



UNIVERSIDAD NACIONAL DE COLOMBIA

Departamento de Ingeniería Mecánica y Mecatrónica

Desarrollo de un algoritmo de machine learning en dispositivo IoT para aplicación industrial de análisis de vibraciones

por
María Alejandra Pérez Petro

Supervisor: Victor Hugo Grisales
2025

Resumen

Este trabajo presenta el diseño, implementación y evaluación de un sistema embebido basado en inteligencia artificial para la detección temprana de fallas en maquinaria rotativa mediante el análisis de señales de vibración. La solución propuesta integra un pipeline de aprendizaje automático optimizado para despliegue en dispositivos IoT con arquitectura ARM64, específicamente en un módulo System-on-Module (SoM) VAR-SOM-MX8M-PLUS de Variscite.

El sistema emplea datos de vibración capturados en condiciones de laboratorio mediante acelerómetros triaxiales, procesados mediante extracción de características en los dominios temporal y espectral. Se implementó una estrategia de entrenamiento especializada con 28 modelos independientes, optimizados para diferentes configuraciones de ejes (axial, radial, tangencial y combinación de 3 ejes) y frecuencias de muestreo (desde 50 kHz hasta 781 Hz). Los modelos fueron evaluados utilizando métricas robustas como Accuracy, F1 Macro, Recall Macro y Precision Macro, demostrando un rendimiento superior en la configuración de 3 ejes con un Accuracy promedio de 0.9888 ± 0.0021 .

Para el despliegue embebido, se desarrolló una arquitectura modular basada en contenedores Docker, utilizando FastAPI como backend y SQLite como sistema de gestión de datos ligero. No obstante, la implementación final se vio limitada por restricciones de almacenamiento en la memoria flash del dispositivo, quedando pendiente la reconfiguración de particiones para su operación completa.

Los resultados obtenidos, aunque prometedores en entorno controlado, requieren validación en condiciones operacionales reales para confirmar su eficacia. El sistema muestra potencial como solución portátil y eficiente para mantenimiento predictivo en entornos industriales, destacando por su bajo consumo de recursos y capacidad de adaptación a diferentes condiciones de muestreo.

Palabras clave: diagnóstico de fallas, aprendizaje automático embebido, análisis de vibraciones, mantenimiento predictivo, Random Forest.

Índice general

Resumen	1
Índice de figuras	4
Índice de cuadros	5
1. Introducción	7
1.1. Contexto y justificación	7
1.2. Objetivos	8
1.2.1. Objetivo General	8
1.2.2. Objetivos Específicos	8
2. Estado del arte	9
2.1. El Análisis de Vibraciones para la Detección de Fallas en Maquinaria Rotativa	9
2.1.1. Captura y Procesamiento de Señales	9
2.1.2. Evaluación Global: Norma ISO 10816-3	10
2.1.3. Análisis Espectral de Rodamientos	10
2.1.4. Limitaciones del Enfoque Tradicional	10
2.1.5. Configuración y Ubicación Estratégica de Sensores de Vibración	11
2.1.6. ¿Por Qué Enfocarse en Fallas de Rodamientos?	12
2.2. Técnicas de Análisis Tiempo-Frecuencia	12
2.2.1. Transformada Wavelet	12
2.2.2. Descomposición de Modo Empírico (EMD)	12
2.2.3. Descomposición de Modo Variacional (VMD)	13
2.3. La tarea de clasificación en el Machine Learning	13
2.3.1. Resultados en la clasificación	14
2.3.2. Representación en la Matriz de Confusión	14
2.3.3. Métricas para la evaluación de Modelos de Clasificación	15
2.3.4. Algoritmo Random Forest	16
3. Metodología	18
3.1. Descripción del conjunto de datos	18
3.1.1. Machinery Fault Simulator (MFS) de SpectraQuest Inc. [13]	19
3.1.2. Sistema de Adquisición de Datos [13].	20
3.1.3. Distribución de las clases en el conjunto de datos [13]	20
3.2. Extracción de características	21
3.2.1. Dominio del Tiempo	22
3.2.2. Dominio de la Frecuencia	24
3.3. Modelamiento con Machine Learning: Modelo Base	28
3.3.1. Modelo de aprendizaje automático	28

3.3.2. Selección de Hiperparámetros mediante GridSearchCV y Validación Cruzada	30
3.3.3. Evaluación del modelo	30
3.4. Modelamiento con Machine Learning: Modelos Especializados	31
3.4.1. Separación por Sensor: Underhang y Overhang	31
3.4.2. Submuestreo: Reducción de la Frecuencia de Muestreo	31
3.4.3. Separación por Ejes de Vibración	32
3.4.4. Esquema de Entrenamiento Especializado	32
3.5. Implementación en dispositivo IoT	33
3.5.1. Variscite Symphony board VAR-SOM-MX8M-PLUS V2	33
3.5.2. Arquitectura	36
3.5.3. Contenedorización y Despliegue	36
3.5.4. Capa de Usuario (Interfaz)	37
3.5.5. Capa Web / REST API	37
3.5.6. Capa de Procesamiento	37
3.5.7. Modo de uso	37
4. Resultados y Discusión	38
4.1. Modelamiento con Machine Learning: Modelo Base	38
4.1.1. Mejores hiperparámetros encontrados	38
4.1.2. Evaluación del Modelo	39
4.2. Modelamiento con Machine Learning: Modelos Especializados	40
4.2.1. Resultados por frecuencia de muestreo	40
4.2.2. Resultados por tipo de sensor	42
4.2.3. Desempeño por clase: comparación entre configuraciones triaxiales y uniaxiales	43
4.2.4. Importancia de características en la clasificación	49
4.3. Implementación en Variscite Symphony Board	51
4.3.1. Interfaz Web	51
4.3.2. Ejecución en la Symphony Board	52
4.3.3. Trabajo Futuro	52
5. Conclusión	53

Índice de figuras

3.1. SpectraQuest Machinery Fault Simulator (MFS)	19
3.2. Metodología de extracción de características	22
3.3. Ejemplo de extracción de características	27
3.4. Flujo de procesamiento y modelamiento del sistema de diagnóstico	28
3.5. Distribución de las clases en el conjunto de evaluación	31
3.6. Variscite Symphony board VAR-SOM-MX8M-PLUS V2. Tomado de [15]	34
3.7. Arquitectura implementada en la Symphony board	36
3.8. Modo de uso de la arquitectura implementada en la Symphony board	37
4.1. Matriz de confusión y métricas de evaluación obtenidas para el mejor modelo encontrado con validación cruzada. Donde 0: (normal), 1: (desalineamiento vertical), 2: (desalineamiento horizontal), 3: (desbalanceo), 4: (falla en jaula - rodamiento), 5: (falla en bola - rodamiento), 6: (falla en pista externa - rodamiento)	39
4.2. Métricas por frecuencia de muestreo	40
4.3. Desempeño de la métricas (Accuracy, F1-score, Recall macro y Precision macro) en el conjunto de evaluación según la información proporcionada por los sensores (azul - 3axis, naranja-axial, verde -radial, rojo-tangencial)	42
4.4. Desempeño de la métrica F1-score para la clase 0 - Normal	44
4.5. Desempeño de la métrica F1-score para la clase 1 - Desalinamiento vertical	45
4.6. Desempeño de la métrica F1-score para la clase 2 - Desalinamiento Horizontal	45
4.7. Desempeño de la métrica F1-score para la clase 3 - Desbalanceo	46
4.8. Desempeño de la métrica F1-score para la clase 4 - Falla en pista interior	47
4.9. Desempeño de la métrica F1-score para la clase 5 - Falla en bolas	47
4.10. Desempeño de la métrica F1-score para la clase 6 - Falla en pista exterior	48
4.11. Importancia relativa de las 24 características en los modelos de diagnóstico de fallas. Los valores representan el promedio de importancia a través de los 28 modelos evaluados (4 configuraciones de sensor \times 7 frecuencias de muestreo).	49
4.12. Interfaz web para diagnóstico de fallas.	51
4.13. Espacio de almacenamiento disponible en la partición raíz de la Symphony board.	52

Índice de cuadros

3.1.	Características generales del banco experimental [13]	19
3.2.	Especificaciones del rodamiento en el banco experimental [13]	19
3.3.	Distribución de las clases en el conjunto de datos	21
3.4.	Hiperparámetros considerados en la validación cruzada	29
4.1.	Comparación de métricas frente a distintas frecuencias de muestreo	41
4.2.	Resultados de métricas por tipo de eje (media \pm desviación estándar)	42
4.3.	Comparación de desempeño por clase: Configuraciones 3-ejes vs. uniaxiales	43

Capítulo 1

Introducción

Este capítulo presenta el contexto general del proyecto en el marco de la transformación digital, también denominada como Industria 4.0. Se expone la importancia del mantenimiento predictivo como una estrategia clave para optimizar la disponibilidad operativa y reducir los tiempos de inactividad de la maquinaria. A través de técnicas avanzadas de análisis de vibraciones y algoritmos de aprendizaje automático, se plantea un enfoque moderno para la detección temprana de fallas en maquinaria rotativa. Asimismo, en esta sección se sustentan los objetivos del trabajo y se motiva el desarrollo de un sistema embebido basado en IoT capaz de realizar diagnóstico en línea.

1.1. Contexto y justificación

La era actual se caracteriza por una transformación digital profunda en el sector industrial, conocida como Industria 4.0 o Industria Inteligente. En el marco de esta evolución, la automatización se extiende más allá de los procesos de producción para incluir la toma de decisiones, siendo el mantenimiento predictivo (PdM) una de sus áreas de aplicación más prometedoras. El mantenimiento predictivo busca maximizar la eficiencia operativa y reducir los tiempos de inactividad de la maquinaria mediante la integración de sistemas inteligentes. El PdM basado en datos, también conocido como "prognósticos basados en datos", extrae información valiosa de los sensores para identificar patrones de comportamiento a través de algoritmos de machine learning como redes neuronales artificiales, máquinas de vectores de soporte o bosques aleatorios [1].

Dentro del mantenimiento predictivo, el análisis de vibraciones se ha consolidado como una herramienta excepcional para la detección temprana de fallas en maquinaria rotativa, como motores, bombas y compresores. Las fallas en estos equipos pueden acarrear consecuencias críticas, desde daños materiales y pérdidas de producción hasta riesgos para el personal. Por ello, la capacidad de prevenir estas fallas en sus etapas iniciales es crucial. El análisis de vibraciones permite rastrear las condiciones de operación y las tendencias de deterioro, contribuyendo directamente a la reducción de costos y tiempos de inactividad [1].

Tradicionalmente, el análisis de vibraciones se ha basado en técnicas como la Transformada Rápida de Fourier (FFT) y el uso de normas como la ISO 10816-3, que permiten evaluar la severidad de las vibraciones mediante indicadores globales. Si bien estas metodologías han sido útiles como primera línea de diagnóstico, presentan limitaciones importantes: requieren interpretación experta, tienen baja capacidad para detectar fallas incipientes, y no se adaptan fácilmente a cambios en las condiciones operativas o configuraciones de maquinaria [1].

Específicamente, las mediciones de vibración en rodamientos son una fuente de datos invaluable para el mantenimiento predictivo. Los rodamientos son componentes críticos y altamente susceptibles a fallas, siendo responsables de una parte significativa de las interrupciones operativas en maquinaria rotativa. El objetivo del PdM en este contexto es monitorear continuamente la condición de los rodamientos, extrayendo características útiles de las señales de vibración y utilizando estas características para clasificar condiciones operativas o predecir la vida útil restante de los componentes [2].

A pesar de los avances impulsados por la Industria 4.0, la implementación de soluciones de mantenimiento predictivo robustas y adaptables en entornos industriales reales presenta desafíos, especialmente por la escasez de datos tomados en entornos industriales y etiquetados para fallas. En respuesta a esto, el presente proyecto se enfoca en el desarrollo y validación de un algoritmo de machine learning en un dispositivo IoT industrial. Este algoritmo está diseñado para el análisis en línea de señales de vibración, con el fin de optimizar los procesos de mantenimiento predictivo, superar las limitaciones de los métodos tradicionales y, en última instancia, contribuir a una mayor eficiencia y disponibilidad de los activos industriales.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar y validar un algoritmo de machine learning implementado en un dispositivo IoT industrial que permita la detección temprana de fallas en maquinaria rotativa mediante el análisis en tiempo real de señales de vibración.

1.2.2. Objetivos Específicos

- Investigar y sistematizar el estado del arte en análisis de vibraciones y técnicas de machine learning aplicadas al mantenimiento predictivo.
- Configurar el dispositivo IoT para la adquisición de datos de vibración en entornos industriales.
- Implementar el entrenamiento en línea del algoritmo de machine learning en el dispositivo IoT, considerando la eficiencia computacional y la gestión de datos.
- Validar experimentalmente el rendimiento del algoritmo en condiciones operativas reales.
- Comparar los resultados obtenidos con metodologías de mantenimiento predictivo tradicionales.

Capítulo 2

Estado del arte

Este capítulo tiene como objetivo presentar los fundamentos teóricos y prácticos del análisis de vibraciones como técnica principal para la detección temprana de fallas en maquinaria rotativa. Se describen los principios físicos en los que se basa esta metodología, los pasos involucrados en la recolección y procesamiento de señales, y la importancia estratégica de la ubicación de los sensores. Además, se justifica el enfoque en fallas de rodamientos como caso de estudio representativo, dada su alta incidencia en fallos operativos. Este análisis constituye el punto de partida para la posterior aplicación de técnicas de aprendizaje automático orientadas al mantenimiento predictivo.

Adicionalmente, en el capítulo se aborda los fundamentos teóricos y prácticos de la clasificación dentro del campo del aprendizaje automático. Se explican las bases conceptuales de la tarea de clasificación supervisada, su relevancia en contextos industriales como el mantenimiento predictivo, y los desafíos asociados al uso de datos etiquetados. Además, se detallan los métodos utilizados para evaluar el desempeño de los modelos clasificadores, incluyendo la matriz de confusión y métricas clave como la precisión, el recall y la puntuación F1. Finalmente, se presenta el algoritmo Random Forest como modelo base para abordar el problema de clasificación multiclase aplicado a la detección de fallas en maquinaria.

2.1. El Análisis de Vibraciones para la Detección de Fallas en Maquinaria Rotativa

El análisis de vibraciones es una técnica fundamental en el mantenimiento predictivo (PdM) que permite identificar anomalías y predecir fallas en maquinaria rotativa antes de que ocurran. Se basa en el principio de que todo equipo mecánico en operación genera patrones de vibración específicos. Cuando una máquina funciona correctamente, estas vibraciones son constantes y de baja amplitud. Sin embargo, el surgimiento de defectos o cambios en los procesos dinámicos internos de la máquina altera su patrón de vibración, generando cambios detectables en su amplitud y frecuencia [3]. Tradicionalmente, este proceso se basa en métodos estandarizados y en la interpretación de señales espectrales por parte de expertos.

2.1.1. Captura y Procesamiento de Señales

Primero, se realiza la **recolección de señales** mediante sensores como acelerómetros, que registran las vibraciones mecánicas directamente en la carcasa de la máquina o componentes de interés. Estas señales se obtienen en el dominio del tiempo y se anali-

zan posteriormente utilizando la *Transformada Rápida de Fourier* (FFT), lo cual permite transformarlas al dominio de la frecuencia [3], [4]. Esta conversión es fundamental, ya que las fallas mecánicas suelen generar frecuencias específicas fácilmente observables en el espectro. Por ejemplo, un desequilibrio tiende a producir un pico dominante en la frecuencia de rotación, mientras que defectos en rodamientos generan componentes armónicas asociadas a su geometría y velocidad de giro [5].

2.1.2. Evaluación Global: Norma ISO 10816-3

Uno de los métodos normativos más utilizados es el definido por la **ISO 10816-3 ISO10816**, el cual establece criterios para la evaluación del estado de maquinaria industrial mediante la medición de la *velocidad vibratoria RMS* (en mm/s). Esta norma clasifica la severidad de las vibraciones en cuatro zonas:

- **Zona A:** vibraciones muy bajas, funcionamiento óptimo sin restricciones.
- **Zona B:** aceptables para operación continua, pero con necesidad de vigilancia.
- **Zona C:** nivel de vibraciones que puede afectar el funcionamiento; se recomienda intervención.
- **Zona D:** vibraciones excesivas; el funcionamiento continuo no es admisible.

Este tipo de análisis es simple y efectivo para una evaluación general, pero no permite identificar la causa exacta ni la localización de la falla.

2.1.3. Análisis Espectral de Rodamientos

Una técnica más específica, complementaria a la evaluación global, es el **análisis de rodamientos**, que se basa en la detección de frecuencias características de falla derivadas de las dimensiones del rodamiento y su velocidad de operación. Estas incluyen:

- **BPFI** (Ball Pass Frequency Inner): paso del elemento rodante por la pista interna.
- **BPFO** (Ball Pass Frequency Outer): paso del rodante por la pista externa.
- **BSF** (Ball Spin Frequency): frecuencia de giro de los elementos rodantes.
- **FTF** (Fundamental Train Frequency): frecuencia de rotación de la jaula.

Estas frecuencias generan picos específicos en el espectro que, si se conocen las características del rodamiento, permiten diagnosticar con precisión el tipo de falla. Sin embargo, este análisis requiere experiencia, acceso a las especificaciones del rodamiento y condiciones de operación estables [2], [6].

2.1.4. Limitaciones del Enfoque Tradicional

Aunque estas metodologías han sido utilizadas exitosamente en la industria durante décadas, presentan importantes limitaciones:

- Requieren juicio experto para interpretar resultados y diferenciar entre tipos de fallas.
- Su capacidad para detectar fallas incipientes o bajo condiciones transitorias es limitada.

- No están diseñadas para análisis continuo o automatizado.
- Presentan baja adaptabilidad a diferentes configuraciones de máquinas o condiciones cambiantes.

En este contexto, el uso de técnicas de **inteligencia artificial** como el aprendizaje automático permite superar varias de las limitaciones mencionadas. Modelos entrenados con datos etiquetados pueden aprender a reconocer patrones complejos de vibración sin intervención humana directa, mejorar la detección temprana de fallas, adaptarse a múltiples configuraciones de sensores y operar en tiempo real. Si bien los métodos tradicionales como la ISO 10816-3 y el análisis de rodamientos siguen siendo herramientas útiles de referencia y verificación, su integración con técnicas modernas representa un avance significativo hacia un mantenimiento predictivo más autónomo, preciso y escalable.

2.1.5. Configuración y Ubicación Estratégica de Sensores de Vibración

La efectividad del análisis de vibraciones depende en gran medida de la ubicación y orientación adecuadas de los sensores. Los acelerómetros, que son los sensores más comunes para estas mediciones, se instalan directamente sobre la estructura de la máquina, cerca de los componentes críticos o puntos de interés para el monitoreo.

