

Predictive Maintenance MCP: An Open-Source Framework for Bridging Large Language Models and Industrial Condition Monitoring via the Model Context Protocol

Original

Predictive Maintenance MCP: An Open-Source Framework for Bridging Large Language Models and Industrial Condition Monitoring via the Model Context Protocol / Di Maggio, L.G.. - In: APPLIED SCIENCES. - ISSN 2076-3417. - 16:6(2026). [10.3390/app16062812]

Availability:

This version is available at: 11583/3013698 since: 2026-07-29T10:17:49Z

Publisher:

Multidisciplinary Digital Publishing Institute (MDPI)

Published

DOI:10.3390/app16062812

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Article

Predictive Maintenance MCP: An Open-Source Framework for Bridging Large Language Models and Industrial Condition Monitoring via the Model Context Protocol

Luigi Gianpio Di Maggio 

Dipartimento di Ingegneria Meccanica e Aerospaziale (DIMEAS), Politecnico di Torino, Corso Duca Degli Abruzzi 24, 10129 Torino, Italy; luigi.dimaggio@polito.it

Featured Application

The proposed MCP server can be deployed as an assistant layer in industrial condition monitoring systems to support vibration-based diagnostics, bearing fault detection, and ISO 20816-3 compliance checks. Maintenance teams can query signals and documentation in natural language while relying on audited analyses and automatically generated diagnostic reports.

Abstract

This paper presents a Proof of Concept (PoC) for *PredictiveMaintenance MCP*, an open-source server based on the Model Context Protocol (MCP) that supports machine condition monitoring and predictive maintenance via natural language interaction with Large Language Models (LLMs). The server constrains the LLM within an explicit perimeter of deterministic resources and tools for vibration-based diagnostics, including FFT spectral analysis with peak identification, envelope analysis for rolling element bearing defects, time-domain indicators, vibration severity assessment consistent with ISO standards and semi-supervised anomaly detection on extracted features. Each tool invocation produces structured outputs and artifacts that record inputs, parameters, and results. The LLM acts as an orchestrator that selects resources, configures parameters, invokes tools, and synthesizes conclusions anchored to computed evidence, thereby improving traceability and repeatability compared to unconstrained text-only interaction. End-to-end workflows are demonstrated in a reproducible package with code, examples, and demo data to support community-driven validation and extension toward industrial requirements. The software is archived on Zenodo and the GitHub repository serves as the collaboration hub.

Keywords: predictive maintenance; condition monitoring; model context protocol; large language models; LLMs; vibration analysis; rolling element bearing diagnostics; envelope analysis; ISO 20816-3; anomaly detection



Academic Editors: Stefan Fischer and Mohamed Benbouzid

Received: 28 January 2026

Revised: 17 February 2026

Accepted: 13 March 2026

Published: 15 March 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Predictive maintenance (PdM) is a key element in modern industrial operations, as it enables planning interventions based on the actual condition of assets, reducing downtime and maintenance costs and increasing availability and production continuity [1,2]. In particular, for rotating machinery, such as electric motors, pumps, compressors, and transmissions, condition monitoring based on sensors such as vibration sensors [3] is an established practice for early detection of failures and degradation. The spread of the

Industrial Internet of Things (IIoT) [4,5] has further amplified the availability of continuous and contextual data, enabling data-driven pipelines for diagnostics [6] and condition monitoring [7], anomaly detection and, ultimately, residual life estimation [2,8]. In this context, rolling bearings are often one of the most informative observation points in rotating machines [9–11].

Despite the effectiveness of many machine learning and deep learning methods [12–14] applied to PdM, obstacles remain that slow down large-scale industrial adoption. These include: a dependency on labeled and representative datasets, which are often difficult to obtain in real-world scenarios [15,16]; difficulty transferring data between different machines and operating conditions and domain shift [12]; the need for interpretability and traceability, especially when outputs influence operational and safety decisions; and the fragmentation of tools and formats, which makes integration into existing maintenance workflows complex. Consequently, even when advanced algorithms are available, their deployment requires an application orchestrator that connects data, procedures, standards, and domain knowledge while maintaining reproducibility and control.

In recent years, Large Language Models (LLMs) have demonstrated remarkable capabilities for generalization and natural language interaction. In the PdM context, these models can enable practical functions [17–19] such as consulting technical documentation and standards, supporting the drafting of reports and procedures with Failure Mode and Effect Analysis [20,21], synthesizing evidence from different tools [4,22], and integrating with knowledge bases via Retrieval Augmented Generation (RAG) [15,20,23,24]. LLM-based assistants are being explored in several asset-intensive sectors such as oil and gas, power generation, and manufacturing to streamline maintenance decision-making and to support engineers in navigating heterogeneous data sources such as manuals, service logs, and monitoring data [22,25]. These early deployments highlight two requirements that are particularly critical for rotating machinery diagnostics. Namely, governance and traceability of the analytical chain, and privacy and controlled access to plant data.

Also, using an LLM as a stand-alone component for diagnostic tasks introduces significant criticalities. The textual output can be non-deterministic, potentially affected by hallucinations or unverifiable inferences, and generally lacks a reproducible chain of evidence. In industrial contexts, this makes an architectural layer essential that explicitly governs which data the model can use and which operations it can invoke, with traceable parameters and inspectable results [26]. Anyone who wanted to implement such a system would also need to produce Application Programming Interface (API) access points, which are application-specific, and each company would therefore have to have its own.

In this direction, the Model Context Protocol (MCP) [27] proposes a standardized approach for connecting LLM-based applications to documents, signals, databases and tools represented by invocable deterministic functions through a host–client–server architecture. The central idea is that the LLM should not invent analyses or calculations, but rather orchestrate explicit procedures [27,28], with structured and potentially auditable inputs and outputs. This paradigm is particularly suited to PdM since diagnosis must remain anchored to reproducible measures and transformations typically related to FFT, envelope analysis, and statistical indicators, and often must include compliance checks against industry standards.

This work presents *Predictive Maintenance MCP*, an open-source MCP server in a Proof of Concept (PoC) that enables diagnostic workflows for rotating machinery focused on vibration analysis and rolling element bearing defects. The server exposes example resources for accessing signals and documentation, and deterministic tools for FFT spectral analysis with peak identification, envelope analysis to highlight typical bearing defect modulations, time-domain indicator extraction, vibration severity assessment consistent

with ISO 20816-3 [29], and semi-supervised anomaly detection modules based on time-domain features [30]. An additional architectural element is the automatic generation of persistent artifacts in the form of interactive HTML reports and metadata, thus separating conversational interaction from the preservation of technical evidence.

This work contributes a governed reference architecture for LLM-assisted predictive maintenance by implementing consolidated diagnostic procedures as deterministic MCP tools within an explicit execution perimeter. The key innovations are reported in the following:

- Governed perimeter for PdM. The LLM is constrained to explicit MCP resources and tool invocations, limiting unsupported inferences and enabling controlled access to plant data.
- Deterministic tool chain. Established analyses are exposed as typed server-side tools, improving reproducibility compared to unconstrained text-only interaction.
- Persistent artifacts for the chain of evidence. Each tool execution produces inspectable reports and metadata that record inputs, parameters, and results, supporting auditability and post hoc review.
- MCP standardization as an integration mechanism. The PoC provides an open-source MCP reference implementation tailored to vibration-based condition monitoring, intended as a baseline for community-driven extensions toward industrial deployment.

