

## Article

# Combining In-Sensor Computing with Reasoning at the Edge for Low-Power Bearing RUL Prediction

Simone Tognocchi <sup>1</sup>, Danilo Pietro Pau <sup>2,\*</sup> and Marco Marcon <sup>1</sup>

<sup>1</sup> Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133, Milan, Italy; simone.tognocchi@polimi.it (S.T.); marco.marcon@polimi.it (M.M.)

<sup>2</sup> System Research and Applications, STMicroelectronics, Via C. Olivetti 2, 20864, Milan, Italy

\* Correspondence: danilo.pau@st.com

## Abstract

The increasing demand for predictive maintenance in industrial environments requires edge-intelligent solutions in which latency, energy consumption, memory footprint, and data movement are strictly constrained. This paper presents a two-stage Edge AI architecture for rolling-bearing prognostics, designed for heterogeneous embedded deployment across two distinct processing levels: a smart programmable sensing unit for local low-complexity signal preprocessing and a low-power embedded multiprocessor for higher-level temporal prognostic reasoning. In the first stage, a tiny neural preprocessor processes high-frequency vibration measurements directly at the sensing level and produces a compact low-dimensional degradation representation, complemented by lightweight health-related physical features. In the second stage, an embedded temporal reasoning model analyzes sequences of these compressed representations to estimate bearing degradation and remaining useful life. In addition to numerical remaining useful life regression, the second stage includes diagnostic reasoning heads that produce maintenance-oriented categories from a restricted vocabulary, enabling interpretable diagnostic summaries without relying on cloud-based language models. The complete pipeline is designed for fully edge-resident operation and exported in deployment-compatible formats, with the objective of supporting practical integration into heterogeneous industrial edge platforms. The proposed framework is trained and evaluated on the PRONOSTIA bearing degradation dataset and positioned against representative recurrent and hybrid prognostic baselines from the literature. From the deployment viewpoint, the sensor-side stage requires 68.62 ms inference time with 2.07 KiB RAM and 1.35 KiB flash/weights, whereas the edge temporal stage runs in 49.2 ms with 90.68 MiB RAM and 67.71 MiB flash/weights. In terms of prognostic performance, the proposed model achieves an average normalized RMSE of 0.1616 and an average normalized MAE of 0.1311 on three held-out bearings, while the weakly supervised diagnostic heads reach accuracies of 0.9066 for degradation trend and 0.8872 for vibration evidence. Experimental results show that the proposed architecture provides an effective trade-off between compact sensor-side processing, temporal prognostic accuracy, monotonic degradation consistency, hardware deployability, and interpretable maintenance-oriented outputs, supporting the feasibility of fully edge-based predictive maintenance systems for rolling-bearing health monitoring.



Academic Editor: Ahmed Abu-Siada

Received: 2 June 2026

Revised: 29 June 2026

Accepted: 20 July 2026

Published: 22 July 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](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

**Keywords:** Generative Edge AI; Interpretable Edge AI; structured diagnostic reasoning; Tiny ML; remaining useful life prediction; intelligent sensor processing unit; two-stage neural architecture

## 1. Introduction

Rolling-bearings are among the most critical components in rotating machinery. They support rotating shafts, transmit mechanical loads, and operate under continuously varying dynamic, thermal, and lubrication conditions. Their degradation can lead to increased vibration, reduced efficiency, secondary damage to surrounding components, unplanned downtime, and, in safety-critical systems, severe operational risks. For this reason, rolling-bearing monitoring has become a central task in Prognostics and Health Management (PHM), where the objective is not only to detect an already developed fault, but also to estimate how much useful operating time remains before functional failure. Bearings are especially relevant in this context because a large portion of rotating-machine failures is associated with these components, and their failure may significantly reduce both machine availability and operational safety [1,2].

Remaining Useful Life (RUL) estimation addresses this problem by predicting the time-to-failure of a component from condition-monitoring data. In bearing applications, RUL prediction is commonly based on vibration signals, temperature measurements, or degradation features extracted from sensor streams. Unlike conventional fault diagnosis, which assigns a discrete health state or fault class, RUL estimation provides a continuous indicator of the degradation process. This makes it particularly useful for predictive maintenance, since maintenance actions can be scheduled before catastrophic failure while avoiding unnecessarily early replacement of still-usable components. Accurate prognostics can therefore reduce unexpected failures, downtime, spare-part waste, maintenance costs, and safety hazards [2–4].

The problem is challenging because bearing degradation trajectories are often non-linear, noisy, and strongly dependent on operating conditions. Bearings operating under different loads and rotational speeds may exhibit different degradation rates, vibration signatures, and durations of the final degradation phase. Moreover, run-to-failure data are usually scarce, since collecting complete degradation histories requires long and expensive experiments or accelerated degradation tests. These characteristics make bearing RUL estimation a difficult generalization problem: a model must learn degradation patterns from a limited number of complete trajectories and transfer them to unseen bearings whose evolution may differ from the training examples.

A further practical requirement is that RUL estimation should be deployable close to the monitored machine. In many industrial settings, continuously transmitting raw high-frequency vibration data to a remote cloud server is not ideal. It may introduce latency, increase bandwidth usage, expose sensitive operational information, and create dependence on network availability. These limitations become even more relevant at scale when monitoring several machines simultaneously or when the system must react quickly to rapidly evolving degradation. Edge computing helps to address these issues by shifting computation closer to the data source [5], reducing communication overhead and enabling faster local decision-making [6,7].

Edge-based RUL estimation imposes additional constraints on neural model design. The model should be accurate, but also compact, computationally efficient, and compatible with limited memory and processing resources. This motivates the development of lightweight architectures that combine informative vibration features with small neural components, rather than relying exclusively on large hyper-parametrized deep neural networks. In this context, one goal is not only to maximize offline predictive accuracy, but also to devise a model that can realistically be executed on embedded or edge hardware for near-real-time condition monitoring.

This work proposes a compact two-stage RUL estimation framework for rolling-bearing prognostics. The proposed model combines compact local vibration representations

extracted from raw acceleration windows with handcrafted statistical features and predicts normalized RUL using a robust regression objective. The design is intentionally lightweight, with the aim of preserving predictive performance while reducing the computational requirement associated with the deployment.

This paper is structured as follows. Section 2 reviews the existing literature on bearing remaining useful life prediction, Edge AI-based predictive maintenance, and interpretable prognostic models. Section 3 summarizes the contributions brought by this work. Section 4 defines the problem addressed in this work and introduces the main requirements for a fully edge-resident two-stage solution. Section 5 describes the proposed architecture, including the sensor-level preprocessing stage, the temporal RUL prediction stage and the diagnostic output. Section 6 presents the experimental hardware setup. Section 7 reports the experimental results, including RUL prediction accuracy, per-bearing analysis, reasoning outputs, and deployment-oriented model export. Section 8 discusses the main findings, limitations, and implications for heterogeneous edge deployment. Finally, Section 9 concludes the paper and outlines possible future research directions.

## 2. Related Works

RUL prediction for rolling-bearings has been widely investigated using both feature-engineering-based approaches and deep learning methods. Public run-to-failure datasets such as PRONOSTIA/PHM2012 [1] and XJTU-SY [8] have played a central role in this research area, since they provide vibration signals acquired under different operating conditions and allow the evaluation of prognostic models on complete degradation trajectories. Despite their widespread use, the comparison among different studies is not straightforward, as the adopted train/test splits, RUL labeling strategies, preprocessing pipelines, and evaluation protocols often differ substantially.

A first group of methods relies on manually designed degradation indicators or handcrafted features. These approaches typically extract statistical, time-domain, frequency-domain, or time-frequency features from vibration signals and then use them to build health indicators or similarity-based prognostic models. For example, Wang et al. [8] proposed a hybrid prognostic framework for the XJTU-SY dataset, combining data-driven degradation representation with model-based life estimation. Sun and Sun [9] used PRONOSTIA with an instance-based learning strategy based on spectrogram features, principal component analysis, data augmentation, and similarity evaluation. These works highlight the importance of informative degradation features, especially when only a limited number of run-to-failure trajectories is available. However, their performance depends strongly on the quality of manually engineered representations and they are not specifically designed for compact deployment on edge hardware.

A second line of research has focused on end-to-end or semi-end-to-end deep learning models. Convolutional neural networks (CNNs) have been used to extract local degradation patterns directly from raw vibration signals or transformed signal representations. Wang et al. [10] proposed a deep separable CNN for machinery RUL prediction, reducing the need for manual feature extraction. Other works transformed the raw vibration signal into time-frequency representations before applying deep neural networks (DNNs). For instance, Liu et al. [11] used short-time Fourier transform representations combined with CNN, LSTM, and attention mechanisms. Similarly, approaches based on wavelet transforms or empirical mode decomposition convert the original signal into richer representations before feeding it to neural architectures [12,13]. These preprocessing strategies can improve the ability of the model to capture non-stationary degradation patterns, but they also increase computational complexity and may be less suitable for resource-constrained edge devices.

Recurrent neural networks (RNNs) and their variants have also been widely adopted for bearing RUL estimation because degradation naturally evolves as a temporal process captured by temporal data series. LSTM, GRU, BiLSTM, and BiGRU architectures are commonly used to model temporal dependencies in vibration features or raw signal windows. More recent works combine RNNs with attention mechanisms to emphasize the most informative time steps in the degradation sequence. Xu et al. [14] proposed a self-attention augmented CNN + GRU network, where a health indicator is first constructed and then processed by a neural model. Yu et al. [15] combined CNN, GRU, multi-head attention, and transfer learning to address limited-data and variable-load scenarios. These methods improve temporal modeling and adaptability, but they often introduce multiple processing stages and do not explicitly address the constraints of real-time edge inference.

