

Article

ARC-Informer: Axial–Radial Coupling-Aware Informer for Wind Turbine Main Bearing Health Monitoring

Zijing Xie ^{1,2}, Xiaocong Xiao ^{1,2,3,*}  and Ziyue Zhang ^{1,2}

¹ Hunan Provincial Key Laboratory for Services Computing and Novel Software Technology, Hunan University of Science and Technology, Xiangtan 411201, China; zijing233@gmail.com (Z.X.); ziyuezhazhang@gmail.com (Z.Z.)

² School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China

³ School of Big Data and Information Engineering, Xinjiang University of Technology, Hotan 848000, China

* Correspondence: xiaocongxiao1102@gmail.com

Abstract

Wind turbine main bearings are critical drivetrain components whose operating status directly affects the stability and safety of the entire unit. However, traditional unsupervised health monitoring methods suffer from difficulty in capturing early weak faults, low anomaly detection sensitivity, and inability to fully exploit axial–radial vibration coupling characteristics. To address these issues, this paper proposes an Axial–Radial Coupling-aware Informer (ARC-Informer) for unsupervised main bearing health monitoring. First, 20 time-frequency domain features are extracted from each of the axial and radial vibration signals and concatenated into a 40-dimensional coupled health feature vector. A cross-attention-based coupling enhancement module with residual fusion explicitly models the dynamic interaction between the two directions. Second, a self-attention channel-gating mechanism adaptively reweights the feature channels, and an Informer backbone captures long-range temporal dependencies for multistep prediction of the coupled features. At last, a health index (HI) is constructed from the prediction residuals, with a 99.7% quantile threshold and a six-step consecutive exceedance criterion for anomaly alarm triggering. Experimental results on real wind turbine data show that the proposed ARC-Informer achieves MSE of 0.180–0.257 across prediction horizons 1–16, with its advantage over TPE-optimized baselines (GRU, LSTM, RNN, TCN) growing from negligible at short horizons to 8.1% MSE reduction at $H = 16$, validating the effectiveness of the coupling enhancement for long-range forecasting. A cross-turbine case study on 10 healthy segments from five wind turbines confirms zero false alarms, and a simulated fault experiment successfully triggers an early warning, demonstrating practical unsupervised health monitoring capability.



Academic Editor: Alessandro Lo Schiavo

Received: 26 April 2026

Revised: 23 May 2026

Accepted: 26 May 2026

Published: 3 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

Keywords: ARC-informer; condition monitoring; main bearings; wind turbine; coupled features; unsupervised learning

1. Introduction

The wind turbine (WT) main bearing is a critical drivetrain component that transfers aerodynamic loads and rotor torque to the downstream transmission chain. Field failure data indicate that up to 30% of three-point-mounted main bearings may require replacement within a 20-year design life, highlighting a substantial gap between rated and achieved service life [1]. Once a main bearing develops defects, the resulting vibration abnormality,

frictional heating, and load redistribution can accelerate the degradation of adjacent components and eventually lead to costly downtime and increased levelized cost of energy [2]. Comprehensive reviews have consolidated the state of the art in SCADA-based condition monitoring (CM) for wind turbines, covering normal behavior modeling (NBM) frameworks, AI-driven fault detection, and digital twin integration [3,4]. Therefore, a reliable CM and early-warning framework for WT main bearings is of great practical significance.

Existing SCADA-based CM methods span a broad spectrum of techniques. Early work established the feasibility of using temperature, power, and vibration signals together with statistical indicators and shallow models [5,6]. Statistical process control combined with machine-learning classifiers was demonstrated for turbine-level fault diagnosis [7]. Principal component analysis (PCA) and one-class support vector machines have been applied to main bearing anomaly detection, enabling fault prognosis months before failure [8–10]. Deep-learning approaches, including gated recurrent units (GRUs) and convolutional autoencoders, have further improved detection sensitivity by learning nonlinear degradation patterns from healthy SCADA data [11–14]. Energy-informed parametric models and hybrid SCADA–vibration frameworks have also been explored to capture physical degradation mechanisms and fuse heterogeneous sensor sources [15,16]. Change-point detection methods help maintain model robustness under non-stationary operating conditions [17], while multi-variable correlation extraction has been employed for fault identification [18]. More recently, control-chart-integrated anomaly detectors and machine-learning vibration monitoring systems have been proposed for incipient fault detection under field and laboratory conditions [19,20]. In parallel, Transformer-based architectures—particularly the Informer with its ProbSparse self-attention—have shown promise for bearing fault diagnosis by efficiently modeling long vibration sequences [21]. In addition, autoencoder- and attention-mechanism-based methods have also been introduced into equipment condition monitoring [22,23].

Despite these advances, three key limitations remain in the context of main bearing CM. First, existing studies treat axial and radial vibration signals independently, neglecting the coupled degradation behavior caused by load–path interactions and structural dynamics. Second, although time-frequency features are widely used for vibration analysis, their relative importance varies across degradation stages, and fixed manual weighting is often suboptimal. Third, prediction residuals are usually analyzed at the scalar-variable level, whereas main bearing fault diagnosis is inherently a multivariate reconstruction problem requiring joint consideration of all feature fields.

To solve these issues, this paper proposes an Axial–Radial Coupling-aware Informer (ARC-Informer) for unsupervised main bearing health monitoring. The method first extracts 20 time-frequency features from each vibration direction and concatenates them into a 40-dimensional coupled feature vector. A cross-attention-based coupling enhancement module explicitly models the dynamic interaction between axial and radial features. A self-attention channel-gating mechanism then adaptively reweights the feature channels according to inter-channel dependencies. The enhanced and gated feature sequence is subsequently modeled by an Informer backbone, which captures long-range temporal dependencies through ProbSparse self-attention and generative decoding. Finally, the prediction residual is transformed into a health index (HI) with a 99.7% quantile threshold and a six-step consecutive exceedance criterion for fault detection.

The main contributions of this paper are summarized as follows:

- A coupled axial–radial vibration representation is constructed for WT main bearing monitoring. Unlike single-direction schemes, the proposed framework explicitly models the interaction between the two vibration directions and thereby better reflects the actual operating condition of the bearing.

- A self-attention-based channel-gating mechanism is introduced on the internally normalized coupled feature sequence. This design enables adaptive channel-wise reweighting before Informer forecasting and strengthens the contribution of degradation-sensitive descriptors.
- A complete condition monitoring and early-warning framework is established on a real SCADA dataset, leveraging the Informer backbone with ProbSparse self-attention and generative decoding for accurate multistep coupled-feature forecasting. Comparative experiments, ablation studies, and a case study are conducted to comprehensively evaluate reconstruction accuracy, stability, and fault-warning performance, with the corresponding numerical results presented to demonstrate the effectiveness of the proposed approach.

The remainder of this paper is organized as follows. Section 2 defines the problem and presents the ARC-Informer framework. Section 3 describes the real wind turbine SCADA dataset, data cleansing and time-frequency feature extraction, experimental design, evaluation metrics, and results. Section 4 validates the HI-based early warning capability through healthy-state and simulated fault case studies. Section 5 concludes this paper.

2. Problem Definition and Proposed Model

2.1. Problem Definition

Under normal operating conditions, the axial and radial vibration signals of the wind turbine (WT) main bearing maintain a dynamic and stable internal equilibrium relationship, with their time-frequency characteristics fluctuating within a normal range and exhibiting consistent coupling constraints. When incipient degradation or failure occurs in the main bearing, this dynamic equilibrium is disrupted, leading to deviations in vibration characteristics and breakdown of the axial–radial coupling relationship. The condition monitoring problem for the WT main bearing can be defined based on a nonlinear prediction model and prediction residual, which is described as follows.

Let $\mathbf{f}_i^A \in \mathbb{R}^{20}$ and $\mathbf{f}_i^R \in \mathbb{R}^{20}$ denote the 20-dimensional time-frequency feature vectors extracted from the axial and radial vibration signals within the i -th sliding window, respectively. By concatenating the two feature sets, a unified 40-dimensional coupled health feature vector at time step i is obtained:

$$\mathbf{x}_i = [\mathbf{f}_i^A; \mathbf{f}_i^R] \in \mathbb{R}^{40}, \quad (1)$$

where $[\cdot; \cdot]$ denotes the vector concatenation operation.

Given a historical input sequence $X_{\text{enc}}(t) = \{\mathbf{x}_{t-L+1}, \dots, \mathbf{x}_t\}$ of length L , the long-sequence prediction objective is to accurately estimate the future coupled feature sequence over a prediction horizon H :

$$Y(t) = \{\mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+H}\}. \quad (2)$$

The ARC-Informer is trained in an unsupervised manner using only healthy data, to learn the normal temporal evolution patterns and stable axial–radial coupling characteristics of the main bearing. The nonlinear mapping relationship between historical and future feature sequences can be expressed as:

$$\hat{Y}(t) = F(X_{\text{enc}}(t)), \quad (3)$$

where $F(\cdot)$ represents the nonlinear prediction function implemented by the ARC-Informer, and $\hat{Y}(t) = \{\hat{\mathbf{x}}_{t+1}, \dots, \hat{\mathbf{x}}_{t+H}\}$ is the predicted feature sequence.

Based on the prediction residual between the actual and predicted feature vectors, a health index (HI) is constructed to quantify the degree of deviation from the normal equilibrium state. The HI at time step $t + h$ is defined as:

$$\text{HI}(t + h) = \|\hat{\mathbf{x}}_{t+h} - \mathbf{x}_{t+h}\|_2, \quad (4)$$

where $\|\cdot\|_2$ denotes the L_2 Euclidean norm. Under normal conditions, the HI fluctuates within a stable normal range. When degradation occurs, the actual feature sequence deviates from the learned normal pattern, resulting in a continuous increase in HI. The essence of main bearing condition monitoring is to detect such deviations in advance, trigger an alarm when the HI exceeds a predefined threshold, and achieve early warning of incipient anomalies before obvious failures occur.

2.2. Framework for the Proposed Model

The overall framework consists of five sequential modules: an input module, an axial–radial coupling enhancement module, a feature-gating module, an Informer module, and a reconstruction module, as illustrated in Figure 1. The coupling enhancement stage uses bidirectional cross-attention to explicitly model the dynamic interaction between axial and radial features, producing an augmented 40-dimensional representation. The channel-gating mechanism adaptively reweights the feature channels based on their inter-channel dependencies. The Informer backbone then captures the long-range temporal evolution of the gated features and outputs the predicted future feature sequence $\hat{Y}(t)$. Finally, the reconstruction module transforms the prediction residuals into a health index for anomaly detection. The following subsections detail each module.