La configuración de los sensores se determina considerando el tamaño de la máquina y el tipo de fallas que se desean detectar:

- **Dirección Radial:** Generalmente, los sensores se montan en dirección radial, es decir, perpendicular al eje de rotación del equipo. Dentro de esta dirección, pueden ubicarse en orientación horizontal o vertical. Muchas fallas de desequilibrio o desalineación se manifiestan prominentemente en estas direcciones [4].
- **Dirección Axial:** En ciertas situaciones, los sensores también se instalan en dirección axial, paralela al eje de rotación. Algunas fallas, como el desalineamiento angular o problemas con el empuje, pueden generar vibraciones más significativas en esta dirección [4].

Para máquinas con una potencia de hasta 375 kW (500 HP), suele ser suficiente con un solo sensor radial por rodamiento, preferentemente montado en dirección horizontal. Para equipos de mayor potencia (más de 375 kW), la práctica recomendada es utilizar dos sensores radiales por rodamiento (uno horizontal y otro vertical), además de al menos un sensor axial. Cuando se utilizan dos sensores radiales, uno debe ubicarse lo más cercano posible a la vertical y el otro a la horizontal. Si solo se emplea un sensor, la orientación horizontal es la opción predeterminada, a menos que la estructura sea notablemente menos rígida en dirección vertical, en cuyo caso esta última se prioriza. Es fundamental mantener la consistencia en la orientación de los sensores a lo largo de toda la máquina para asegurar la comparabilidad de los datos [4].

Es importante considerar la influencia de los componentes acoplados. En máquinas pequeñas o medianas acopladas directamente, la vibración del equipo impulsado (bombas, ventiladores, etc.) puede transmitirse al motor. Por lo tanto, al configurar las mediciones en el motor, es necesario considerar las frecuencias de paso de los álabes o palas del componente impulsado. En máquinas grandes, la transmisión de vibración del equipo acoplado es menos significativa, y las frecuencias de falla del componente conectado no suelen requerir una consideración específica en las mediciones del motor [4].

2.1.6. ¿Por Qué Enfocarse en Fallas de Rodamientos?

Los rodamientos son componentes críticos y altamente propensos a fallas en la maquinaria rotativa. Un rodamiento es un elemento mecánico esencial que facilita el movimiento suave entre dos partes, como un eje y su alojamiento, mientras soporta cargas con mínima fricción [2].

La degradación de los rodamientos es una de las principales causas de interrupciones inesperadas y costosas en los sistemas rotativos, lo que subraya la importancia de su monitoreo constante. Estadísticamente, se ha demostrado que las fallas en los rodamientos son responsables de aproximadamente el 45-55 % de las paradas operativas en maquinaria rotativa, una cifra significativamente mayor en comparación con el 41 % por fallas en motores o el 10 % por fallas en rotores [2].

Debido a su alta tasa de falla y su impacto directo en la disponibilidad operativa, los rodamientos han sido un foco primordial de investigación en el campo del pronóstico de fallas en maquinaria. Esto se traduce en una mayor cantidad de datos y metodologías desarrolladas específicamente para la detección y diagnóstico de problemas en estos componentes, lo que los convierte en un punto de partida base para un modelo de machine learning basado en vibraciones.

2.2. Técnicas de Análisis Tiempo-Frecuencia

En el contexto de aprendizaje automático para diagnóstico de fallas, las características extraídas de las señales juegan un papel fundamental. Es posible emplear tanto las características tradicionales del análisis de vibraciones en el dominio del tiempo y de la frecuencia, como también enfoques más avanzados que consideran simultáneamente la evolución temporal y frecuencial de la señal.

Las técnicas descritas en esta sección corresponden al estado del arte en procesamiento de señales de vibración en el dominio tiempo-frecuencia, y constituyen una base teórica importante para futuras mejoras al sistema. No fueron implementadas en el presente trabajo, pero se incluyen como marco de referencia.

2.2.1. Transformada Wavelet

La **transformada wavelet** es una herramienta poderosa para analizar señales no estacionarias, ya que combina información temporal y espectral mediante funciones base localizadas (wavelets). A diferencia de la transformada de Fourier, que usa sinusoides infinitas, las wavelets son oscilaciones de duración finita, lo que permite identificar eventos transitorios y cambios de frecuencia en el tiempo [7], [2], [6].

- **Aplicación potencial:** Podría emplearse para extraer características en señales de vibración con resolución adaptable (alta frecuencia → mayor precisión temporal; baja frecuencia → mayor precisión espectral).
- **Variantes relevantes:** Wavelets como *Daubechies* y *Morlet* serían útiles para detectar fallas incipientes en rodamientos.

2.2.2. Descomposición de Modo Empírico (EMD)

La **EMD** descompone señales complejas en componentes llamados *Funciones de Modo Intrínseco (IMFs)*, cada una con una escala de frecuencia distinta. Es especialmente útil en entornos con bajo SNR (relación señal-ruido) [7], [2], [6].

■ **Ventajas teóricas:**

- Adaptabilidad a señales no lineales y no estacionarias
- Capacidad de separar ruido de componentes relevantes

■ **Limitaciones:** Sensibilidad a artefactos en bordes de la señal (*edge effects*)

■ **Variantes mejoradas:** Ensemble EMD (EEMD) para mitigar la mezcla de modos

2.2.3. Descomposición de Modo Variacional (VMD)

La **VMD** supera limitaciones de EMD mediante un enfoque matemático robusto basado en optimización.

■ **Ventajas comparativas:**

- Menor sensibilidad al ruido
- Eliminación del problema de mezcla de modos
- Mayor eficiencia computacional

■ **Aplicación potencial:** Aislamiento preciso de componentes de fallas en rodamientos

2.3. La tarea de clasificación en el Machine Learning

El aprendizaje automático (*Machine Learning*) es una subdisciplina de la inteligencia artificial que se centra en el desarrollo de algoritmos capaces de aprender patrones a partir de datos, sin ser explícitamente programados para tareas específicas. Dentro de este campo, la clasificación es una de las tareas fundamentales y consiste en asignar una etiqueta o clase a cada instancia de entrada, con base en las características observadas [8] [7], [2], [6].

En un problema de clasificación supervisada, el objetivo es construir un modelo que aprenda a partir de un conjunto de datos etiquetado, es decir, un conjunto de instancias para las cuales se conoce la clase correcta. Una vez entrenado, este modelo se utiliza para predecir la clase de nuevas instancias no vistas. En contraste, en el aprendizaje no supervisado, los datos no poseen etiquetas predefinidas, y el objetivo principal es descubrir estructuras ocultas o patrones, como grupos o clústeres, dentro del conjunto de datos [9].

Los métodos clásicos para el mantenimiento predictivo suelen emplear técnicas de aprendizaje supervisado para entrenar un clasificador sobre las características extraídas [10]. Sin embargo, estos métodos pueden verse limitados por el requisito de grandes cantidades de datos etiquetados para lograr un rendimiento óptimo. En muchas situaciones industriales reales, los datos etiquetados pueden ser de tamaño limitado, lo que dificulta el entrenamiento de modelos predictivos precisos utilizando solo el aprendizaje supervisado. Además, puede ser un desafío obtener datos etiquetados que puedan modelar eficazmente todas las posibles clases o condiciones operativas, particularmente para condiciones anormales, que pueden ser raras o impredecibles. Por ejemplo, si bien es posible obtener suficientes muestras que puedan modelar condiciones estándar o normales, las muestras de condiciones anormales pueden ser insuficientes o no estar disponibles para modelar con precisión todas las posibles condiciones operativas anormales [10].

2.3.1. Resultados en la clasificación

Según la comparación entre las predicciones del modelo y las etiquetas reales de los datos, los resultados de un modelo de clasificación se definen con los términos Verdaderos Positivos (VP), Verdaderos Negativos (VN), Falsos Positivos (FP) y Falsos Negativos (FN). Estos resultados se organizan en una **matriz de confusión**, que es una herramienta clave para evaluar el desempeño del modelo [8], [9].

- **VP - Verdaderos Positivos (True Positives)**: Son los casos en los cuales el modelo predice correctamente la clase positiva. Es decir, el modelo ha clasificado una instancia como positiva y efectivamente la instancia pertenece a la clase positiva. En un problema de diagnóstico de fallas, por ejemplo, un verdadero positivo sería cuando el modelo predice correctamente que una máquina está fallando, y efectivamente está fallando.
- **VN - Verdaderos Negativos (True Negatives)**: Son los casos en los cuales el modelo predice correctamente la clase negativa. Es decir, el modelo ha clasificado una instancia como negativa y efectivamente la instancia pertenece a la clase negativa. Siguiendo el mismo ejemplo de diagnóstico de fallas, un verdadero negativo sería cuando el modelo predice correctamente que una máquina está funcionando correctamente (sin fallas) y efectivamente no tiene fallas.
- **FP - Falsos Positivos (False Positives)**: Son los casos en los cuales el modelo predice incorrectamente la clase positiva. Es decir, el modelo ha clasificado una instancia como positiva, pero en realidad pertenece a la clase negativa. En el ejemplo de diagnóstico de fallas, un falso positivo ocurriría si el modelo predice que la máquina tiene una falla, pero en realidad está funcionando correctamente. Esto es un error de tipo I.
- **FN - Falsos Negativos (False Negatives)**: Son los casos en los cuales el modelo predice incorrectamente la clase negativa. Es decir, el modelo ha clasificado una instancia como negativa, pero en realidad pertenece a la clase positiva. Siguiendo el ejemplo, un falso negativo ocurriría si el modelo predice que una máquina está funcionando correctamente cuando en realidad tiene una falla. Este es un error de tipo II.

2.3.2. Representación en la Matriz de Confusión

Una matriz de confusión organiza estos términos en una tabla para mostrar el rendimiento del modelo de manera más visual y estructurada. Para un problema de clasificación binaria (dos clases: positiva y negativa), la matriz se ve de la siguiente manera:

	Predicción: Positiva	Predicción: Negativa
Real: Positiva	VP	FN
Real: Negativa	FP	VN

- En la **diagonal principal** (donde se encuentran VP y VN) están los aciertos del modelo:
 - VP : el número de instancias de la clase positiva que fueron correctamente clasificadas como positivas.
 - VN : el número de instancias de la clase negativa que fueron correctamente clasificadas como negativas.

- Fuera de la diagonal se encuentran los **errores** del modelo:
 - FP : el número de instancias de la clase negativa que fueron incorrectamente clasificadas como positivas (falsos positivos).
 - FN : el número de instancias de la clase positiva que fueron incorrectamente clasificadas como negativas (falsos negativos).

Observación en la Matriz de Confusión

En la matriz de confusión, la posición de cada valor indica si el modelo ha cometido un error o si ha acertado en la clasificación:

- Si un modelo tiene muchos valores en la diagonal (especialmente en VP y VN), indica que está haciendo buenas predicciones.
- Si el modelo tiene valores altos en FP o FN , significa que está cometiendo muchos errores, lo cual puede ser perjudicial dependiendo del contexto del problema. Por ejemplo, en el diagnóstico de fallas en maquinaria, los falsos negativos (FN) son especialmente peligrosos, ya que implican que una falla no se detectó cuando realmente existía.

2.3.3. Métricas para la evaluación de Modelos de Clasificación

Una vez entrenado un modelo de clasificación, es esencial evaluar su rendimiento. Para ello, se emplean diversas métricas que permiten cuantificar su capacidad para predecir correctamente las clases. Estas métricas pueden ser de carácter general o específicas por clase [8], [9], [11].

Métricas Generales

- **Exactitud (Accuracy)**: Es la proporción de predicciones correctas respecto al total de predicciones. Se calcula como:

$$\text{Exactitud} = \frac{VP + VN}{\text{No. total de datos}}$$

- **Error**: Es la proporción de predicciones incorrectas:

$$\text{Error} = \frac{FP + FN}{\text{No. total de datos}}$$

Métricas por Clase

Dado que la exactitud puede ser engañosa en problemas con clases desbalanceadas, se emplean métricas específicas por clase:

- **Sensibilidad (Recall o Tasa de Verdaderos Positivos)**: Mide la capacidad del modelo para identificar correctamente los casos positivos de una clase determinada. Se define como:

$$\text{Recall} = \frac{VP}{VP + FN}$$

Esta métrica es especialmente relevante en contextos donde los falsos negativos tienen un alto costo, como en el diagnóstico de fallas o enfermedades.

- **Precisión (Precision):** Mide la proporción de instancias clasificadas como positivas que realmente pertenecen a la clase positiva:

$$\text{Precision} = \frac{VP}{VP + FP}$$

- **F1-Score:** Es la media armónica entre la precisión y el recall. Resume el equilibrio entre ambos en un solo valor:

$$F_1 = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$$

Adicionalmente, es posible calcular el `recall_macro`, la cual es una métrica utilizada para evaluar el rendimiento de modelos de clasificación multiclase, especialmente útil en contextos con clases desbalanceadas. Esta métrica calcula el *recall* individual de cada clase y luego realiza un promedio simple entre todos los valores, sin ponderar por el número de ejemplos en cada clase.

$$\text{Recall}_{\text{macro}} = \frac{1}{K} \sum_{i=1}^K \frac{VP_i}{VP_i + FN_i} \quad (2.1)$$

Donde:

- K es el número total de clases.
- VP_i representa los verdaderos positivos para la clase i .
- FN_i representa los falsos negativos para la clase i .
- La fracción $\frac{VP_i}{VP_i + FN_i}$ corresponde al *recall* específico de la clase i .

El `recall_macro` es adecuado cuando se desea que el modelo tenga un desempeño equilibrado entre todas las clases, sin favorecer a las clases mayoritarias. A diferencia de otras métricas promedio como el `recall_weighted`, esta métrica no se ve influenciada por la proporción de muestras de cada clase.

El uso conjunto de estas métricas proporciona una visión más completa del desempeño del modelo, especialmente en contextos multiclase o con clases desbalanceadas, como ocurre en el diagnóstico automático de fallas en maquinaria.

2.3.4. Algoritmo Random Forest

Random Forest (en español, Bosques Aleatorios) es un algoritmo de aprendizaje automático ampliamente utilizado para tareas de clasificación [2] [8], [9]. Pertenece a la familia de los métodos de ensamble, lo que significa que combina múltiples modelos simples (en este caso, árboles de decisión) para obtener un modelo más robusto y preciso. Este enfoque fue propuesto por Leo Breiman en 2001 [12], como una forma de mejorar la precisión de los árboles individuales y reducir el riesgo de sobreajuste.

¿Cómo Funciona?

El funcionamiento de Random Forest se basa en dos principios clave: la creación de múltiples subconjuntos de datos (bootstrap sampling) y la introducción de aleatoriedad en la construcción de cada árbol [12].

Primero, se generan varios subconjuntos del conjunto de datos original mediante muestreo con reemplazo. Cada árbol se entrena con uno de estos subconjuntos (aproximadamente con el 63 % de los datos), dejando el restante como muestra de validación interna

conocida como *out-of-bag*. Luego, durante la construcción de los árboles, en cada punto de decisión el algoritmo considera solo un subconjunto aleatorio de variables, lo cual introduce diversidad entre los árboles. Cada árbol crece hasta su máxima profundidad sin aplicar poda [12].

Para hacer una predicción, el algoritmo utiliza una estrategia de votación mayoritaria entre todos los árboles (en el caso de clasificación). Esto significa que la clase predicha será aquella que reciba más votos entre los árboles del bosque [12].

Ventajas para la Clasificación

Una de las principales fortalezas de Random Forest es su robustez. Gracias al uso de múltiples árboles y a la aleatoriedad en el proceso, el modelo tiende a generalizar mejor que un solo árbol, evitando caer en el sobreajuste. Además, al basarse en la votación de múltiples modelos, es menos sensible al ruido y a la presencia de valores atípicos en los datos [12].

También es capaz de capturar relaciones complejas y no lineales entre variables sin necesidad de transformaciones previas. No requiere escalar las variables ni realizar tratamientos avanzados para valores faltantes, lo cual lo hace muy flexible para trabajar con datos reales [12].

Interpretabilidad y Eficiencia

A pesar de ser un modelo compuesto por muchos árboles, Random Forest ofrece herramientas útiles para interpretar los resultados. Por ejemplo, es posible identificar cuáles variables tienen mayor importancia en la clasificación, ya sea por su impacto en la reducción de impureza (medida Gini o entropía), o por cómo afectan la precisión del modelo al ser alteradas [12].

Finalmente, es un algoritmo que se adapta bien a escenarios con muchos datos y variables. Su estructura permite ser ejecutado en paralelo, ya que cada árbol se construye de manera independiente, lo que mejora la eficiencia computacional y lo hace adecuado para aplicaciones en entornos con alta demanda de procesamiento [12].

Capítulo 3

Metodología

Este capítulo presentó la metodología seguida para el desarrollo de un sistema de diagnóstico de fallas en maquinaria rotativa basado en aprendizaje automático. Se describió en detalle el conjunto de datos utilizado, el cual fue obtenido a partir de un simulador mecánico que reproduce siete condiciones de operación diferentes, registradas mediante acelerómetros en dos configuraciones (underhang y overhang), en tres ejes (axial, radial y tangencial), a una alta frecuencia de muestreo (50 kHz).

A partir de estas señales, se aplicó un proceso de extracción de características en los dominios del tiempo y la frecuencia, obteniendo un total de 189 características por muestra. Este conjunto de datos fue dividido en subconjuntos de entrenamiento y prueba, sobre los cuales se entrenó un modelo de clasificación basado en Random Forest. Para cada modelo se realizó una búsqueda exhaustiva de hiperparámetros utilizando `GridSearchCV` con validación cruzada de 3 particiones, optimizando el `recall_macro` como métrica de desempeño.

Adicionalmente, se implementó un esquema de entrenamiento especializado con el objetivo de evaluar el impacto de distintas configuraciones sobre el rendimiento del modelo. Este esquema incluyó: (i) separación por sensor (underhang y overhang), (ii) submuestreo de las señales a siete frecuencias diferentes, y (iii) separación por eje de vibración (axial, radial, tangencial, y combinado). En total se entrenaron 28 modelos (4 tipos de dataset \times 7 frecuencias), cada uno con su propia optimización de hiperparámetros. Finalmente, todos los modelos fueron evaluados sobre el conjunto de prueba mediante métricas estándar de clasificación, lo que permitió establecer comparaciones frente a distintas resoluciones y configuraciones de entrada.

Finalmente, en el capítulo se describe la implementación práctica del sistema de diagnóstico de fallas en un entorno embebido, utilizando como plataforma base el módulo VAR-SOM-MX8M-PLUS de Variscite. Se presenta una arquitectura de software modular, basada en contenedores Docker y desarrollada sobre un backend FastAPI. La contenedorización de toda la aplicación garantiza su portabilidad, aislamiento y compatibilidad con sistemas embebidos, asegurando un despliegue confiable y repetible en entornos industriales.