Transformer-based and attention-based architectures have recently become prominent in bearing prognostics. These models are designed to capture long-range dependencies and dynamically assign different importance to different temporal regions or feature channels. Lyu et al. [16] proposed a BiGRU–ASSA–iTransformer architecture that processes raw vibration signals through a BiGRU with adaptive sparse self-attention and models handcrafted feature sequences through an iTransformer branch. The model integrates deep representations and handcrafted degradation features, showing the potential of hybrid architectures for bearing RUL prediction. Other studies adopt Transformer or Transformer–BiLSTM structures combined with signal decomposition techniques such as empirical mode decomposition [13]. While these methods are powerful, they are generally designed and evaluated as offline prognostic models, with limited discussion of model size, memory footprint, inference latency, or deployability on embedded hardware.

The main difference with the present work is therefore not the use of hybrid representations alone, but the deployment granularity of the hybrid architecture. While these methods combine neural and handcrafted information within a single offline prognostic model, the proposed framework explicitly partitions the computation across two embedded processing levels: an ISPU-based sensing stage for local vibration compression and an STM32MP2-class edge stage for temporal RUL prediction and structured diagnostic output.

Multi-branch architectures represent another important direction. These methods process different signal representations in parallel, such as raw vibration sequences, handcrafted features, denoised signals, or time-frequency maps. Tran et al. [17], for example, proposed a robust multi-branch DNN model that combines denoising, one-dimensional features, two-dimensional representations, and sequence modeling. Multi-branch models can capture complementary degradation information, but their complexity may become a limitation when the target application requires lightweight inference close to the sensing element. In practical industrial scenarios, especially when high-frequency vibration streams must be processed continuously, computational cost and memory requirements are primary aspects posing central design constraints.

Thus, although prior hybrid and multi-branch models demonstrate the benefit of combining complementary signal representations, they do not address the same system-level question considered here: how to distribute the prognostic pipeline between a programmable smart sensor and a low-power embedded edge processor.

Overall, the literature shows a clear trend toward increasingly complex preprocessing pipelines and neural architectures. Many existing methods rely on time-frequency image generation, empirical mode decomposition, autoencoder-based denoising, attention mechanisms, Transformer blocks, or multi-branch fusion networks. These strategies can be effective for offline accuracy, but they rarely provide a deployable edge-oriented analysis. In particular, most prior works do not explicitly evaluate whether the model can be deployed on a resource-constrained device, nor do they report deployment-oriented

aspects such as compactness, inference cost, memory usage, or suitability for local real-time monitoring. Some works have started to investigate edge-oriented RUL estimation architectures [18]; however, these solutions often remain based on a mixed edge–cloud paradigm, where local controllers perform feature extraction while the actual degradation modeling and RUL prediction are executed on remote cloud or supervisory platforms.

### 3. Contributions

The present work differs from the approaches reviewed in Section 2 by focusing on a compact, interpretable, and edge-oriented RUL estimation pipeline for rolling-bearing prognostics. Instead of relying on computationally intensive time-frequency images, cloud-side inference, or large Transformer-based architectures, the proposed method adopts a two-stage architecture in which a tiny sensor-level preprocessor extracts compact neural and physical representations from raw vibration data, while an embedded temporal model estimates the normalized RUL from sequences of these representations.

A first contribution of this work is the formulation of a compact two-stage prognostic pipeline suitable for fully edge-resident operation. The first stage reduces the dimensionality of high-frequency vibration acquisitions by processing short raw-signal chunks and generating a low-dimensional neural embedding together with lightweight physical health-related descriptors. The second stage performs temporal RUL prediction from compact representations and handcrafted vibration features, thereby limiting raw-data movement beyond the sensing level.

A second contribution is the integration of maintenance-oriented interpretability into the RUL estimation process. In addition to numerical RUL regression, the proposed pipeline generates diagnostic outputs associated with degradation stage, degradation trend, vibration evidence, confidence level, and suggested maintenance action. This restricted-vocabulary formulation provides constrained and interpretable diagnostic categories, while the final rule-based layer maps them into reproducible maintenance-oriented messages without relying on unconstrained language generation, which would be unsuitable for constrained embedded environments. These diagnostic outputs are trained and evaluated using weak pseudo-labels derived from the available degradation trajectory and vibration-feature trends. They are intended as auxiliary interpretability outputs and not as independently validated fault-diagnosis labels.

A third contribution is the use of a robust and physically consistent training and inference formulation. The main RUL regression branch is trained with a Huber objective, which reduces the influence of large residuals in noisy degradation trajectories, while auxiliary structured-diagnostic heads are trained with weak pseudo-labels. During inference, the predicted RUL sequence is refined through monotonic post-processing.

Finally, this work provides a deployment-oriented evaluation on heterogeneous edge hardware. The first stage is designed for sensor-side execution on an ISPU-based smart sensing device, whereas the second stage is mapped to an STM32MP2-class embedded processor. This allows the proposed architecture to be evaluated not only in terms of prediction accuracy, but also with respect to model compactness, memory footprint, inference latency, data movement reduction, and compatibility with practical Edge AI platforms.

### 4. Problem Definition

This work aims to address the following question: *is it feasible to deploy a two-stage Edge AI architecture for rolling-bearing prognostics, in which a tiny model performs local preprocessing at the smart sensor level while a second embedded model performs temporal RUL prediction and diagnostic reasoning on an edge microprocessor?*

The target application is predictive maintenance of rolling-bearings using vibration data. In this scenario, high-frequency sensor signals may generate a large amount of data, making continuous transmission to a cloud server or to a high-performance computing platform inefficient in terms of bandwidth, latency, energy consumption, and system autonomy. Therefore, the objective is to design a fully edge-resident pipeline able to estimate the remaining useful life of the bearing while also providing interpretable maintenance-oriented outputs.

The requirements for a candidate solution are the following:

- **Sensor-level preprocessing:** the first stage shall be lightweight enough to run close to the sensing element, for example on an STMicroelectronics Intelligent Sensor Processing Unit (ISPU), which integrates a programmable processing core suitable for tiny AI models.
- **Compact data representation:** the first stage shall reduce the dimensionality of the high-frequency vibration signal by producing a compact neural embedding and a small set of physical health-related features.
- **Edge-side temporal prediction:** the second stage shall process temporal sequences of compact representations on an embedded edge processor, such as a low-power STM32MP2-class platform, and estimate the normalized RUL of the bearing.
- **Interpretable diagnostic output:** in addition to numerical RUL prediction, the system shall generate maintenance-oriented diagnostic outputs, such as degradation stage, trend, vibration evidence, confidence level, and suggested maintenance action.
- **Physics consistency:** the predicted RUL trajectory shall be consistent with the physical nature of run-to-failure experiments, where the remaining useful life is expected to decrease over time.
- **Deployability:** the two stages shall be exportable as independent embedded models, supporting integration into heterogeneous edge hardware.
- **Competitive prediction accuracy:** the proposed architecture shall maintain acceptable RUL prediction performance compared with conventional one-stage models, despite the constraints introduced by sensor-level compression.

Despite significant advances in deep-learning-based RUL prediction, many existing approaches are not designed for deployment on constrained edge hardware, such as the ISPU and STM32MP2 targets considered in this work. Server-oriented models often process raw vibration windows directly and rely on relatively large neural architectures. Although such models can achieve high prediction accuracy, they may require substantial memory, computational resources, and data movement, which limits their applicability in low-power industrial edge systems.

Furthermore, most prognostic models focus only on numerical RUL estimation. This limits their practical usefulness in maintenance scenarios, where operators also require an interpretable indication of the degradation stage and the recommended action. Fully generative language models are not suitable for this task in constrained embedded environments, due to their computational cost and the risk of uncontrolled outputs. A restricted-vocabulary diagnostic mechanism is therefore preferable for edge deployment.

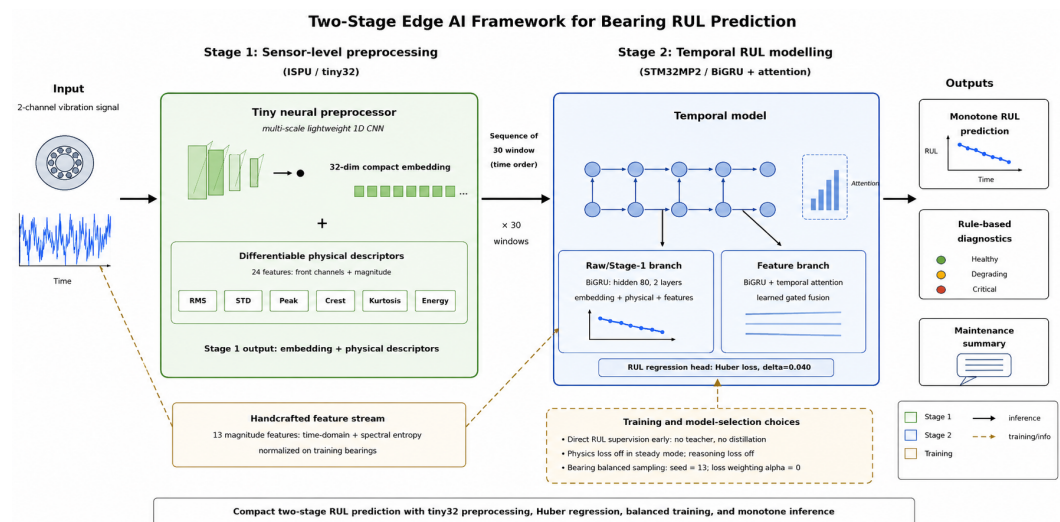
Therefore, the problem addressed in this work is the design of a compact, interpretable, physically consistent, and fully edge-resident RUL prediction pipeline for rolling-bearing prognostics. The solution must support sensor-level preprocessing, embedded temporal inference, structured maintenance-oriented diagnostic output, and heterogeneous hardware deployment, while preserving competitive prediction performance under the constraints imposed by Edge AI execution.