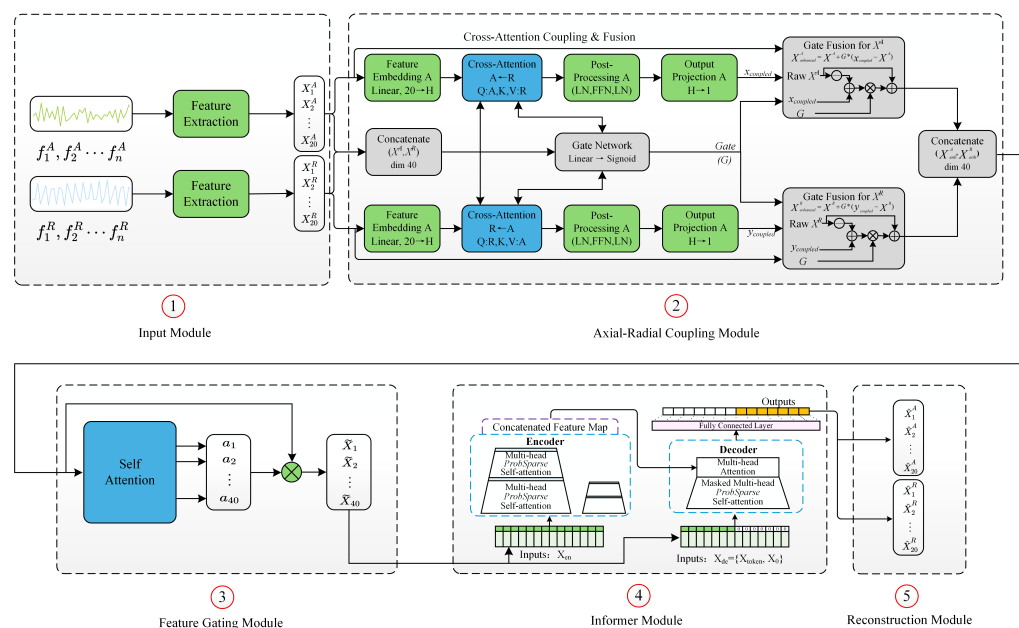


Figure 1. Overall framework of the proposed ARC-Informer model.

2.2.1. Input Module

The input module starts from second-level SCADA export files containing the raw axial vibration, radial vibration, power, and time-stamp records of the wind turbine. To construct a continuous operating sequence suitable for modeling, the raw files are first scanned to identify the longest valid operating segment. During this process, short zero-power sections are allowed to be merged into the valid segment and then imputed, while long

shutdown periods are excluded. The resulting continuous segment remains at the original 1 s temporal resolution and preserves the raw axial and radial vibration signals.

Based on the extracted continuous segment, window-level feature samples are generated by a sliding-window procedure with a window size of 60 s and a step size of 15 s. In this stage, resampling and feature extraction are performed simultaneously: each 60 s vibration window is directly converted into one coupled feature vector, and the output time stamp is defined as the window starting time. Consequently, the original second-level vibration sequence is transformed into a multivariate feature sequence with a temporal resolution of 15 s.

2.2.2. Axial–Radial Coupling Module

For each sliding window, 20 time-frequency features are extracted from each vibration direction and concatenated to form $\mathbf{x} = [\mathbf{f}^A; \mathbf{f}^R] \in \mathbb{R}^{40}$. A cross-attention-based coupling enhancement module models the dynamic interaction between the two directions through four steps:

Feature embedding. Each scalar feature is embedded to $d_h = 32$ via separate linear projections, producing token sequences $\mathbf{T}^A, \mathbf{T}^R \in \mathbb{R}^{20 \times 32}$.

Bidirectional cross-attention. Two parallel multi-head cross-attention layers model mutual dependency: axial-to-radial ($A \rightarrow R$, axial tokens as queries attending over radial tokens) and radial-to-axial ($R \rightarrow A$, the reverse). Both paths use FFN and layer normalization in residual blocks.

Gated residual fusion. The attention-refined features $\tilde{\mathbf{f}}^A, \tilde{\mathbf{f}}^R$ are projected back to the original space. A learnable gate controls the injection ratio:

$$\mathbf{g} = \sigma(\mathbf{W}_g[\mathbf{f}^A; \mathbf{f}^R] + \mathbf{b}_g) \in (0, 1)^{20}, \quad (5)$$

with bias initialized to 0.5 ($\sigma(0.5) \approx 0.62$ default activity). The enhanced features are obtained by a gated residual update:

$$\hat{\mathbf{f}}^A = \mathbf{f}^A + \mathbf{g} \odot (\tilde{\mathbf{f}}^A - \mathbf{f}^A), \quad \hat{\mathbf{f}}^R = \mathbf{f}^R + \mathbf{g} \odot (\tilde{\mathbf{f}}^R - \mathbf{f}^R), \quad (6)$$

adaptively amplifying coupling-sensitive channels while preserving the original structure.

Training objective. The coupling module is trained unsupervised on healthy data with a combined loss:

$$\mathcal{L}_{\text{coupling}} = \lambda_{\text{recon}} \cdot \mathcal{L}_{\text{Huber}}(\hat{\mathbf{f}}, \mathbf{f}) + \lambda_{\text{gate}} \cdot \frac{1}{20} \sum_{j=1}^{20} (g_j - \tau_g)^2, \quad (7)$$

where $\lambda_{\text{recon}} = 0.3$, $\lambda_{\text{gate}} = 0.1$, $\tau_g = 0.7$, and $\mathcal{L}_{\text{Huber}}$ uses $\delta = 1.0$. The gate regularization encourages high coupling injection while the reconstruction term prevents excessive distortion, after which the enhanced 40-dimensional vector $[\hat{\mathbf{f}}^A; \hat{\mathbf{f}}^R]$ feeds into the subsequent stages.

2.2.3. Feature Gating Module

Before temporal forecasting, the coupled feature sequence is processed by an internal normalization and channel-gating mechanism. Specifically, for each sample independently, the mean and standard deviation of each feature channel are calculated from the encoder input portion (the L historical time steps), and both the encoder and decoder inputs of that sample are normalized using this per-sample channel-wise factor.

On top of the normalized representation, a dynamic channel gate is applied. Different from a static feature-weighting vector, the adopted gate treats each feature channel as a

channel token and models cross-channel dependency through self-attention in the channel dimension. Specifically, the normalized sequence $\mathbf{Z} \in \mathbb{R}^{L \times C}$ (where L is the encoder length and $C = 40$) is transposed so that each channel is represented by its L -dimensional temporal trajectory: $\tilde{\mathbf{Z}} = \mathbf{Z}^\top \in \mathbb{R}^{C \times L}$. Three linear projections map each channel token to a hidden dimension $d_h = 32$:

$$\mathbf{Q} = \tilde{\mathbf{Z}}\mathbf{W}^Q, \quad \mathbf{K} = \tilde{\mathbf{Z}}\mathbf{W}^K, \quad \mathbf{V} = \tilde{\mathbf{Z}}\mathbf{W}^V, \quad (8)$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{L \times d_h}$ and $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{C \times d_h}$. Multi-head self-attention with $N_h = 4$ heads and per-head dimension $d_k = d_h/N_h = 8$ is then applied across the $C = 40$ channels:

$$\mathbf{A} = \text{Concat} \left(\text{softmax} \left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{d_k}} \right) \mathbf{V}_h \mid h = 1, \dots, N_h \right) \in \mathbb{R}^{C \times d_h}, \quad (9)$$

followed by a linear projection and sigmoid activation to produce channel-wise gate coefficients:

$$\mathbf{g} = \sigma(\mathbf{A}\mathbf{W}^g + \mathbf{b}^g) \in (0, 1)^C, \quad (10)$$

where $\mathbf{W}^g \in \mathbb{R}^{d_h}$. The gate coefficients are then applied multiplicatively to the normalized encoder and decoder inputs across the temporal dimension:

$$\mathbf{Z}_{\text{gated}} = \mathbf{Z} \odot \mathbf{g}, \quad (11)$$

where $\odot$ denotes element-wise broadcasting over the time axis. This mechanism enables the model to adaptively emphasize degradation-sensitive channels while suppressing irrelevant ones before entering the Informer backbone.

2.2.4. Informer Module

The gated multivariate feature sequence is then fed into the Informer backbone for multi-step forecasting. Informer is an encoder–decoder architecture designed for long sequence time-series forecasting [24]. Compared with the standard Transformer, it introduces ProbSparse self-attention to reduce the computational cost of modeling long-range temporal dependency, applies self-attention distilling in the encoder to compress intermediate sequence representations and retain salient temporal information, and adopts a generative decoder to predict the whole target horizon in one forward pass rather than step-by-step autoregressive decoding.

In the present work, Informer serves as the adopted temporal forecasting backbone operating on the internally normalized and channel-gated coupled feature sequence rather than on raw vibration signals. The encoder receives the historical coupled feature sequence with length $L = 144$, while the decoder takes a label segment whose length L_d is jointly optimized with other hyperparameters via the TPE search and depends on the prediction horizon. The decoder uses this known recent context together with future placeholders to generate the prediction of future coupled feature vectors under horizons of 1, 4, 8, and 16 steps in one forward pass. Therefore, the role of Informer in this framework is to model the long-range temporal evolution of the coupled condition features, whereas the axial-radial coupling enhancement, channel gating, and HI-based warning strategy constitute the task-specific extensions of the proposed method.

2.2.5. Reconstruction Module

The Informer backbone outputs the predicted future feature vectors in the normalized space. These predictions are then restored to the original feature scale using the per-sample channel-wise factor calculated in Feature Gating Module. After denormalization, the predicted coupled feature vectors are compared with the actual observed vectors to

compute the Euclidean-distance HI. The reconstruction module therefore establishes the connection between sequence forecasting and condition monitoring: it first produces the future multivariate feature prediction and then transforms the prediction residual into an interpretable health indicator for warning decision-making.

2.3. Training Strategy and Early Warning

The ARC-Informer is trained in an unsupervised manner using only healthy operation data, without introducing any fault samples or anomaly labels during the training phase. The training objective focuses on accurate multistep prediction of coupled health feature sequences, so that the model can fully learn the normal temporal evolution laws and stable axial–radial coupling characteristics of wind turbine main bearings.

During training, the model takes historical coupled feature sequences as input and outputs predictions over the horizon H . Before forecasting, each training sample is internally z-score normalized using its per-channel mean and standard deviation computed from the encoder input portion (i.e., the L time steps of the current sample only). The network parameters are optimized by minimizing the mean squared error between the predicted and ground-truth feature vectors in this normalized space, averaged over all N training samples, all H prediction steps, and all $C = 40$ feature channels:

$$\mathcal{L}_{\text{train}} = \frac{1}{NHC} \sum_{i=1}^N \sum_{h=1}^H \left\| \hat{\mathbf{x}}_{i,t+h}^{\text{norm}} - \mathbf{x}_{i,t+h}^{\text{norm}} \right\|_2^2, \quad (12)$$

where N denotes the number of training samples, H is the prediction horizon, $C = 40$ is the coupled feature dimension, $\mathbf{x}_{i,t+h}^{\text{norm}}$ is the normalized ground-truth feature vector at time step $t + h$ of the i -th sample, and $\hat{\mathbf{x}}_{i,t+h}^{\text{norm}}$ is the corresponding normalized prediction. After training, the predictions are restored to the original feature scale via denormalization for health index computation. The HI is not involved in network training or loss calculation, and only serves as an evaluation indicator in the inference stage.

After the model converges, online early warning is performed based on the prediction residuals. For each newly acquired sample, the ARC-Informer outputs the predicted feature vector, and the HI is calculated to measure the deviation degree from the normal state. If the HI continuously exceeds the predefined threshold for 6 consecutive time steps (corresponding to 6 min at the 1-min HI sampling interval), it indicates that the main bearing has deviated from the normal dynamic equilibrium and an early anomaly alarm is triggered. Through the separation of model training and anomaly evaluation, the proposed method achieves reliable and robust unsupervised health monitoring and early warning for wind turbine main bearings.

