, et al., "FaultFormer: Transformer-based Prediction of Bearing Faults," *arXiv preprint arXiv:2312.02380*, 2023.
4. A. Vaswani, et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.
5. Z. Zhao, et al., "Deep Learning for Bearing Fault Diagnosis: A Review," *IEEE Access*, 2020.

Appendix: Code Snippets

A.1 Data Preprocessing (Normalization)

```
def z_score_normalize(signal):
    mean = np.mean(signal)
    std = np.std(signal)
    return (signal - mean) / std
```

A.2 Transfer Learning Setup

```
# Load Pre-trained Model
base_model = load_model('cwru_pretrained.h5')

# Freeze CNN Tokenizer Layers
for layer in base_model.layers[:2]:
    layer.trainable = False

# Add New Classification Head
x = base_model.layers[-2].output
output = Dense(3, activation='softmax')(x)
transfer_model = Model(inputs=base_model.input, outputs=output)
```