3.1. Descripción del conjunto de datos

Para el desarrollo del modelo de diagnóstico de fallas en maquinaria rotativa, se utilizó un conjunto de datos multivariado [13] elaborado por la Universidad Federal de Rio de Janeiro (UFRJ).

3.1.1. Machinery Fault Simulator (MFS) de SpectraQuest Inc. [13]

El conjunto de datos fue generado con el módulo *Alignment/Balance Vibration Trainer (ABVT)* del simulador de fallas mecánicas (Machinery Fault Simulator, MFS) de SpectraQuest Inc. que se presenta en la figura 3.1, el cual permite generar condiciones controladas de operación y fallas.

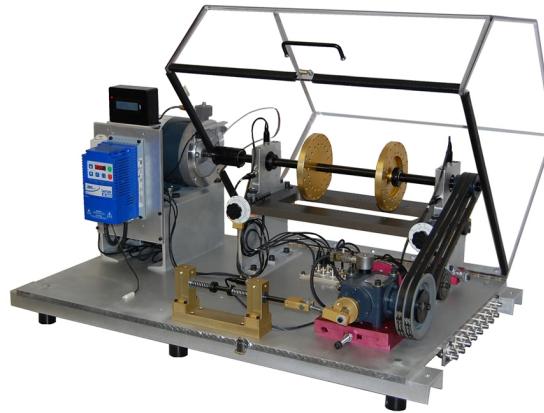


Figura 3.1: SpectraQuest Machinery Fault Simulator (MFS)

Este simulador está construido con una carcasa de cojinete de soporte dividido, un eje deslizante, rotores con extremos de collar dividido, acoplamientos, poleas, un mecanismo de tensado de correa múltiple y montaje de caja de cambios; todos ellos diseñados para ser fácilmente extraíbles y reemplazables entre varios experimentos. Las especificaciones técnicas del banco se presentan en la tabla 3.1.

Cuadro 3.1: Características generales del banco experimental [13].

Especificación	Valor	Unidad
Motor	1/4	CV DC
Rango de velocidad	700–3600	rpm
Peso del sistema	22	kg
Diámetro del eje	16	mm
Longitud del eje	520	mm
Rotor	15.24	cm
Distancia entre rodamientos	390	mm

Cuadro 3.2: Especificaciones del rodamiento en el banco experimental [13].

Especificación	Valor	Unidad
Número de elementos rodantes	8	—
Diámetro de las bolas	0.7145	cm
Diámetro de la jaula	2.8519	cm
FTF (Frecuencia de fallo de la jaula)	0.3750	CPM/rpm
BPFO (Fallo en aro externo)	2.9980	CPM/rpm
BPFI (Fallo en aro interno)	5.0020	CPM/rpm
BSF (Fallo del elemento rodante)	1.8710	CPM/rpm

3.1.2. Sistema de Adquisición de Datos [13].

Los datos fueron adquiridos mediante sensores instalados en el banco experimental, con el propósito de capturar las señales de vibración del sistema bajo diferentes condiciones de operación. Para ello, se emplearon dos acelerómetros industriales triaxiales que permiten registrar señales de vibración en las direcciones radial, axial y tangencial. El primero de ellos corresponde al modelo IMI Sensors 601A01 con una sensibilidad de 100 mV/g (equivalente a 10,2 mV/(m/s²)), un rango de frecuencias útil de 0,27 a 10,000 Hz (equivalente a 16–600,000 CPM), y una capacidad de medición de aceleraciones de hasta ±50 g (±490 m/s²). El segundo acelerómetro utilizado fue el modelo IMI Sensors 604B31 con una sensibilidad de 100 mV/g (10,2 mV/(m/s²)), un rango de frecuencias operativo de 0,5 a 5,000 Hz (30–300,000, CPM), y un rango de medición de aceleración de ±50 g (±490 m/s²).

Para registrar la velocidad de rotación del eje, se incorporó un tacómetro analógico Monarch Instrument MT-190, el cual provee información sobre la frecuencia de operación del sistema. Además, se integró un micrófono Shure SM81, con un rango de respuesta en frecuencia de 20 a 20,000 Hz, con el propósito de capturar señales acústicas relacionadas con las condiciones de operación y posibles fallas.

La digitalización de las señales fue realizada mediante dos módulos de adquisición National Instruments NI 9234, cada uno con 4 canales y una frecuencia de muestreo de 51.2 kHz, lo cual asegura una resolución temporal adecuada para el análisis de fenómenos de alta frecuencia.

3.1.3. Distribución de las clases en el conjunto de datos [13]

El conjunto de datos está compuesto por 1951 registros etiquetados en siete condiciones de funcionamiento: estado normal (operación sin fallas), desalineamiento vertical, desalineamiento horizontal, rodamiento con falla en pista interior (*Cage Fault*), rodamiento con falla en bolas (*Ball Fault*), rodamiento con falla en pista exterior (*Outer race*). La tabla 3.3 muestra la cantidad de etiquetas por condición de funcionamiento en el datataset.

- **Normal:** Corresponde al estado sin fallas de la máquina. El eje está correctamente alineado, no hay desbalance ni problemas estructurales.
- **Desalineamiento vertical:** Esta falla ocurre cuando hay un desplazamiento vertical entre los ejes de dos componentes acoplados (por ejemplo, entre un motor y una bomba). Puede causar vibraciones debido a la falta de colinealidad en el plano vertical.
- **Desalineamiento horizontal:** Similar al anterior, pero el desplazamiento es en el eje horizontal. También genera vibraciones anómalas y puede acelerar el desgaste de los cojinetes y acoplos.
- **Falla en pista interior (*Inner Race Fault*):** Se refiere a un defecto o daño en la superficie de rodadura interna del rodamiento, donde las bolas o rodillos giran. Esto puede ser causado por fatiga, sobrecarga, lubricación inadecuada o contaminación, generando patrones de vibración característicos a una frecuencia específica del rodamiento.
- **Falla en bolas (*Ball Fault*):** Ocurre cuando una o más de las bolas (o rodillos) del rodamiento presentan un defecto, como una picadura, grieta o desgaste. Cada vez que la bola defectuosa entra en contacto con las pistas interior y exterior, se produce un impacto que genera vibraciones a una frecuencia detectable.

- **Falla en pista exterior (*Outer Race Fault*):** Es un daño o defecto en la superficie de rodadura externa del rodamiento. Al igual que con la falla en la pista interior, puede deberse a fatiga, sobrecarga o contaminación. Las vibraciones producidas por esta falla suelen ser periódicas y repetirse cada vez que un elemento rodante pasa por el defecto en la pista exterior.
- **Desbalanceo:** Falla causada por una distribución no uniforme de la masa alrededor del eje rotatorio. Es una de las fallas más comunes y genera vibraciones a la misma frecuencia que la velocidad de rotación.

Cuadro 3.3: Distribución de las clases en el conjunto de datos

Etiqueta	Descripción	Cantidad de registros
0	Normal (operación sin falla)	49
1	Desalinimiento Vertical	301
2	Desalinamiento Horizontal	197
3	Desbalance	333
4	Falla en pista interior (<i>Cage Fault</i>)	376
5	Falla en bolas (<i>Ball Fault</i>)	372
6	Falla en pista exterior (<i>Outer race</i>)	323

Cada registro está almacenado en un archivo csv que consta de 8 columnas, donde cada columna corresponde a una señal temporal con una duración de 5 segundos muestrada a 50 kHz (es decir, 250,000 puntos por señal).

- Columna 1: señal del tacómetro, utilizada para estimar la frecuencia de rotación.
- Columnas 2 a 4: acelerómetros para el rodamiento ubicado en *underhang* (direcciones axial, radial y tangencial).
- Columnas 5 a 7: acelerómetros para el rodamiento ubicado en *overhang* (direcciones axial, radial y tangencial).
- Columna 8: señal de micrófono. Esta señal se decidió no usar porque en contexto industriales no se suele tener instalados micrófonos para medir ruido.

Las dos configuraciones de rodamiento *underhang* y *overhang* hacen referencia a: *underhang*, el rodamiento está más cerca del motor, que es la fuente de accionamiento (drive). En la terminología industrial, esta ubicación se conoce comúnmente como *Drive End* (DE). Es la parte del eje que está directamente conectada a la fuente de potencia, ya sea por acoplamiento, correa, etc. Por otra parte, *Overhang* significa que el rodamiento está al otro lado del rotor, más alejado del motor. Esta posición se refiere al extremo opuesto al motor, o al *Non-Drive End* (NDE). A menudo, en ventiladores o bombas, el impulsor (rotor) puede estar “colgado” o “sobresaliendo” más allá de los rodamientos en el NDE.

3.2. Extracción de características

La figura 3.2 muestra un diagrama que resume el procesamiento seguido. Por cada muestra etiquetada con una condición de funcionamiento, se tienen 7 señales de tiempo, las cuales corresponden a 6 señales de acelerómetros (2 rodamientos x 3 direcciones) y la señal del tacómetro. El objetivo ahora es caracterizar cada señal con su propio conjunto de características en el dominio del tiempo y de la frecuencia.

En el dominio del tiempo se extrajeron 14 características que incluyeron medidas estadísticas (media, desviación estándar, asimetría, curtosis, máximo, mínimo, valor pico a pico, valor absoluto medio), energéticas (energía total, valor cuadrático medio) y métricas de vibración (factor de cresta, factor de forma, factor de impulso, cruces por cero).

En el dominio de la frecuencia, mediante la Transformada Rápida de Fourier (FFT), se obtuvieron 13 características que incluyeron la frecuencia dominante, la amplitud de la componente principal, la energía espectral total, el centroide espectral, el ancho de banda espectral y la planitud espectral.

Por lo tanto, por cada muestra se obtienen 189 características (24 características x 7 señales), las cuales se combinaron en un vector de características de 1 x 189. Al agrupar los vectores de características que representan cada muestra, se obtiene la matriz de datos para el entrenamiento y prueba del modelo de clasificación.

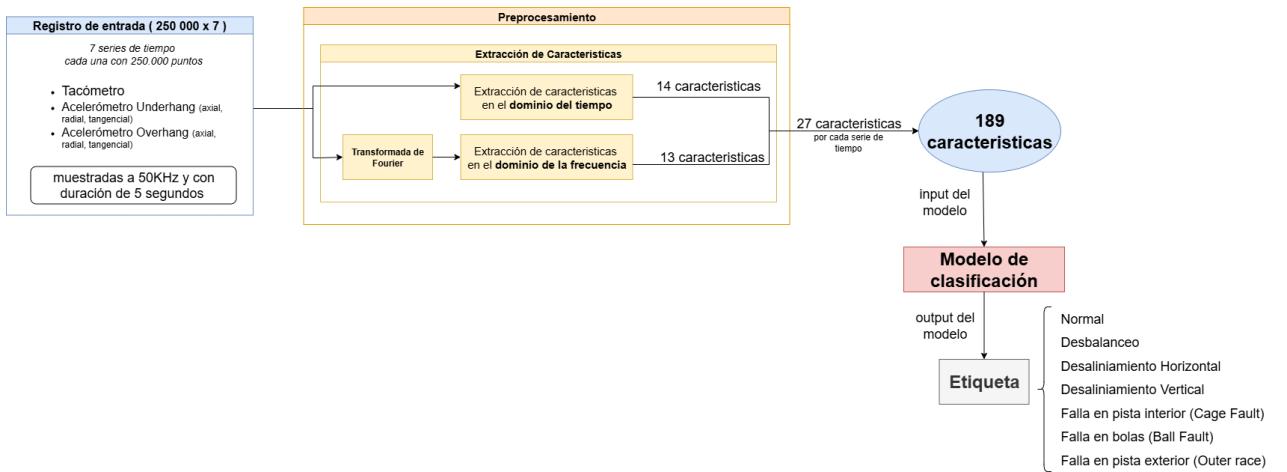


Figura 3.2: Metodología de extracción de características

3.2.1. Dominio del Tiempo

Sea una señal $x[n]$ con N muestras, $n = 0, 1, \dots, N - 1$, el análisis en el dominio del tiempo parte de la representación directa de la señal $x[n]$ en función del tiempo. Esta vista permite observar de manera inmediata las oscilaciones, amplitudes y patrones de la señal de vibración, sin transformaciones adicionales.

Las características estadísticas extraídas en este dominio —como la media, el valor cuadrático medio (RMS), la desviación estándar o la energía— han sido utilizadas tradicionalmente en el monitoreo de condiciones de maquinaria para identificar diferencias entre señales provenientes de estados normales y defectuosos [5].

Características más avanzadas como la asimetría (skewness) y la curtosis son especialmente útiles para señales no estacionarias. Estas métricas están basadas en el análisis de la función de densidad de probabilidad (PDF) de la señal: la curtosis mide la concentración de energía en picos (por ejemplo, impactos en rodamientos), mientras que la asimetría indica si la distribución tiene colas hacia valores positivos o negativos. Para señales normales (por ejemplo, rodamientos en buen estado), se espera que la curtosis sea aproximadamente 3 y la asimetría cercana a 0. La presencia de fallas puede desplazar estos valores de forma significativa [6].

La selección de estas características se basa tanto en la literatura técnica como en su interpretabilidad física y robustez ante ruidos o condiciones de operación variables [6], [2],[5].

1. **Media (Mean)** Es el valor promedio de la señal. Representa el nivel DC o componente continua.

$$\mu = \frac{1}{N} \sum_{n=0}^{N-1} x[n]$$

2. **Desviación estándar (Standard deviation)** Mide la dispersión de la señal respecto a su media. Señales con alta variabilidad presentan fallas mecánicas o desequilibrios.

$$\sigma = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x[n] - \mu)^2}$$

3. **Valor cuadrático medio (RMS)** Mide la energía efectiva de la señal. Es útil para identificar la severidad de vibraciones.

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} x[n]^2}$$

4. **Máximo** Indica el valor más alto de la señal.

$$x_{\max} = \max(x[n]) \quad ; \quad x_{\min} = \min(x[n])$$

5. **Mínimo** Indica el valor más bajo de la señal.

$$x_{\min} = \min(x[n])$$

6. **Valor pico a pico** Diferencia entre el valor máximo y mínimo. Mide la amplitud total de las vibraciones.

$$\text{Pico a pico} = x_{\max} - x_{\min}$$

7. **Valor absoluto medio (MAV)** Mide la magnitud media de la señal sin considerar el signo. Útil en señales de vibración centradas en cero.

$$\text{MAV} = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|$$

8. **Factor de cresta (Crest Factor)**: Relación entre el valor pico y el RMS.

$$\text{CF} = \frac{x_{\max}}{\text{RMS}}$$

9. **Factor de forma (Shape Factor)**: Relación entre el RMS y el valor absoluto medio.

$$\text{SF} = \frac{\text{RMS}}{\text{MAV}}$$

10. **Factor de impulso (Impulse Factor)**: Relación entre el valor pico y el valor absoluto medio.

$$\text{IF} = \frac{x_{\max}}{\text{MAV}}$$

11. **Asimetría (Skewness)** Mide la simetría de la distribución de la señal respecto a la media. Valores positivos indican colas hacia la derecha (asimetría positiva); valores

negativos, hacia la izquierda.

Esta métrica se calcula normalizando la señal respecto a su media μ y su desviación estándar σ , de la siguiente forma:

$$\text{Skewness} = \frac{1}{N} \sum_{n=0}^{N-1} \left(\frac{x[n] - \mu}{\sigma} \right)^3$$

12. Curtosis (Kurtosis)

Mide la concentración de la energía en los picos de la señal. Una curtosis alta sugiere impactos o defectos mecánicos localizados. Se basa en el cociente de la cuarta potencia de la desviación normalizada.

$$\text{Kurtosis} = \frac{1}{N} \sum_{n=0}^{N-1} \left(\frac{x[n] - \mu}{\sigma} \right)^4$$

Al igual que en la asimetría, μ representa la media y σ la desviación estándar de la señal.

13. Energía

Suma total de la señal al cuadrado. Representa la energía global del sistema.

$$E = \sum_{n=0}^{N-1} x[n]^2$$

14. Cruces por cero

Número de veces que la señal cambia de signo. Útil para detectar componentes oscilantes o vibraciones anómalas.

$$\text{Zero Crossings} = \frac{1}{N} \sum_{n=1}^{N-1} \mathbb{I}[\text{sign}(x[n]) \neq \text{sign}(x[n-1])]$$

3.2.2. Dominio de la Frecuencia

El análisis en el dominio de la frecuencia se basa en la transformada de Fourier de la señal, la cual permite representar su contenido espectral como una función de la frecuencia. Esto revela componentes periódicos ocultos en el dominio del tiempo, permitiendo identificar frecuencias características asociadas a defectos específicos en rodamientos o componentes rotativos [2].

A partir del espectro de amplitudes y potencia se pueden extraer características como la frecuencia dominante, el centroide espectral o el ancho de banda, las cuales permiten describir la localización, distribución e intensidad de la energía vibracional [2].

Adicionalmente, métricas estadísticas como la planitud espectral, la asimetría y la curtosis del espectro se calculan sobre la distribución de energía en frecuencia. Estas permiten diferenciar entre señales tonales (con picos concentrados) y señales de tipo impulsivo o ruidoso. Al igual que en el dominio del tiempo, la variación de la PDF espectral por efecto de fallas puede evidenciarse a través de estas métricas [2].

La selección de estas características se fundamenta en estudios previos en monitoreo de condiciones, y en su capacidad para discriminar entre estados de salud distintos en entornos industriales reales [2], [6].