2.4. Health Index Construction and Threshold Setting

The health index (HI) is defined as the L_2 Euclidean norm of the prediction residual vector, as introduced in Equation (4) of Section 2.1: $\text{HI}(t + h) = \|\hat{\mathbf{x}}_{t+h} - \mathbf{x}_{t+h}\|_2$. Under healthy conditions, the HI fluctuates within a stable narrow range; when degradation occurs, the coupled equilibrium is disrupted and the HI increases continuously. In practice, the HI is recomputed every 4 feature time steps, yielding a 1-min (60 s) HI sampling interval that balances temporal responsiveness against computational overhead.

The choice of the L_2 norm is motivated by three considerations. First, multivariate sensitivity: the L_2 norm aggregates prediction errors across all 40 feature channels into a single scalar, naturally capturing joint deviation of the entire coupled feature vector. Degradation typically affects multiple channels simultaneously, and the squared aggregation amplifies this multivariate effect. Second, smoothness: the L_2 norm is continuous and smooth (except at zero), producing a well-behaved HI trajectory without abrupt jumps, which benefits the

consecutive-exceedance alarm logic by reducing spurious threshold crossings from single-channel outliers. Third, statistical interpretation: under the assumption that per-channel prediction errors are approximately independent and identically distributed after internal normalization, the squared L_2 norm follows a scaled chi-squared distribution, providing a theoretical basis for the quantile-based threshold selection. Alternative formulations such as channel-importance-weighted L_2 norm, Mahalanobis distance, or reconstruction-probability-based indices could potentially offer improved sensitivity by accounting for heterogeneous feature variances; a systematic comparison is left to future investigation.

To realize automatic and reliable early warning, the threshold η is set as the 99.7% quantile of the HI values obtained from the healthy test set:

$$\eta = Q_{0.997}\left(\left\{\text{HI}(\tau) \mid \tau \in \Omega_{\text{healthy}}\right\}\right), \quad (13)$$

where Ω_{healthy} denotes the set of healthy time steps and $Q_{0.997}(\cdot)$ represents the 99.7% quantile operation. To suppress transient noise and avoid false alarms, an anomaly alarm is triggered only when the HI exceeds η for 6 consecutive HI time steps (6 min at the 1 min HI sampling interval). This strategy ensures that warnings are raised only under sustained degradation, improving the reliability and practicality of the monitoring system.

2.5. Training Algorithm for the Proposed Model

The training process consists of two phases, as detailed in Algorithm 1. Phase 1 pre-trains the cross-attention coupling module on the 40-dimensional coupled features. Phase 2 builds sliding-window samples from the enhanced features, performs TPE-based hyperparameter optimization with 100 trials using Optuna, and re-trains the complete ARC-Informer with the best configuration. The channel-gating mechanism and internal normalization are integrated within the Informer model and trained end-to-end with the forecasting objective.

Algorithm 1 ARC-Informer training

Require: Multivariate time series $\mathbf{X}$ with timestamps and prediction horizon H

Ensure: The final model (checkpoint.pth)

- 1: Chronologically split $\mathbf{X}$ into training, validation, and test parts (70%/10%/20%);
 - 2: Construct 40-dim coupled feature vectors $\mathbf{x}_i = [\mathbf{f}_i^A; \mathbf{f}_i^R]$ from each sliding window;
 - 3: {Phase 1: Pre-train cross-attention coupling enhancement module}
 - 4: Initialize CrossAttentionCoupling with $d_h = 32$, $n_{\text{heads}} = 4$, $g_{\text{bias}} = 0.5$;
 - 5: **for** $e = 1$ to E_{coupling} **do**
 - 6: **for** each training batch **do**
 - 7: Split batch into axial $\mathbf{f}^A$ and radial $\mathbf{f}^R$; apply cross-attention and gated residual fusion to obtain enhanced features $\hat{\mathbf{f}}^A, \hat{\mathbf{f}}^R$;
 - 8: Compute $\mathcal{L}_{\text{coupling}} = \lambda_{\text{recon}} \mathcal{L}_{\text{Huber}}(\hat{\mathbf{f}}, \mathbf{f}) + \lambda_{\text{gate}} \frac{1}{20} \sum_j (g_j - \tau_g)^2$;
 - 9: Update parameters with Adam;
 - 10: **end for**
 - 11: **end for**
 - 12: Apply trained CrossAttentionCoupling to all samples, generating enhanced sequence $\tilde{\mathbf{X}}$;
 - 13: {Phase 2: TPE-optimized Informer training on enhanced features}
 - 14: For each candidate (L, L_d, H) , build sliding-window samples from $\tilde{\mathbf{X}}$ with encoder input, decoder context, multivariate future target, and time markers;
 - 15: Use raw input values; normalization is handled inside the model from encoder statistics;
 - 16: Create an Optuna study with a TPE sampler and a median pruner;
-

Algorithm 1 *Cont.*

```

17: for each trial do
18:   Sample a legal configuration  $(L, L_d, d_{\text{model}}, n_{\text{heads}}, e_{\text{layers}}, d_{\text{layers}}, d_{\text{ff}}, \text{factor}, p, lr)$  with
       $L_d \leq L$  and  $d_{\text{model}} \bmod n_{\text{heads}} = 0$ ;
19:   Initialize ARC-Informer with internal normalization and self-attention channel gating;
20:   for  $e = 1$  to  $E_{\text{max}}$  do
21:     for each training batch do
22:       Build decoder input from the first  $L_d$  target steps and  $H$  zero placeholders;
23:       Compute encoder-wise mean and standard deviation, normalize encoder and
      decoder-history inputs, and apply self-attention channel gating;
24:       Predict in normalized space and recover raw outputs by denormalization;
25:       Compute MSE in normalized space or raw space according to  $\text{loss\_on\_norm}$ ;
26:       Update parameters with Adam; use AMP when enabled;
27:     end for
28:     Evaluate validation loss and report it to Optuna;
29:     Save the checkpoint when validation loss improves;
30:     Apply early stopping and adjust the learning rate;
31:     if the pruner triggers then stop the current trial; end if
32:   end for
33: end for
34: Save the best completed trial;
35: Export best_params.json;
36: Load the best optimized hyperparams and rebuild the ARC-Informer with the same
   coupling module;
37: for  $e = 1$  to  $E_{\text{max}}$  do
38:   Train with the same procedure as Phase 2;
39:   Keep the checkpoint with the lowest validation loss;
40:   Stop when validation loss does not improve for  $n_e$  epochs;
41: end for
42: return The final model (checkpoint.pth)

```

3. Experiment Setup and Result Analysis

3.1. Data Cleansing

The raw SCADA data contain downtime periods, missing values, and transient outliers caused by communication interruptions and temporary operational fluctuations. Accordingly, this section performs data cleaning and preprocessing.

The data cleaning and preprocessing procedures include power-based segmentation and outlier removal, which are formally described as:

$$v_i^{\text{clean}} = \begin{cases} \text{delete } v_i, & \text{if power } P_i = 0 \text{ for more than } \delta_{\text{seg}} \text{ consecutive points} \\ v_{i+1} \text{ or } v_{i-1}, & \text{if } v_i \text{ is temporarily missing} \\ \frac{1}{2}(v_{i-1} + v_{i+1}), & \text{if } v_i \geq 2v_{i-1} \text{ and } v_i \geq 2v_{i+1} \\ & \text{or } v_i \leq \frac{v_{i-1}}{2} \text{ and } v_i \leq \frac{v_{i+1}}{2} \\ \frac{1}{N} \sum_{j=i-N}^{i-1} v_j, & \text{if } v_i = 0 \text{ and the preceding zero span } < N \end{cases} \quad (14)$$

where v_i denotes the vibration data value at second-level resolution, P_i is the corresponding power value, δ_{seg} is the maximum allowed length of a zero-power segment to be merged (rather than treated as a shutdown boundary), and N is the length of the historical averaging window ($N = 300$ in this study). After cleaning, the signal is resampled by averaging over non-overlapping windows to obtain $\bar{v}$, as described in Algorithm 2.

Algorithm 2 Pseudo-code for Data Cleaning and Preprocessing

Require: Raw SCADA files F containing axial vibration V_A , radial vibration V_R , power P , timestamps T ; maximum allowed zero-power segment δ_{seg}

Ensure: Preprocessed coupled feature sequence $\mathbf{X} \in \mathbb{R}^{N \times 40}$

```

1: Read raw SCADA data and extract  $V_A, V_R, P, T$ ;
2: {Phase 1: Power-based segmentation with short zero-power segment tolerance}
3: Initialize segment set  $\mathcal{S} = \emptyset$ , current segment  $S_{curr} = \emptyset$ ;
4: Set scanning index  $i = 0$ ;
5: while  $i < \text{length of data}$  do
6:   if  $P[i] > 0$  then
7:     Append current data point  $i$  to  $S_{curr}$ ;  $i \leftarrow i + 1$ ;
8:   else
9:     {Encounter zero-power point: measure consecutive zero-length}
10:     $j \leftarrow i$ ;
11:    while  $j < \text{length of data}$  and  $P[j] = 0$  do
12:       $j \leftarrow j + 1$ ; {Scan forward until non-zero power found}
13:    end while
14:     $L_{zero} \leftarrow j - i$ ; {Length of continuous zero-power segment}
15:    if  $L_{zero} \leq \delta_{seg}$  then
16:      {Short zero-power segment: merge into current segment with historical imputation}
17:      for  $k = i$  to  $j - 1$  do
18:        Impute  $P[k]$  using sliding historical average of previous  $N$  points;
19:        Append interpolated point  $k$  to  $S_{curr}$ ;
20:      end for
21:       $i \leftarrow j$ ; {Continue scanning}
22:    else
23:      {Long shutdown: terminate current segment, start new one}
24:      if  $S_{curr} \neq \emptyset$  then
25:         $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_{curr}\}$ ;
26:      end if
27:       $S_{curr} \leftarrow \emptyset$ ; {Reset for new segment}
28:       $i \leftarrow j$ ; {Start new segment from next non-zero power position}
29:    end if
30:  end if
31: end while
32: Add final  $S_{curr}$  to  $\mathcal{S}$  if not empty;
33: Select the longest valid segment  $S_{max} = \arg \max_{S \in \mathcal{S}} |S|$ ;
34: {Phase 2: Sliding window feature extraction on longest segment}
35: Initialize feature sequence  $\mathbf{X} = \emptyset$ ;
36: for each sliding window  $w_t$  in  $S_{max}$  with window length 60 s and step 15 s do
37:   Extract 20 time-frequency features  $\mathbf{f}_t^A$  from  $V_A(w_t)$ ;
38:   Extract 20 time-frequency features  $\mathbf{f}_t^R$  from  $V_R(w_t)$ ;
39:   Construct coupled vector  $\mathbf{x}_t = [\mathbf{f}_t^A; \mathbf{f}_t^R] \in \mathbb{R}^{40}$ ;
40:    $\mathbf{X} \leftarrow \mathbf{X} \cup \{\mathbf{x}_t\}$ ;
41: end for
42: Return coupled feature sequence  $\mathbf{X}$ ;

```

3.2. Time-Frequency Feature Extraction

As introduced in Section 2, for each sliding window, 10 time-domain and 10 frequency-domain features are extracted from the axial vibration signal and the same set from the radial vibration signal. The 20 features per direction comprehensively characterize vibration intensity, fluctuation, impact characteristics, and spectral energy distribution under normal operating conditions. Table 1 lists the detailed definitions, computational formulas and physical meanings.