## 5. Proposed Solution

The proposed solution for rolling-bearing RUL prediction is a two-stage edge-oriented pipeline. It combines a compact sensor-level neural preprocessor, lightweight handcrafted vibration features, temporal recurrent modeling, and monotonic post-processing of the predicted RUL trajectory.

### 5.1. Architecture Overview

The proposed framework (Figure 1) is composed of two stages. Stage 1 processes each raw vibration acquisition and extracts a compact representation containing learned neural features and explicit physical descriptors. Stage 2 receives a temporal sequence of these representations, together with handcrafted vibration features, and predicts the normalized remaining useful life.



**Figure 1.** Overall two-stage edge-oriented RUL prediction pipeline.

### 5.2. Stage 1: Tiny Sensor-Level Preprocessor

Let  $\mathbf{x}_t \in \mathbb{R}^{L \times 2}$  denote the two-channel vibration acquisition at time step  $t$ , where  $L = 2560$  in the PRONOSTIA acquisitions considered in this work. To make the sensor-level processing compatible with a streaming edge implementation, the raw acquisition is not processed as a single long input. Instead, it is divided into  $B$  shorter non-overlapping chunks:

$$\mathbf{x}_t = [\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,B}], \quad \mathbf{x}_{t,b} \in \mathbb{R}^{L_c \times 2},$$

where  $L_c = 128$  is the chunk length and  $B = L/L_c = 20$ .

Each chunk is processed by the same tiny neural encoder:

$$\mathbf{z}_{t,b} = g_{\theta_1}(\mathbf{x}_{t,b}), \quad \mathbf{z}_{t,b} \in \mathbb{R}^8.$$

The encoder  $g_{\theta_1}$  is a compact one-dimensional convolutional network composed of three Conv1D–ReLU–AveragePooling blocks followed by a small fully connected projection. This design avoids expensive operations such as attention, layer normalization, or non-linearities that are difficult to deploy on very small sensor-side targets.

The local chunk embeddings are then aggregated over the full acquisition to obtain a compact neural representation:

$$\mathbf{e}_t = A(\mathbf{z}_{t,1}, \mathbf{z}_{t,2}, \dots, \mathbf{z}_{t,B}), \quad \mathbf{e}_t \in \mathbb{R}^{32},$$

where  $A(\cdot)$  denotes the aggregation module applied across the 20 chunk-level embeddings. In the proposed implementation, the aggregation combines summary statistics of the local embeddings in order to represent the full vibration acquisition while reusing the same lightweight chunk encoder.

In parallel, Stage 1 computes lightweight physical descriptors from the raw vibration signal. These include mean absolute value, standard deviation, RMS, peak value, peak-to-peak value, crest factor, kurtosis, and energy, computed over the two vibration channels and their magnitude. The physical descriptor vector is denoted as:

$$\mathbf{p}_t \in \mathbb{R}^{24}.$$

The final Stage 1 representation is obtained by concatenating the aggregated neural embedding and the physical descriptor vector:

$$\mathbf{u}_t = [\mathbf{e}_t, \mathbf{p}_t], \quad \mathbf{u}_t \in \mathbb{R}^{56}.$$

This formulation preserves the role of Stage 1 as a compact sensor-level representation module, while making the neural part suitable for streaming execution on short raw-signal chunks. In particular, the chunk encoder takes an input of size  $128 \times 2$  and produces an 8-dimensional local embedding, allowing the same encoder to be applied sequentially over the 20 chunks of each acquisition.

Stage 1 is initialized through an auxiliary pretraining phase in which the representation is encouraged to reconstruct handcrafted vibration features and provide a coarse RUL estimate:

$$\mathcal{L}_{pre} = \mathcal{L}_{feat} + \beta \mathcal{L}_{auxRUL}.$$

### 5.3. Handcrafted Feature Stream

In addition to the Stage 1 representation, the model uses a lightweight handcrafted feature stream. For each vibration acquisition, a compact feature vector  $\mathbf{h}_t$  is extracted from the vibration magnitude. The selected feature set includes time-domain and spectral descriptors such as mean value, standard deviation, variance, RMS, peak value, peak-to-peak value, skewness, kurtosis, crest factor, impulse factor, shape factor, margin factor, and spectral entropy.

These features are normalized using statistics computed from the training bearings and are passed to Stage 2 as a second temporal input.

### 5.4. Stage 2: Temporal RUL Prediction

Stage 2 processes temporal windows of consecutive acquisitions. In the selected configuration, each input sequence contains 30 consecutive acquisition steps. The Stage 1 representations and handcrafted features are analyzed by two recurrent branches.

The first branch processes the Stage 1 representation:

$$\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_T\},$$

using a bidirectional GRU with two recurrent layers and hidden size 88. The second branch processes the handcrafted feature sequence using a smaller bidirectional GRU-based feature encoder. Temporal attention is applied to both branches in order to summarize the most informative parts of the degradation trajectory.

The two temporal contexts are then combined through a learned gating mechanism:

$$\mathbf{c} = g\mathbf{c}^{raw} + (1 - g)\mathbf{c}^{feat},$$

where  $\mathbf{c}^{raw}$  and  $\mathbf{c}^{feat}$  are the contexts extracted from the Stage 1 and handcrafted-feature branches, respectively.

The fused representation is passed to a regression head that predicts the normalized RUL:

$$\hat{y}_t = \sigma(a_t),$$

where  $a_t$  is the network output logit and  $\sigma(\cdot)$  constrains the prediction to the interval  $[0, 1]$ .

### 5.5. Training Objective

The selected model is trained using a multi-task objective composed of the main normalized RUL regression loss and auxiliary structured-diagnostic losses. The primary regression term is the Huber loss:

$$\mathcal{L}_{RUL} = \frac{1}{N} \sum_{i=1}^N \text{Huber}_{\delta}(\hat{y}_i, y_i).$$

Let the prediction residual be:

$$r_i = \hat{y}_i - y_i.$$

The Huber loss is defined as:

$$\text{Huber}_{\delta}(\hat{y}_i, y_i) = \begin{cases} \frac{1}{2}r_i^2, & \text{if } |r_i| \leq \delta, \\ \delta\left(|r_i| - \frac{1}{2}\delta\right), & \text{otherwise.} \end{cases}$$

This loss behaves quadratically for small residuals and linearly for large residuals. Therefore, it combines two useful properties: near the optimum it behaves similarly to the mean squared error, encouraging precise regression, while for larger errors it reduces the influence of outliers compared with a purely quadratic loss. This is particularly useful in RUL prediction, where degradation trajectories may contain abrupt transitions, noisy windows, or bearings whose failure evolution differs from the dominant training pattern. The Huber loss was originally introduced in the context of robust estimation by Huber [19].

In addition to the RUL regression term, the diagnostic reasoning heads are trained using cross-entropy losses computed on the weak pseudo-labels described in Section 5.7. The total training objective is therefore:

$$\mathcal{L} = \mathcal{L}_{RUL} + \lambda_R \sum_{h \in \mathcal{H}} \mathcal{L}_{CE}^{(h)},$$

where  $\mathcal{H}$  denotes the set of auxiliary diagnostic heads and  $\lambda_R = 0.02$  in the selected configuration. These auxiliary losses are used only to encourage structured diagnostic consistency and do not replace the main RUL regression objective.

No late-life reweighting is used and therefore, all training samples contribute uniformly to the regression objective. A bearing-balanced sampler is used during training to reduce the dominance of bearings with a larger number of temporal windows.

### 5.6. Monotonic Post-Processing

Since PRONOSTIA consists of run-to-failure bearing experiments, the RUL is expected to decrease over time. For this reason, the selected inference variant applies a monotonic correction independently to each bearing after the raw RUL predictions are obtained:

$$\tilde{y}_t = \min(\hat{y}_t, \tilde{y}_{t-1}).$$

This post-processing step prevents physically inconsistent increases in the predicted RUL trajectory while preserving the learned degradation trend. The correction is applied independently to each bearing and only during inference. No exponential moving average is used in the selected paper-ready model.

### 5.7. Diagnostic Output

The primary model output is the normalized RUL estimate. In addition, the model includes lightweight reasoning heads associated with maintenance-oriented diagnostic categories. These categories include life stage, degradation trend, vibration evidence, confidence level, and suggested maintenance action.