Beyond the software description, this manuscript frames the PoC as a governed architecture for LLM-assisted diagnostics, and evaluates it through workflow properties rather than through fault-classification accuracy on a specific dataset. The software is archived and versioned on Zenodo (<https://doi.org/10.5281/zenodo.17611542>, accessed on 15 February 2026) [31]. This paper introduces the version 0.4.1, however newest packages have been released. The GitHub repository serves as the collaboration hub for issues, discussions, and contributions.

2. Background

Recent advances in Large Language Models (LLMs) have enabled the development of AI agents. Those systems extend beyond text generation by coordinating external tools and data sources to support multi-step tasks that involve perception, reasoning, planning, and action [26,32]. In Predictive Maintenance (PdM), agentic systems are particularly interesting because they can provide a unified natural language interface to heterogeneous assets, including sensor signals, knowledge repositories, and analytical procedures for signal processing, diagnostics and compliance checks. Potentially, this supports end-to-end workflows from inspection to reporting. However, increasing autonomy also amplifies governance requirements since, without explicit constraints on accessible data, permitted actions, and procedural boundaries, an agent may produce non-verifiable conclusions or execute unintended operations. Therefore, PdM-oriented agents benefit from a controlled execution perimeter in which tools, permissions, and evidence are explicitly defined and auditable. Figure 1 summarizes this paradigm. The LLM coordinates the workflow, while data, tools, and policies define the technical and procedural boundary within which the system can operate [32].

LLMs are increasingly investigated as assistant layers for condition monitoring workflows, primarily because they provide a natural language interface to heterogeneous assets such as sensor signals, maintenance logs, and technical documentation [4]. In rotating machinery applications, this can support practical tasks such as retrieving and summarizing procedures from manuals and standards, guiding non-expert users through established analyses and synthesizing evidence from multiple analysis outputs into structured reports [22,25]. However, diagnostic use also exposes well-known limitations since textual

outputs may be non-deterministic, may omit critical assumptions, and can include unsupported inferences if the model is not constrained to computed evidence. Therefore, rotating machinery diagnostics particularly benefit from architectures that enforce traceability and restrict the assistant to auditable data access and deterministic computations [26]. This motivates research artifacts that enable reproducible experimentation under explicit governance constraints, as pursued in this work.

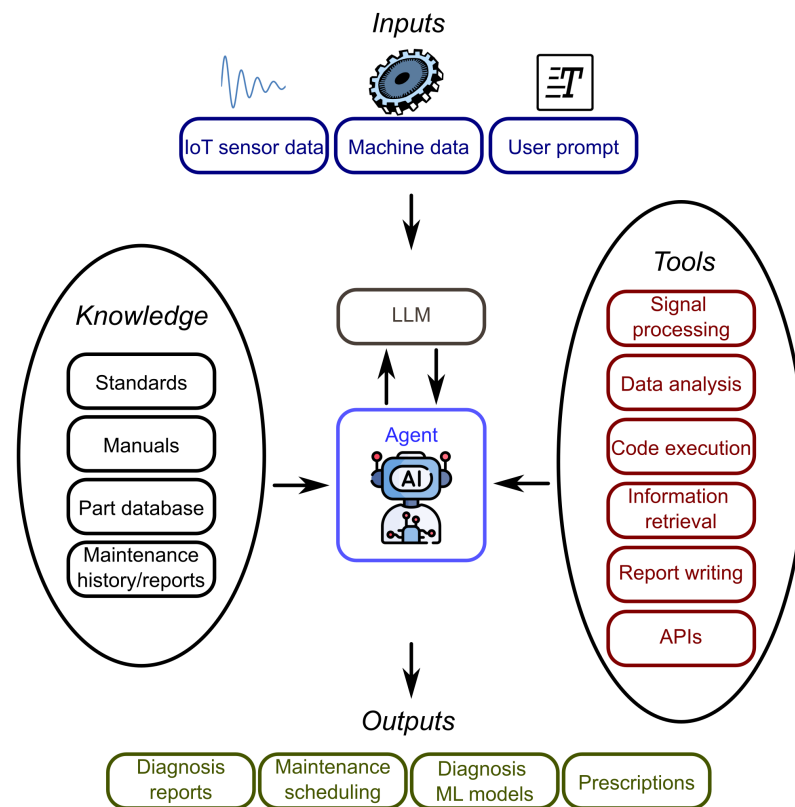


Figure 1. Conceptual overview of a predictive maintenance agent: the LLM orchestrates the workflow while tools, data resources, and policies define an auditable execution perimeter [32].

Table 1 summarizes representative research branches on LLM-based assistants for PdM together with their main advantages and limitations. A broader consolidation of evidence and use cases is available in [32]. Moreover, while research on LLM assistants and agentic systems for PdM is rapidly growing, PdM implementations explicitly adopting the MCP for governed tools and resources invocation are still scarce. This PoC therefore contributes as an open-source MCP reference implementation tailored to vibration-based condition monitoring, with deterministic tools and persistent artifacts to support traceability and repeatability.

2.1. Model Context Protocol (MCP)

Deploying LLM-based agents in industrial contexts requires a reliable mechanism to connect the assistant to data sources and computational tools while reducing integration fragmentation and improving portability, control, and security. The Model Context Protocol (MCP) [27,28] addresses this need by standardizing how an AI application can access external context and invoke functions through a host–client–server architecture. MCP formalizes three primitives (Figure 2):

- Resources are endpoints for direct and structured access to data and context, including local files, databases, and technical documents;

- Tools are server-side functions with typed inputs and outputs that perform deterministic operations such as signal analysis, normative calculations and database queries;
- Prompts are reusable templates that guide multi-step workflows and standardize recurring procedures [27,28].

Table 1. Representative research streams on LLM-based assistants and agentic systems for predictive maintenance.

Topic	Key Points	Representative Sources
Enterprise LLM assistants for maintenance decision support	Pros: Natural language access to procedures, maintenance records, and operational context; reported productivity gains in an internal study; conversational access can streamline dashboard workflows. Cons: Risk of incorrect and untrustworthy answers on uncontextualized industrial data; strong dependence on data quality and quantity and secure enterprise integration.	[22,25]
Context-aware, tool-using prescriptive agents	Pros: Agentic LLM for context-aware maintenance decision support via tool usage; case study shows fault detection without extra training or fine-tuning with performance comparable to baselines. Cons: The prediction-to-prescription transition is still described as not yet mature.	[33]
Domain-specialized LLMs for PdM	Pros: Foundation models and adaptation pipelines for rotating machinery show few-shot, cross-condition and cross-dataset potential; approaches include feature textualization or numeric encoding; a multimodal maintenance LLM reports improved accuracy vs generalist LLMs in curated tests. Cons: Requires domain data and dedicated signal-to-text or encoding pipelines; evaluation is often performed on constructed datasets and conditions.	[34–37]
Digital Twins (DT) based on Generative AI for diagnosis and PdM	Pros: Generative AI supports DT-based simulation and synthetic data generation; DT and LLM integration can enhance temporal feature learning and multimodal reasoning in industrial scenarios; DT and LLM pipelines are proposed for inspection tasks with good reported performance in case studies. Cons: Challenges highlighted include computational complexity and data security; DT maturity gaps and missing components for fully automated PdM remain.	[4,5,38–40]

Table 1. Cont.