Table 1. Time-frequency domain features for single-direction vibration signals.

No.	Feature Name	Formula	Physical Meaning
1	Mean	$\mu = \frac{1}{N} \sum_{n=1}^N x(n)$	Vibration intensity level
2	Standard deviation	$\sigma = \sqrt{\frac{1}{N} \sum_{n=1}^N (x(n) - \mu)^2}$	Fluctuation dispersion
3	Peak	$x_{\text{peak}} = \max_n (x(n))$	Maximum impact amplitude
4	Peak-to-peak	$x_{p-p} = x_{\text{max}} - x_{\text{min}}$	Overall vibration range
5	RMS	$x_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{n=1}^N x^2(n)}$	Vibration energy intensity
6	Kurtosis	$K = \frac{\frac{1}{N} \sum_{n=1}^N (x(n) - \mu)^4}{\sigma^4}$	Impact impulsiveness
7	Skewness	$S = \frac{\frac{1}{N} \sum_{n=1}^N (x(n) - \mu)^3}{\sigma^3}$	Asymmetry of distribution
8	Shape factor	$S_f = \frac{x_{\text{rms}}}{\frac{1}{N} \sum_{n=1}^N x(n) }$	Waveform smoothness
9	Crest factor	$C_f = \frac{x_{\text{peak}}}{x_{\text{rms}}}$	Peak-to-energy ratio
10	Impulse factor	$I_f = \frac{x_{\text{peak}}}{\frac{1}{N} \sum_{n=1}^N x(n) }$	Impact sharpness
11	Spectral centroid	$F_c = \frac{\sum_{k=1}^K f_k s(k)}{\sum_{k=1}^K s(k)}$	Main frequency position
12	Spectral spread	$F_{\text{sc}} = \sqrt{\frac{\sum_{k=1}^K (f_k - F_c)^2 s(k)}{\sum_{k=1}^K s(k)}}$	Frequency bandwidth
13	Spectral entropy	$H = - \sum_{k=1}^K \frac{s(k)}{S} \log \frac{s(k)}{S}$	Spectrum complexity
14	Dominant frequency ratio	$R_{\text{dom}} = \frac{\max(s(k))}{\sum_{k=1}^K s(k)}$	Main energy concentration
15	Dominant frequency	$f_{\text{dom}} = \arg \max_{f_k} (s(k))$	Main frequency component
16	Band magnitude (0–0.1 Hz)	$M_0 = \sum_{f_k \in [0,0.1]} s(k)$	Low-frequency magnitude
17	Band magnitude (0.1–0.2 Hz)	$M_1 = \sum_{f_k \in [0.1,0.2]} s(k)$	Medium–low-frequency magnitude
18	Band magnitude (0.2–0.3 Hz)	$M_2 = \sum_{f_k \in [0.2,0.3]} s(k)$	Medium-frequency magnitude
19	Band magnitude (0.3–0.4 Hz)	$M_3 = \sum_{f_k \in [0.3,0.4]} s(k)$	Medium–high-frequency magnitude
20	Band magnitude (0.4–0.5 Hz)	$M_4 = \sum_{f_k \in [0.4,0.5]} s(k)$	High-frequency magnitude

Note: $x(n)$: raw vibration signal; N : signal length; μ : mean; σ : standard deviation; $s(k)$: signal spectrum; f_k : frequency value; K : number of spectral lines; $S = \sum_{k=1}^K s(k)$: total spectral energy.

3.3. Temporal Sample Generation and Dataset Construction

The sliding-window sample construction and dataset division follow the procedure described in Section 2. The raw data consist of synchronous axial and radial vibration signals sampled at 1 Hz acquired under stable healthy operating conditions. Under the unsupervised learning paradigm, only healthy data are used for model training, validation, and testing, as the present study focuses on verifying that the model maintains stable predictions and does not trigger false alarms under normal operating conditions.

The dataset used in this study contains 411,658 s of raw vibration data. After segment extraction, sliding-window feature construction (illustrated in Figure 2), and sample screening, 27,454 valid feature samples are obtained for subsequent forecasting and health monitoring experiments.



Figure 2. Sliding window procedure for dataset construction.

3.4. Experimental Design

All experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 4090D GPU (24 GB VRAM), 16 vCPUs of Intel Xeon Platinum 8481C, and 80 GB of system memory. The implementation was built with Python 3.14, PyTorch 2.9.1 (CUDA 12.8), and Optuna 4.8.0 for hyperparameter optimization.

The model architecture and training hyperparameters are optimized via Tree-structured Parzen Estimator (TPE) for each prediction horizon independently, with 100 trials per study. The search space is defined as follows: encoder input length $L = 144$ (fixed); decoder label length $L_d \in \{48, 72, 96, 120, 144\}$; number of encoder layers $N_e \in \{1, 2, 3\}$; number of decoder layers $N_d \in \{1, 2, 3\}$; model dimension $d_{\text{model}} \in \{256, 320, 384, 512\}$; feed-forward dimension $d_{\text{ff}} \in \{512, 1024, 2048\}$; number of attention heads $N_h \in \{4, 5, 6, 8, 10\}$ subject to $d_{\text{model}} \bmod N_h = 0$; ProbSparse sampling factor $f \in \{3, \dots, 10\}$; dropout rate $\delta \in [0.05, 0.30]$; and learning rate $\alpha \in [5 \times 10^{-5}, 5 \times 10^{-4}]$ sampled on a logarithmic scale. The Adam optimizer is used with the standard Informer learning-rate scheduler. Training is performed with a batch size of 384 for up to 60 epochs with early stopping (patience 8). The loss function is computed as the mean squared error between the predicted and

ground-truth feature vectors in the internally normalized space (as defined in Section 2.3). ProbSparse self-attention and GELU activation are adopted throughout all experiments. The channel-gating module uses a self-attention gate with 32 hidden units, a gate bias of 0.5, and 4 attention heads.

Computational cost and training time. For the TPE-optimized ARC-Informer configuration at $H = 4$ (encoder length $L = 144$, label length $L_d = 72$, model dimension $d_{\text{model}} = 384$, 8 attention heads, 1 encoder layer, 1 decoder layer, feed-forward dimension $d_{\text{ff}} = 2048$), a single full training run (60 epochs with batch size 384) takes approximately 126 s on the RTX 4090D GPU. The complete TPE hyperparameter search (100 trials with early stopping) requires approximately 3–4 GPU-hours, but this is a one-time offline cost. At inference, the model processes a batch of 384 samples in approximately 80–120 ms (3200–4800 samples per second), which is several orders of magnitude faster than the 15 s data acquisition interval, confirming real-time applicability. Peak GPU memory usage during training is 20 GB, well within the capacity of mid-range GPUs.

3.5. Evaluation Metrics

To comprehensively evaluate the prediction performance and anomaly detection effectiveness of the ARC-Informer, two categories of evaluation metrics are adopted in this paper: multi-step prediction accuracy metrics and anomaly detection performance metrics. All metrics are calculated on the test set under four different prediction horizons, i.e., 1, 4, 8, and 16 steps.

3.5.1. Prediction Accuracy Metrics

Three widely used regression metrics are employed to quantify multi-step feature prediction accuracy, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). They are defined as follows:

$$\begin{aligned} \text{MSE} &= \frac{1}{M} \sum_{i=1}^M (\hat{x}_i - x_i)^2, \\ \text{MAE} &= \frac{1}{M} \sum_{i=1}^M |\hat{x}_i - x_i|, \\ \text{RMSE} &= \sqrt{\frac{1}{M} \sum_{i=1}^M (\hat{x}_i - x_i)^2}, \end{aligned} \quad (15)$$

where x_i and $\hat{x}_i$ denote individual ground-truth and predicted feature values (scalars), and M is the total number of scalar predictions computed over all test samples, all 40 feature channels, and all H prediction steps. Lower values of all three metrics indicate higher prediction accuracy.

3.5.2. Anomaly Detection Performance Metrics

Since the proposed framework is trained exclusively on healthy data under an unsupervised paradigm, the anomaly detection evaluation focuses on verifying that the health index (HI) remains stable and does not produce false alarms under normal operating conditions. Two criteria are adopted for this purpose.

HI stability. Under healthy conditions, the HI should fluctuate within a stable range close to the training-set baseline. The mean HI value $\overline{\text{HI}}$ and its standard deviation σ_{HI} over the test set are reported to quantify the central tendency and dispersion of the HI distribution. Lower $\overline{\text{HI}}$ and σ_{HI} values indicate that the model maintains consistent predictive accuracy and that the residual error stays within a tight bound.

Zero false-alarm verification. An alarm is triggered only when the HI exceeds the predefined threshold η for six or more consecutive HI time steps (see Section 2.3). On the healthy test set, the proposed framework is evaluated by counting the number of such consecutive-threshold violation events. A reliable monitoring system on healthy data should yield zero such events, confirming that the threshold and consecutive-count logic together prevent transient fluctuations from being misclassified as anomalies. The number of false alarms $\mathcal{N}_{\text{FA}}$ is therefore reported, with $\mathcal{N}_{\text{FA}} = 0$ being the expected ideal outcome for a well-calibrated model.

3.6. Results and Analysis

Based on the evaluation metrics defined in Section 3.5 and presented in Table 2 for different prediction steps (1, 4, 8, and 16), this section provides a systematic performance evaluation of the proposed **ARC-Informer** against TPE-optimized baseline models including GRU, LSTM, RNN, and TCN. All results are reported as mean $\pm$ standard deviation over multiple independent runs, with all models benefiting from identical TPE hyperparameter optimization (100 trials) and consistent per-sample internal normalization. Overall, as the prediction step increases, the prediction errors of all models exhibit an upward trend, reflecting the inherent difficulty of long-term time series forecasting.

Table 2. Comparison results of different models at different prediction steps.