The reasoning heads are trained using weak pseudo-labels derived from the normalized RUL and vibration-feature trends. These pseudo-labels encode deterministic assumptions linking the normalized RUL trajectory and vibration-feature evolution to a small set of maintenance-oriented categories. They are therefore not equivalent to expert-annotated diagnostic labels and should be interpreted as auxiliary structured explanations rather than as independently validated diagnostic classes.

The final monotonic RUL trajectory can also be converted into fixed rule-based messages. This rule-based diagnostic layer maps the predicted RUL and its temporal trend into reproducible maintenance-oriented templates. It avoids unconstrained language generation and provides deterministic explanations linked to the predicted degradation stage.

### 5.8. Inference and Deployment

During inference, each vibration acquisition is first processed by Stage 1, which produces a compact neural–physical representation. Stage 2 then analyses a temporal sequence of these representations and handcrafted features to estimate the RUL. Finally, the monotonic post-processing step is applied to obtain the physically consistent RUL trajectory.

The resulting pipeline is suitable for Edge AI deployment. Stage 1 can operate close to the sensing element to reduce the dimensionality of the raw vibration signal, while Stage 2 can run on a more capable embedded multi-core processor responsible for temporal degradation modeling and RUL estimation.

## 6. Hardware Deployment

The two-stage architecture is designed for heterogeneous edge deployment using two STMicroelectronics platforms, as shown in Table 1. Stage 1 is intended to run on an ST smart sensor embedding an Intelligent Sensor Processing Unit (ISPU; STMicroelectronics N.V., Geneva, Switzerland) [20]. The ISPU is an ultra-low-power programmable processing core integrated in selected ST MEMS sensors ( $2.5 \times 3.0 \times 0.83$  mm) and is designed to execute signal-processing and AI algorithms closer to the sensing level. In the considered device, the ISPU operates at 5 MHz, making it suitable for always-on, low-power preprocessing tasks rather than computationally intensive deep neural inference.

The ISPU is built upon the STRED Cross Development Tools, including the GCC-based C/C++ compiler version 8.3.1 (Cosmic Software, Chelmsford, MA, USA) [21], a proprietary reduced-energy instruction-set micro-architecture developed by ST. This architecture enables local execution of compact neural and signal-processing workloads inside the sensor package, reducing the need to continuously transfer raw high-frequency vibration streams to an external processor. For this reason, the Stage 1 model is deliberately designed as a very small chunk-based encoder, operating on short two-channel vibration segments and using only lightweight operations compatible with constrained sensor-side execution.

In the implemented system, this stage performs local vibration preprocessing and extracts a compact neural embedding together with lightweight physical health indicators. The resulting representation is then transmitted to the second processing level instead of the original raw signal. This reduces data movement, limits bandwidth requirements between the sensing unit and the edge processor, and supports a more energy-efficient fully edge-resident prognostic pipeline.

Stage 2 is mapped to the STM32MP257F-EV1 evaluation platform (STMicroelectronics, Geneva, Switzerland), based on the STM32MP257FAI3 microprocessor. The microprocessor integrates a dual-core Arm Cortex-A35 processor operating at up to 1.5 GHz, a 3D GPU, and a VeriSilicon neural processing unit (NPU) operating at up to 900 MHz. The evaluation platform is equipped with 4 GB of off-chip DDR4 RAM and 32-Gbit eMMC v5.0 storage. This platform represents the higher-level edge-processing node of the proposed architecture, complementing the sensor-side ISPU stage with substantially larger memory and compute resources for temporal prognostic inference.

In the proposed pipeline, the STM32MP257F-EV1 executes the Stage 2 temporal RUL model, including the BiGRU-based sequence processing, temporal attention, RUL regression head, and diagnostic reasoning outputs. Unlike Stage 1, which operates directly on short raw vibration chunks, Stage 2 receives only compact temporal sequences generated from the sensor-level preprocessing stage, together with the handcrafted feature stream. This design reduces the amount of high-frequency raw data transferred beyond the sensing layer and allows the STM32MP2 platform to focus on higher-level degradation modeling.

The resource-utilization analysis performed on the ST Edge AI Developer Cloud indicates that the workload is mainly assigned to the platform acceleration resources, with the NPU accounting for approximately 78.19% of the reported utilization and the remaining 21.81% assigned to the GPU.

These results show that the second stage can be executed on an embedded MPU-class edge device, enabling local RUL estimation and maintenance-oriented diagnostic generation without requiring cloud-side inference.

Both stages are exported as independent deployable models. This separation allows the first model to be integrated into the sensor-side processing flow, while the second model can be executed on the embedded Linux environment of the STM32MP2. The resulting architecture avoids cloud-side inference and supports local predictive maintenance with reduced data movement, lower latency, and improved deployment flexibility.

**Table 1.** Hardware deployment summary of the proposed two-stage Edge AI pipeline.

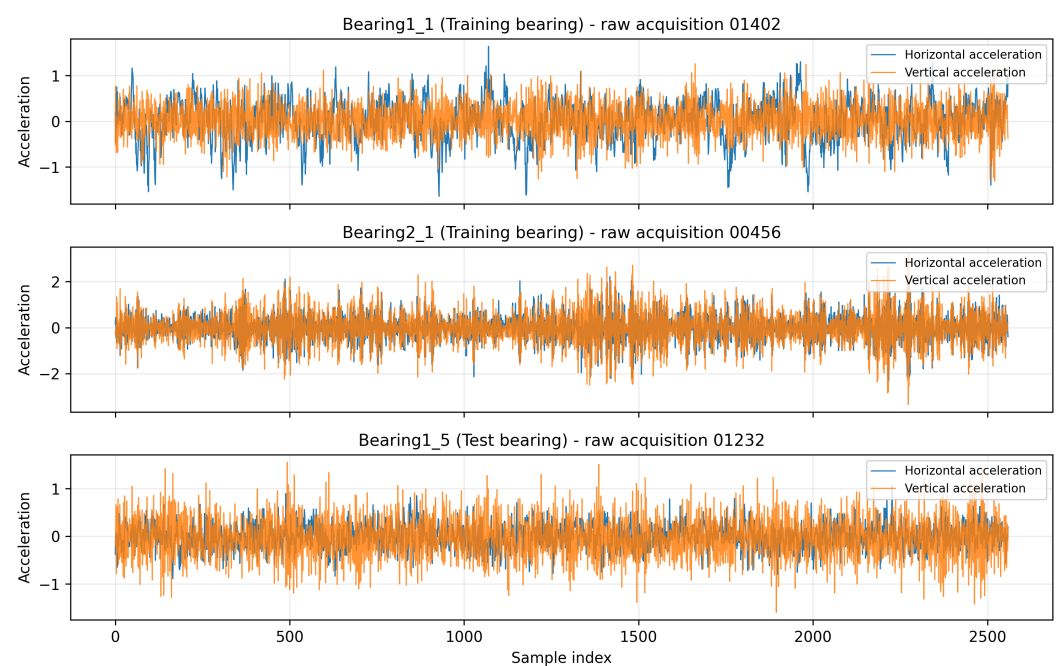
| Model Stage                      | Target Hardware          | Inference Time | RAM       | Flash     | Weights   | Parameters |
|----------------------------------|--------------------------|----------------|-----------|-----------|-----------|------------|
| Stage 1: Tiny Preprocessor       | LSM6DSO16IS @ 5MHz       | 68.62 ms       | 2.07 KiB  | 1.35 KiB  | 1.35 KiB  | 346        |
| Stage 2: RUL and Reasoning Model | STM32MP257F-EV1 @ 1.5GHz | 49.2 ms        | 90.68 MiB | 67.71 MiB | 67.71 MiB | 665,806    |

## 7. Experiments and Results

This section reports the experimental setup and quantitative results obtained with the selected edge-oriented RUL prediction model. All RUL values are normalized in the interval  $[0, 1]$ . The selected model is evaluated on a held-out PRONOSTIA test split composed of three unseen bearings: *Bearing1\_5*, *Bearing2\_5*, and *Bearing3\_3*. The evaluation includes global window-weighted metrics, per-bearing metrics, life-stage-specific errors, and auxiliary diagnostic reasoning outputs.

### 7.1. Dataset

The experimental evaluation is conducted on the PRONOSTIA bearing degradation dataset, also known as the IEEE PHM 2012 Data Challenge dataset. It contains run-to-failure vibration measurements collected from rolling-bearings operating under accelerated degradation conditions. The signals are acquired along two vibration axes and describe the complete evolution of each bearing from healthy operation to failure (Figure 2). The dataset includes three operating conditions, defined by different rotational speeds and radial loads. For each condition, multiple bearings are available, allowing the evaluation of both within-condition and cross-condition generalization. In this work, the vibration acquisitions are divided into temporal windows and each window is associated with a normalized RUL target. The dataset is particularly suitable for Edge AI prognostics because it combines high-frequency sensor data, nonlinear degradation behavior, and strong variability among different bearings.



**Figure 2.** Three examples (two from the training set and one from the test set) of raw vibration data of the bearings on the two axes.