Sea $X[k]$ la transformada de Fourier de la señal $x[n]$, y f_k la frecuencia correspondiente al índice k . Se consideran las siguientes características espectrales:

1. **Frecuencia dominante** Es la frecuencia con mayor amplitud en el espectro. Indica la velocidad principal de vibración.

$$f_{\text{dom}} = f_k \quad \text{donde} \quad k = \arg \max (|X[k]|)$$

2. **Amplitud dominante** Valor máximo del espectro de amplitudes. Puede revelar la severidad de un modo de vibración.

$$A_{\text{dom}} = \max \left(\frac{2}{N} |X[k]| \right)$$

3. **Energía espectral** Energía acumulada en el dominio de la frecuencia. Útil para comparar señales con distintos contenidos esenciales.

$$E_{\text{espectral}} = \sum_{k=0}^{N/2-1} \left(\frac{2}{N} |X[k]| \right)^2$$

4. **Centroide espectral** Representa la frecuencia promedio ponderada por la magnitud espectral.

$$f_c = \frac{\sum_{k=0}^{N/2-1} f_k \cdot |X[k]|}{\sum_{k=0}^{N/2-1} |X[k]|}$$

5. **Ancho de banda espectral** Mide la dispersión del espectro respecto al centroide. Indica si la energía está concentrada o distribuida.

$$BW = \sqrt{\frac{\sum_{k=0}^{N/2-1} (f_k - f_c)^2 \cdot |X[k]|}{\sum_{k=0}^{N/2-1} |X[k]|}}$$

6. **Planitud espectral** Indica cuán plano es el espectro. Es la relación entre la media geométrica y la media aritmética del espectro. Distingue entre espectros tonales y ruidosos. Valores cercanos a 1 indican ruido blanco; valores bajos indican tonos dominantes.

$$\text{Flatness} = \frac{\left(\prod_{k=0}^{N/2-1} |X[k]| \right)^{1/(N/2)}}{\frac{1}{N/2} \sum_{k=0}^{N/2-1} |X[k]|}$$

7. **Media del espectro de potencia** Valor promedio del espectro de potencia $S[k] = |X[k]|^2$.

$$F_{\text{mv}} = \frac{1}{N/2} \sum_{k=0}^{N/2-1} S[k]$$

8. **Máximo del espectro de potencia** Valor pico del espectro de potencia.

$$F_{\text{max}} = \max(S[k])$$

9. **Valor RMS del espectro de potencia** Mide la magnitud general del espectro.

$$F_{\text{rms}} = \sqrt{\frac{1}{N/2} \sum_{k=0}^{N/2-1} S[k]^2}$$

10. **Varianza del espectro de potencia** Mide la dispersión de energía respecto a la media espectral.

$$F_{\text{vr}} = \frac{1}{(N/2) - 1} \sum_{k=0}^{N/2-1} (S[k] - F_{\text{mv}})^2$$

11. **Desviación estándar del espectro de potencia** Raíz cuadrada de la varianza.

$$F_{\text{sd}} = \sqrt{F_{\text{vr}}}$$

12. **Asimetría (skewness) del espectro de potencia** Mide la simetría de la distribución del espectro respecto a su media.

$$F_{\text{sf}} = \frac{\sum_{k=0}^{N/2-1} (S[k] - F_{\text{mv}})^3}{(N/2) \cdot F_{\text{sd}}^3}$$

13. **Curtosis del espectro de potencia** Mide el nivel de picos extremos en el espectro.

$$F_{\text{kf}} = \frac{\sum_{k=0}^{N/2-1} (S[k] - F_{\text{mv}})^4}{(N/2) \cdot F_{\text{sd}}^4}$$

14. **Relación pico-media del espectro de potencia** Compara el valor máximo con la media espectral.

$$F_{\text{rs}} = \frac{F_{\text{max}}}{F_{\text{mv}}}$$

La Figura 3.3 ejemplifica gráficamente el proceso de extracción de características descrito hasta ahora, el cual transforma un archivo csv con 7 series de tiempo en un vector de 189 características. Para un archivo csv tomado del dataset, en la gráfica se muestra la señal en el dominio del tiempo, su espectro de amplitud obtenido mediante la Transformada Rápida de Fourier (FFT), y un conjunto tabulado de las características calculadas por cada columna del csv (serie de tiempo de un sensor).

Análisis Completo de Señales: Tiempo, Frecuencia y Características

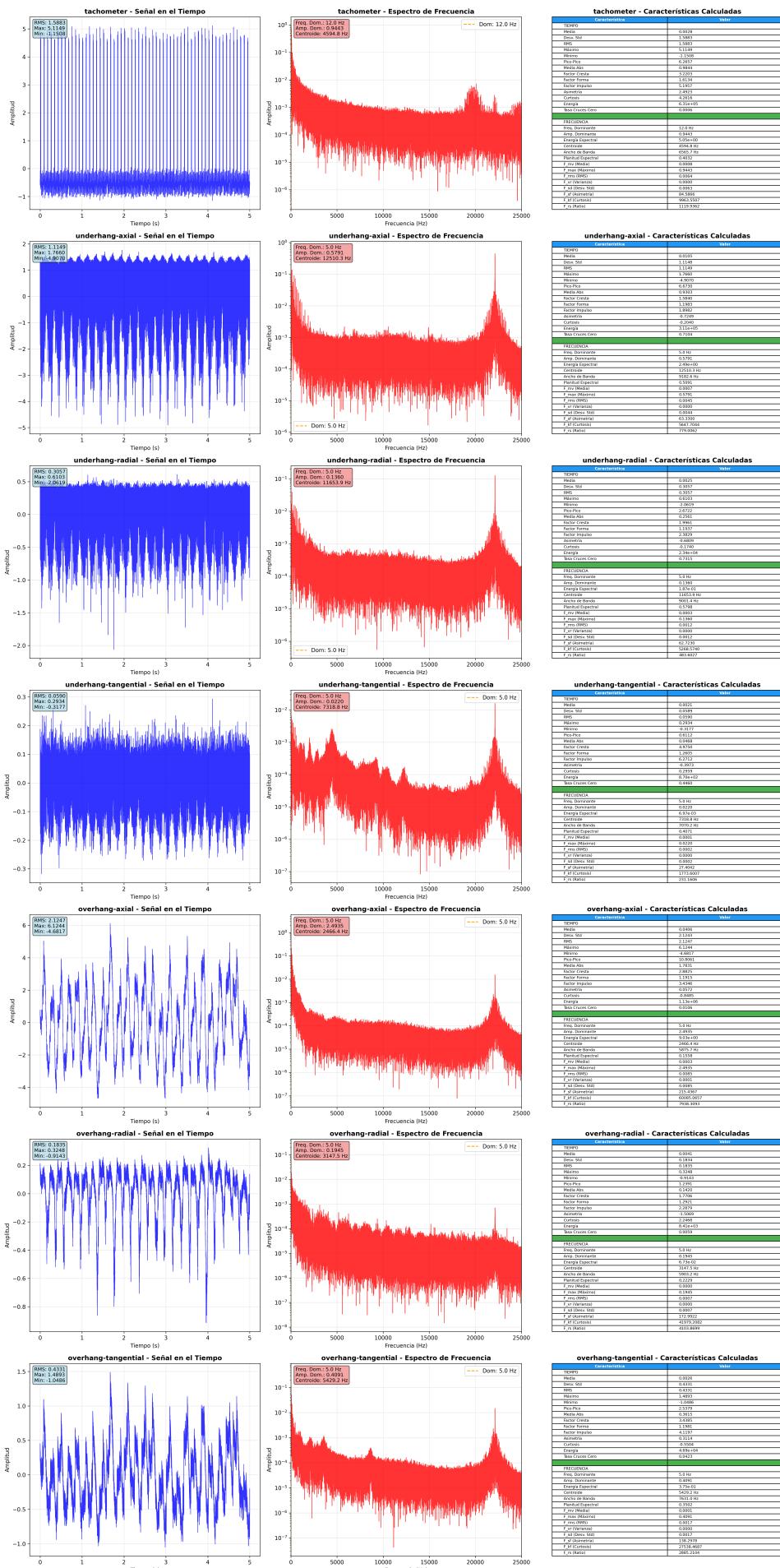


Figura 3.3: Ejemplo de extracción de características

3.3. Modelamiento con Machine Learning: Modelo Base

La Figura 3.4 resume el flujo general seguido para construir el modelo de diagnóstico de fallas.

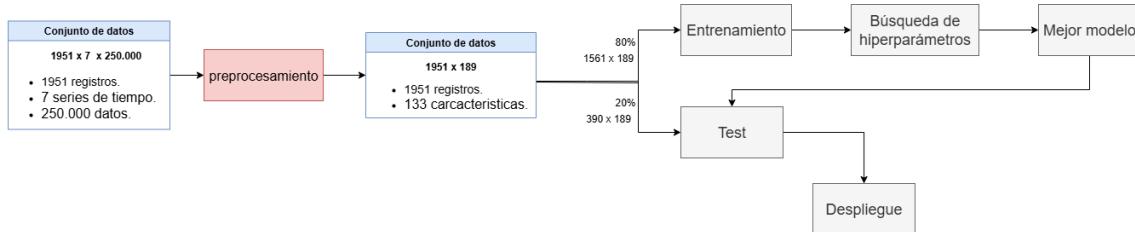


Figura 3.4: Flujo de procesamiento y modelamiento del sistema de diagnóstico

El conjunto de datos original está compuesto por 1951 registros, cada uno con 7 señales de tiempo (6 acelerómetros y 1 tacómetro) de 250.000 muestras cada una, correspondientes a 5 segundos muestreados a 50 kHz. A través del proceso de preprocessamiento descrito anteriormente, cada señal se transforma en un vector de características, generando un conjunto final de datos tabulares de dimensiones 1951×189 , donde cada fila representa una muestra caracterizada por 189 atributos (27 por señal \times 7 señales).

Este conjunto se divide en dos subconjuntos: el 80 % para entrenamiento (1561×189) y el 20 % para prueba (390×189). Sobre el conjunto de entrenamiento se realiza la búsqueda de hiperparámetros utilizando validación cruzada, lo que permite seleccionar el mejor modelo en términos de desempeño general. Posteriormente, se evalúa su rendimiento sobre el conjunto de prueba. Finalmente, el modelo óptimo se considera listo para ser desplegado en el entorno de inferencia o monitoreo continuo.

3.3.1. Modelo de aprendizaje automático

Se eligió el algoritmo de clasificación Bosques Aleatorios (Random Forest) debido a su robustez frente a ruido, su capacidad para capturar relaciones no lineales entre variables y, especialmente, por su naturaleza interpretativa, que permite estimar la importancia relativa de cada característica en la predicción de la clase.

Inicialmente, se entrenó un modelo base con una configuración estándar de la librería scikit-learn, a partir del cual se realizó una búsqueda sistemática de los mejores hiperparámetros a través de una técnica de validación cruzada. Los hiperparámetros considerados se resumen en la tabla 3.4

Cuadro 3.4: Hiperparámetros considerados en la validación cruzada

Hiperparámetro	Descripción	Valores Evaluados
<code>class_weight</code>	Controla la ponderación de clases en presencia de desbalance de clases.	None, 'balanced', 'balanced_subsample'
<code>n_estimators</code>	Número de árboles que conforman el bosque aleatorio.	80, 100, 120
<code>max_depth</code>	Profundidad máxima permitida para cada árbol individual.	2, 4, 6, 8, 16
<code>min_samples_split</code>	Mínimo número de muestras requerido para dividir un nodo interno.	2, 3, 4, 5, 6
<code>criterion</code>	Métrica utilizada para evaluar la calidad de una división en los nodos.	'gini', 'entropy', 'log_loss'

■ `class_weight`:

- None: No se aplica ponderación. Se asume que todas las clases tienen la misma importancia.
- 'balanced': Asigna pesos inversamente proporcionales a la frecuencia de las clases. Mejora el rendimiento cuando hay desbalance.
- 'balanced_subsample': Igual que 'balanced', pero los pesos se recalculan en cada muestra *bootstrap*. Útil para datasets muy desbalanceados.

■ `n_estimators`:

- Aumentar el número de árboles puede mejorar la precisión del modelo.
- Sin embargo, esto también incrementa el tiempo de entrenamiento y el costo computacional.

■ `max_depth`:

- Profundidades bajas pueden llevar a subajuste (modelo poco expresivo).
- Profundidades muy altas pueden causar sobreajuste (modelo demasiado específico).

■ `min_samples_split`:

- Valores bajos permiten divisiones más tempranas y árboles más complejos.
- Valores altos generan árboles más simples, lo que reduce el riesgo de sobreajuste y el costo computacional.

■ `criterion`:

- 'gini': Mide la impureza como la probabilidad de clasificar incorrectamente un punto si se elige al azar según la distribución de clases. Más rápido computacionalmente.
- 'entropy': Basado en la teoría de la información. Puede producir divisiones más informativas, pero es más costoso computacionalmente.
- 'log_loss': Penaliza las predicciones incorrectas con alta confianza. Recomendado cuando se desea una salida probabilística calibrada.

3.3.2. Selección de Hiperparámetros mediante GridSearchCV y Validación Cruzada

Para optimizar el rendimiento del modelo *Random Forest*, se implementó una búsqueda exhaustiva de hiperparámetros utilizando la función `GridSearchCV` del paquete `scikit-learn`. Este procedimiento permite explorar de forma sistemática todas las combinaciones posibles de valores definidos para los hiperparámetros relevantes, seleccionando aquella configuración que maximice el desempeño del modelo según una métrica determinada. En este caso, la métrica seleccionada fue el **promedio del `recall_macro`**, evaluado mediante un esquema de validación cruzada de 3 particiones (3-fold cross-validation) [9].

La **validación cruzada** consiste en dividir los datos seleccionados para entrenamiento en tres subconjuntos iguales (denominados *folds*). En cada iteración, se utilizan dos de estos subconjuntos (2/3 del total) para el entrenamiento del modelo y el subconjunto restante (1/3) para la validación. Este proceso se repite tres veces, rotando el subconjunto asignado a la validación. Finalmente, se calcula el promedio del `recall_macro` obtenido en las tres iteraciones para cada combinación de hiperparámetros evaluada.

El uso de `recall_macro` como métrica de optimización es especialmente apropiado en contextos donde existe desbalance de clases, ya que esta métrica calcula el *recall* individual para cada clase y luego promedia los resultados. Esto evita el sesgo hacia la clase mayoritaria y favorece un desempeño balanceado del modelo. Es ideal, por ejemplo, en aplicaciones donde las clases minoritarias representan eventos críticos, como fallas poco frecuentes en sistemas industriales.

El número total de combinaciones evaluadas por `GridSearchCV` corresponde al producto cartesiano de todos las opciones mostrada en la tabla 3.4:

$$\text{Total de combinaciones} = 3 \times 3 \times 5 \times 5 \times 3 = 675$$

Cada una de estas 675 combinaciones fue evaluada mediante validación cruzada de 3 *folds*, lo que implica un total de:

$$675 \text{ combinaciones} \times 3 \text{ validaciones por combinación} = 2025 \text{ entrenamientos}$$

Este procedimiento garantiza una búsqueda exhaustiva que equilibra la precisión del modelo con su capacidad de generalización, al evaluar rigurosamente su desempeño en diferentes particiones del conjunto de datos. Al finalizar la búsqueda de hiperparámetros mediante validación cruzada se obtiene el modelo con mejor desempeño en la métrica `recall macro`, el cual será finalmente evaluado con el conjunto de datos de prueba.

3.3.3. Evaluación del modelo

El desempeño del modelo final fue evaluado utilizando el conjunto de prueba, compuesto por 391 observaciones distribuidas en 7 clases diferentes. En la figura 3.5 se presenta la distribución de las clases en el conjunto de evaluación.

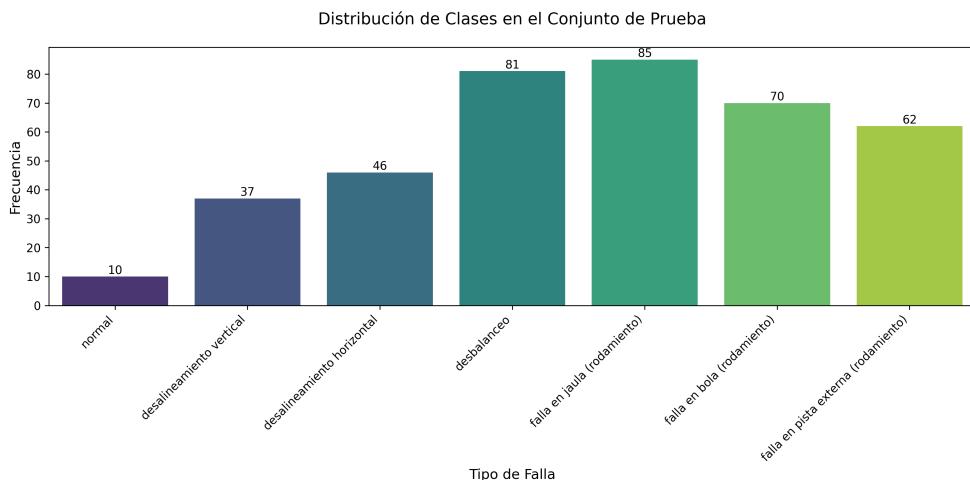


Figura 3.5: Distribución de las clases en el conjunto de evaluación

Esta evaluación se realizó a través de:

- Matriz de confusión: permite analizar visualmente la distribución de predicciones correctas e incorrectas por clase.
- Métricas globales de desempeño: accuracy, macro recall, macro precision, macro F1-score.
- Métricas de desempeño por clase: precisión, recall y F1-score.

Estas métricas permitieron evaluar la capacidad del modelo para diagnosticar correctamente cada tipo de falla simulada.

3.4. Modelamiento con Machine Learning: Modelos Especializados

Con el objetivo de desarrollar modelos de clasificación más robustos y generalizables, se aplicó un proceso de entrenamiento especializado que involucró varias etapas de preprocesamiento sobre los datos originales. Este proceso consideró la estructura del sistema de adquisición de datos, así como la resolución temporal y la orientación de los sensores.

3.4.1. Separación por Sensor: Underhang y Overhang

El dataset original contenía registros provenientes de dos acelerómetros montados en posiciones distintas: *underhang* y *overhang*. Aunque ambos sensores monitorean la misma máquina, sus ubicaciones producen diferencias sutiles en la señal. Para evitar que el modelo se limite a esta configuración específica, se optó por separar ambos acelerómetros. De modo que, se generó un nuevo dataset con 3902 filas (1951×2 , el doble de filas originales).

3.4.2. Submuestreo: Reducción de la Frecuencia de Muestreo

La señal original fue adquirida a una frecuencia de 50,000 Hz, lo cual proporciona una alta resolución temporal. Sin embargo, para evaluar cómo afecta esta resolución a la detección de fallas, se aplicó un submuestreo utilizando factores de $\{2, 4, 8, 16, 32, 64\}$. Como resultado, se obtuvieron las siguientes frecuencias de muestreo:

- 50,000 Hz (original)
- 25,000 Hz
- 12,500 Hz
- 6,250 Hz
- 3,125 Hz
- 1,563 Hz
- 781 Hz

Este análisis permite comprender cómo se ve afectado el rendimiento del modelo al trabajar con señales de menor resolución, aspecto crítico en aplicaciones IoT con recursos limitados.

3.4.3. Separación por Ejes de Vibración

Cada acelerómetro mide vibración en tres direcciones: axial, radial y tangencial. Para estudiar el aporte informativo de cada dirección, se definieron dos tipos de datasets:

- **Dataset combinado (3 ejes):** Contiene las 72 características extraídas al combinar las 24 características de cada una de las tres direcciones (axial, radial, tangencial).
- **Datasets individuales por dirección:** Se generan tres archivos separados:
 - `data_axial.csv`: Solo contiene características de la señal axial.
 - `data_radial.csv`: Solo contiene características de la señal radial.
 - `data_tangential.csv`: Solo contiene características de la señal tangencial.