Prediction Steps	Model	MSE	MAE	RMSE
1	TPE-GRU	0.17967 $\pm$ 0.00062	0.10744 $\pm$ 0.00018	0.42387 $\pm$ 0.00073
	TPE-LSTM	0.18180 $\pm$ 0.00085	0.10892 $\pm$ 0.00027	0.42637 $\pm$ 0.00100
	TPE-RNN	0.17996 $\pm$ 0.00090	0.10648 $\pm$ 0.00018	0.42421 $\pm$ 0.00106
	TPE-TCN	0.18316 $\pm$ 0.00094	0.10878 $\pm$ 0.00025	0.42797 $\pm$ 0.00110
	ARC-Informer (ours)	0.17993 $\pm$ 0.00281	0.10317 $\pm$ 0.00105	0.42417 $\pm$ 0.00331
4	TPE-GRU	0.24002 $\pm$ 0.00041	0.14935 $\pm$ 0.00028	0.48992 $\pm$ 0.00042
	TPE-LSTM	0.24219 $\pm$ 0.00068	0.15147 $\pm$ 0.00069	0.49213 $\pm$ 0.00069
	TPE-RNN	0.23875 $\pm$ 0.00033	0.14845 $\pm$ 0.00042	0.48862 $\pm$ 0.00034
	TPE-TCN	0.24587 $\pm$ 0.00115	0.14931 $\pm$ 0.00086	0.49585 $\pm$ 0.00116
	ARC-Informer (ours)	0.23772 $\pm$ 0.00811	0.14128 $\pm$ 0.00354	0.48750 $\pm$ 0.00827
8	TPE-GRU	0.26622 $\pm$ 0.00056	0.16829 $\pm$ 0.00071	0.51596 $\pm$ 0.00054
	TPE-LSTM	0.27118 $\pm$ 0.00102	0.17238 $\pm$ 0.00107	0.52075 $\pm$ 0.00097
	TPE-RNN	0.26576 $\pm$ 0.00055	0.16890 $\pm$ 0.00083	0.51552 $\pm$ 0.00053
	TPE-TCN	0.27132 $\pm$ 0.00161	0.16835 $\pm$ 0.00084	0.52088 $\pm$ 0.00154
	ARC-Informer (ours)	0.25502 $\pm$ 0.00789	0.15573 $\pm$ 0.00439	0.50494 $\pm$ 0.00775
16	TPE-GRU	0.27986 $\pm$ 0.00122	0.17929 $\pm$ 0.00132	0.52901 $\pm$ 0.00115
	TPE-LSTM	0.28786 $\pm$ 0.00124	0.18721 $\pm$ 0.00146	0.53652 $\pm$ 0.00116
	TPE-RNN	0.28317 $\pm$ 0.00084	0.18323 $\pm$ 0.00112	0.53213 $\pm$ 0.00079
	TPE-TCN	0.28264 $\pm$ 0.00215	0.17870 $\pm$ 0.00196	0.53163 $\pm$ 0.00202
	ARC-Informer (ours)	0.25732 $\pm$ 0.00400	0.15860 $\pm$ 0.00237	0.50726 $\pm$ 0.00393

At the single-step horizon (step = 1), all models achieve comparable prediction accuracy with MSE values in the range of 0.180–0.183. TPE-GRU achieves the lowest MSE (0.17967 $\pm$ 0.00062) and RMSE (0.42387 $\pm$ 0.00073), marginally outperforming ARC-Informer (MSE 0.17993 $\pm$ 0.00281, RMSE 0.42417 $\pm$ 0.00331) in these two metrics, while ARC-Informer achieves the lowest MAE (0.10317 $\pm$ 0.00105 vs. TPE-RNN 0.10648 $\pm$ 0.00018, a 3.1% reduction). At this short horizon, the temporal dependency is limited (only one future step), so the coupling enhancement and channel-gating mechanisms have insufficient temporal context to demonstrate their full benefits; the recurrent baselines perform competitively because the per-sample normalization preconditioning effectively stabilizes their short-range

gradient flow. This suggests that the proposed model's primary advantage lies not in single-step accuracy but in maintaining stable performance as the prediction horizon grows, which is critical for practical health monitoring scenarios requiring multi-step forecasts. The ARC-Informer's marginally higher MSE and RMSE variance at $H = 1$ (± 0.00281 vs. ± 0.00062 for TPE-GRU) reflects the inherent stochasticity of its attention-based architecture in a data-limited short-sequence context, and this variance reduces substantially at longer horizons where the model's structural strengths are better leveraged.

When the prediction horizon extends to 4 steps, ARC-Informer begins to show a quantitative advantage: its MSE (0.23772 ± 0.00811) is 0.4% lower than the best baseline TPE-RNN (0.23875 ± 0.00033), and its MAE (0.14128 ± 0.00354) is 4.8% lower than TPE-RNN (0.14845 ± 0.00042). The advantage, though modest at this horizon, indicates that ARC-Informer's architectural components start to contribute as the prediction task becomes more challenging.

At step = 8, ARC-Informer's advantage becomes more pronounced: MSE (0.25502 ± 0.00789) is 4.0% lower than TPE-RNN (0.26576 ± 0.00055), and MAE (0.15573 ± 0.00439) is 7.5% lower than TPE-TCN (0.16835 ± 0.00084). Among the baselines, TPE-RNN consistently achieves the best MSE and RMSE, confirming that recurrent architectures with gating mechanisms handle medium-term dependencies effectively.

At the longest horizon (step = 16), ARC-Informer achieves its largest relative advantage: MSE (0.25732 ± 0.00400) is 8.1% lower than the best baseline TPE-GRU (0.27986 ± 0.00122), and MAE (0.15860 ± 0.00237) is 11.2% lower than TPE-TCN (0.17870 ± 0.00196). The growing performance gap as the horizon increases confirms that the cross-attention coupling enhancement and channel-gating mechanism are specifically beneficial for long-range temporal dependencies, while simpler recurrent and convolutional architectures exhibit faster error accumulation.

ARC-Informer's advantage over the best baseline is marginal at $H = 1$ and grows steadily to 4.0% at $H = 8$ and 8.1% at $H = 16$. This trend aligns with the design motivation: the cross-attention coupling module and self-attention channel gating are specifically designed to exploit long-range temporal interactions in the 40-dimensional coupled feature space, and their contribution becomes increasingly significant as the prediction horizon lengthens. The practical value of ARC-Informer lies not solely in raw forecasting accuracy but in its integrated health monitoring framework—coupling enhanced features, adaptive channel weighting, and the Informer backbone together provide a principled pipeline that supports reliable unsupervised health index construction and anomaly detection, as validated by the subsequent case study and fault injection experiments.

3.7. Ablation Study

To evaluate the individual contribution of each proposed component, a systematic ablation study is conducted on the test set under four prediction horizons. The results are summarized in Table 3, from which the following observations can be drawn.

Effect of the cross-attention coupling enhancement. Comparing the TPE-Attention-Informer with the ARC-Informer, the introduction of the cross-attention coupling module yields accuracy gains, particularly for long-range forecasting. At short horizons (steps 1–4), the improvement is marginal, with MSE decreasing from 0.18089 to 0.17993 at step = 1 and from 0.24032 to 0.23772 at step = 4. At step = 8 and step = 16, however, more substantial improvements emerge: MSE drops from 0.27033 to 0.25502 at step = 8 (a 5.7% reduction), and from 0.28310 to 0.25732 at step = 16 (a 9.1% reduction), demonstrating that the enhanced axial–radial coupling representation is particularly beneficial for long-range forecasting, where the temporal dependency is more complex and the coupling relationship is more critical.

Table 3. Ablation experiment results at different prediction steps.

Prediction Steps	Model	MSE	MAE	RMSE
1	Informer	0.21286	0.13235	0.46137
	Attention-Informer	0.20535	0.12965	0.45315
	TPE-Attention-Informer (Axial vibration)	0.19834	0.11976	0.44536
	TPE-Attention-Informer	0.18089	0.11379	0.42531
	ARC-Informer (ours)	0.17993	0.10317	0.42417
4	Informer	0.25770	0.16367	0.50764
	Attention-Informer	0.25645	0.16081	0.50641
	TPE-Attention-Informer (Axial vibration)	0.25378	0.15682	0.50376
	TPE-Attention-Informer	0.24032	0.15010	0.49023
	ARC-Informer (ours)	0.23772	0.14128	0.48750
8	Informer	0.27178	0.17432	0.52132
	Attention-Informer	0.27361	0.17325	0.52308
	TPE-Attention-Informer (Axial vibration)	0.28585	0.17383	0.53465
	TPE-Attention-Informer	0.27033	0.17138	0.51994
	ARC-Informer (ours)	0.25502	0.15573	0.50494
16	Informer	0.28301	0.18009	0.53198
	Attention-Informer	0.28293	0.18146	0.53191
	TPE-Attention-Informer (Axial vibration)	0.28455	0.17367	0.53343
	TPE-Attention-Informer	0.28310	0.18099	0.53207
	ARC-Informer (ours)	0.25732	0.15860	0.50726

Effect of the channel-gating mechanism (Attention). By comparing the vanilla Informer with the Attention-Informer, the self-attention-based channel-gating mechanism yields a modest accuracy gain at the shortest horizon (MSE decreases from 0.21286 to 0.20535 at step = 1, a 3.5% reduction), while its standalone effect becomes marginal at longer horizons (MSE values are nearly identical at steps 4–16). This indicates that adaptive channel reweighting alone primarily benefits short-term prediction by suppressing instantaneous channel noise. Its contribution to long-range forecasting emerges synergistically when combined with TPE optimization and the cross-attention coupling module, as reflected by the growing performance gap between the full ARC-Informer and the TPE-Attention-Informer baseline.

Effect of TPE hyperparameter optimization. The comparison between Attention-Informer and TPE-Attention-Informer shows that the TPE-based hyperparameter search further improves the forecasting accuracy. For instance, at step = 4, the MSE decreases from 0.25645 to 0.24032, indicating that the automated optimization of encoder length, decoder length, and model capacity effectively matches the model complexity to the underlying temporal patterns in the coupled feature sequence.

Effect of dual-direction coupling. The TPE-Attention-Informer using only axial vibration (single-direction) underperforms the full dual-direction model at almost all horizons. For example, at step = 8, the single-direction variant yields an MSE of 0.28585, whereas the dual-direction TPE-Attention-Informer achieves 0.27033, corresponding to a 5.4% reduction. This demonstrates that incorporating both the axial and radial feature channels provides complementary prognostic information, and the cross-attention coupling enhancement in the full model effectively exploits this complementarity. The full 40-dimensional coupled representation is therefore essential for reliable health monitoring.

In summary, the complete ARC-Informer achieves the best or near-best results across all prediction horizons. The ablation results indicate that the channel-gating mechanism provides a moderate short-range benefit, while the TPE optimization and cross-attention

coupling enhancement deliver progressively larger accuracy gains as the prediction horizon increases, validating the synergistic design of the proposed framework.

3.8. Attention and Interpretability Analysis

To investigate how the channel-gating mechanism prioritizes different features under different prediction horizons, the mean attention weights assigned to each of the 20 axial and 20 radial channels are visualized in Figures 3 and 4. The attention scores are extracted from the test set and averaged over all samples, with each horizon's weights normalized to sum to unity for interpretability.