### 7.2. Experimental Setup

The final model is a compact two-stage architecture. Each input sample consists of a two-channel vibration acquisition. Consecutive acquisitions are grouped into temporal sequences of length:

$$T = 30.$$

A bearing-held-out split was used for the experiments reported in this paper. The test set was composed of Bearing1\_5, Bearing2\_5, and Bearing3\_3, with one held-out bearing from each PRONOSTIA operating-condition family. The validation set was composed of Bearing1\_2, Bearing2\_2, and Bearing3\_2. The remaining bearings used for training were Bearing1\_1, Bearing1\_3, Bearing1\_4, Bearing1\_6, Bearing1\_7, Bearing2\_1, Bearing2\_3, Bearing2\_4, Bearing2\_6, Bearing2\_7, and Bearing3\_1. Temporal windows were generated with length  $T = 30$  acquisitions and stride 10 acquisitions. For a bearing trajectory with  $N_b$  acquisitions, the normalized RUL label associated with a window ending at acquisition index  $i$  was defined as

$$y_i = \frac{N_b - 1 - i}{N_b - 1},$$

so that the first acquisition has normalized RUL close to 1 and the final acquisition has normalized RUL equal to 0. The label of each temporal window was therefore assigned according to the last acquisition in the window.

The selected model uses a 32-dimensional neural embedding in Stage 1, concatenated with differentiable physical descriptors. A handcrafted feature stream composed of 13 vibration descriptors is also provided to the temporal prediction module. Stage 2 uses recurrent temporal modeling and attention to predict the normalized RUL.

The RUL regression term is trained using the Huber loss with threshold:

$$\delta = 0.040.$$

This relatively small threshold makes the objective more robust to large residuals, since the loss enters its linear regime earlier.

The final inference variant applies direct monotonic post-processing to the predicted RUL trajectory. Table 2 summarizes the selected model configuration.

**Table 2.** Selected model configuration used for the final PRONOSTIA evaluation.

| Component                                             | Selected Setting                          |
|-------------------------------------------------------|-------------------------------------------|
| Architecture                                          | Two-stage compact tiny32 model            |
| Stage 1 neural embedding                              | 32 dimensions                             |
| Stage 1 neural–physical output                        | 56 dimensions                             |
| Handcrafted feature stream                            | 13 vibration features                     |
| Temporal sequence length                              | 30 acquisitions                           |
| RUL regression loss                                   | Huber loss                                |
| Huber threshold $\delta$                              | 0.040                                     |
| Auxiliary diagnostic weight $\lambda_{\text{reason}}$ | 0.020                                     |
| Post-processing                                       | Monotonic correction                      |
| Test bearings                                         | <i>Bearing1_5, Bearing2_5, Bearing3_3</i> |

The training workflow was kept fixed for all main experiments. Stage 1 was initialized by auxiliary pretraining for eight epochs using AdamW with learning rate  $10^{-3}$  and weight decay  $10^{-3}$ . Stage 2 was trained using AdamW with learning rate  $6 \times 10^{-4}$ , weight decay  $8 \times 10^{-3}$ , cosine-annealing learning-rate scheduling with minimum learning rate  $7 \times 10^{-5}$ , and batch size 16. Training was run for at most 96 epochs, with early stopping based on validation RMSE and a patience of 24 epochs. The checkpoint with the lowest validation RMSE was selected for final test evaluation.

The configuration reported in Table 2 was selected through a validation-based grid search over compact edge-oriented model variants. The grid search considered both feature-level and architecture-level variants, including different combinations of Stage 1 neural chunk encoding, physical descriptors, handcrafted vibration features, temporal attention, gated fusion, and auxiliary structured-diagnostic heads. The selected configuration corresponds to the compact tiny32 two-stage model with Stage 1 neural and physical representations, the handcrafted feature stream, temporal attention, gated fusion, and structured diagnostic heads with  $\lambda_R = 0.020$ . This configuration was then used for the main results reported in the paper, while the ablation study reports the effect of removing individual components from the selected architecture.

### 7.3. Evaluation Metrics

The RUL prediction performance is evaluated using root mean squared error (RMSE) and mean absolute error (MAE):

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2},$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|.$$

### 7.4. RUL Prediction Results

Table 3 reports the RUL prediction error separately for each held-out bearing. This bearing-level evaluation provides a detailed view of the prediction performance across the different degradation trajectories.

**Table 3.** Per-bearing normalized RUL prediction results of the selected model.

| Bearing           | Windows | RMSE   | MAE    | Mean Error | Median Absolute Error |
|-------------------|---------|--------|--------|------------|-----------------------|
| <i>Bearing1_5</i> | 244     | 0.1276 | 0.1036 | 0.0008     | 0.0947                |
| <i>Bearing2_5</i> | 229     | 0.2609 | 0.2115 | −0.1989    | 0.2176                |
| <i>Bearing3_3</i> | 41      | 0.0963 | 0.0783 | 0.0473     | 0.0701                |
| <b>Average</b>    | –       | 0.1616 | 0.1311 | –          | –                     |

The selected model achieves an average RMSE of 0.1616 and an average MAE of 0.1311 across the three held-out bearings.

### 7.5. Ablation Study

Table 4 reports the contribution of the main architectural and inference components using the same bearing-average RMSE and MAE adopted for the main results. The selected model achieves the lowest average RMSE, while the raw output without monotonic post-processing obtains a slightly lower average MAE. This indicates a small trade-off between numerical accuracy metrics and physical consistency. In particular, the monotonic correction slightly improves the average RMSE from 0.1639 to 0.1616, while the average MAE increases from 0.1274 to 0.1311. This behavior is possible because RMSE is more sensitive to large errors, whereas MAE weights all absolute deviations linearly. Therefore, enforcing monotonicity can reduce larger non-physical fluctuations while introducing small corrections over several windows.

In the experiments, the raw RUL trajectories contained 226 monotonicity violations over 511 adjacent within-bearing prediction transitions. Thus, 44.2% of the within-bearing temporal transitions showed a local increase in the predicted remaining useful life. The causal correction  $\tilde{y}_t = \min(\hat{y}_t, \tilde{y}_{t-1})$  removed all these violations, reducing their number to zero. Importantly, this correction uses only the current raw prediction and the previous corrected output, and can therefore be applied during online inference without access to future samples.

The component ablations show that removing the handcrafted feature stream causes the largest degradation, confirming that the compact vibration descriptors provide important complementary information for temporal RUL prediction. Removing the chunk encoder, physical descriptors, temporal attention, reasoning heads, or gated fusion also increases the average error with respect to the selected configuration. Overall, these results support the final combination described in this work.

**Table 4.** Ablation study on the selected PRONOSTIA split composed of Bearing1\_5, Bearing2\_5, and Bearing3\_3. The reported RMSE and MAE values are averages across the three held-out bearings. Lower values are better.

| Setting                       | Chunk | Phys. | Hand. | Attn. | Reason. | Gating | Mono. | Avg. RMSE     | Avg. MAE      |
|-------------------------------|-------|-------|-------|-------|---------|--------|-------|---------------|---------------|
| Selected model                | ✓     | ✓     | ✓     | ✓     | ✓       | ✓      | ✓     | <b>0.1616</b> | 0.1311        |
| No monotonic post-processing  | ✓     | ✓     | ✓     | ✓     | ✓       | ✓      | ×     | 0.1639        | <b>0.1274</b> |
| No handcrafted feature stream | ✓     | ✓     | ×     | ✓     | ✓       | ×      | ✓     | 0.2366        | 0.1897        |
| No physical descriptors       | ✓     | ×     | ✓     | ✓     | ✓       | ✓      | ✓     | 0.1761        | 0.1484        |
| No temporal attention         | ✓     | ✓     | ✓     | ×     | ✓       | ✓      | ✓     | 0.2040        | 0.1650        |
| No reasoning heads            | ✓     | ✓     | ✓     | ✓     | ×       | ✓      | ✓     | 0.1926        | 0.1597        |
| No chunk encoder              | ×     | ✓     | ✓     | ✓     | ✓       | ✓      | ✓     | 0.1909        | 0.1588        |
| No gated fusion               | ✓     | ✓     | ✓     | ✓     | ✓       | ×      | ✓     | 0.1874        | 0.1513        |

Chunk: Stage 1 neural chunk encoder; Phys.: physical descriptors; Hand.: handcrafted feature stream; Attn.: temporal attention; Reason.: auxiliary diagnostic reasoning heads; Mono.: monotonic RUL post-processing. Best values in each column are shown in bold.

### 7.6. Diagnostic Analysis

The selected model includes auxiliary reasoning heads. These heads provide maintenance-oriented diagnostic outputs associated with life stage, degradation trend, vibration evidence, confidence level, and suggested maintenance action.

To evaluate the model across the degradation trajectory, the test windows are grouped into life-stage bins according to their normalized RUL. Table 5 reports the prediction error for each life-stage bin.

**Table 5.** Normalized RUL prediction error grouped by life-stage bin.

| Life-Stage Bin | Windows | RMSE   | MAE    | Mean Error |
|----------------|---------|--------|--------|------------|
| Healthy        | 97      | 0.1008 | 0.0787 | −0.0708    |
| Early          | 104     | 0.2637 | 0.1989 | −0.1939    |
| Intermediate   | 131     | 0.2442 | 0.2015 | −0.1776    |
| Advanced       | 103     | 0.1386 | 0.1233 | −0.0186    |
| Critical       | 79      | 0.1565 | 0.1204 | 0.1115     |

The life-stage evaluation shows the normalized error distribution across the full degradation trajectory, from healthy operation to the critical end-of-life region.

Table 6 reports the classification metrics of the other four auxiliary heads.