Topic	Key Points	Representative Sources
LLM-assisted documentation for maintenance (e.g., FMEA drafting)	Pros: Expert-guided generation of structured maintenance artifacts; empirical analysis reports that models can generate a substantial portion of key FMEA content. Cons: Requires expert-in-the-loop supervision for correctness and completeness.	[20]
Governance, safety, and deployment architectures	Pros: Governing frameworks frame delegation, oversight and liability risks for AI agents; surveys and reference architectures discuss industrial deployment along edge–fog–cloud and trust aspects; edge constraints and challenges are explicitly discussed. Cons: Works emphasize governance gaps and the need for new technical infrastructure; recurring constraints include resource limits, security, and operational deployment complexity.	[26,41–44]

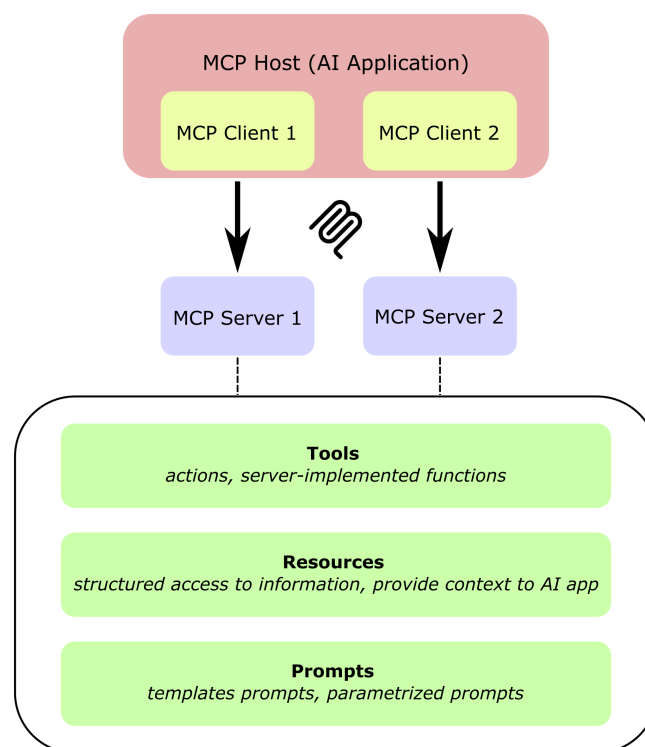


Figure 2. Model Context Protocol (MCP) architecture and main primitives [32].

Architecturally, the MCP separates the definition and exposure of capabilities (server) from their orchestration (client). This separation is particularly relevant for PdM, where trustworthy diagnostics depend on the ability to anchor conclusions to measured data and reproducible procedures. In practice, a workflow based on the MCP reduces reliance on

implicit assumptions and constrains the LLM to explicitly permitted operations, improving traceability and supporting artifacts produced by deterministic tools.

2.2. Vibration and Envelope Analysis for Rolling Bearing Diagnostics

Vibration analysis is a well-established approach for diagnosing rotating machinery, as several mechanical phenomena manifest as characteristic patterns in the frequency domain and in time-domain indicators [3]. For rolling bearings, localized defects often generate impulsive excitations that activate structural resonances at higher frequencies [45]. These impulses are modulated by defect frequencies related to bearing kinematics and shaft speed, and may be partially masked in the spectrum of the raw signal. Envelope analysis [46,47] is a standard procedure to highlight such modulations [3]. In brief, a resonant frequency band is isolated via band-pass filtering, the signal envelope is extracted typically via the Hilbert transform, and the envelope spectrum is analyzed to identify defect-related components and harmonics.

Let $x(t)$ be the measured vibration signal and $x_f(t)$ its band-pass filtered version in a selected resonance band. The operator $\mathcal{H}\{\cdot\}$ denotes the Hilbert transform, and j denotes the imaginary unit. Given a vibration signal $x(t)$, after band-pass filtering, we obtain $x_f(t)$. The envelope can be computed from the analytic signal:

$$z(t) = x_f(t) + j \mathcal{H}\{x_f(t)\}, \quad (1)$$

where $\mathcal{H}\{\cdot\}$ denotes the Hilbert transform. The envelope is:

$$e(t) = |z(t)| = \sqrt{x_f^2(t) + \mathcal{H}\{x_f(t)\}^2}. \quad (2)$$

The envelope spectrum emphasizes the modulation components typical of bearing defects. Key theoretical defect frequencies include BPFO (outer race), BPFI (inner race), BSF (rolling element spin), and FTF (cage). Assuming shaft rotation frequency f_r (Hz), the number of rolling elements n , rolling element diameter d , pitch diameter D , and contact angle ϕ , these frequencies can be expressed as [3]:

$$\text{BPFO} = \frac{n}{2} f_r \left(1 - \frac{d}{D} \cos \phi \right), \quad (3)$$

$$\text{BPFI} = \frac{n}{2} f_r \left(1 + \frac{d}{D} \cos \phi \right), \quad (4)$$

$$\text{BSF} = \frac{D}{2d} f_r \left(1 - \left(\frac{d}{D} \cos \phi \right)^2 \right), \quad (5)$$

$$\text{FTF} = \frac{1}{2} f_r \left(1 - \frac{d}{D} \cos \phi \right). \quad (6)$$

Comparing observed peaks with these frequencies supports the diagnostic hypothesis, especially when components related to the fault are weak or masked by other excitations in the conventional FFT spectrum.

2.3. Vibration Severity Assessment

Beyond fault type identification, PdM workflows often require estimating vibration severity in a manner consistent with industrial standards. ISO 20816-3 [29] provides guidelines for measuring and evaluating vibration on industrial machinery within defined power classes and operating speed ranges, introducing assessment criteria typically based on broadband indicators as the RMS velocity and severity zones associated with operational recommendations.

In practice, ISO assessment requires selecting the appropriate machine group and support conditions and computing the normative indicator and mapping it to the corresponding severity zone [29]. Encapsulating this assessment as a deterministic tool on the MCP server supports procedural consistency and enables traceable reporting by making assumptions, parameters, and results explicit within the analysis artifacts.

2.4. Anomaly Detection with Semi-Supervised Models

In industrial settings, labeled fault data are often scarce since failures are relatively rare, operating conditions evolve over time, and annotation requires specialized expertise. Consequently, PdM pipelines commonly adopt anomaly detection strategies trained primarily on reference data under nominal conditions [30]. In such approaches, the model learns a representation of normal behavior and flags observations that deviate significantly from this reference.