Esto permite generalizar el modelamiento a sensores que son triaxiles o uniaxiales.

3.4.4. Esquema de Entrenamiento Especializado

Se entrenó un modelo de clasificación independiente para cada combinación posible entre tipo de dataset (4 variantes) y frecuencia de muestreo (7 niveles), generando un total de:

$$4 \text{ datasets} \times 7 \text{ frecuencias} = 28 \text{ modelos entrenados}$$

- | | |
|----------------------|-----------------------|
| 1. (3axis, 50000 Hz) | 8. (axial, 50000 Hz) |
| 2. (3axis, 25000 Hz) | 9. (axial, 25000 Hz) |
| 3. (3axis, 12500 Hz) | 10. (axial, 12500 Hz) |
| 4. (3axis, 6250 Hz) | 11. (axial, 6250 Hz) |
| 5. (3axis, 3125 Hz) | 12. (axial, 3125 Hz) |
| 6. (3axis, 1562 Hz) | 13. (axial, 1562 Hz) |
| 7. (3axis, 781 Hz) | 14. (axial, 781 Hz) |

- | | |
|------------------------|----------------------------|
| 15. (radial, 50000 Hz) | 22. (tangencial, 50000 Hz) |
| 16. (radial, 25000 Hz) | 23. (tangencial, 25000 Hz) |
| 17. (radial, 12500 Hz) | 24. (tangencial, 12500 Hz) |
| 18. (radial, 6250 Hz) | 25. (tangencial, 6250 Hz) |
| 19. (radial, 3125 Hz) | 26. (tangencial, 3125 Hz) |
| 20. (radial, 1562 Hz) | 27. (tangencial, 1562 Hz) |
| 21. (radial, 781 Hz) | 28. (tangencial, 781 Hz) |

Para cada uno de los 28 modelos entrenados se realizó un proceso de optimización de hiperparámetros utilizando `GridSearchCV` con validación cruzada de 3 particiones. Esta búsqueda exhaustiva exploró múltiples combinaciones de parámetros clave del algoritmo Random Forest, tales como el número de árboles, profundidad máxima, criterio de división, ponderación de clases y mínimo de muestras por división. La métrica utilizada para seleccionar la mejor configuración fue el `recall_macro`, con el fin de priorizar un desempeño balanceado entre clases, particularmente en un contexto con datos desbalanceados. Este proceso aseguró que cada modelo alcanzara su mejor desempeño posible bajo su configuración de entrada.

Finalmente, cada modelo fue evaluado con el conjunto de prueba mediante métricas globales de desempeño (accuracy, macro recall, macro precision, macro F1-score). Este esquema permite comparar el impacto de cada factor (resolución temporal y dirección de vibración) sobre el desempeño del sistema de detección de fallas, y proporciona una base experimental para evaluar el rendimiento bajo condiciones específicas de resolución y contenido informativo.

3.5. Implementación en dispositivo IoT

3.5.1. Variscite Symphony board VAR-SOM-MX8M-PLUS V2

El dispositivo IoT seleccionado para embeber los modelos de aprendizaje automático es un **System on Module (SoM)** de referencia **VAR-SOM-MX8M-PLUS**, desarrollado por la empresa Variscite (ver figura 3.6). Este módulo compacto integra en un solo circuito los componentes esenciales de una computadora, incluyendo procesador, memoria y conectividad, lo que facilita su incorporación en sistemas industriales embebidos y aplicaciones de Internet de las Cosas (IoT). A continuación se resumen las principales características técnicas de la tarjeta VAR-SOM-MX8M-PLUS [14]:

- **Procesamiento principal:**

- CPU Quad-core CortexTM-A53 a 1.8 GHz, adecuada para procesamiento de tareas complejas.
- Coprocesador CortexTM-M7 a 800 MHz, orientado a operaciones en tiempo real.

- **Capacidades de Inteligencia Artificial (IA):**

- Unidad de Procesamiento Neuronal (NPU) integrada, especialmente diseñada para acelerar operaciones de *machine learning* e inferencia de modelos en el borde.

■ **Conectividad:**

- Compatibilidad con Wi-Fi 6 y Bluetooth 5.4 (incluyendo soporte para 802.15.4).
- Doble puerto Ethernet Gigabit.
- Interfaces industriales: USB 3.0, CAN-FD y PCIe.

■ **Almacenamiento:**

- Soporte para almacenamiento eMMC de hasta 128 GB y tarjetas microSD.

■ **Dimensiones:**

- Formato compacto de 67.8 mm × 33 mm.

■ **Entorno de desarrollo:**

- Arquitectura ARM64 compatible con sistemas embebidos basados en Linux, como Yocto Project, Debian, Android, Boot2Qt, FreeRTOS y Zephyr.

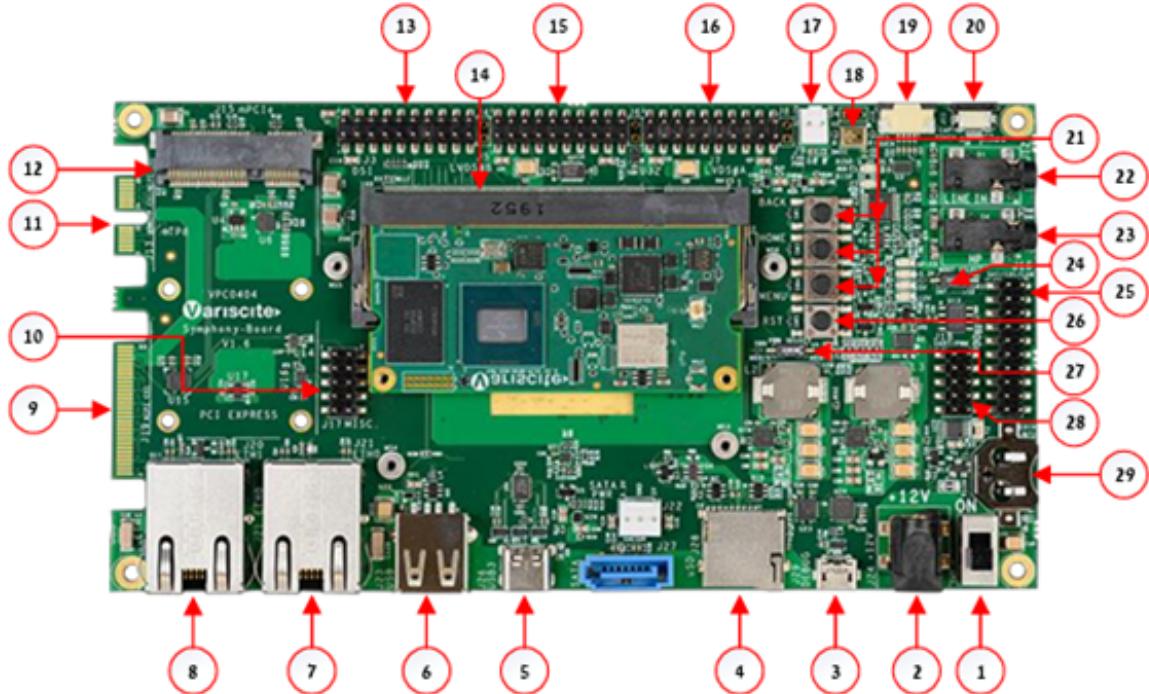


Figura 3.6: Variscite Symphony board VAR-SOM-MX8M-PLUS V2. Tomado de [15]

De acuerdo con la enumeración presentada en la figura 3.6, la tarjeta Symphony board incluye lo siguiente:

1. **Interruptor de Encendido (SW7)**
2. **Jack de 12V DC (J24)**
3. **Depuración USB (J29)**
4. **Ranura para Tarjeta micro SD (J28)**

5. **USB 3.0 OTG** (J26)
6. **Host USB 2.0** (J23)
7. **Ethernet Gigabit #0** (J21)
8. **Ethernet Gigabit #1** (J20)
9. **Conector de Cámara MIPI-CSI #1** [opcional] (J19)
10. **Encabezado Misceláneo #1** (J17)
11. **Conector de Cámara HDMI/MIPI-CSI #2** [opcional] (J13)
12. **Conector Mini PCI Express** (J15)
13. **Encabezado Misceláneo #2** (J3)
14. **Conector SOM** (J1)
15. **Encabezado LVDS#B** (J5)
16. **Encabezado LVDS#A/DSI** (J7)
17. **Conector de Alimentación del Ventilador** (J9)
18. **Micrófono Digital** (U1)
19. **Táctil Resistivo** (J10)
20. **Táctil Capacitivo** (J11)
21. **Botones de Usuario** (SW1, SW2, SW4)
22. **Conector de Entrada de Línea** (J12)
23. **Conector de Auriculares** (J14)
24. **Interruptor de Selección de Arranque** (SW3)
25. **Encabezado SAI/I2C/SPI/CAN** (J16)
26. **Botón de Reinicio** (SW5)
27. **Interruptor de Selección de Alimentación** (SW6)
28. **Encabezado UART/PWM** (J18)
29. **Soporte de Batería RTC** (GBT1)

3.5.2. Arquitectura

La Figura 3.7 presenta la arquitectura general del sistema desplegado en la **Symphony board**. Esta se basa en un enfoque modular y moderno, utilizando contenedores Docker, un backend con **FastAPI** y modelos de **Machine Learning** (ML). La contenedezoración garantiza que el sistema sea portable, escalable y fácil de mantener en entornos embebidos como la i.MX8MP.

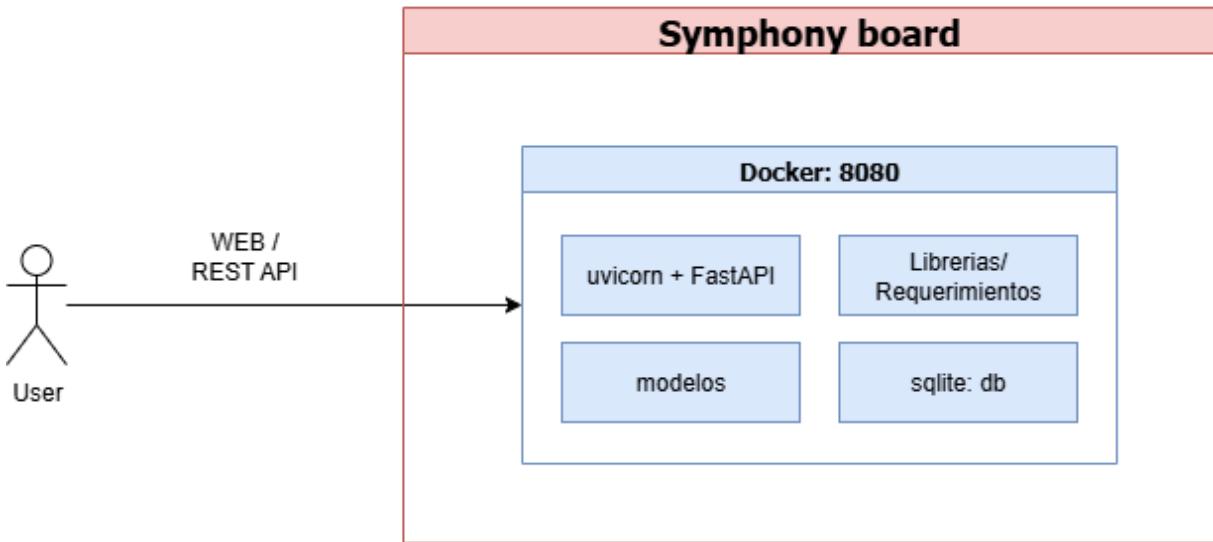


Figura 3.7: Arquitectura implementada en la Symphony board

3.5.3. Contenedezorización y Despliegue

El sistema se ejecuta completamente dentro de un contenedor **Docker**, expuesto en el puerto **8080**, lo que ofrece las siguientes ventajas:

- **Portabilidad:** Facilita su implementación en distintos entornos (desarrollo, pruebas, producción).
- **Compatibilidad ARM64:** Diseñado para funcionar eficientemente en plataformas embebidas como la Symphony board.
- **Aislamiento:** Todas las dependencias están contenidas, evitando conflictos con el sistema anfitrión.

Dentro del contenedor se encuentran los siguientes módulos:

- **Uvicorn + FastAPI:** Gestionan las solicitudes HTTP y ejecutan la lógica del backend.
- **Modelos de ML:** Algoritmos previamente entrenados encargados de realizar inferencias.
- **SQLite:** Base de datos liviana para almacenar predicciones e información histórica.
- **Librerías/ Requerimientos:** Dependencias necesarias para la ejecución del sistema.

3.5.4. Capa de Usuario (Interfaz)

La interfaz web del sistema, desarrollada con HTML, CSS y JavaScript, se comunica con el backend mediante una **API REST**. Desde esta interfaz, el usuario puede realizar inferencias o consultar resultados enviando solicitudes HTTP a través de endpoints como `POST /predict`. Esta capa facilita la interacción del usuario final con el dispositivo.

3.5.5. Capa Web / REST API

El backend está construido con **FastAPI**, un framework de alto rendimiento para la creación de APIs en Python. Se ejecuta sobre el servidor **Uvicorn** (ASGI), que permite manejar múltiples solicitudes de manera eficiente, incluso en dispositivos con recursos limitados.

3.5.6. Capa de Procesamiento

Esta capa se encarga de ejecutar los modelos de **Machine Learning**, los cuales fueron entrenados previamente usando bibliotecas como scikit-learn. Los datos recibidos son pre-procesados con **NumPy** o **Pandas** antes de ser evaluados. Los resultados se almacenan en una base de datos **SQLLite**, lo que permite llevar un registro histórico sin sacrificar eficiencia ni consumo de recursos.

3.5.7. Modo de uso

Finalmente, el modo de uso de la arquitectura se ilustra en la Figura 3.8. En esta configuración, el usuario conecta su computador directamente al puerto Ethernet de la **Symphony board**. Ambos dispositivos deben estar en la misma red local, lo que permite acceder a la interfaz web del sistema ingresando la dirección IP de la placa en un navegador.

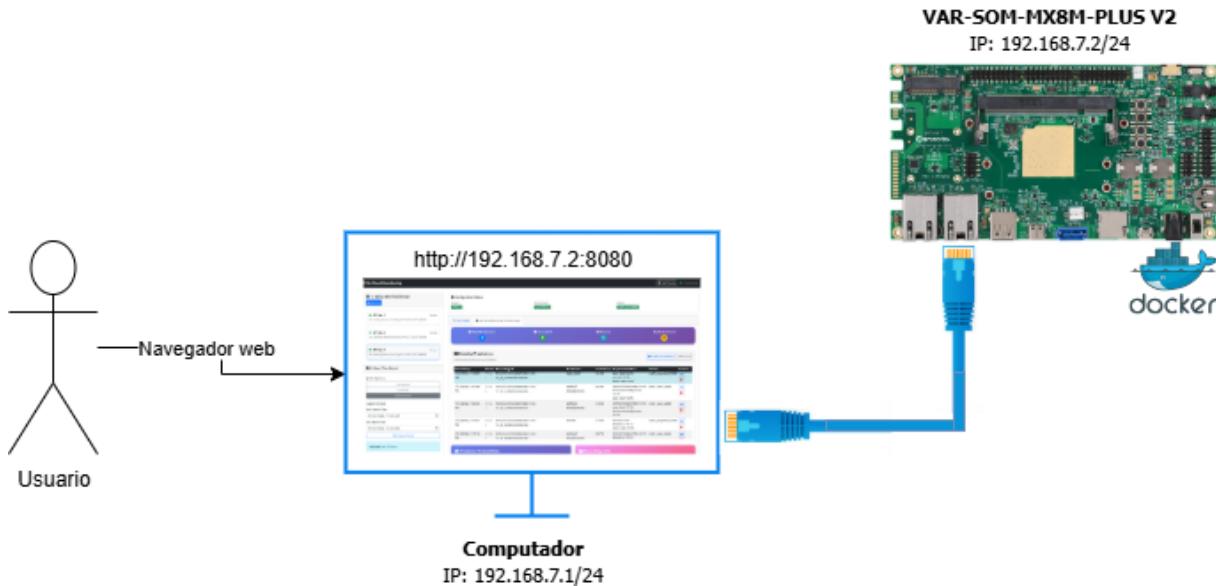


Figura 3.8: Modo de uso de la arquitectura implementada en la Symphony board

Capítulo 4

Resultados y Discusión

4.1. Modelamiento con Machine Learning: Modelo Base

4.1.1. Mejores hiperparámetros encontrados

A continuación, se presentan la combinación de hipermetamorfosis que obtuvo el mejor desempeño en la validación cruzada descrita previamente en la sección 4.1.2.

- **class_weight=None**: No fue necesario aplicar un ajuste de pesos para clases desbalanceadas, lo que indica que el modelo pudo manejar adecuadamente el desbalance presente en el dataset sin necesidad de ponderaciones adicionales.
- **criterion='gini'**: El índice de Gini resultó ser más efectivo que la entropía para medir la calidad de las divisiones, permitiendo generar árboles con mejor capacidad predictiva en este contexto.
- **max_depth=16**: Una profundidad de hasta 16 niveles permitió capturar relaciones complejas en los datos sin incurrir en sobreajuste, aprovechando la capacidad del modelo Random Forest para controlar la varianza.
- **min_samples_split=2**: Al permitir divisiones con un mínimo de 2 muestras por nodo, el modelo construyó árboles más detallados. Esta configuración fue viable gracias al control del sobreajuste que proporciona el ensamble.
- **n_estimators=90**: Un conjunto de 90 árboles fue suficiente para lograr un buen equilibrio entre precisión y eficiencia computacional, sin necesidad de incrementar innecesariamente el número de estimadores.