**Table 6.** Performance of the auxiliary diagnostic reasoning heads using the weak pseudo-labels.

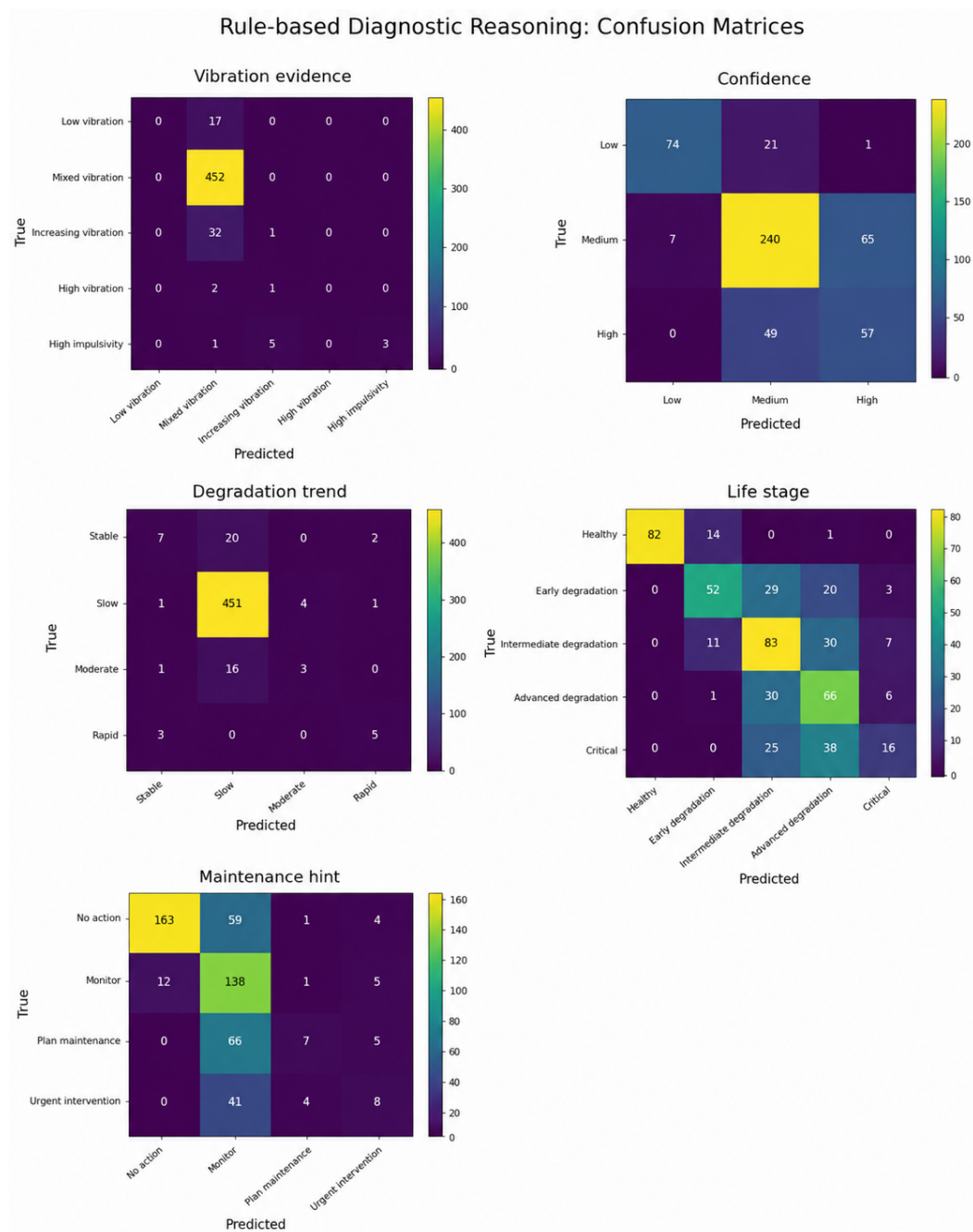
| Reasoning Head     | Windows | Accuracy |
|--------------------|---------|----------|
| Degradation trend  | 514     | 0.9066   |
| Vibration evidence | 514     | 0.8872   |
| Confidence level   | 514     | 0.7218   |
| Maintenance hint   | 514     | 0.6148   |

The auxiliary reasoning heads provide diagnostic information alongside the RUL estimate. The highest accuracy values are obtained for degradation trend and vibration evidence. These values should not be interpreted as expert-level diagnostic accuracy, since the reference categories are automatically derived from RUL and vibration-feature trends.

Figure 3 reports the confusion matrices obtained for the rule-based diagnostic reasoning outputs associated with the selected RUL trajectory. The matrices summarize five diagnostic attributes: vibration evidence, confidence level, degradation trend, life stage, and maintenance recommendation. For each attribute, rows indicate the pseudo-reference

diagnostic category and columns indicate the predicted category generated by the deterministic diagnostic layer. The values inside the cells represent the number of windows assigned to each pseudo-reference–predicted category pair.

The matrices show that vibration evidence and degradation trend are mostly concentrated in the dominant low-to-moderate categories, with limited off-diagonal dispersion. Life-stage errors mainly occur between adjacent stages, which is expected because these categories are obtained by thresholding a continuous RUL trajectory. The maintenance-hint matrix shows a conservative behavior, with most predictions assigned to no action, monitoring, or maintenance planning rather than to urgent intervention. Overall, the matrices serve as a consistency check of the deterministic diagnostic layer with respect to the rule-derived pseudo-labels, rather than as validation against expert-annotated diagnostic ground truth.



**Figure 3.** Confusion matrices of the diagnostic analysis.

### 7.7. Rule-Based Diagnostic Output

The final post-processed RUL trajectory is also converted into deterministic rule-based diagnostic messages. These messages provide a maintenance-oriented description of the predicted bearing state. Each output includes the estimated life stage, degradation trend, vibration evidence, confidence level, and suggested maintenance action.

Table 7 reports representative examples of the generated rule-based diagnostic outputs.

**Table 7.** Examples of deterministic rule-based diagnostic outputs generated from the selected RUL trajectory.

| Bearing           | Window End | True RUL | Pred. RUL | Abs. Error | Diagnostic Output                                                                                                     |
|-------------------|------------|----------|-----------|------------|-----------------------------------------------------------------------------------------------------------------------|
| <i>Bearing2_5</i> | 349        | 0.8489   | 0.8483    | 0.0006     | Healthy stage, moderate trend, low vibration evidence, high confidence, no action.                                    |
| <i>Bearing1_5</i> | 1499       | 0.3911   | 0.3918    | 0.0006     | Intermediate degradation stage, stable trend, increasing vibration evidence, medium confidence, maintenance planning. |
| <i>Bearing3_3</i> | 299        | 0.3095   | 0.3138    | 0.0043     | Advanced degradation stage, rapid trend, high vibration evidence, medium confidence, maintenance planning.            |

## 8. Discussion

The experimental results of Section 7 show that the two-stage architecture provides a compact and deployment-oriented solution for bearing RUL estimation. The main objective of the proposed framework is not only to reduce the normalized RUL prediction error, but also to obtain a model structure that can be executed across heterogeneous edge-processing levels. In this sense, the results reported in Table 1 are central to the interpretation of the proposed approach: Stage 1 is small enough to be analyzed on the LSM6DSO16IS ISPU, while Stage 2 is exported and benchmarked on the STM32MP257F-EV1 platform. This confirms that the architecture is not limited to an offline experimental setting, but follows a concrete deployment path in which raw vibration processing is moved close to the sensing element and temporal prognostic inference is performed on a more capable embedded processor.

A further aspect concerns the relation between inference latency and the temporal structure of the vibration data. In the PRONOSTIA acquisition setting, each vibration record contains 2560 samples per axis acquired at 25.6 kHz, corresponding to 0.1 s of signal. Since Stage 1 divides each acquisition into 20 non-overlapping chunks of 128 samples, each chunk represents 5 ms of vibration data. The measured Stage 1 latency on the ISPU target is 68.62 ms. If this latency is interpreted as the execution time of one chunk-level encoder call, processing the 20 chunks of one complete acquisition requires approximately 1.37 s. Therefore, the current sensor-side implementation should not be interpreted as supporting uninterrupted hard real-time processing of a continuous 25.6 kHz vibration stream.

However, the PRONOSTIA protocol is based on short vibration records acquired at discrete observation intervals rather than on continuous streaming acquisition. Under the standard 10 s observation period, the 1.37 s required by Stage 1, combined with the 49.2 ms latency of Stage 2, results in an end-to-end update latency of approximately 1.42 s. This corresponds to about 14.2% of the 10 s acquisition period, leaving a substantial time margin before the next vibration record is available. From this perspective, the proposed deployment is compatible with real-time operation at the acquisition-cycle level, while continuous sample-rate processing at the full vibration sampling frequency would require further optimization of the sensor-side stage.

The per-bearing results in Table 3 show that the selected model achieves an average normalized RMSE of 0.1616 and an average normalized MAE of 0.1311 across the three

held-out bearings. The results are especially consistent on Bearing1\_5 and Bearing3\_3, where the prediction error remains relatively limited and the mean error is close to zero. This indicates that, for these two trajectories, the model is able to follow the degradation evolution without introducing a strong systematic bias. Bearing2\_5 is more challenging and contributes most of the global error, as shown by its higher RMSE and negative mean error. Nevertheless, this behavior is useful from an experimental point of view because it highlights the difficulty of generalizing to unseen degradation trajectories in PRONOSTIA, where different bearings can exhibit substantially different degradation rates and end-of-life dynamics.