In the proposed server, anomalies are not estimated directly on raw waveforms but on a features extracted from time segments. A feature matrix $X \in \mathbb{R}^{M \times p}$ is constructed, where M is the number of segments and p is the number of indicators. To improve numerical stability and reduce redundancy, features are standardized and projected via Principal Component Analysis (PCA) [48], yielding $Z \in \mathbb{R}^{M \times q}$ with $q \ll p$ and configurable retained variance.

On this compact representation, the server includes two consolidated model families:

- One-Class SVM (OC-SVM) which learns a decision boundary enclosing the normal region in feature space with radial basis function kernel and controlling the expected fraction of outliers through parameter ν . At inference, segments outside the learned boundary are flagged as anomalous [49].
- Local Outlier Factor (LOF): estimates anomalies by comparing the local density of each point with that of its neighbors, enabling detection of local deviations even when the global distribution is multimodal or nonlinearly separable [30,50].

These methods are widely available in standard implementations, facilitating reproducibility and software integration.

In this work, the term semi-supervised is used in an operational sense. Training is primarily performed on healthy data. When fault examples exist, they can support parameter selection and consistency checks. Otherwise, the system remains a novelty detection model.

Within the MCP paradigm, the LLM does not autonomously estimate anomalies. It orchestrates a reproducible workflow by selecting reference signals, invoking feature extraction and model training, applying inference to new signals, and summarizing results in natural language anchored to the tool outputs. In this phase, the LLM primarily contributes by:

- guiding the user in selecting the baseline and parameters while avoiding arbitrary assumptions;
- interpreting numerical outputs with conservative language;
- linking novelty detection outcomes to subsequent physical verification such as FFT and envelope analysis and reporting, reducing the risk of conclusions not supported by evidence.

Sections 2.2–2.4 summarize consolidated diagnostic procedures that are widely used in industry. The contribution of this manuscript is not to introduce new fault-detection algorithms, but to implement these procedures as deterministic MCP tools to expose them within a controlled execution perimeter. Within this paradigm, the LLM interacts through natural language. The final textual synthesis is constrained to the structured outputs

and saved artifacts, improving traceability and repeatability compared to unconstrained conversational interaction.

3. System Architecture

The system architecture is shown in Figure 3 and adopts a modular design in which the MCP server constitutes the controlled execution perimeter within which an LLM can orchestrate diagnostic workflows. The file system plays the role of persistence and a traceability layer. It includes directories dedicated to signals (`data/signals/`), textual resources (`resources/`), generated reports (`reports/`), and trained models (`models/`). The server exposes these contents as MCP resources with vibration signals and machine manuals, allowing the assistant to consult data and context in a structured and inspectable way, reducing reliance on implicit descriptions or conversational memory alone.

At the computational level, the server exposes MCP tools that encapsulate the main PdM and condition monitoring operations in deterministic functions with formalized inputs and outputs:

- FFT analysis and spectral component identification;
- Envelope analysis for rolling element bearing diagnostics;
- Vibration severity assessment according to ISO 20816-3;
- Anomaly detection based on machine learning models trained on reference baselines;
- Generation of reports and persistent artifacts for auditing and reuse.

The implementation of the tools is primarily concentrated in the signal analysis module, which integrates: an FFT engine, filters and demodulation for envelope analysis, the extraction of statistical indicators in the time domain, an anomaly detection pipeline (feature extraction and models), and the generation of graphs and tables. This separation between orchestration (LLM) and computation (deterministic tools) aims to make the analyses reproducible, reduce the non-deterministic nature typical of purely textual interaction, and produce inspectable artifacts to support traceability.

In parallel, a document reader module manages documentation-based workflows, supporting the extraction of information from manuals and technical resources, such as component identifiers and operating parameters useful for diagnosis. In the current PoC, PDF parsing capabilities and integration with databases and bearing catalogs are demonstrative and require further development to ensure robustness across heterogeneous formats and to enable complete chains that lead from manuals to technical specifications, characteristic frequencies, and spectral verification reliably.

In this context, the LLM acts as an orchestrator. It selects resources, invokes tools, connects evidence, and returns a reasoned summary. However, diagnosis remains tied to the results produced by the tools and the saved artifacts, increasing auditability and control and reducing the risk of unsupported conclusions compared to an ungoverned conversational interaction [26–28].

Table 2 summarizes which capabilities are implemented in the current PoC and which ones are planned.

In summary, Section 3 introduces the system design and the current implementation scope, Section 4 specifies the available MCP resources/tools and their outputs, and Section 5 illustrates how these components are combined into reproducible workflows.

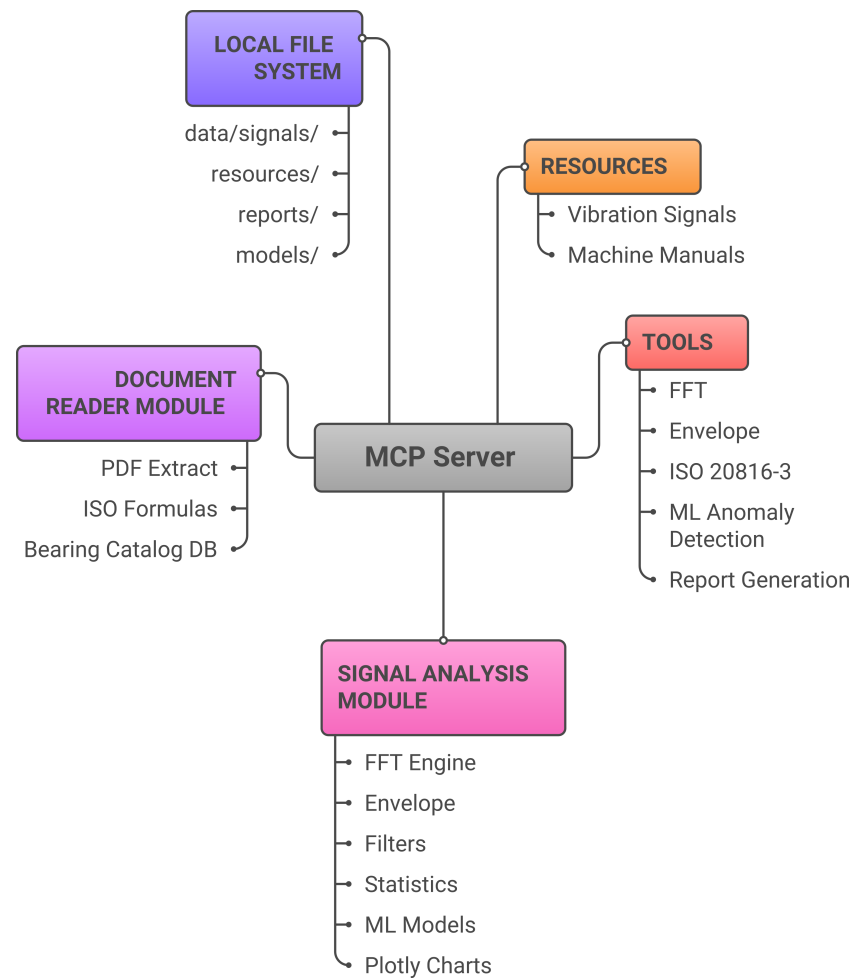


Figure 3. Architecture of the *Predictive Maintenance MCP* server.