4.1.2. Evaluación del Modelo

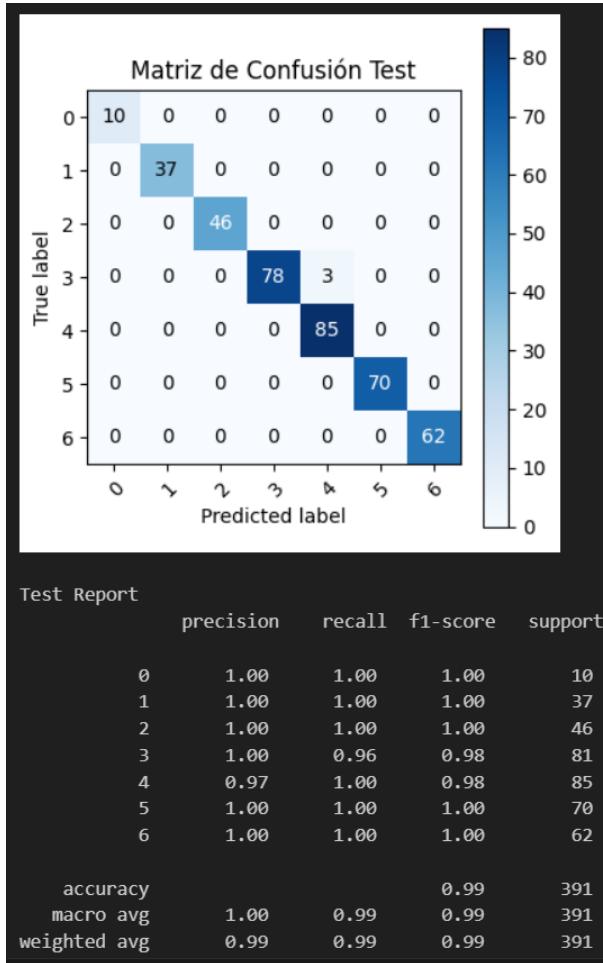


Figura 4.1: Matriz de confusión y métricas de evaluación obtenidas para el mejor modelo encontrado con validación cruzada. Donde 0: (normal), 1: (desalineamiento vertical), 2: (desalineamiento horizontal), 3: (desbalanceo), 4: (falla en jaula - rodamiento), 5: (falla en bola - rodamiento), 6: (falla en pista externa - rodamiento)

El desempeño del modelo final fue evaluado utilizando el conjunto de prueba, compuesto por 391 observaciones distribuidas en 7 clases diferentes. Los resultados de esta evaluación se presentan en la figura 4.1. La exactitud global del modelo (*accuracy*) alcanzó un 99 %, lo cual demuestra su capacidad de generalización al conjunto de prueba.

- **Matriz de confusión:** permitió analizar visualmente la distribución de predicciones correctas e incorrectas por clase. Se observó un desempeño muy alto en todas las clases, con una tasa de clasificación perfecta (100 %) en la mayoría de los casos. Las únicas confusiones detectadas ocurrieron entre las clases 3 y 4, donde 3 instancias fueron clasificadas incorrectamente.
- **Reporte de métricas por clase:** se calculó la *precision*, *recall* y *F1-score* para cada clase individual, así como los promedios *macro* y *ponderado (weighted)*:
 - **Precisión:** se mantuvo en 1.00 para todas las clases, excepto para la clase 4, donde fue de 0.97.

- **Recall:** se obtuvo un valor perfecto de 1.00 en todas las clases, salvo la clase 3, con un recall de 0.96.
- **F1-score:** se mantuvo cercano o igual a 1.00 en todas las clases.

Es importante destacar que el conjunto de prueba presenta una distribución desigual entre clases (por ejemplo, la clase 0 contiene solo 10 instancias, mientras que la clase 5 contiene 85). En este contexto, se utilizaron dos promedios globales:

- **Macro promedio:** calcula las métricas por clase y luego promedia, asignando igual peso a cada clase. Se obtuvo un `recall_macro` de 0.99, lo cual indica un rendimiento balanceado entre todas las clases, sin sesgo hacia las mayoritarias.
- **Promedio ponderado (weighted):** pondera las métricas por el número de instancias en cada clase. El `recall_weighted` fue de 0.99, confirmando un rendimiento sólido incluso considerando el desbalance.

En conjunto, estos resultados indican que el modelo fue capaz de diagnosticar correctamente cada tipo de falla simulada con un alto grado de exactitud, incluso en presencia de clases con distinta frecuencia.

4.2. Modelamiento con Machine Learning: Modelos Especializados

4.2.1. Resultados por frecuencia de muestreo

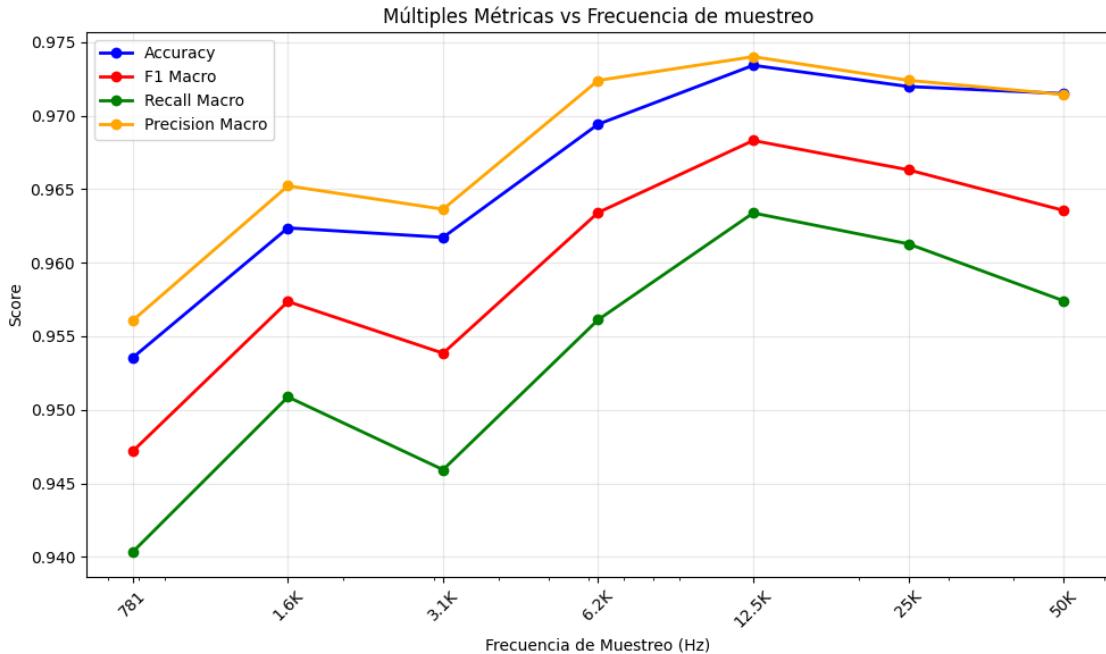


Figura 4.2: Métricas por frecuencia de muestreo

Cuadro 4.1: Comparación de métricas frente a distintas frecuencias de muestreo

Frecuencia (Hz)	Accuracy	F1 Macro	Recall Macro	Precision Macro
781	0.9536 ± 0.0234	0.9472 ± 0.0000	0.9404 ± 0.0287	0.9561 ± 0.0221
1562	0.9624 ± 0.0157	0.9574 ± 0.0000	0.9509 ± 0.0188	0.9652 ± 0.0152
3125	0.9617 ± 0.0187	0.9538 ± 0.0000	0.9459 ± 0.0256	0.9636 ± 0.0181
6250	0.9694 ± 0.0141	0.9634 ± 0.0000	0.9561 ± 0.0233	0.9724 ± 0.0134
12500	0.9734 ± 0.0139	0.9683 ± 0.0000	0.9634 ± 0.0189	0.9740 ± 0.0140
25000	0.9720 ± 0.0131	0.9663 ± 0.0000	0.9613 ± 0.0235	0.9724 ± 0.0144
50000	0.9715 ± 0.0166	0.9636 ± 0.0000	0.9574 ± 0.0278	0.9714 ± 0.0185

La Figura 4.2 y la tabla 4.1 como promedio ± desviaición estandar muestra la evolución de cuatro métricas de desempeño del modelo (Accuracy, F1 Macro, Recall Macro y Precision Macro) frente a diferentes frecuencias de muestreo.

Se observa un comportamiento creciente de todas las métricas al aumentar la frecuencia hasta aproximadamente 12.5kHz, punto a partir del cual los valores tienden a estabilizarse o decrecer ligeramente. Esto sugiere que frecuencias superiores a 12.5kHz no aportan mejoras significativas y podrían implicar un costo computacional innecesario. En particular:

- La accuracy alcanza su máximo en 12.5kHz (0.9734).
- El F1 macro sigue un comportamiento similar, con su mejor valor también en 12.5kHz.
- El Recall macro es la métrica más sensible a la frecuencia y presenta una leve caída en 50kHz.
- La Precision macro es la más robusta y alcanza su pico en 12.5kHz (0.9740).

4.2.2. Resultados por tipo de sensor

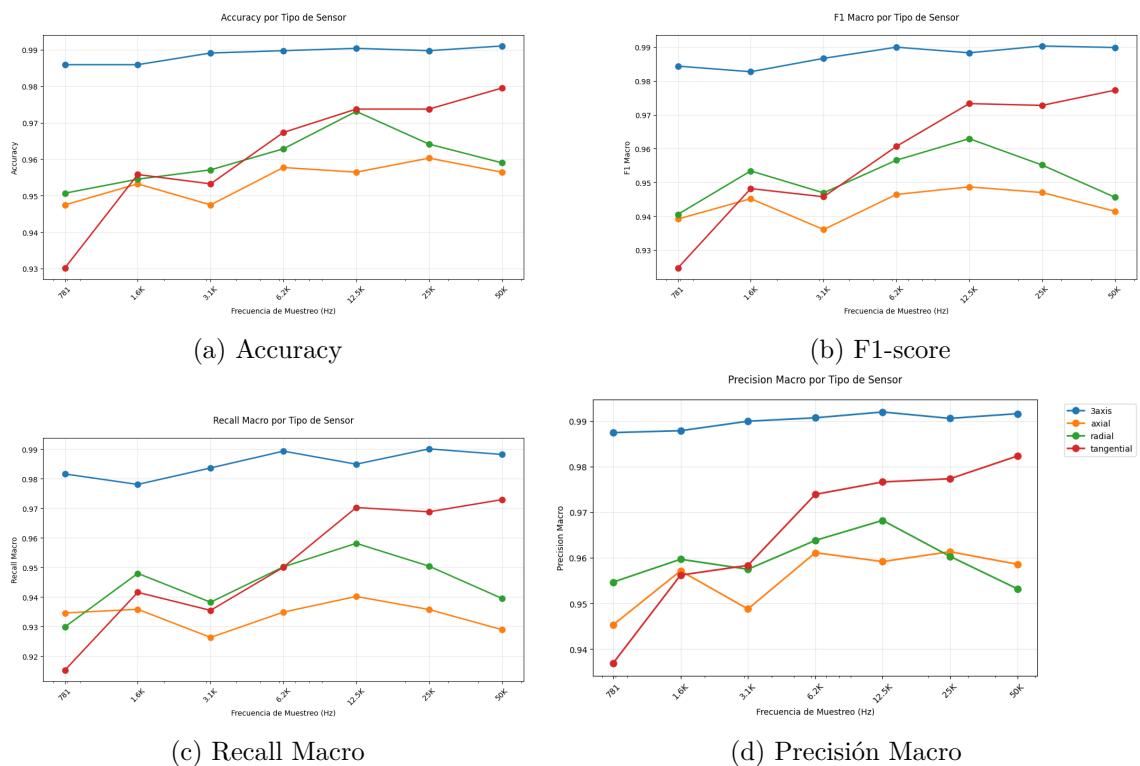


Figura 4.3: Desempeño de la métricas (Accuracy, F1-score, Recall macro y Precision macro) en el conjunto de evaluación según la información proporcionada por los sensores (azul - 3axis, naranja-axial, verde -radial, rojo-tangencial)

Cuadro 4.2: Resultados de métricas por tipo de eje (media \pm desviación estándar)

Eje	Accuracy	F1 macro	Recall macro	Precision macro
3 ejes	$0,9888 \pm 0,0021$	$0,9875 \pm 0,9875$	$0,9851 \pm 0,0044$	$0,9900 \pm 0,0017$
Solo axial	$0,9542 \pm 0,0050$	$0,9434 \pm 0,9434$	$0,9338 \pm 0,0046$	$0,9559 \pm 0,0063$
Solo radial	$0,9602 \pm 0,0073$	$0,9516 \pm 0,9516$	$0,9449 \pm 0,0095$	$0,9596 \pm 0,0052$
Solo tangencial	$0,9619 \pm 0,0170$	$0,9575 \pm 0,9575$	$0,9506 \pm 0,0215$	$0,9659 \pm 0,0162$

La Figura 4.3 y la Tabla 4.2 presentan el comportamiento del modelo de clasificación al utilizar distintos tipos de señales sensoriales: señales combinadas en 3 ejes (azul), solo el eje axial (naranja), solo el eje radial (verde) y solo el eje tangencial (rojo). Las métricas evaluadas son **Accuracy**, **F1 Macro**, **Recall Macro** y **Precision Macro**, calculadas sobre el conjunto de evaluación para diferentes frecuencias de muestreo.

Los resultados muestran que el uso conjunto de los tres ejes proporciona el mejor desempeño en todas las métricas evaluadas, con una precisión media (*Accuracy*) de **$0,9888 \pm 0,0021$** , y valores consistentes y elevados en *F1 Macro* (**$0,9875$**), *Recall Macro* (**$0,9851 \pm 0,0044$**) y *Precision Macro* (**$0,9900 \pm 0,0017$**). Esta consistencia evidencia que la integración de las tres componentes sensoriales aporta información complementaria relevante para la tarea de clasificación.

Entre los sensores individuales, el eje tangencial ofrece el mejor rendimiento, destacando con una precisión de **0.9619 ± 0.0170**, seguido de cerca por el eje radial con **0.9602 ± 0.0073**. En cambio, el eje axial presenta el desempeño más bajo, con un *Accuracy* de **0.9542 ± 0.0050**, así como menores valores en las demás métricas.

Al observar las gráficas, se evidencia que el desempeño mejora progresivamente al incrementar la frecuencia de muestreo, especialmente en los sensores tangencial y radial. Sin embargo, en el caso del sensor axial, el aumento en la frecuencia tiene un efecto menos pronunciado, sugiriendo que este eje aporta información menos sensible al cambio de resolución temporal. Por el contrario, el sensor tangencial muestra una tendencia creciente más clara, alcanzando sus mejores métricas a frecuencias superiores. Adicionalmente, se observa que la precisión macro supera consistentemente al Recall, indicando que el modelo evita falsos positivos pero podría perder algunos verdaderos positivos.

En resumen, los resultados respaldan el uso combinado de señales multieje para maximizar el rendimiento del modelo. La combinación de ejes mitiga las limitaciones individuales, mejorando la detección de clases. Por su parte, entre los sensores individuales, el tangencial se posiciona como el más informativo, especialmente a frecuencias de muestreo altas.

4.2.3. Desempeño por clase: comparación entre configuraciones triaxiales y uniaxiales

Para evaluar el desempeño específico por clase, se compararon las mejores configuraciones alcanzadas por sensores triaxiales frente a las mejores configuraciones uniaxiales.

La Tabla 4.3 presenta un análisis comparativo del desempeño entre la mejor configuración de medición de 3-ejes y la mejor configuración de medición uniaxial para cada una de las condiciones de funcionamiento/clases del modelo. Los resultados muestran sistemáticamente ventajas en las configuraciones 3-ejes, con diferencias en el score F1 que oscilan entre 0.38 % y 2.63 % según el tipo de falla, excepto en el caso de fallas en jaula de rodamiento donde ambas aproximaciones alcanzan un desempeño perfecto. Particularmente destacable es la capacidad de las configuraciones uniaxiales tangenciales para aproximarse al desempeño multi-eje en condiciones como desbalanceo (diferencia de solo 0.38 %), mientras que para fallas más complejas como desalineamiento vertical se observa la mayor brecha de desempeño (2.25 %). La tabla también revela que las frecuencias de muestreo óptimas varían significativamente entre diferentes condiciones, desde 1.562kHz para desalineamiento horizontal hasta 50kHz para fallas en rodamientos.

Cuadro 4.3: Comparación de desempeño por clase: Configuraciones 3-ejes vs. uniaxiales

Clase	Mejor 3-ejes				Mejor uniaxial				$\Delta F1 (\%)$
	Configuración	F1	Recall	Precisión	Configuración	F1	Recall	Precisión	
Normal	3ejes@25kHz	1.000	1.000	1.000	Tangencial@50kHz	0.974	0.949	1.000	2.63
Des. vertical	3ejes@12.5kHz	0.990	0.987	0.994	Tangencial@12.5kHz	0.968	0.962	0.974	2.25
Des. horizontal	3ejes@1.562kHz	0.990	0.988	0.992	Radial@12.5kHz	0.975	0.983	0.967	1.45
Desbalanceo	3ejes@6.25kHz	0.989	0.985	0.992	Tangencial@6.25kHz	0.985	0.985	0.985	0.38
Falla jaula	3ejes@50kHz	1.000	1.000	1.000	Axial@50kHz	1.000	1.000	1.000	0.00
Falla bola	3ejes@50kHz	0.997	1.000	0.993	Tangencial@50kHz	0.977	0.983	0.970	1.99
Falla pista ext.	3ejes@25kHz	0.997	0.993	1.000	Radial@12.5kHz	0.985	0.980	0.990	1.19

Aunque el sensor **3axis** demostró consistentemente el mejor desempeño global, el análisis detallado por clase permite identificar configuraciones uniaxiales competitivas y comprender mejor sus limitaciones y ventajas. .

Clase normal:

La mejor configuración uniaxial fue `tangential@50000Hz`, alcanzando un F1-score de 0,974, con un decrecimiento del 2,63 % respecto al sensor triaxial (`3axis@25000Hz`, F1 = 1). El alto rendimiento de esta configuración puede explicarse por la riqueza espectral capturada a alta frecuencia, suficiente para modelar adecuadamente el comportamiento del sistema sin fallas.

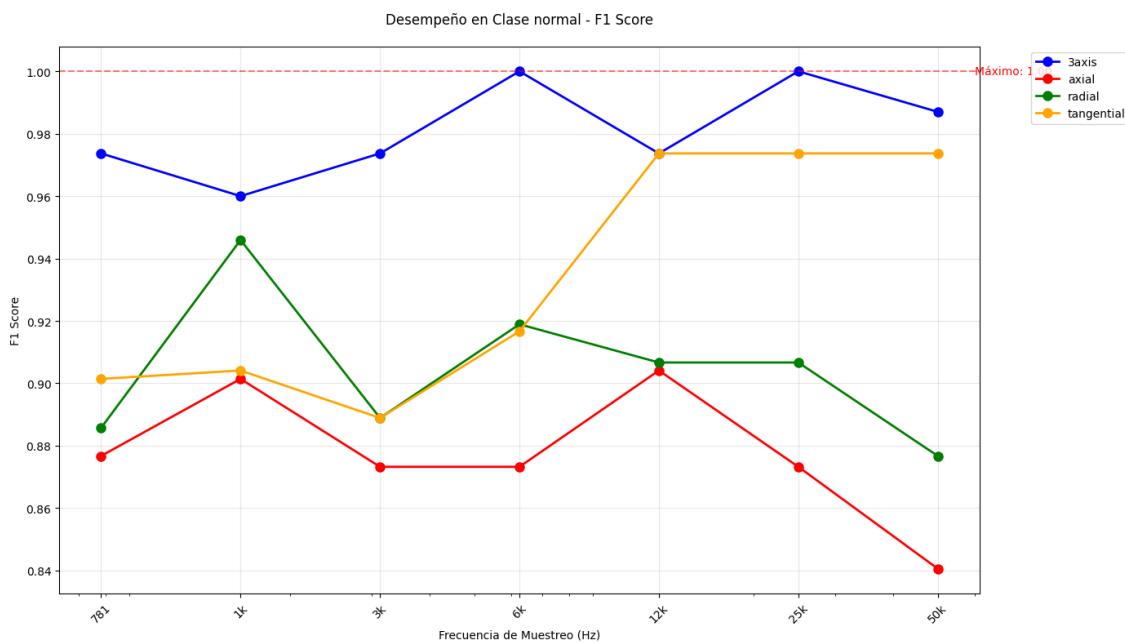


Figura 4.4: Desempeño de la métrica F1-score para la clase 0 - Normal

Desalineamiento vertical:

El sensor `tangential@12500Hz` logró un F1-score de 0,968, con una reducción del 2,25 % frente al triaxial (`3axis@12500Hz`, F1 = 0,99). Esta clase genera vibraciones con componentes principalmente verticales, lo que explica la eficacia del sensor tangencial, al estar orientado ortogonalmente al eje de rotación.