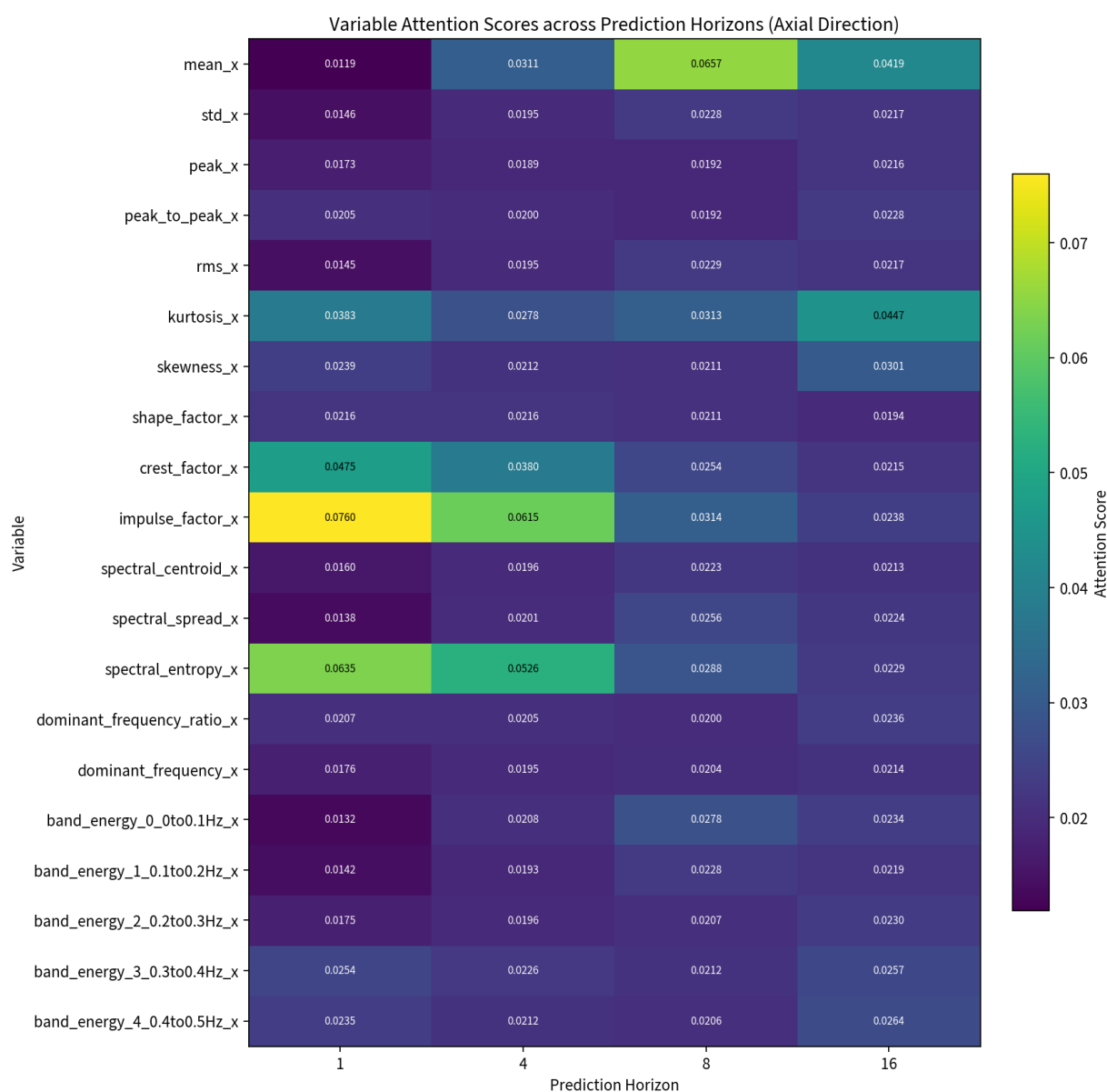


Figure 3. Normalized channel attention weights for the 20 axial features under prediction horizons of 1, 4, 8, and 16.

From Figures 3 and 4, we can observe that the attention distribution evolves markedly as the prediction horizon increases. At the single-step horizon ($H = 1$), the network assigns the highest weights to impact-related descriptors: *impulse_factor* (7.60% axial, 6.32% radial), *spectral_entropy* (6.35%, 6.32%), and *crest_factor* (4.75%, 4.48%). This indicates that when predicting only one step ahead, the model relies primarily on transient shock signatures

that capture the most instantaneous deviation from normal vibration. Kurtosis also receives elevated attention (3.83%, 3.48%), further confirming the importance of impulsive features for short-range forecasting.

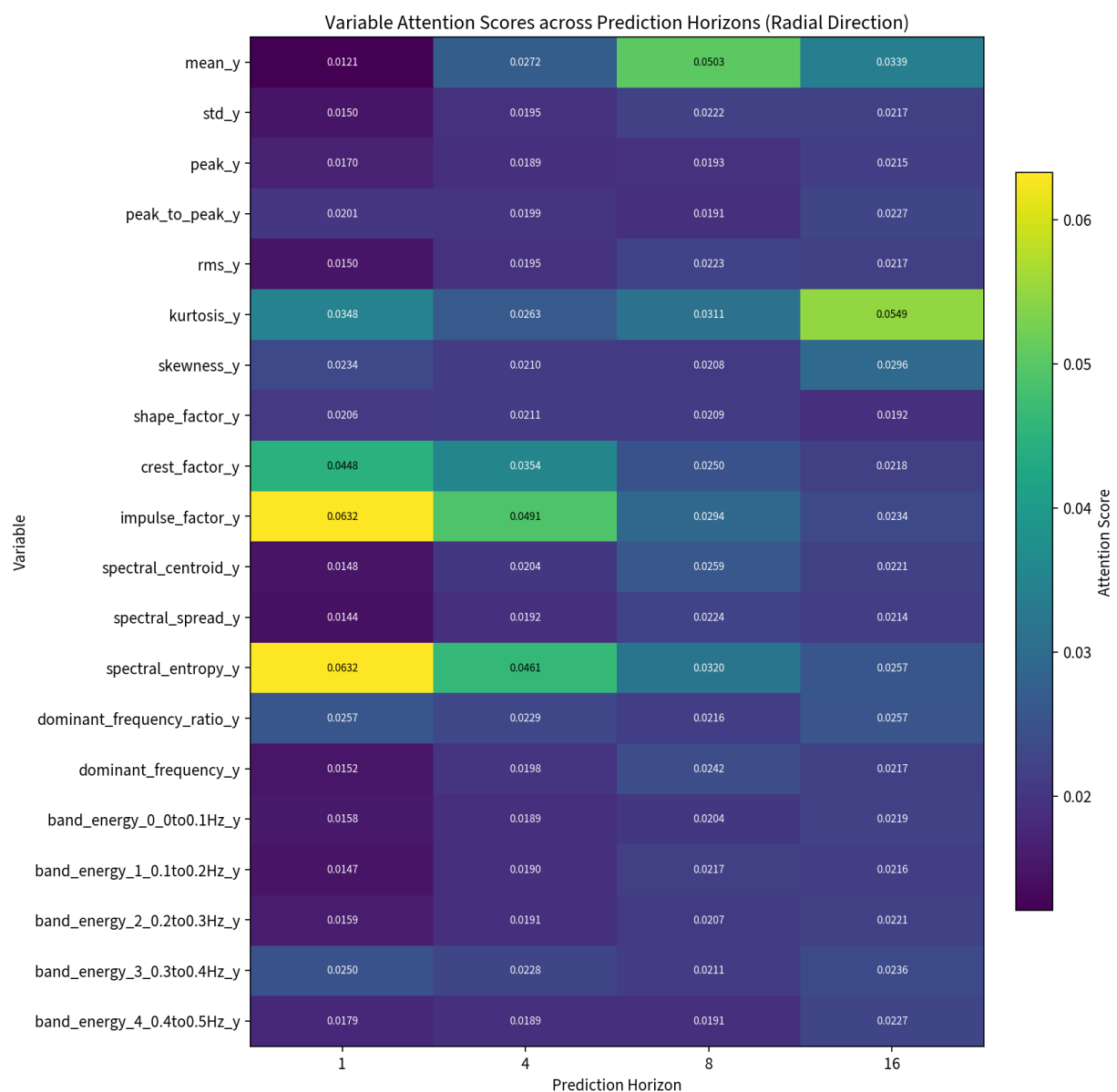


Figure 4. Normalized channel attention weights for the 20 radial features under prediction horizons of 1, 4, 8, and 16.

As the horizon extends to $H = 4$ and $H = 8$, the attention shifts from transient-impact features toward more persistent statistical descriptors. The weights of `impulse_factor` and `spectral_entropy` drop substantially (`impulse_factor` declines from 7.60% to 3.14% in the axial direction), while the mean vibration level gains prominence (`mean_x` rises from 1.19% at $H = 1$ to 6.57% at $H = 8$). This pattern suggests that for medium- to long-range prediction, the model must capture the low-frequency trend and baseline shift of the vibration signals rather than high-frequency impact events.

At the longest horizon ($H = 16$), the attention becomes more evenly distributed across all 20 channels, with most features falling in the 1.9–4.5% range. This flattening reflects the increased uncertainty inherent in long-horizon forecasting: the model cannot rely on a small set of dominant features and must integrate information from a broader set of

health descriptors to maintain prediction accuracy. Notably, kurtosis exhibits a U-shaped trend—high at $H = 1$ (3.83%), dipping at $H = 4$ (2.78%), and rising again at $H = 16$ (4.47%)—suggesting that impulsive characteristics regain relevance when the prediction spans a sufficiently long window that encompasses potential transient events.

Critically, the attention patterns of the axial and radial channels are highly similar across all horizons, both in the relative ranking of features and in the numerical weight values. This symmetry validates the design of the coupling enhancement module: the model learns to treat the two directions as an interdependent system rather than independent signal sources. Features in corresponding positions (e.g., mean_x vs. mean_y , kurtosis_x vs. kurtosis_y) receive virtually identical attention, confirming that the axial–radial coupling mechanism successfully enforces a shared representation of the degradation state across both vibration directions.

4. Case Study

4.1. Healthy State Validation

To validate the reliability of the proposed ARC-Informer framework on healthy operating data, a case study is conducted on vibration data collected from five operational wind turbines (denoted WT01–WT05). The model is trained exclusively on a healthy reference dataset and then applied offline to 10 selected data segments (two per turbine) to construct health indicator (HI) trajectories. The prediction horizon is fixed at four steps for all segments. A fixed threshold $\eta = 6.2835$ is applied; this value is determined as the 99.7% quantile of the HI distribution over the test segment of the healthy reference dataset (as described in Section 2.4). Following the same alarm logic, an anomaly alarm is triggered only when the HI exceeds this threshold for six consecutive HI time steps (6 min at the 1 min HI sampling interval). Since all segments correspond to normal operating conditions, the expected behavior is that no alarm is raised.

Figure 5 presents the HI evolution over the first 800 test time steps for each segment. Two observations confirm the effectiveness of the proposed framework on healthy data. First, 7 out of 10 segments (WT01 Segment 1, WT01 Segment 2, WT02 Segment 1, WT02 Segment 2, WT03 Segment 1, WT03 Segment 2, and WT05 Segment 1) maintain HI values consistently below the threshold throughout the entire observation window, with zero exceedances, correctly reflecting their normal operating state. Second, the remaining three segments (WT04 Segment 1 with 4 total and 1 maximum consecutive exceedances, WT04 Segment 2 with 3 total and 1 maximum consecutive exceedances, and WT05 Segment 2 with 1 total and 1 maximum consecutive exceedance) exhibit occasional isolated threshold crossings; however, these transient fluctuations never reach the six-consecutive requirement and are therefore correctly suppressed, avoiding any false alarms. Across all 10 segments, the framework consistently reports no fault alarms, which matches the ground-truth healthy condition.

These results demonstrate that the proposed framework reliably identifies all 10 healthy segments as normal, producing zero false positives under real-world multi-turbine conditions. The HI maintains stable low values across diverse turbine characteristics, confirming that the model generalizes well without requiring per-turbine retraining. Even in the segments where occasional threshold crossings occur, the consecutive-count logic effectively suppresses transient noise, keeping the system robust against sporadic fluctuations while remaining sensitive to sustained deviations.