Table 2. Summary of implemented and planned capabilities in the current PoC.

Capability	Status	Output
FFT analysis with peak identification	Implemented	HTML report, peak tables
Envelope analysis for bearings	Implemented	Envelope spectrum report
ISO 20816-3 severity assessment	Implemented	Severity zone report
Feature-based anomaly detection (OC-SVM/LOF)	Implemented	Scores, report
Manual parsing and bearing catalog integration	Prototype	Demonstrative utilities
Streaming and edge deployment and SCADA and CMMS integration	Planned	Not in current PoC

4. Implemented Capabilities

This section details the capabilities summarized in Section 3 by describing the exposed MCP resources and deterministic tools, with emphasis on how they support reproducible and auditable workflows. In the MCP paradigm, resources provide structured access to data and documents, while tools encapsulate deterministic procedures with typed inputs and outputs. This allows for guided and reproducible workflows [27,28]. The capabilities are organized into data and context access, analysis and verification and report generation.

4.1. Resources: Data and Context Access

Resources allow you to enumerate and retrieve:

- Vibration signals in common formats, with optional metadata;
- Technical documentation useful for extracting machine and component parameters to support diagnosis.

Context recovery occurs by avoiding implicit assumptions and facilitating reproducibility of analyses across sessions.

4.2. Tools

MCP tools implement server-side computations and return structured results, which can be combined in multi-step workflows. Current tools cover six families.

- **Frequency Domain Analysis.** Performs FFT with configurable options and automatically identifies dominant and harmonic components, producing peak and indicator tables useful for initial spectral characterization.
- **Envelope Analysis for Rolling Bearing Defects.** Implements band-pass filtering, envelope extraction using the Hilbert transform, and an envelope spectrum, allowing comparison between observed peaks and characteristic frequencies.
- **Vibration Severity Assessment.** Calculates the broadband normative indicator given by the RMS of the speed and determines the severity zone, making explicit the required assumptions such as the machine group and installation and support conditions [29].
- **Anomaly Detection.** Builds features through sliding windows, applies standardization and PCA, and trains classical models such as OC-SVM and LOF on healthy baselines. The output includes scores and anomaly labels by segment and aggregates that can be used as a screening module.
- **Documentation Utility.** Supports the extraction of identifiers and parameters from manuals and the calculation of characteristic defect frequencies from catalog data, which can be used as a reference for spectral interpretation (e.g., overlay on the envelope spectrum).
- **Report Generation.** For the main analyses, tools are available that generate interactive HTML reports (graphs, peak tables, markers) with execution metadata. The reports are designed as inspectable and reusable technical output, complementary to the textual summary produced by the assistant.

The LLM does not perform diagnostic calculations but composes workflows by selecting resources and invoking tools with explicit parameters, and produces a conservative summary constrained by the structured outputs of the tools [27,28].

5. Use Cases and Workflows

This section describes end-to-end workflows that can be executed with the MCP server, derived from the operational examples in the repository (file `EXAMPLES.md`) and therefore reproducible by the [31] community. The use cases are presented as chains of steps consisting of the selection of a resource (signal or document), invocation of one or more deterministic tools, and production of persistent artifacts (reports and metadata) that document parameters and results. In these examples, the LLM acts as the coordinator of the interaction and composition of the steps, while the diagnostic evidence remains anchored to the structured outputs of the tools and the generated reports. The aim of this work is not to describe experimentation relating to a specific dataset; therefore, reference is made to commonly used data available among online resources for verification and reuse.

The numerical outputs shown in the use cases are produced by conventional signal-processing methods. Compared to running the same analyses as stand-alone scripts, the MCP-based workflow provides explicit parameterization for each step by using natural language.

5.1. Demonstration Data and Provenance

The examples use the signals included in the repository as a demonstration dataset. These signals are derived from the public dataset *RollingElementBearingFaultDiagnosis-Data*

made available by MathWorks® on GitHub [51], which contains experimental vibration measurements under healthy conditions and in the presence of internal or external raceway defects. In the PoC, the signals are accompanied by metadata, including the sampling rate and operating conditions used by the tools to parameterize the analyses and make the results reproducible.

5.2. Use Case 1: FFT Analysis for Spectral Baseline

The objective is to build an initial baseline of the frequency content of a signal under healthy conditions. The workflow is based on: selecting the signal and reading metadata, including the sampling rate, duration, and rotation speed when available; invoking the `analyze_fft` tool; and generating the HTML report using `generate_fft_report`. An example of a conversational request is:

```
“Generate FFT report for real_train/baseline_1.csv”
```

The report explicitly specifies the frequency resolution, main peaks and harmonics, as well as the analysis parameters. This facilitates comparisons between acquisitions and the traceability of assumptions.

In a stand-alone workflow, FFT analysis typically requires manual scripting. In the proposed MCP-based workflow, the same analysis becomes a deterministic tool invocation by means of natural language.

5.3. Use Case 2: Envelope Analysis for BPFI and BPFO Identification

The objective is to highlight typical modulations of bearing defects using envelope demodulation. The workflow is based on: band-pass filtering on a resonant band; envelope extraction via Hilbert transform; envelope spectrum calculation; and comparison between observed peaks and expected characteristic frequencies. An example of a conversational request is the following. The output is reported in Figure 4.

```
“Generate envelope report for real_train/InnerRaceFault_vload_1.csv”
```

Parameters such as the filter bandwidth, use of random segments, and availability of characteristic frequencies are explicit in the tool inputs. The LLM guides the choice of parameters and formulates the synthesis exclusively based on the outputs produced.

Envelope analysis is sensitive to parameter choices. Here, the MCP makes these choices explicit in the tool inputs and stores them alongside results in the generated report, reducing hidden assumptions and enabling engineers to verify that defect-frequency markers and peaks are consistent with the selected configuration.

5.4. Use Case 3: Evaluation of Vibration Severity According to ISO 20816-3

The objective is to obtain a reproducible estimate of vibration severity consistent with ISO 20816-3. The LLM workflow operates with signal selection, the calculation of the required normative indicator, and the determination of the severity zone and classification thresholds. An example of a conversational request is the following.

Example of a conversational request.

```
“Evaluate real_train/baseline_1.csv against ISO 20816-3”
```

The tool returns the indicator, thresholds, and severity zone (Figure 5). The HTML report displays the position of the measurement relative to the limits, making the parameters and assumptions used transparent.



Figure 4. Example of an envelope analysis report: the peaks are consistent with the BPFI and related harmonics, highlighted by markers in the spectrum.