The final evaluation was conducted on three held-out PRONOSTIA bearings, selected to include one unseen trajectory from each operating-condition family. This setting provides a compact bearing-held-out test case for assessing both prognostic performance and edge deployability on complete run-to-failure trajectories. Although the results support the feasibility of the proposed two-stage edge pipeline, further evaluation on additional bearings and datasets would be useful to assess generalization under a wider range of degradation behaviors.

The life-stage analysis in Table 5 provides a more detailed view of how the prediction error is distributed along the degradation trajectory. The model obtains the lowest errors in the healthy region, while larger errors are observed in the early and intermediate degradation stages. This is coherent with the fact that these phases often correspond to transitions in which the degradation process has started but its future evolution is still uncertain. In contrast, the advanced and critical regions show lower errors than the early and intermediate stages. This is an important result for predictive maintenance, because the later degradation stages are the most relevant for maintenance planning and failure avoidance. The monotonic post-processing also contributes to this behavior by enforcing a physically consistent RUL trajectory during inference, preventing unrealistic increases in the estimated remaining life.

The reasoning results in Table 6 show that the auxiliary diagnostic heads provide structured categorical information alongside the numerical RUL estimate. These results were obtained using weak pseudo-labels generated from normalized RUL thresholds and vibration-feature trends, rather than from expert-annotated diagnostic labels. Therefore, the reported accuracies should be interpreted as agreement with the pseudo-labeling scheme, not as independently validated diagnostic correctness. This choice was necessary because the PRONOSTIA protocol used in this work provides run-to-failure vibration trajectories but does not include expert annotations for categories such as degradation trend, vibration evidence, confidence level, or maintenance action. Moreover, the RUL studies reviewed in Section 2 mainly report numerical prognostic performance and do not provide directly comparable expert-validated diagnostic output benchmarks. For this reason, the diagnostic layer should be considered an interpretable weakly supervised decision-support mechanism, whose rules and outputs would require validation by domain experts before use in real maintenance applications. The highest accuracies are obtained for degradation trend and vibration evidence, suggesting that the model captures useful information about the global direction of degradation and the vibration-related condition of the bearing. Confidence level and maintenance hint are more difficult categories, which is expected because they depend on higher-level interpretation of the RUL trajectory and on thresholds that compress a continuous degradation process into a small number of discrete maintenance-oriented labels. However, even in this case, the diagnostic layer provides a structured and reproducible output, which is preferable to unconstrained text generation in embedded predictive-maintenance applications.

The confusion matrices in Figure 3 further clarify the behavior of the diagnostic layer. For degradation trend and vibration evidence, most samples are concentrated around dominant and physically plausible categories, indicating that the reasoning outputs are stable rather than randomly distributed. The life-stage matrix shows that most errors occur between neighboring degradation stages, such as early, intermediate, and advanced degradation. This is an acceptable behavior for a rule-based or weakly supervised diagnostic representation, because the boundaries between consecutive degradation stages are inherently gradual. Similarly, the maintenance-hint matrix shows that the model mainly distributes predictions among adjacent maintenance actions, such as no action, monitoring, and maintenance planning, rather than producing inconsistent transitions across distant categories.

Table 7 provides representative examples of the final deterministic diagnostic messages generated from the post-processed RUL trajectory. These examples illustrate the intended role of the diagnostic layer: the output is not meant to replace the numerical RUL estimate, but to translate it into a maintenance-oriented summary that can be interpreted by an operator or by a higher-level monitoring system. The examples cover different degradation conditions and show that the same pipeline can provide a normalized RUL value, an absolute prediction error, a life-stage description, a degradation trend, vibration evidence, confidence information, and an associated maintenance action. This supports the use of the proposed architecture as a complete edge-oriented prognostic module rather than as a pure regression model.

Overall, the results confirm the feasibility of the proposed asymmetric Edge AI architecture. The main strength of the system is the combination of compact sensor-side preprocessing, temporal RUL modeling, monotonic physical consistency, and structured diagnostic output. The approach is particularly relevant for applications in which raw vibration data cannot be continuously transmitted or processed by large cloud-based models. Although the prediction accuracy is still affected by difficult bearing trajectories, the proposed framework offers a practical trade-off between regression performance, interpretability, and hardware feasibility. This trade-off is central to the goal of deploying RUL estimation directly at the edge, where memory footprint, latency, data movement, and model complexity must be considered together with predictive accuracy.

#### Comparison with Other Works

The results in Table 8 provide a contextual comparison with representative methods reported in [16]. This comparison should be interpreted with caution, since the preprocessing pipeline and deployment constraints are not necessarily equivalent across the compared studies. Therefore, the objective of this comparison is not to claim a strictly controlled ranking among methods, but rather to position the proposed approach with respect to commonly used offline RUL predictors and more complex accuracy-oriented architectures.

**Table 8.** Comparison with selected literature methods on the three held-out test bearings considered in this work. Literature results are reported from [16].

| Method                      | Bearing1_5   |              | Bearing2_5   |              | Bearing3_3   |              | Average      |              | Model Characteristics                                                                            | Edge-Deployment Evidence   |
|-----------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------------------------------------------------------------------------------------------|----------------------------|
|                             | RMSE         | MAE          | RMSE         | MAE          | RMSE         | MAE          | RMSE         | MAE          |                                                                                                  |                            |
| LSTM                        | 0.123        | 0.112        | 0.165        | 0.152        | 0.089        | 0.076        | 0.126        | 0.113        | Recurrent temporal baseline.                                                                     | Not reported               |
| BiGRU–Transformer–Attention | 0.068        | 0.056        | 0.126        | 0.091        | 0.182        | 0.171        | 0.125        | 0.106        | Hybrid recurrent-transformer architecture with attention mechanisms.                             | Not reported               |
| BAIT-RUL                    | <b>0.024</b> | <b>0.016</b> | <b>0.082</b> | <b>0.062</b> | <b>0.029</b> | <b>0.021</b> | <b>0.045</b> | <b>0.033</b> | Accuracy-oriented model combining BiGRU, ASSA, iTransformer, and handcrafted feature extraction. | Not reported               |
| This work                   | 0.128        | 0.104        | 0.261        | 0.212        | 0.096        | 0.078        | 0.162        | 0.131        | Compact hierarchical model with ISPU-oriented local encoding and MPU-level temporal inference.   | <b>Reported in Table 1</b> |

Best values in each column are shown in bold.

The selected baselines cover three different levels of model complexity. The LSTM represents a standard recurrent temporal model. The BiGRU–Transformer–Attention method introduces a more complex hybrid structure based on recurrent units, Transformer components, and attention mechanisms. BAIT-RUL represents a stronger accuracy-oriented reference, combining BiGRU, adaptive sparse self-attention, iTransformer, and handcrafted feature extraction in a multi-module framework [16]. In particular, BAIT-RUL is explicitly designed to maximize prediction performance by integrating several complementary temporal and feature-extraction mechanisms, whereas the present work follows a different design objective, maintaining competitive RUL prediction while enabling a concrete hierarchical edge-deployment path.

As shown in Table 8, the proposed model achieves prediction errors in the same order of magnitude as the recurrent and hybrid attention-based baselines on the selected test bearings. In particular, the proposed method provides competitive results on Bearing1\_5 and Bearing3\_3, while Bearing2\_5 remains the most challenging case. More complex methods, especially BAIT-RUL, achieve lower RMSE and MAE values; however, these approaches are primarily reported as offline prediction models and do not provide hardware-oriented evidence such as sensor-level memory footprint, inference latency, or compatibility with embedded deployment targets.

The main distinction of this work is therefore not only the predictive accuracy, but the explicit deployment-oriented design. The proposed Stage 1 encoder is reformulated as an ISPU-compatible chunk-based neural module operating on short raw vibration segments, while the temporal Stage 2 model is converted and benchmarked on an STM32MP257F-EV1 edge platform. This enables the RUL estimation pipeline to be evaluated not only in terms of regression metrics, but also in terms of practical deployability, memory footprint, and inference latency. Consequently, the proposed approach offers a trade-off between predictive performance and hardware feasibility, targeting edge-oriented RUL estimation rather than purely offline accuracy maximization.

## 9. Conclusions and Future Improvements

This work presented a compact two-stage Edge AI framework for rolling-bearing RUL prediction. The proposed architecture separates the prognostic task into two complementary processing levels: a tiny sensor-level preprocessing stage, designed to extract compact vibration representations from short raw-signal chunks, and an embedded temporal inference stage, responsible for RUL estimation and structured maintenance-oriented diagnostic output. This design enables high-frequency vibration data to be processed close to the sensing unit while preserving the temporal modeling capability required for degradation analysis.

The experimental results on the PRONOSTIA dataset show that the proposed model provides a practical trade-off between predictive performance, physical consistency, interpretability, and deployment feasibility. From the deployment viewpoint, the two stages were exported and evaluated on their respective target edge platforms, with Stage 1 targeting an ISPU-based smart sensing device and Stage 2 targeting an STM32MP2-class embedded platform. These results indicate that the proposed pipeline is not only an offline prognostic model, but a technically deployable edge-resident implementation path for local bearing-health monitoring under the acquisition-cycle assumptions considered in this work.