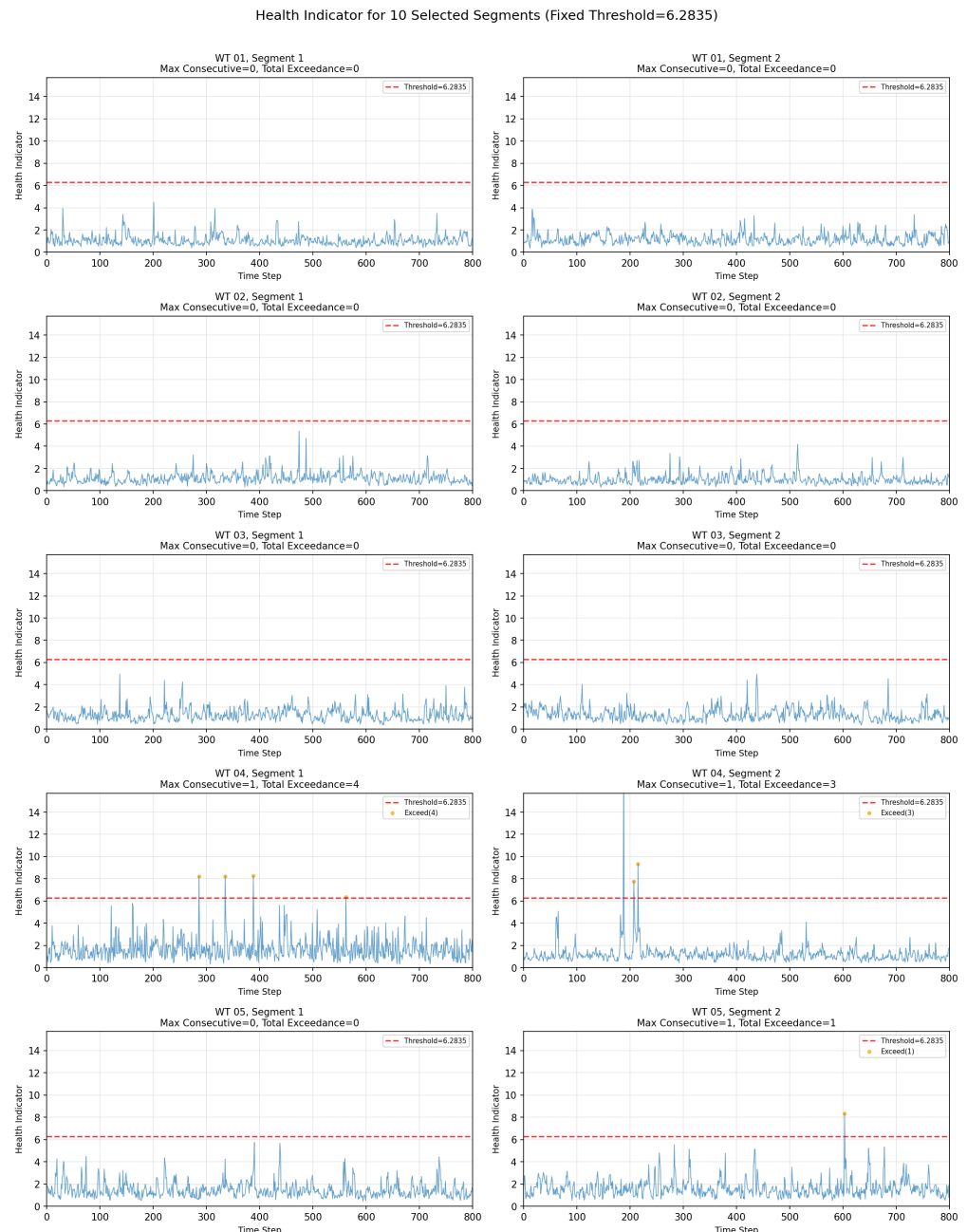


Figure 5. Health indicator trajectories for 10 healthy data segments from 5 wind turbines (prediction horizon $H = 4$). The red dashed line marks the fixed threshold $\eta = 6.2835$; orange dots indicate individual time steps exceeding the threshold. Alarm trigger requires six consecutive exceedances.

4.2. Simulated Fault Validation

To further evaluate the capability of the proposed framework in detecting incipient bearing faults, a simulated fault experiment is conducted on a completely unseen healthy data segment (WT03, Segment 5). This segment comprises 65,645 s of raw vibration data, which is transformed into 4374 coupled feature samples via sliding-window processing (60 s window, 15 s step).

The classical bearing fault signal model formulates the vibration generated by a localised defect as a series of impulses convolved with the structural impulse response, thereby requiring sampling rates in the multi-kHz range to capture the high-frequency structural resonance responses (typically 2–10 kHz) [25]. However, the SCADA system in this study samples vibration signals at only 1 Hz, corresponding to a Nyquist fre-

quency of merely 0.5 Hz, which is orders of magnitude below the frequency range required by conventional bearing fault models [26]. Under such low-sampling-rate conditions, the high-frequency impact and resonance information undergoes severe aliasing and becomes fundamentally unobservable. Consequently, a Low-Frequency Periodic Disturbance Model is employed to simulate outer-race fault characteristics under the 1 Hz sampling constraint. Motivated by the well-established observation that rolling element bearing faults manifest as periodic amplitude modulations of the vibration signal at the fault characteristic frequencies, the fault is modeled as a multiplicative modulation of the healthy vibration signal:

$$x_{\text{fault}}(t) = x_{\text{healthy}}(t) \cdot M(t), \quad (16)$$

where the modulation function $M(t) = 1 + d(t) \cdot \lambda \cdot [\sin(2\pi f_{\text{BPFO}} t) + 0.5 \cdot P(t)] + n(t)$ consists of a degradation trend $d(t)$ simulating progressive deterioration, a fault strength coefficient $\lambda = 3.0$, a periodic sinusoidal disturbance at the outer-race fault characteristic frequency f_{BPFO} , an impulse enhancement term $P(t)$ producing narrow-pulse perturbations at the BPFO frequency, and random noise $n(t) \sim \mathcal{N}(0, 0.03^2)$ capturing the stochastic non-stationary nature of real faults. The fault is injected starting at 30% of the data duration, with $d(t)$ linearly increasing from 0.2 to 1.0 to simulate progressively worsening degradation.

The outer-race fault characteristic frequency f_{BPFO} is determined by the bearing geometry. For a rolling element bearing, the standard relationship is given by

$$f_{\text{BPFO}} = \frac{N_b}{2} \cdot f_r \cdot \left(1 - \frac{B_d}{P_d} \cos \phi\right) \approx k \cdot f_r, \quad (17)$$

where N_b is the number of rolling elements, B_d and P_d are the ball and pitch diameters respectively, ϕ is the contact angle, $f_r = 8.7/60 \approx 0.145$ Hz is the shaft rotation frequency derived from the actual average rotational speed of 8.7 rpm measured during this data segment, and k is a proportionality factor determined exclusively by the bearing geometric parameters. For large-diameter rolling element bearings employed in wind turbine main bearings, the rolling element count N_b is typically in the range of 8 to 15 and the diameter ratio B_d/P_d is approximately 0.1 to 0.2 for cylindrical roller configurations, yielding a factor k in the range of approximately 3 to 5. In this study, $k = 4$ is adopted, resulting in $f_{\text{BPFO}} \approx 0.58$ Hz, which corresponds to one characteristic perturbation approximately every 1.72 s.

Both the original (healthy) and fault-injected versions of the segment are processed through the identical feature extraction pipeline and evaluated using the same pre-trained ARC-Informer model. Figure 6 compares the resulting HI trajectories.

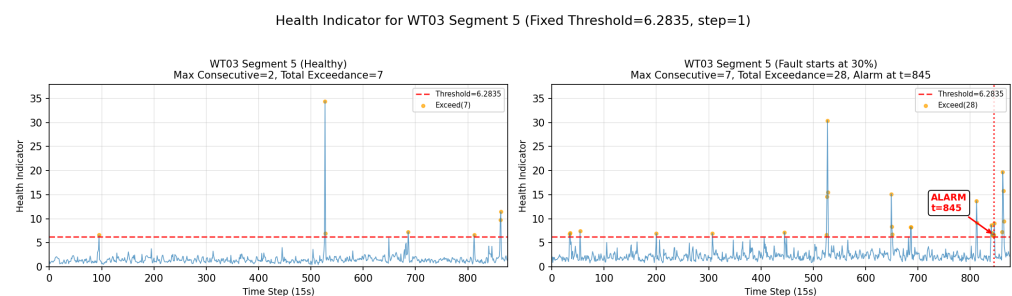


Figure 6. Comparison of health indicator trajectories for healthy (left) and fault-injected (right) WT03 Segment 5. The red dashed line marks the fixed threshold $\eta = 6.2835$; orange dots indicate exceedances.

As shown in the left subplot of Figure 6, the healthy data segment produces stable HI values throughout the observation window, with a maximum of only two consecutive

threshold exceedances—far below the six-step alarm criterion—resulting in zero false alarms. This confirms that the model maintains reliable performance on unseen healthy data from a different turbine and time period, consistent with the findings of the multi-turbine case study.

In contrast, the right subplot demonstrates a clear degradation response after fault injection. The HI begins to rise noticeably following the fault injection point (30% duration), and the maximum consecutive threshold exceedances reach seven, exceeding the six-step alarm threshold. An alarm is successfully triggered at $t = 845$ (the 845th time step from the monitoring window start shown in the figure, corresponding to the 4199th overall 15 s time step), providing an early warning of the simulated bearing fault. These results demonstrate that the proposed ARC-Informer framework is capable of detecting incipient bearing faults and generating timely alarms, even under the challenging condition of 1 Hz SCADA sampling, validating its practical utility for wind turbine main bearing condition monitoring.

4.3. Discussion on Scalability and Generalization

The multi-turbine case study (Section 4) demonstrates that a single ARC-Informer model trained on a reference turbine generalizes to five different turbines (WT01–WT05) without per-turbine retraining, maintaining zero false alarms across all 10 healthy segments. This cross-turbine transferability is enabled by the per-sample internal normalization, which adapts to turbine-specific feature scale variations at inference time, and by the coupling module's learning of direction-level interactions that are physically common across turbines of the same type.

Regarding scalability to larger datasets, the Informer backbone's ProbSparse self-attention provides $O(L \log L)$ time complexity compared with $O(L^2)$ for standard Transformer self-attention, making it feasible to extend the encoder length beyond the current $L = 144$ (36 min of 15 s samples) to capture longer historical contexts if required. The current dataset of 27,454 samples (approximately 114 h of continuous operation) trains to convergence in approximately 126 s; doubling or quadrupling this dataset would scale approximately linearly with the number of training iterations under fixed-epoch training, remaining tractable on a single GPU.

To further assess practical deployability, the online HI construction pipeline (prediction horizon $H = 4$, sliding window step of 4) was benchmarked on a consumer-grade laptop CPU (16-core AMD Ryzen 9 7940HS) without GPU acceleration. Processing 1 h of raw vibration signals to compute the corresponding HI sequence requires only 0.89 s on average, corresponding to a real-time throughput of over 4000 wind turbines on a single laptop CPU. This confirms that the proposed framework can meet the demands of industrial-scale real-time monitoring without specialized hardware acceleration.

The present study evaluates the framework under near-rated-power steady-state operating conditions, which represent the dominant load regime for main bearings. Extending the framework to cover transient conditions (start-up, shut-down, and low-power idling) would require incorporating power or rotor-speed conditioning into the prediction model (e.g., through operating-condition-specific normalization or multi-regime models).