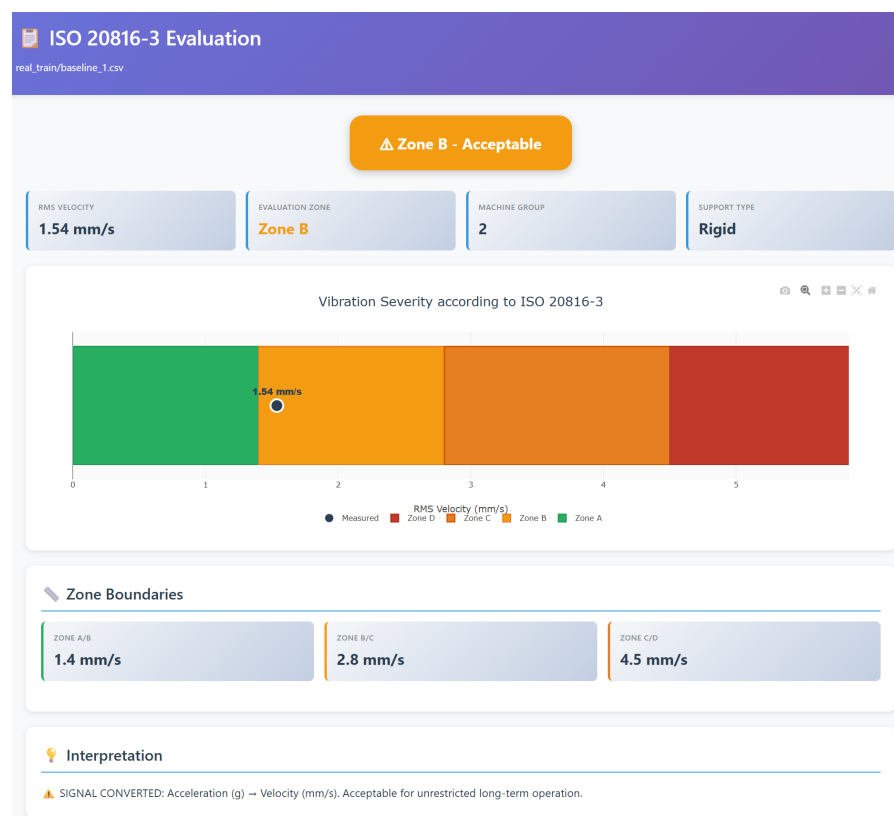


Figure 5. Example of an ISO 20816-3 report with severity zones and calculated indicator.

ISO 20816-3 assessment requires explicit assumptions. Encapsulating the procedure as a deterministic tool forces these assumptions to be stated and recorded, and the resulting report provides an auditable severity classification that can be referenced in maintenance documentation.

5.5. Use Case 4: Complete Diagnosis via Guided Workflow

The objective is to build a multi-indicator workflow to obtain a diagnostic synthesis based on consistent evidence. The LLM workflow is a combination of ISO assessment for global severity, FFT for spectral baseline and dominant components, envelope analysis to highlight characteristic modulations, and final synthesis that integrates the results of the tools and recalls the generated reports as evidence. This use case highlights LLMs as a glue between deterministic tools, while the conclusion is formulated only after collecting multiple and consistent indicators.

5.6. Use Case 5: Anomaly Detection on Features in the Time Domain

The objective is to build a semi-supervised quantitative screening module, trained on healthy baselines and applied to potentially degraded signals. The LLM workflow is based on: feature extraction on sliding windows; standardization and PCA; the training of OC-SVM or LOF on healthy baselines; inference on new signals; the calculation of segment and aggregate labels; and the possible generation of comparative reports. The output can be used to prioritize which acquisitions to subject to physical verification, avoiding premature conclusions.

5.7. Use Case 6: Reading Manuals and Calculating Characteristic Frequencies

The objective is to enable documentation-based workflows in which information extracted from manuals and catalogs supports spectral diagnosis. The LLM workflow is based on: extracting bearing information from documents, such as the identification number, geometry, or parameters, and rotational speed; calculating defect characteristic frequencies; and using these frequencies as a reference for reading spectra. This use case illustrates the MCP advantage of integrating access to resources and extraction/calculation tools within a controlled and traceable perimeter.

The numerical results shown in the use cases are produced by established signal-processing procedures. The goal of this PoC is to demonstrate how these procedures can be reliably executed and documented through a natural language assistant under explicit governance constraints. In particular, the effectiveness of the proposed software should be evaluated at the workflow level, i.e., whether the assistant's conclusions can be traced back to computed evidence and whether the full analytical chain can be reproduced with the same inputs and parameters.

6. Discussion

The proposed system is a PoC whose evaluation is not oriented towards a performance comparison of the diagnosis, but rather towards workflow properties relevant to the adoption of agents in industrial contexts. The assessments include: the verifiability of the conclusions with respect to the tool outputs; the reproducibility of the analyses with the same inputs and parameters; and auditability, understood as the availability of persistent artifacts documenting parameters, transformations, and results. This framework is consistent with the MCP's objective of constraining the LLM's action to explicit resources and deterministic tools [27,28], and with the governance needs typical of agentic systems [26].

In the workflows presented (Section 5), the server enforces a clear separation between orchestration (LLM) and computation (server-side tools). The LLM selects resources, configures parameters, and executes tool chains, while diagnostic evidence is produced

by deterministic functions and preserved in interactive HTML reports with metadata. This reduces the ambiguity typical of a text-only response and allows for post hoc review. Dominant frequencies, peaks, markers, normative assumptions, and computed indicators can be inspected independently of conversational interaction. However, the results must be interpreted in light of intrinsic PoC limitations and security considerations. In particular:

- The demonstration package is limited and does not represent typical plant heterogeneity. Therefore, the workflows should be considered as reproducible examples and not as field-validated procedures.
- Analyses such as envelope analysis, peak picking, and anomaly detection depend on parameterized choices. In the PoC, these choices are explicit and traceable, but not yet standardized for machine families and failure modes.
- Extraction from manuals and catalogs, and the automatic derivation of characteristic frequencies, requires consolidation to avoid propagation of errors in spectral readings.
- The current architecture is oriented towards offline analysis and report generation. Extensions to streaming, real-time constraints, edge deployment, and integration with SCADA and CMMS require additional components.
- The use of LLMs in maintenance contexts requires caution because the outputs can influence operational decisions with impacts on people, assets, and production continuity. The system assumes that the LLM is solely for operator support. It coordinates deterministic tools and synthesizes computed results, while decision-making remains human [32]. In real-world deployments, additional checks and mitigations against prompt injection or untrusted input are necessary. Furthermore, ISO functions support computation and classification but are not a substitute for the correct selection of the machine class and measurement conditions [29].

The open-source nature and MCP approach foster incremental development. Adding new resources and tools can extend capabilities without altering the orchestration paradigm [27,28]. Priorities include expanding and diversifying datasets, validating them on real-world industrial cases, extending them to additional faults and standards, improving the test suite, and defining evaluation protocols for PdM agents that include not only diagnostic metrics, but also traceability, robustness, and safety requirements [26]. A quantitative comparison against commercial condition monitoring platforms is outside the scope of this PoC, mainly due to differences in proprietary implementations and evaluation protocols. However, since the implemented analyses rely on consolidated procedures, outputs can be cross-validated with standard engineering tool chains. Benchmarking against selected commercial platforms is considered a relevant direction for future work once evaluation datasets and protocols are consolidated.