The monotonic post-processing improves the physical coherence of the predicted RUL trajectory by preventing non-physical increases in the estimated remaining life. The structured diagnostic outputs provide additional maintenance-oriented information about degradation stage, trend, vibration evidence, confidence level, and maintenance recommen-

dation. However, these diagnostic outputs are based on weak pseudo-labels derived from RUL thresholds and vibration-feature trends, rather than on expert-annotated diagnostic labels. Therefore, they should be interpreted as an interpretable decision-support layer and not as independently validated diagnostic ground truth.

The main contribution of this work is therefore not limited to numerical RUL regression, but exists in the design, implementation, and deployment-oriented evaluation of a compact edge prognostic pipeline. Unlike purely offline prognostic models, the proposed solution explicitly considers compactness, data reduction, local inference, hardware compatibility, and interpretable maintenance support. This makes the approach suitable for industrial scenarios in which raw vibration data cannot be continuously transmitted to cloud infrastructure and where low-latency local decision support is required.

Future work will focus on improving the predictive accuracy and robustness of the framework, especially on more challenging degradation trajectories, cross-condition generalization scenarios, and additional bearing datasets. Further work may also include compression, quantization, and pruning of the temporal Stage 2 model to reduce its memory footprint and improve deployment efficiency. Although the proposed stages have already been evaluated on the target edge hardware, broader experimental campaigns will be required to assess long-term robustness across different machines, operating regimes, sensors, and failure modes. Finally, the diagnostic layer should be refined and validated with the support of domain experts, incorporating expert-approved maintenance rules and uncertainty-aware confidence estimation before being used as an operational maintenance-decision module.

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

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are available at the following link: <https://www.kaggle.com/datasets/egorefimow/smallone> (accessed on 1 May 2026).

**Conflicts of Interest:** The authors declare no conflicts of interest. Danilo Pau is with STMicroelectronics, System Research and Applications in Agrate Brianza, Italy

## Abbreviations

The following abbreviations are used in this manuscript:

|        |                                      |
|--------|--------------------------------------|
| AI     | Artificial Intelligence              |
| ASSA   | Adaptive Sparse Self-Attention       |
| BiGRU  | Bidirectional Gated Recurrent Unit   |
| BiLSTM | Bidirectional Long Short-Term Memory |
| CNN    | Convolutional Neural Network         |
| DNN    | Dense Neural Network                 |
| DSP    | Digital Signal Processor             |
| GRU    | Gated Recurrent Unit                 |
| ISPU   | Intelligent Sensor Processing Unit   |
| LSTM   | Long Short-Term Memory               |
| MAE    | Mean Absolute Error                  |
| MEMS   | Micro-Electro-Mechanical Systems     |
| MLP    | Multi-Layer Perceptron               |

|         |                                   |
|---------|-----------------------------------|
| MPU     | Microprocessor Unit               |
| ONNX    | Open Neural Network Exchange      |
| PHM     | Prognostics and Health Management |
| ReLU    | Rectified Linear Unit             |
| RMSE    | Root Mean Squared Error           |
| RMS     | Root Mean Square                  |
| RUL     | Remaining Useful Life             |
| STD     | Standard Deviation                |
| Tiny ML | Tiny Machine Learning             |

## References

1. Nectoux, P.; Gouriveau, R.; Medjaher, K.; Ramasso, E.; Morello, B.; Zerhouni, N.; Varnier, C. PRONOSTIA: An Experimental Platform for Bearings Accelerated Degradation Tests. In Proceedings of the IEEE International Conference on Prognostics and Health Management, Denver, CO, USA, 18–21 June 2012; pp. 1–8.
2. Heng, A.; Zhang, S.; Tan, A.C.C.; Mathew, J. Rotating Machinery Prognostics: State of the Art, Challenges and Opportunities. *Mech. Syst. Signal Process.* **2009**, *23*, 724–739. <https://doi.org/10.1016/j.ymssp.2008.06.009>.
3. Jardine, A.K.S.; Lin, D.; Banjevic, D. A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance. *Mech. Syst. Signal Process.* **2006**, *20*, 1483–1510. <https://doi.org/10.1016/j.ymssp.2005.09.012>.
4. Lee, J.; Wu, F.; Zhao, W.; Ghaffari, M.; Liao, L.; Siegel, D. Prognostics and Health Management Design for Rotary Machinery Systems: Reviews, Methodology and Applications. *Mech. Syst. Signal Process.* **2014**, *42*, 314–334. <https://doi.org/10.1016/j.ymssp.2013.06.004>.
5. Pau, D.P.; Tognocchi, S. Reviewing progresses on In-Sensor AI Computing. *Microprocess. Microsyst.* **2025**, in press, 105156. <https://doi.org/10.1016/j.micpro.2025.105156>.
6. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. <https://doi.org/10.1109/JIOT.2016.2579198>.
7. Satyanarayanan, M. The Emergence of Edge Computing. *Computer* **2017**, *50*, 30–39. <https://doi.org/10.1109/MC.2017.9>.
8. Wang, B.; Lei, Y.; Li, N.; Li, N. A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings. *IEEE Trans. Reliab.* **2020**, *69*, 401–412. <https://doi.org/10.1109/TR.2018.2882682>.
9. Sun, Q.; Sun, J. Bearing Prognostics: An Instance-Based Learning Approach with Feature Engineering, Data Augmentation, and Similarity Evaluation. *Signals* **2021**, *2*, 662–687. <https://doi.org/10.3390/signals2040040>.
10. Wang, B.; Lei, Y.; Li, N.; Yan, T. Deep Separable Convolutional Network for Remaining Useful Life Prediction of Machinery. *Mech. Syst. Signal Process.* **2019**, *134*, 106330. <https://doi.org/10.1016/j.ymssp.2019.106330>.
11. Liu, B.; Gao, Z.; Lu, B.; Dong, H.; An, Z. Deep Learning-Based Remaining Useful Life Estimation of Bearings with Time-Frequency Information. *Sensors* **2022**, *22*, 7402. <https://doi.org/10.3390/s22197402>.
12. Jin, Z.; Chen, C.; Syntetos, A.; Liu, Y. Enhanced Bearing RUL Prediction Based on Dynamic Temporal Attention and Mixed MLP. *Auton. Intell. Syst.* **2025**, *5*, 2. <https://doi.org/10.1007/s43684-024-00088-4>.
13. Jin, C.; Li, B.; Yang, Y.; Yuan, X.; Tu, R.; Qiu, L.; Chen, X. Remaining Useful Life Prediction of Rolling Bearings Based on Empirical Mode Decomposition and Transformer Bi-LSTM Network. *Appl. Sci.* **2025**, *15*, 9529. <https://doi.org/10.3390/app15179529>.
14. Xu, J.; Duan, S.; Chen, W.; Wang, D.; Fan, Y. SACGNet: A Remaining Useful Life Prediction of Bearing with Self-Attention Augmented Convolution GRU Network. *Lubricants* **2022**, *10*, 21. <https://doi.org/10.3390/lubricants10020021>.
15. Yu, J.; Shao, J.; Peng, X.; Liu, T.; Yao, Q. Remaining Useful Life of the Rolling Bearings Prediction Method Based on Transfer Learning Integrated with CNN-GRU-MHA. *Appl. Sci.* **2024**, *14*, 9039. <https://doi.org/10.3390/app14199039>.
16. Lyu, Y.; Qiu, Q.; Chu, Y.; Zhang, J. An Adaptive BiGRU-ASSA-iTransformer Method for Remaining Useful Life Prediction of Bearing in Aerospace Manufacturing. *Actuators* **2025**, *14*, 238. <https://doi.org/10.3390/act14050238>.
17. Tran, K.; Vu, H.C.; Pham, L.; Boudaoud, N.; Nguyen, H.S.H. Robust-MBDL: A Robust Multi-Branch Deep-Learning-Based Model for Remaining Useful Life Prediction of Rotating Machines. *Mathematics* **2024**, *12*, 1569. <https://doi.org/10.3390/math12101569>.
18. Barbieri, M.; Nguyen, K.T.P.; Diversi, R.; Medjaher, K.; Tilli, A. RUL Prediction for Automatic Machines: A Mixed Edge-Cloud Solution Based on Model-of-Signals and Particle Filtering Techniques. *J. Intell. Manuf.* **2021**, *32*, 1421–1440. <https://doi.org/10.1007/s10845-020-01696-6>.
19. Huber, P.J. Robust Estimation of a Location Parameter. In *Breakthroughs in Statistics: Methodology and Distribution*; Kotz, S., Johnson, N.L., Eds.; Springer: New York, NY, USA, 1992; pp. 492–518. [https://doi.org/10.1007/978-1-4612-4380-9\\_35](https://doi.org/10.1007/978-1-4612-4380-9_35).

20. STMicroelectronics. *IMUs with Intelligent Sensor Processing Unit: Decision-Making in the Edge*; STMicroelectronics N.V.: Geneva, Switzerland. Available online: [https://www.st.com/content/st\\_com/en/campaigns/ispu-ai-in-sensors.html](https://www.st.com/content/st_com/en/campaigns/ispu-ai-in-sensors.html) (accessed on 12 May 2026).
21. Cosmic Software. *STRED Cross Development Tools, Including a C/C++ Compiler Based on GCC v8.3.1*; Cosmic Software: Chelmsford, MA, USA. Available online: <https://www.cosmicsoftware.com/stred.php> (accessed on 9 May 2026).

**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.