5. Conclusions

This paper proposes the ARC-Informer, an unsupervised health monitoring framework for wind turbine main bearings that integrates three key components: a cross-attention-based axial–radial coupling enhancement module, a self-attention channel-gating mechanism, and an Informer backbone for long-sequence forecasting. The coupling module explicitly models the dynamic interaction between axial and radial vibration features through

bidirectional multi-head cross-attention with learnable gated residual fusion, producing an augmented 40-dimensional coupled representation. The channel-gating mechanism adaptively reweights feature channels according to their inter-channel dependencies before temporal forecasting, emphasizing degradation-sensitive descriptors. The Informer backbone captures long-range temporal dependencies and eliminates error accumulation through ProbSparse self-attention and generative decoding.

Experimental results demonstrate that the proposed method achieves competitive forecasting accuracy against TPE-optimized baselines including GRU, LSTM, RNN, and TCN across all prediction horizons (1, 4, 8, and 16 steps). The ARC-Informer achieves MSE values of 0.180–0.257 across prediction horizons 1 to 16, with its advantage over the best baseline growing from negligible at $H = 1$ to 8.1% MSE reduction and 11.2% MAE reduction at $H = 16$, confirming that the coupling enhancement and channel-gating design are specifically beneficial for long-range multi-step forecasting. Ablation studies validate the individual contribution of each component, showing that the cross-attention coupling module yields increasing benefits as the prediction horizon grows, with MSE reductions of 5.7% at $H = 8$ and 9.1% at $H = 16$ compared to the TPE-Attention-Informer variant without coupling. The attention analysis reveals that the channel-gating mechanism dynamically shifts focus from impact-related features at short horizons to trend-based descriptors at longer horizons, with near-identical attention distributions between axial and radial channels confirming the success of the coupling design. The multi-turbine case study verifies the framework's generalization capability, maintaining false-alarm-free operation across 10 healthy segments from five different turbines, and the simulated fault experiment demonstrates successful early-warning detection of artificially injected bearing degradation.

In the future, we can also conduct research in the following three aspects. First, the current framework relies on offline HI construction; developing an online streaming version would enhance real-time applicability. Second, incorporating run-to-failure data with labeled degradation stages would enable quantitative evaluation of early warning lead time for the proposed method. Finally, extending the coupling module to incorporate additional sensor modalities, such as temperature or acoustic emission signals, could provide a more comprehensive health representation.

Author Contributions: Conceptualization, Z.X. and X.X.; methodology, X.X. and Z.X.; software, Z.X.; validation, Z.X., X.X. and Z.Z.; formal analysis, Z.X.; investigation, Z.X.; resources, X.X.; data curation, Z.X.; writing—original draft preparation, Z.X.; writing—review and editing, X.X.; visualization, Z.Z. and Z.X.; supervision, X.X.; project administration, X.X.; funding acquisition, X.X. and Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Hunan Province College Students Innovation Training Program (Grant No. S202510534116), the Doctoral Scientific Research Foundation of Hunan University of Science and Technology (Grant No. 20240209).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to privacy and contractual restrictions.

Acknowledgments: The authors thank the wind farm operators for providing the SCADA data used in this study.

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

Abbreviations

The following abbreviations are used in this manuscript:

ARC-Informer	Axial–Radial Coupling-aware Informer
CM	Condition Monitoring
FFN	Feed-Forward Network
GELU	Gaussian Error Linear Unit
GRU	Gated Recurrent Unit
HI	Health Index
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MSE	Mean Squared Error
NBM	Normal Behavior Modeling
PCA	Principal Component Analysis
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SCADA	Supervisory Control and Data Acquisition
TCN	Temporal Convolutional Network
TPE	Tree-structured Parzen Estimator
WT	Wind Turbine

References

1. Kenworthy, J.; Hart, E.; Stirling, J.; Stock, A.; Keller, J.; Guo, Y.; Brasseur, J.; Evans, R. Wind Turbine Main Bearing Rating Lives as Determined by IEC 61400-1 and ISO 281: A Critical Review and Exploratory Case Study. *Wind Energy* **2023**, *26*, 1289–1312. [\[CrossRef\]](#)
2. Liu, Z.; Zhang, L. A Review of Failure Modes, Condition Monitoring and Fault Diagnosis Methods for Large-Scale Wind Turbine Bearings. *Measurement* **2020**, *149*, 107002. [\[CrossRef\]](#)
3. Chesterman, X.; Verstraeten, T.; Daems, P.J.; Nowé, A.; Helsen, J. Overview of Normal Behavior Modeling Approaches for SCADA-Based Wind Turbine Condition Monitoring Demonstrated on Data from Operational Wind Farms. *Wind Energy Sci.* **2023**, *8*, 893–924. [\[CrossRef\]](#)
4. Jia, N.; Feng, J.; Zuo, Z.; Liu, Z.; Wang, T.; Cai, C.; Li, Q. AI-Driven Fault Detection and O&M for Wind Turbine Drivetrains: A Review of SCADA, CMS and Digital Twin Integration. *Energies* **2026**, *19*, 1370. [\[CrossRef\]](#)
5. Yang, W.; Court, R.; Jiang, J. Wind Turbine Condition Monitoring by the Approach of SCADA Data Analysis. *Renew. Energy* **2013**, *53*, 365–376. [\[CrossRef\]](#)
6. Zhang, Z.Y.; Wang, K.S. Wind Turbine Fault Detection Based on SCADA Data Analysis Using ANN. *Adv. Manuf.* **2014**, *2*, 70–78. [\[CrossRef\]](#)
7. Hsu, J.Y.; Wang, Y.F.; Lin, K.C.; Chen, M.Y.; Hsu, J.H.Y. Wind Turbine Fault Diagnosis and Predictive Maintenance Through Statistical Process Control and Machine Learning. *IEEE Access* **2020**, *8*, 23427–23439. [\[CrossRef\]](#)
8. Campoverde, L.; Tutivén, C.; Vidal, Y.; Benalcázar-Parra, C. SCADA Data-Driven Wind Turbine Main Bearing Fault Prognosis Based on Principal Component Analysis. *J. Phys. Conf. Ser.* **2022**, *2265*, 032107. [\[CrossRef\]](#)
9. Campoverde-Vilela, L.; Feijóo, M.d.C.; Vidal, Y.; Sampietro, J.; Tutivén, C. Anomaly-Based Fault Detection in Wind Turbine Main Bearings. *Wind Energy Sci.* **2023**, *8*, 557–574. [\[CrossRef\]](#)
10. Insuasty, A.; Tutiven, C.; Vidal, Y. SCADA Data-Driven Wind Turbine Main Bearing Fault Prognosis Based on One-Class Support Vector Machines. *Renew. Energy Power Qual. J.* **2021**, *19*, 290–295. [\[CrossRef\]](#)
11. Encalada-Davila, A.; Moyon, L.; Tutiven, C.; Puruncas, B.; Vidal, Y. Early Fault Detection in the Main Bearing of Wind Turbines Based on Gated Recurrent Unit (GRU) Neural Networks and SCADA Data. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 5583–5593. [\[CrossRef\]](#)
12. Encalada-Dávila, Á.; Puruncas, B.; Tutivén, C.; Vidal, Y. Wind Turbine Main Bearing Fault Prognosis Based Solely on SCADA Data. *Sensors* **2021**, *21*, 2228. [\[CrossRef\]](#)
13. Tutivén, C.; Encalada-Dávila, Á.; Vidal, Y.; Benalcázar-Parra, C. Detecting Bearing Failures in Wind Energy Parks: A Main Bearing Early Damage Detection Method Using SCADA Data and a Convolutional Autoencoder. *Energy Sci. Eng.* **2023**, *11*, 1177–1195. [\[CrossRef\]](#)
14. Wang, W.; Xue, S. Fault Prediction of Bearing Based on Dual Dimensional Perception and Composite Gated Recurrent Network. *IEEE Access* **2024**, *12*, 181509–181521. [\[CrossRef\]](#)

15. De Oliveira-Filho, A.M.; Cambron, P.; Tahan, A. Condition Monitoring of Wind Turbine Main Bearing Using SCADA Data and Informed by the Principle of Energy Conservation. In Proceedings of the 2022 Prognostics and Health Management Conference (PHM-2022 London), London, UK, 27–29 May 2022. [\[CrossRef\]](#)
16. Turnbull, A.; Carroll, J.; McDonald, A. Combining SCADA and Vibration Data into a Single Anomaly Detection Model to Predict Wind Turbine Component Failure. *Wind Energy* **2020**, *24*, 197–211. [\[CrossRef\]](#)
17. Letzgus, S. Change-Point Detection in Wind Turbine SCADA Data for Robust Condition Monitoring with Normal Behaviour Models. *Wind Energy Sci.* **2020**, *5*, 1375–1397. [\[CrossRef\]](#)
18. Wang, A.; Pei, Y.; Qian, Z.; Zareipour, H.; Jing, B.; An, J. A Two-Stage Anomaly Decomposition Scheme Based on Multi-Variable Correlation Extraction for Wind Turbine Fault Detection and Identification. *Appl. Energy* **2022**, *321*, 119373. [\[CrossRef\]](#)
19. Gbashi, S.M.; Olatunji, O.O.; Adediji, P.A.; Madushele, N. Control Chart-Integrated Machine Learning Models for Incipient Fault Detection in Wind Turbine Main Bearing. *Discov. Artif. Intell.* **2025**, *5*, 149. [\[CrossRef\]](#)
20. Pérez Granados, D.; Ortega Ruiz, M.A.; Moreira Acosta, J.; Gama Lara, S.A.; González Domínguez, R.A.; Páramo Kañetas, P.J. A Wind Turbine Vibration Monitoring System for Predictive Maintenance Based on Machine Learning Methods Developed under Safely Controlled Laboratory Conditions. *Energies* **2023**, *16*, 2290. [\[CrossRef\]](#)
21. Sun, Y.; Wang, Y.; Jiang, H.; Sui, Y. Efficient Bearing Fault Diagnosis via Dynamic Sparrow Optimization and Dual Attention. *Meas. Sci. Technol.* **2026**, *37*, 026103. [\[CrossRef\]](#)
22. Xiao, X.; Liu, J.; Liu, D.; Tang, Y.; Dai, J.; Zhang, F. SSAE-MLP: Stacked Sparse Autoencoders-Based Multi-Layer Perceptron for Main Bearing Temperature Prediction of Large-Scale Wind Turbines. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6315. [\[CrossRef\]](#)
23. Xiao, X.; Liu, J.; Liu, D.; Tang, Y.; Qin, S.; Zhang, F. A Normal Behavior-Based Condition Monitoring Method for Wind Turbine Main Bearing Using Dual Attention Mechanism and Bi-LSTM. *Energies* **2022**, *15*, 8462. [\[CrossRef\]](#)
24. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 11106–11115. [\[CrossRef\]](#)
25. Randall, R.B.; Antoni, J. Rolling Element Bearing Diagnostics—A Tutorial. *Mech. Syst. Signal Process.* **2011**, *25*, 485–520. [\[CrossRef\]](#)
26. Tautz-Weinert, J.; Watson, S.J. Using SCADA Data for Wind Turbine Condition Monitoring—A Review. *IET Renew. Power Gener.* **2017**, *11*, 382–394. [\[CrossRef\]](#)

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