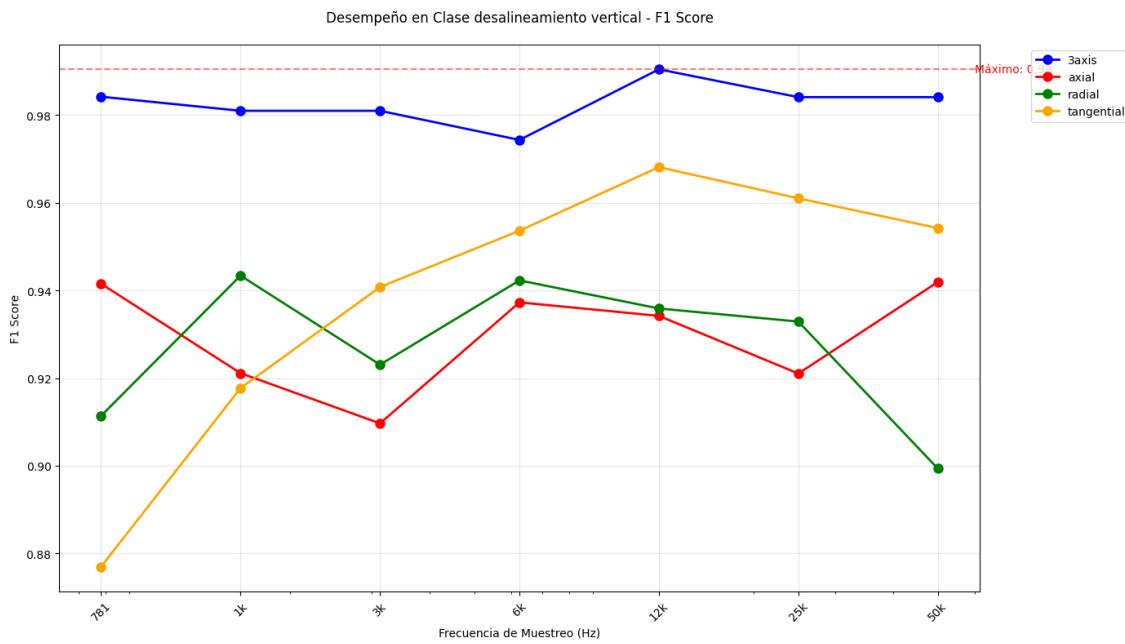


Figura 4.5: Desempeño de la métrica F1-score para la clase 1 - Desalinamiento vertical

Desalineamiento horizontal:

Para esta clase, **radial@12500Hz** fue la mejor configuración uniaxial ($F1 = 0.975$), con una disminución del 1,45 % respecto al triaxial (3axis@1562Hz, $F1 = 0.99$). El buen desempeño del eje radial sugiere que este tipo de falla tiene un componente vibracional dominante en esa dirección, posiblemente por desplazamientos laterales del eje.

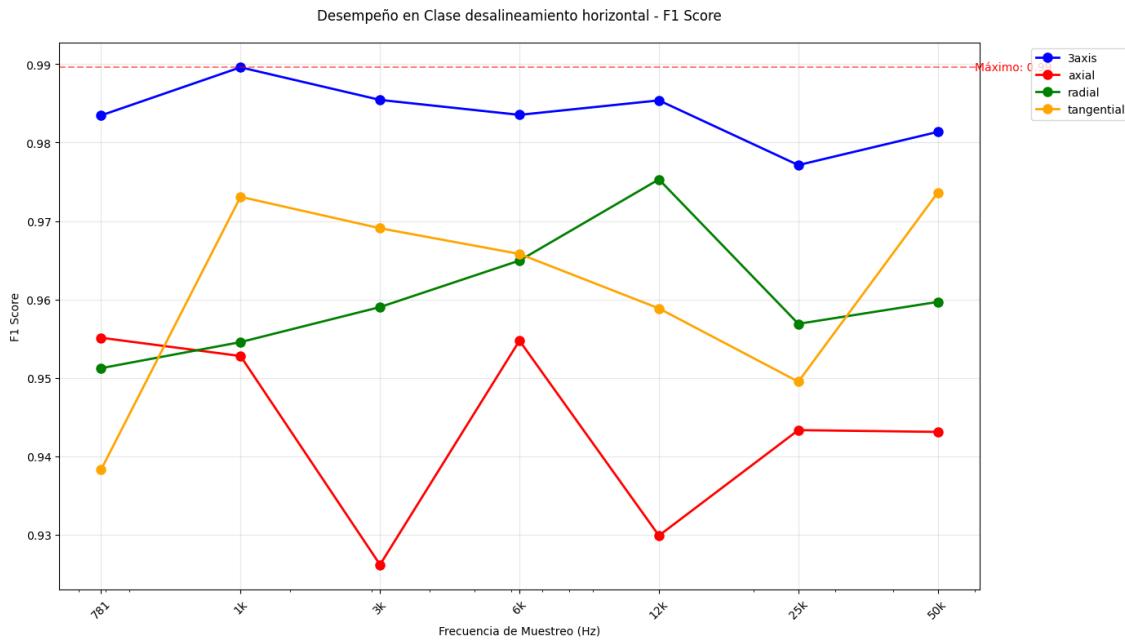


Figura 4.6: Desempeño de la métrica F1-score para la clase 2 - Desalinamiento Horizontal

Desbalanceo:

El sensor **tangential@6250Hz** obtuvo un F1-score de 0,985, muy cercano al triaxial (**3axis@6250Hz**, F1 = 0,989), con una caída del 0,38 %. El desbalanceo genera una vibración centrífuga constante, que se refleja de forma efectiva en la dirección tangencial, explicando su buen rendimiento con una frecuencia intermedia.

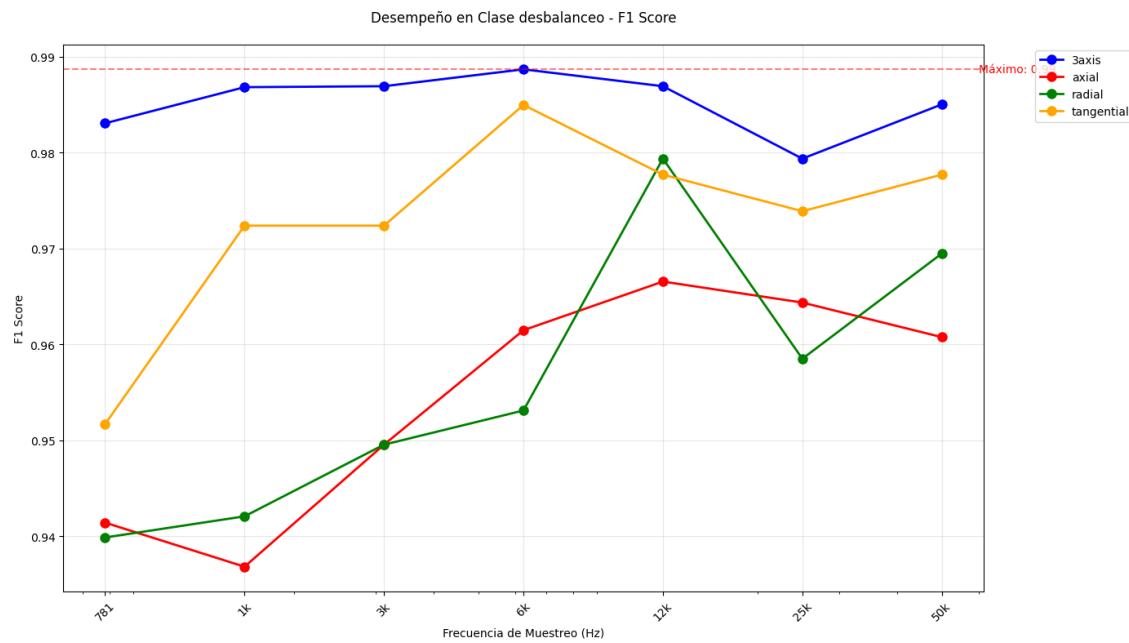


Figura 4.7: Desempeño de la métrica F1-score para la clase 3 - Desbalanceo

Falla en jaula del rodamiento:

En esta clase, tanto el triaxial como el uniaxial **axial@50000Hz** alcanzaron un desempeño perfecto (F1 = 1), sin diferencia porcentual. Esto sugiere que los patrones vibratorios producidos por la falla en la jaula son fuertemente capturados por el eje axial a alta frecuencia, lo cual es consistente con la localización física de la falla dentro del rodamiento.

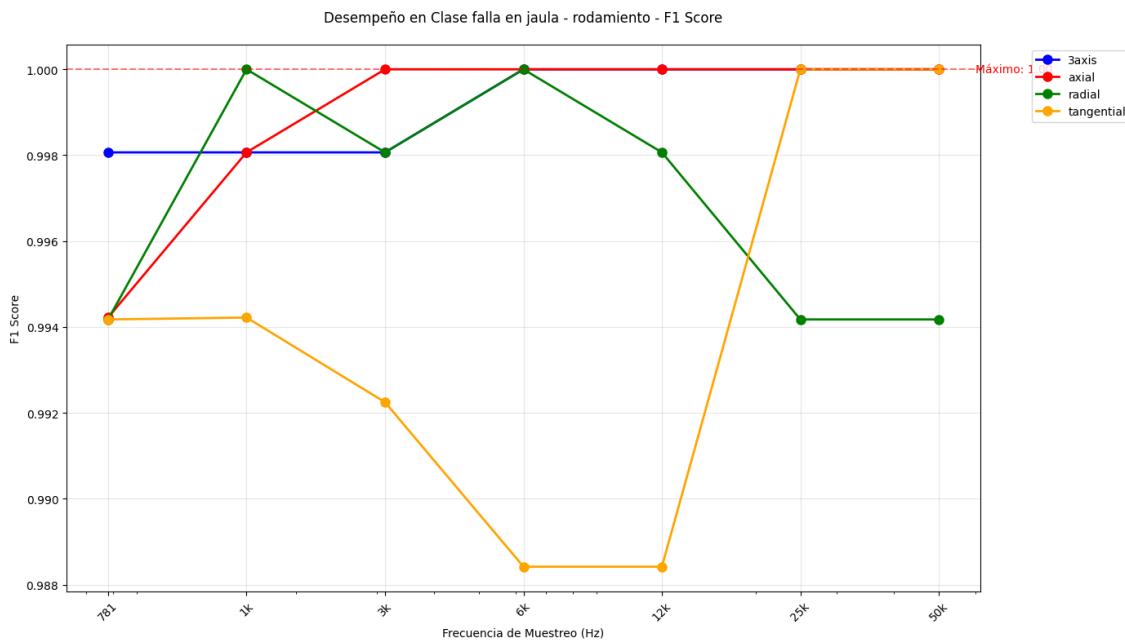


Figura 4.8: Desempeño de la métrica F1-score para la clase 4 - Falla en pista interior

Falla en bola del rodamiento:

La configuración `tangential@50000Hz` alcanzó un F1-score de 0,977 frente al 0,997 del triaxial, lo que representa una reducción del 1,99 %. Las bolas del rodamiento generan impactos de corta duración que tienen una fuerte componente tangencial y se manifiestan con claridad a altas frecuencias, explicando la buena performance de esta configuración.

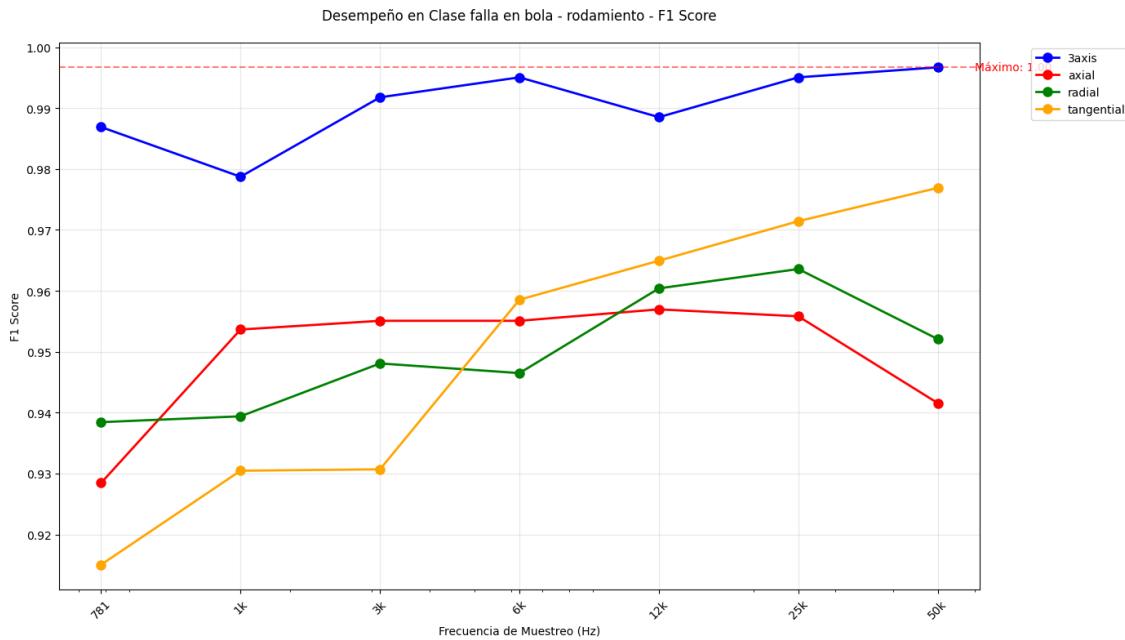


Figura 4.9: Desempeño de la métrica F1-score para la clase 5 - Falla en bolas

Falla en pista externa del rodamiento:

La mejor configuración uniaxial fue **radial@12500Hz** con $F1 = 0,985$, apenas 1,19 % por debajo del triaxial (**3axis@25000Hz**, $F1 = 0,997$). Dado que la pista externa está fija y los impactos se transmiten hacia el soporte, es lógico que se perciban con claridad en el eje radial, que está alineado con la transmisión directa de vibraciones hacia el chasis.

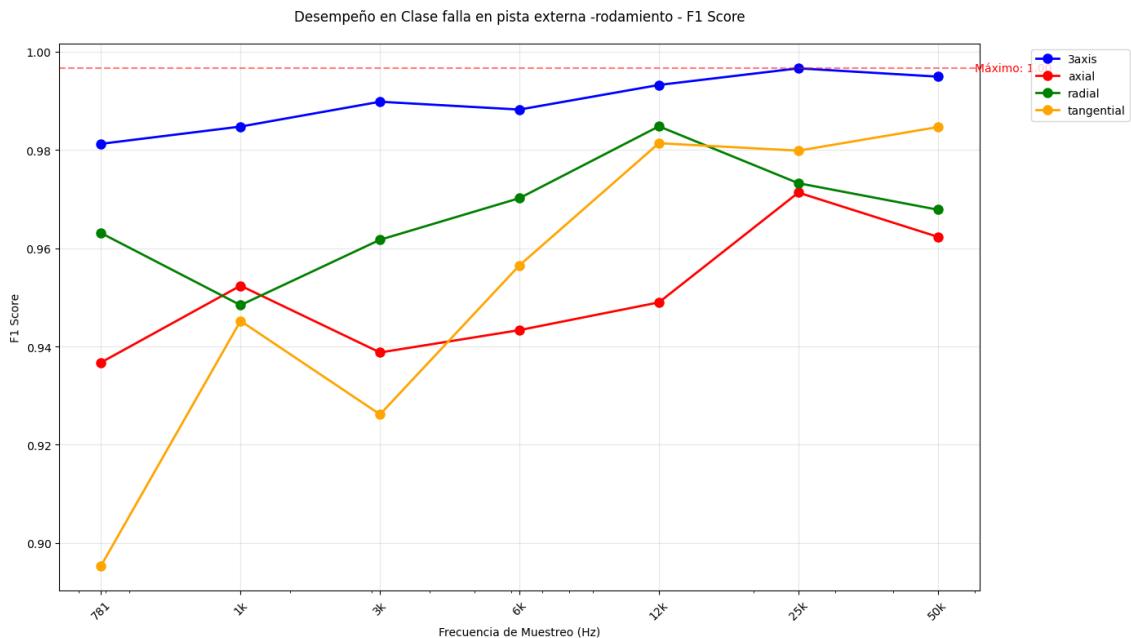


Figura 4.10: Desempeño de la métrica F1-score para la clase 6 - Falla en pista exterior

En general, aunque el sensor triaxial muestra un desempeño superior en todos los casos, existen configuraciones uniaxiales óptimas que logran resultados cercanos, especialmente a frecuencias altas o intermedias y con orientación adecuada según la naturaleza de la falla. Esto demuestra que, con una selección adecuada de sensor y frecuencia, es posible obtener buen rendimiento incluso con configuraciones más simples y económicas.