The proposed PoC is evaluated through workflow properties rather than fault-classification accuracy. Repeatability is ensured by design because the analytical chain is executed through deterministic and typed server-side tools within the MCP perimeter. However, client-side variations should be investigated. Traceability and auditability are supported by persistent artifacts produced at each tool invocation. Reports and associated metadata explicitly capture tool identity and input resource identifiers, enabling post hoc inspection and reproduction of the analytical chain. Future work will extend this evaluation with repeated-run statistical analyses across environments to quantify variability, and to explicitly characterize potential sources of non-determinism when optional randomness is enabled or when model-training pipelines are extended.

LLM-assisted PdM workflows are exposed to security and governance risks whenever the assistant consumes untrusted inputs and is connected to operational tools. In such settings, text sources may contain malicious instructions aimed at overriding the assistant's policies by means of prompt injection, inducing tool misuse, or eliciting disclosure of

sensitive information. Moreover, even benign content may cause unsafe recommendations if the assistant is allowed to act without grounding to verifiable evidence. The proposed PoC mitigates these risks by design through a governed perimeter enabled by the MCP. Only an explicit allowlist of deterministic and typed tools is exposed, resources can be configured as read-only and are treated as data rather than trusted instructions, tool calls are constrained by strict schemas, and parameter validation and the narrative output are grounded to computed results and persisted artifacts rather than to unverified textual content.

7. Conclusions

Predictive Maintenance MCP was presented as an open-source Proof of Concept that connects LLM-based interaction to condition monitoring and predictive maintenance workflows through an MCP perimeter of deterministic resources and tools. Reproducible diagnostic workflows were enabled by tool-based analyses, including FFT, envelope analysis for rolling bearings, ISO 20816-3 severity assessment, and semi-supervised anomaly detection on time-domain features. Interactive HTML reports and execution metadata were generated to improve the traceability and repeatability of diagnostic statements by anchoring the final synthesis to computed evidence. The software was archived on Zenodo (<https://doi.org/10.5281/zenodo.17611542>) and released as an executable baseline to support community-driven validation and extension toward industrial requirements. Future work is expected to include validation on heterogeneous field datasets and operating conditions, extension to additional faults, sensors and standards, support for streaming and edge deployment, improved robustness and test coverage, and the definition of evaluation protocols that explicitly incorporate governance and chain of evidence constraints.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in GitHub at <https://github.com/mathworks/RollingElementBearingFaultDiagnosis-Data> (accessed on 12 March 2026).

Conflicts of Interest: The author declares no conflict of interest.

References

1. Ali, M.I.; Lai, N.S.; Abdulla, R. Predictive maintenance of rotational machinery using deep learning. *Int. J. Electr. Comput. Eng. (IJECE)* **2024**, *14*, 1112. [\[CrossRef\]](#)
2. Zhou, J.; Yang, J.; Qian, Q.; Qin, Y. A comprehensive survey of machine remaining useful life prediction approaches based on pattern recognition: Taxonomy and challenges. *Meas. Sci. Technol.* **2024**, *35*, 062001. [\[CrossRef\]](#)
3. Randall, R.B. *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2011. [\[CrossRef\]](#)
4. Mikołajewska, E.; Mikołajewski, D.; Mikołajczyk, T.; Paczkowski, T. Generative AI in AI-Based Digital Twins for Fault Diagnosis for Predictive Maintenance in Industry 4.0/5.0. *Appl. Sci.* **2025**, *15*, 3166. [\[CrossRef\]](#)
5. Ma, S.; Flanigan, K.A.; Bergés, M. State-of-the-Art Review: The Use of Digital Twins to Support Artificial Intelligence-Guided Predictive Maintenance. *arXiv* **2024**. [\[CrossRef\]](#)
6. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process.* **2020**, *138*, 106587. [\[CrossRef\]](#)
7. Surucu, O.; Gadsden, S.A.; Yawney, J. Condition Monitoring using Machine Learning: A Review of Theory, Applications, and Recent Advances. *Expert Syst. Appl.* **2023**, *221*, 119738. [\[CrossRef\]](#)
8. Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech. Syst. Signal Process.* **2018**, *104*, 799–834. [\[CrossRef\]](#)
9. Hendriks, J.; Dumond, P.; Knox, D. Towards better benchmarking using the CWRU bearing fault dataset. *Mech. Syst. Signal Process.* **2022**, *169*, 108732. [\[CrossRef\]](#)
10. Farooq, U.; Ademola, M.; Shaalan, A. Comparative Analysis of Machine Learning Models for Predictive Maintenance of Ball Bearing Systems. *Electronics* **2024**, *13*, 438. [\[CrossRef\]](#)

11. Giraudo, L.; Di Maggio, L.G.; Giorio, L.; Delprete, C. Dynamic Multibody Modeling of Spherical Roller Bearings with Localized Defects for Large-Scale Rotating Machinery. *Sensors* **2025**, *25*, 2419. [\[CrossRef\]](#)
12. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Di Maggio, L.G. Intelligent Fault Diagnosis of Industrial Bearings Using Transfer Learning and CNNs Pre-Trained for Audio Classification. *Sensors* **2023**, *23*, 211. [\[CrossRef\]](#)
14. Eang, C.; Lee, S. Predictive Maintenance and Fault Detection for Motor Drive Control Systems in Industrial Robots Using CNN-RNN-Based Observers. *Sensors* **2024**, *25*, 25. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Yang, X.; Fang, C.; Liao, Y.; Yang, J.; Gryllias, K.; Chronopoulos, D. Deep Generative Models in Condition and Structural Health Monitoring: Opportunities, Limitations and Future Outlook. *arXiv* **2025**. [\[CrossRef\]](#)
16. DI MAGGIO, L.G.; Brusa, E.; Delprete, C. Zero-Shot Generative AI for Rotating Machinery Fault Diagnosis: Synthesizing Highly Realistic Training Data via Cycle-Consistent Adversarial Networks. *Appl. Sci.* **2023**, *13*, 12458. [\[CrossRef\]](#)
17. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165. [\[CrossRef\]](#)
18. OpenAI. GPT-4 Technical Report. *arXiv* **2023**. [\[CrossRef\]](#)
19. Wu, J.; Gan, W.; Chen, Z.; Wan, S.; Yu, P.S. Multimodal Large Language Models: A Survey. *arXiv* **2023**, arXiv:2311.13165. [\[CrossRef\]](#)
20. Lynch, K.; Lorenzi, F.; Sheehan, J.; Kabakci-Zorlu, D.; Eck, B. FMEA Builder: Expert Guided Text Generation for Equipment Maintenance. *arXiv* **2024**, arXiv:2411.05054. [\[CrossRef\]](#)
21. Yigin, B.; Celik, M. A Prescriptive Model for Failure Analysis in Ship Machinery Monitoring Using Generative Adversarial Networks. *J. Mar. Sci. Eng.* **2024**, *12*, 493. [\[CrossRef\]](#)
22. Reddicharla, N.; Ali, M.S. Innovating Oil and Gas Field Operations - Harnessing the Power of Generative Ai for Supporting Workforce Towards Achieving Autonomous Operations. In Proceedings of the ADIPEC, Abu Dhabi, United Arab Emirates, 4–7 November 2024; p. D011S020R005. [\[CrossRef\]](#)
23. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Proceedings of the Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 9459–9474.
24. Moran, G.E.; Aragam, B. Towards Interpretable Deep Generative Models via Causal Representation Learning. *arXiv* **2025**. [\[CrossRef\]](#)
25. Nuti, L.; Veneri, G.; Porciani, D.; Catalano, L.; Iasparro, D.; Lauriola, M.; Capanni, M.; Vitale, M. Agentic AI on Customer Dashboards in Remote Monitoring and Diagnostic Services. In Proceedings of the ADIPEC, Abu Dhabi, United Arab Emirates, 3–6 November 2025; p. D021S080R004. [\[CrossRef\]](#)
26. Kolt, N. Governing AI Agents. *arXiv* **2025**, arXiv:2501.07913. [\[CrossRef\]](#)
27. Hou, X.; Zhao, Y.; Wang, S.; Wang, H. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. *arXiv* **2025**. [\[CrossRef\]](#)
28. Ray, P.P. A Survey on Model Context Protocol: Architecture, State-of-the-art, Challenges and Future Directions. *TechRxiv* **2025**. [\[CrossRef\]](#)
29. ISO 20816-3:2022(en); Mechanical vibration—Measurement and Evaluation of Machine Vibration Part 3: Industrial Machinery with a Power Rating Above 15 kW and Operating Speeds Between 120 r/min and 30 000 r/min. ISO: London, UK, 2022.
30. Di Maggio, L.G.; Brusa, E.; Delprete, C. Novelty Detection in Rotating Machinery: Assessment of Unsupervised Machine Learning Models for Medium-Sized Industrial Bearings. In Proceedings of the 2025 International Conference on Control, Automation and Diagnosis (ICCAD), Barcelona, Spain, 1–3 July 2025; pp. 1–7. [\[CrossRef\]](#)
31. Di Maggio, L.G. *Predictive Maintenance MCP Server: An Open-Source Framework for Integrating Large Language Models with Predictive Maintenance and Fault Diagnosis Workflows*; Zenodo: Geneva, Switzerland, 2025. [\[CrossRef\]](#)
32. Di Maggio, L.G. Toward Autonomous LLM-Based AI Agents for Predictive Maintenance: State of the Art, Challenges, and Future Perspectives. *Appl. Sci.* **2025**, *15*, 11515. [\[CrossRef\]](#)
33. Deng, H.; Namono, B.; Zheng, B.; Khan, S.; Ahmet Erkoyuncu, J. From Prediction to Prescription: Large Language Model Agent for Context-Aware Maintenance Decision Support. *PHM Soc. Eur. Conf.* **2024**, *8*, 10. [\[CrossRef\]](#)
34. Wang, Y.; Yu, Y.; Sun, K.; Lei, P.; Zhang, Y.; Zio, E.; Xia, A.; Li, Y. RmGPT: A Foundation Model With Generative Pre-trained Transformer for Fault Diagnosis and Prognosis in Rotating Machinery. *IEEE Internet Things J.* **2025**, *12*, 41562–41573. [\[CrossRef\]](#)
35. Tao, L.; Liu, H.; Ning, G.; Cao, W.; Huang, B.; Lu, C. LLM-based Framework for Bearing Fault Diagnosis. *arXiv* **2024**. [\[CrossRef\]](#)
36. Qaid, H.A.A.M.; Zhang, B.; Li, D.; Ng, S.K.; Li, W. FD-LLM: Large Language Model for Fault Diagnosis of Machines. *arXiv* **2024**. [\[CrossRef\]](#)

37. He, H.; Huang, J.; Li, Q.; Wang, X.; Zhang, F.; Yang, K.; Meng, L.; Chu, F. MaintAGT:Sim2Real-Guided Multimodal Large Model for Intelligent Maintenance with Chain-of-Thought Reasoning. *arXiv* **2024**. [CrossRef]
38. Sun, Y.; Zhang, Q.; Bao, J.; Lu, Y.; Liu, S. Empowering digital twins with large language models for global temporal feature learning. *J. Manuf. Syst.* **2024**, *74*, 83–99. [CrossRef]
39. Zhang, X.; Sun, W.; Chen, K.; Jiang, R. A multimodal expert system for the intelligent monitoring and maintenance of transformers enhanced by multimodal language large model fine-tuning and digital twins. *IET Collab. Intell. Manuf.* **2024**, *6*, e70007. [CrossRef]
40. Ferdousi, R.; Hossain, M.A.; Yang, C.; Saddik, A.E. DefectTwin: When LLM Meets Digital Twin for Railway Defect Inspection. *arXiv* **2024**. [CrossRef]
41. Kok, I.; Demirci, O.; Ozdemir, S. When IoT Meet LLMs: Applications and Challenges. *arXiv* **2024**. [CrossRef]
42. Mahr, F.; Angeli, G.; Sindel, T.; Schmidt, K.; Franke, J. A Reference Architecture for Deploying Large Language Model Applications in Industrial Environments. In Proceedings of the 2024 IEEE 30th International Symposium for Design and Technology in Electronic Packaging (SIITME), Sibiu, Romania, 16–18 October 2024; pp. 19–23. [CrossRef]
43. Friha, O.; Amine Ferrag, M.; Kantarci, B.; Cakmak, B.; Ozgun, A.; Ghoualmi-Zine, N. LLM-Based Edge Intelligence: A Comprehensive Survey on Architectures, Applications, Security and Trustworthiness. *IEEE Open J. Commun. Soc.* **2024**, *5*, 5799–5856. [CrossRef]
44. Dashdamirli, N. Integration of Large Language Models with microcontrollers to optimize industrial automation processes. *InterConf* **2024**, *46*, 500–507. [CrossRef]
45. Smith, W.A.; Randall, R.B. Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mech. Syst. Signal Process.* **2015**, *64–65*, 100–131. [CrossRef]
46. Brusa, E.; Bruzzone, F.; Delprete, C.; Di Maggio, L.G.; Rosso, C. Health indicators construction for damage level assessment in bearing diagnostics: A proposal of an energetic approach based on envelope analysis. *Appl. Sci.* **2020**, *10*, 8131. [CrossRef]
47. Abboud, D.; Antoni, J.; Sieg-Zieba, S.; Eltabach, M. Envelope analysis of rotating machine vibrations in variable speed conditions: A comprehensive treatment. *Mech. Syst. Signal Process.* **2017**, *84*, 200–226. [CrossRef]
48. Brusa, E.; Delprete, C.; DI MAGGIO, L.G. Eigen-spectrograms: An interpretable feature space for bearing fault diagnosis based on artificial intelligence and image processing. *Mech. Adv. Mater. Struct.* **2022**, *33*, 4639–4651. [CrossRef]
49. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
50. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the Proceedings of the 2000 ACM SIGMOD international conference on Management of data, Dallas TX, USA, 16–18 May 2000; pp. 93–104. [CrossRef]
51. The Mathworks Inc. Rolling Element Bearing Fault Diagnosis Dataset. GitHub Repository. 2023. Available online: <https://github.com/mathworks/RollingElementBearingFaultDiagnosis-Data> (accessed on 28 January 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.