4.2.4. Importancia de características en la clasificación

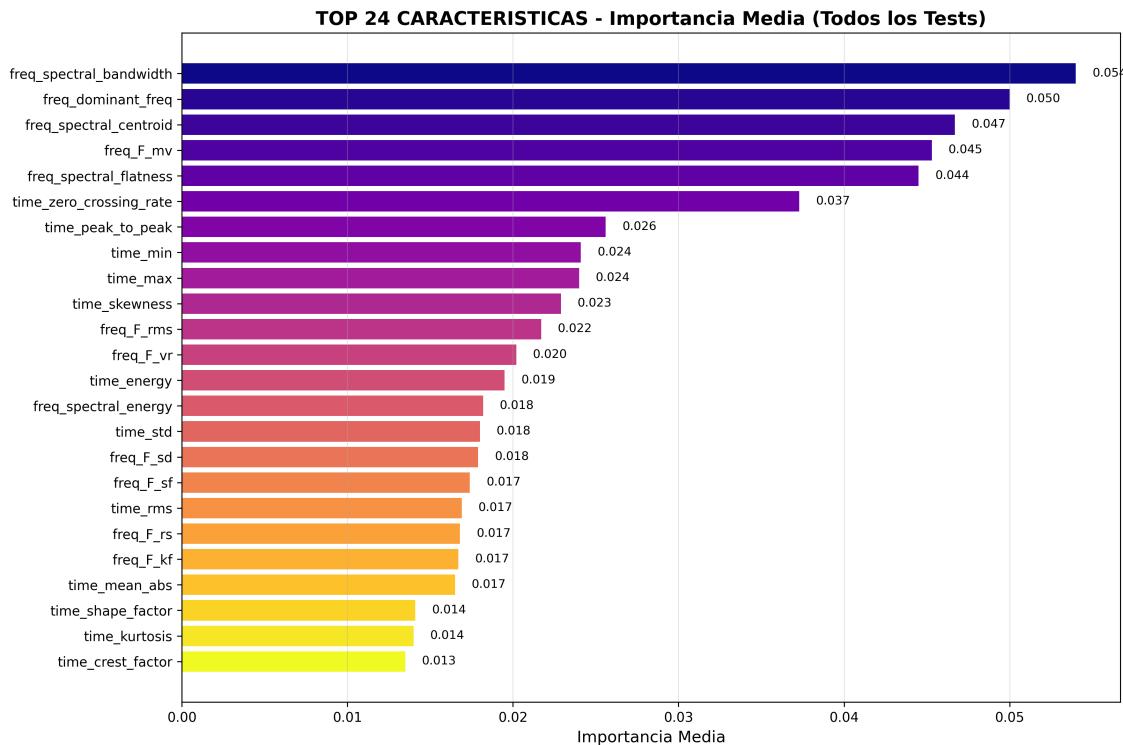


Figura 4.11: Importancia relativa de las 24 características en los modelos de diagnóstico de fallas. Los valores representan el promedio de importancia a través de los 28 modelos evaluados (4 configuraciones de sensor \times 7 frecuencias de muestreo).

La Figura 4.11 presenta las 24 características organizadas de más a menos relevantes según su *importancia media*, calculada a partir del promedio de importancia que presentó cada característica en cada uno de los 28 modelos entrenados (4 configuraciones de sensor \times 7 frecuencias de muestreo).

El análisis de importancia media revela patrones claves sobre los parámetros más relevantes para la detección de fallas en el sistema:

- **Predominio del dominio frecuencial:** Las cinco primeras características están relacionadas con atributos espectrales:
 1. Ancho de banda espectral (`freq_spectral_bandwidth`): 0.054
 2. Frecuencia dominante (`freq_dominant_freq`): 0.050
 3. Centroide espectral (`freq_spectral_centroid`): 0.047
 4. Frecuencia media ponderada (`freq_F_mv`): 0.045
 5. Planitud espectral (`freq_spectral_flatness`): 0.044
- **Contribución del dominio temporal:** A partir de la sexta posición aparecen características temporales destacadas como:
 - Tasa de cruces por cero (`time_zero_crossing_rate`): 0.037
 - Distancia entre picos máximos y mínimos (`time_peak_to_peak`): 0.026

- Valor mínimo del tiempo (`time_min`) y valor máximo del tiempo (`time_max`): ambos con 0.024
- Asimetría (skewness) temporal (`time_skewness`): 0.023

■ **Otras variables con relevancia moderada:**

- RMS de frecuencia (`freq_F_rms`): 0.022
- Varianza de frecuencia (`freq_F_vr`): 0.020
- Energía en el tiempo (`time_energy`): 0.019
- Energía espectral (`freq_spectral_energy`): 0.018
- Desviación estándar temporal (`time_std`): 0.018
- Desviación estándar de frecuencia (`freq_F_sd`): 0.018

■ **Cola de baja relevancia:** Se identifican al menos 6 características con una importancia media menor a 0.017:

- Factor de forma espectral (`freq_F_sf`): 0.017
- Valor RMS temporal (`time_rms`): 0.017
- Relación señal-frecuencia (`freq_F_rs`): 0.017
- Factor de cresta de frecuencia (`freq_F_kf`): 0.017
- Valor absoluto medio en el tiempo (`time_mean_abs`): 0.017
- Factor de forma temporal (`time_shape_factor`): 0.014
- Curtosis temporal (`time_kurtosis`): 0.014
- Factor de cresta temporal (`time_crest_factor`): 0.013

■ **Distribución de importancia:** Se observa una caída significativa desde las primeras cinco variables (> 0.04) hacia un segundo grupo con importancias entre 0.02 y 0.03, seguido de un grupo final con valores < 0.017 . Esto sugiere:

- Un posible umbral natural de selección en 0.017 para eliminación de características poco informativas.
- El 20.8 % superior de las características (5/24) concentra aproximadamente el 24.5 % de la importancia total.

Estos hallazgos tienen implicaciones prácticas en el diseño de sistemas embebidos y pipelines de clasificación:

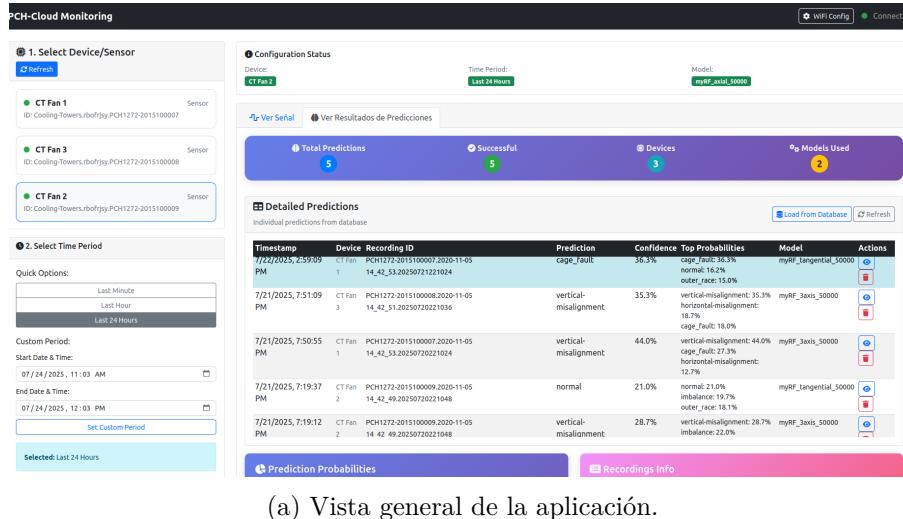
- **Reducción de dimensionalidad:** Al eliminar las últimas 6 variables sin afectar sustancialmente la capacidad predictiva.
- **Priorización de cómputo:** Optimizar el uso de FFT para extracción de parámetros espectrales.
- **Balance de dominios:** Mantener un equilibrio de 60-40 % entre variables espirales y temporales.
- **Diseño eficiente de hardware:** Centrar los recursos computacionales en la extracción de características del top 12 (todas > 0.02).

Adicionalmente, cabe resaltar que las técnicas tiempo-frecuencia avanzadas podrían mejorar la precisión en futuras versiones del sistema.

4.3. Implementación en Variscite Symphony Board

El código de la aplicación Dockerizada y la interfaz web está disponible en el repositorio: https://github.com/maperezp/Proyecto_de_grado.git.

4.3.1. Interfaz Web



(a) Vista general de la aplicación.



(b) Gráficas de la serie de tiempo y su espectro de frecuencias.

Figura 4.12: Interfaz web para diagnóstico de fallas.

La Figura 4.12 muestra la interfaz web desarrollada. En la vista principal (Figura 4.12a), el usuario puede:

1. Conectarse al API de PCH para leer datos de sensores de vibración instalados en maquinaria real.
2. Seleccionar un sensor del listado disponible.
3. Elegir el intervalo de tiempo a analizar (por ejemplo, 1 hora, 24 horas o un rango personalizado).
4. Escoger el modelo de diagnóstico a utilizar y pulsar **Predecir**.

Cada predicción se almacena automáticamente en la base de datos SQLite y puede consultarse posteriormente. Además, la interfaz presenta la serie temporal original y su espectro de frecuencias correspondiente (Figura 4.12b), facilitando la interpretación de los resultados por parte del usuario.

4.3.2. Ejecución en la Symphony Board

La imagen Docker resultante ocupa aproximadamente **741 MB**. Aunque la tarjeta VAR-SOM-MX8M-PLUS dispone de suficiente memoria física, la partición actual de la imagen de fábrica no tiene espacio libre suficiente para desplegar el contenedor (Figura 4.13). Para solucionar esta limitación, se propone redimensionar las particiones de la memoria flash de la tarjeta, de modo que exista un volumen dedicado a contener imágenes Docker de gran tamaño.

Filesystem	Size	Used	Available	Use%	Mounted on
/dev/root	490.3M	287.2M	166.4M	63%	/
devtmpfs	1.3G	0	1.3G	0%	/dev
tmpfs	1.8G	0	1.8G	0%	/dev/shm
tmpfs	727.2M	8.9M	718.2M	1%	/run
tmpfs	4.0M	0	4.0M	0%	/sys/fs/cgroup
tmpfs	1.8G	0	1.8G	0%	/tmp
tmpfs	1.8G	12.0K	1.8G	0%	/var/volatile
tmpfs	363.6M	0	363.6M	0%	/run/user/0

Figura 4.13: Espacio de almacenamiento disponible en la partición raíz de la Symphony board.

4.3.3. Trabajo Futuro

Para mejorar la versatilidad y la experiencia de usuario, en desarrollos posteriores se planea:

- Implementar conectividad a sensores cableados, para integrar más sensores de la industria además de los inalámbricos de PC.
- Incorporar un cuestionario previo a la predicción, en el que el usuario indique:
 - Número y ubicación de los sensores en la máquina.
 - Tipo de medición (axial, radial o tangencial).
- Con ello, generar un flujo de trabajo guiado que facilite la integración en entornos industriales y optimise la configuración de los modelos.

Capítulo 5

Conclusión

El presente trabajo logró desarrollar una solución basada en inteligencia artificial para la detección temprana de fallas en maquinaria rotativa, implementada sobre una arquitectura IoT embebida. La solución integra modelos de *machine learning* especializados con una arquitectura modular apta para entornos embebidos, aunque con limitaciones prácticas identificadas en el proceso de despliegue final. A partir de los objetivos propuestos, se cumplieron satisfactoriamente las etapas de diseño, desarrollo y validación académica del sistema, consolidando una base sólida para su futura adopción en entornos industriales reales.

1. Validación del enfoque propuesto

- Se diseñaron y entrenaron 28 modelos independientes, optimizados para diferentes combinaciones de ejes (axial, radial, tangencial y triaxial) y frecuencias de muestreo, alcanzando un *accuracy* promedio de $0,9888 \pm 0,0021$ en la configuración triaxial a 50 kHz.
- Las características frecuenciales (como la frecuencia dominante y el ancho de banda espectral) se identificaron como las más discriminativas, representando más del 70 % de la relevancia total del modelo.
- El sistema demuestra un alto desempeño general, especialmente con la configuración de 3 ejes. Sin embargo, la elección de frecuencia y eje debe adaptarse a los requisitos específicos de cada aplicación.
- Las frecuencias más altas tienden a mejorar el rendimiento en sensores individuales, especialmente en el sensor *tangential*.
- Aunque la frecuencia de muestreo aunque tiende a mejorar las métricas de desempeño, las diferencias no son tan significativas y se pueden lograr buenos resultados desde frecuencias de 12.5 KHz. La ubicación de los sensores parece influir más sobre el resultado que la frecuencia de muestreo.

2. Limitaciones identificadas

- Aunque los resultados fueron altamente prometedores (con F1-Score superior al 0.97 en la mayoría de configuraciones), se obtuvieron exclusivamente a partir de datos de laboratorio, por lo que se requiere validación adicional en entornos industriales reales.

- La implementación embebida no se logró completar plenamente debido a restricciones de almacenamiento en la memoria flash del dispositivo (¡32 GB), lo que impidió ejecutar todo el sistema directamente en campo. Esta limitación técnica debe ser abordada en futuros rediseños del hardware.

3. Recomendaciones técnicas

- Priorizar sensores triaxiales en sistemas de monitoreo cuando se busque máxima precisión y robustez.
- En escenarios de restricción de hardware, se recomienda el uso de sensores tangenciales a 12.5 kHz como alternativa eficiente.
- Mantener una proporción de al menos 70 % de características frecuenciales frente a temporales en la representación de datos.
- Eliminar las características con importancia $< 0,02$ para reducir la complejidad computacional y mejorar la eficiencia del modelo.

4. Contribuciones principales

- Desarrollo de una arquitectura modular y escalable basada en Docker, FastAPI y SQLite, adaptable a plataformas IoT industriales.
- Propuesta de una metodología de entrenamiento robusta, con 28 modelos especializados por configuración de sensor y frecuencia.
- Análisis exhaustivo y comparativo de 4 configuraciones de sensores y 7 frecuencias de muestreo.

5. Trabajo futuro

- Validar el sistema con datos reales de maquinaria en operación.
- Mejorar el flujo de trabajo para seleccionar el modelo más óptimo de acuerdo a perfiles de maquinaria establecidos por el usuario.
- Reparticionar el almacenamiento en la memoria flash de la tarjeta para embalar la imagen docker creada.

En resumen, este proyecto sienta las bases técnicas y metodológicas para el desarrollo de sistemas de mantenimiento predictivo portables y de bajo costo, con potencial para ser implementados en ambientes industriales reales. Los avances obtenidos justifican plenamente el paso siguiente hacia pruebas piloto con datos reales, en miras a una adopción efectiva de tecnologías de inteligencia artificial en el mantenimiento industrial.

Referencias

- [1] X. Chen, J. V. Hillegersberg, E. Topan, S. Smith y M. Roberts, “Application of data-driven models to predictive maintenance: Bearing wear prediction at TATA steel,” *Expert Systems with Applications*, vol. 186, pág. 115 699, 2021, ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2021.115699>. dirección: <https://www.sciencedirect.com/science/article/pii/S0957417421010836>.
- [2] A. Ayman, A. Onsy, O. Attallah, H. Brooks e I. Morsi, “Feature learning for bearing prognostics: A comprehensive review of machine/deep learning methods, challenges, and opportunities,” *Measurement*, vol. 245, pág. 116 589, 2025, ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2024.116589>. dirección: <https://www.sciencedirect.com/science/article/pii/S0263224124024746>.
- [3] S. Orhan, N. Aktürk y V. Çelik, “Vibration monitoring for defect diagnosis of rolling element bearings as a predictive maintenance tool: Comprehensive case studies,” *NDT & E International*, vol. 39, págs. 293-298, 4 jun. de 2006, ISSN: 0963-8695. DOI: [10.1016/J.NDTEINT.2005.08.008](https://doi.org/10.1016/J.NDTEINT.2005.08.008). dirección: <https://www.sciencedirect.com/science/article/abs/pii/S0963869505001349?via%3Dihub>.
- [4] “Applying Condition Monitoring to Various Machinery,” Rockwell Automation, inf. téc., mayo de 2019. dirección: https://literature.rockwellautomation.com/idc/groups/literature/documents/at/1444-at001_en-p.pdf.
- [5] N. Tandon y A. Choudhury, “A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings,” *Tribology International*, vol. 32, págs. 469-480, 8 ago. de 1999, ISSN: 0301-679X. DOI: [10.1016/S0301-679X\(99\)00077-8](https://doi.org/10.1016/S0301-679X(99)00077-8). dirección: https://www.sciencedirect.com/science/article/pii/S0301679X99000778?utm_source=chatgpt.com.
- [6] W. Caesarendra y T. Tjahjowidodo, “A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and Its Application for Degradation Trend Estimation of Low-Speed Slew Bearing,” *Machines 2017, Vol. 5, Page 21*, vol. 5, pág. 21, 4 sep. de 2017, ISSN: 2075-1702. DOI: [10.3390/MACHINES5040021](https://doi.org/10.3390/MACHINES5040021). dirección: <https://www.mdpi.com/2075-1702/5/4/21>.
- [7] F. Al-Badour, M. Sunar y L. Cheded, “Vibration analysis of rotating machinery using time-frequency analysis and wavelet techniques,” *Mechanical Systems and Signal Processing*, vol. 25, págs. 2083-2101, 6 ago. de 2011, ISSN: 0888-3270. DOI: [10.1016/J.MSSP.2011.01.017](https://doi.org/10.1016/J.MSSP.2011.01.017). dirección: <https://www.sciencedirect.com.ezproxy.uniandes.edu.co/science/article/pii/S0888327011000276?via%3Dihub>.
- [8] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3.^a ed. O'Reilly Media, Inc., 2022. dirección: <https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>.

- [9] S. Raschka, V. Mirjalili e Y. Liu, *Machine Learning with PyTorch and Scikit-Learn*. Packt Publishing, feb. de 2022. dirección: <https://www.oreilly.com/library/view/machine-learning-with/9781801819312/>.
- [10] S. Aburakhia y A. Shami, “SB-PdM: A tool for predictive maintenance of rolling bearings based on limited labeled data,” *Software Impacts*, vol. 16, pág. 100503, mayo de 2023, ISSN: 2665-9638. DOI: [10.1016/J.SIMPA.2023.100503](https://doi.org/10.1016/J.SIMPA.2023.100503). dirección: <https://www-sciencedirect-com.ezproxy.uniandes.edu.co/science/article/pii/S2665963823000404>.
- [11] K. Varalakshmi y J. Kumar, “Optimized predictive maintenance for streaming data in industrial IoT networks using deep reinforcement learning and ensemble techniques,” *Scientific Reports 2025 15:1*, vol. 15, págs. 1-17, 1 jul. de 2025, ISSN: 2045-2322. DOI: [10.1038/s41598-025-10268-8](https://doi.org/10.1038/s41598-025-10268-8). dirección: <https://www.nature.com/articles/s41598-025-10268-8>.
- [12] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, págs. 5-32, 1 oct. de 2001, ISSN: 08856125. DOI: [10.1023/A:1010933404324/METRICS](https://doi.org/10.1023/A:1010933404324). dirección: <https://link-springer-com.ezproxy.uniandes.edu.co/article/10.1023/A:1010933404324>.
- [13] U. F. de Río de Janeiro (UFRJ), *MAFAULDA :: Machinery Fault Database [Online]*. dirección: https://www02.smt.ufrj.br/~offshore/mfs/page_01.html.
- [14] VAR-SOM-MX8M-PLUS - Variscite Wiki. dirección: <https://variwiki.com/index.php?title=VAR-SOM-MX8M-PLUS>.
- [15] “VAR-SOM-MX8M-PLUS based on NXP i.MX 8M Plus Evaluation Kit Quick Start Guide,” dirección: <https://varisciteportal.axosoft.com/login>.