EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación y Desarrollo Tecnológico

Tesis de Doctorado

Mantenimiento Predictivo Utilizando Machine Learning

presentada por

MC. Jose Luis Molina Salgado

como requisito para la obtención del grado de

Doctor en Ciencias de la Computación

Director de tesis

Dr. Máximo López Sánchez

Codirector de tesis

Dr. René Santaolaya Salgado

Cuernavaca, Morelos, México. Febrero de 2025.

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

cenidet® Centro Nacional de Investigación y Desarrollo Tecnológico	ACEPTACIÓN DE IMPRESIÓN DEL DOCUMENTO DE TESIS DOCTORAL	Código: CENIDET-AC-006-D20
		Revisión: O
	Referencia a la Norma ISO 9001:2008 7.1, 7.2.1, 7.5.1, 7.6, 8.1, 8.2.4	Página 1 de 1

Cuernavaca, Mor., a 22 de enero de 2025

DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
P R E S E N T E

AT'n: DR. JUAN GABRIEL GONZÁLEZ SERNA
PRESIDENTE DEL CLAUSTRO DOCTORAL DEL DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Los abajo firmantes, miembros del Comité Tutorial del estudiante **M.C. JOSÉ LUIS MOLINA SALGADO** manifiestan que después de haber revisado el documento de tesis titulado **"MANTENIMIENTO PREDICTIVO UTILIZANDO MACHINE LEARNING"**, realizado bajo la dirección del **Dr. Máximo López Sánchez** y codirección del **Dr. René Santaolaya Salgado,** el trabajo se ACEPTA para proceder a su impresión.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al Servicio de México"

DR. MÁXIMO LÓPEZ SÁNCHEZ
TECNM/CENIDET

DR. RENÉ SANTAOLAYA SALGADO
TECNM/CENIDET

DR. JUAN GABRIEL GONZÁLEZ SERNA
TECNM/CENIDET

DR. NOÉ ALEJANDRO CASTRO SÁNCHEZ
TECNM/CENIDET

DRA. OLIVIA GRACIELA FRAGOSO DÍAZ
TECNM/CENIDET

DRA. MARÍA CLARA GÓMEZ ÁLVAREZ
UNIVERSIDAD NACIONAL DE COLOMBIA/MEDELLÍN

c.c.p: C María Elena Gómez Torres/ Jefa del Departamento de Servicios Escolares
c.c.p: C Nóe Alejandro Castro Sánchez / Jefe del Departamento de Ciencias Computacionales
c.c.p: Expediente

Centro Nacional de Investigación y Desarrollo tecnológico
Subdirección Académica

Cuernavaca Mor, 04/febrero/2025
Oficio No. SAC/051/2025
Asunto: Autorización de impresión de tesis

JOSÉ LUIS MOLINA SALGADO
CANDIDATO AL GRADO DE DOCTOR
EN CIENCIAS DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"MANTENIMIENTO PREDICTIVO UTILIZANDO MACHINE LEARNING"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al Servicio de México"

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SUBDIRECCIÓN ACADÉMICA

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

c.c.p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/lmz

Interior Internado Palmira S/N, Col. Palmira,
C. P. 62490, Cuernavaca, Morelos Tel. 01 (777) 3627770, ext. 4104,
e-mail: acad_cenidet@tecnm.mx tecnm.mx | cenidet.tecnm.mx

Agradecimientos

Al programa de becas del Consejo Nacional de Humanidades Ciencias y Tecnologías (CONAHCYT) por brindarme el apoyo económico y poder cumplir una de mis metas en la vida.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por aceptarme en el programa de Doctorado en Ciencias De La Computación.

A mi esposa que me motiva a seguir adelante pase lo que pase, con su amor y cariño.

A mi madre que siempre me ha dado todo su apoyo y cariño incondicional.

A mi hermano que ha estado ahí para aconsejarme cuando lo necesito.

A mi amiga y compañera Karen que nos apoyamos y motivamos a seguir adelante durante este trayecto.

A mi director de tesis, Dr. Máximo López Sánchez por sus consejos, observaciones y por la confianza que siempre ha tenido en mí.

A mi codirector de tesis, Dr. René Santaolaya Salgado por su aportación durante la realización de la tesis.

A mis revisores, Dr. Noé Alejandro Castro Sánchez, Dr. Juan Gabriel González Serna, Dra. Olivia Fragoso Díaz y Dra. María Clara Gómez Álvarez por sus consejos y revisiones.

A excelentes personas que conocí en el centro de investigación tanto de vigilancia, intendencia, administrativos, directivos e investigadores

Abstract

In modern industry, unexpected failures in critical equipment, such as industrial bearings, can lead to high costs and compromise operational efficiency. Predictive maintenance emerges as an essential solution by enabling the anticipation of failures and improving maintenance planning. However, the effectiveness of prediction models based on Machine Learning is limited by the quality of the data used. This thesis aims to address this challenge by proposing a methodology that optimizes the preprocessing of sensor data, considering their origin and operational context.

The thesis is developed through four key stages: identification and collection of relevant data, processing of the information to eliminate noise and highlight useful patterns, iterative adjustment to refine the applied techniques, and validation of predictive models using algorithms such as Decision Trees, Random Forest, and Neural Networks. The data come from vibration sensors selected from a real dataset that reflects the condition of bearings under different samples.

The results show that the proposed approach significantly improves the accuracy and reliability of predictions, allowing companies to reduce operational costs, minimize downtime, and maximize the lifespan of their assets. Furthermore, this thesis offers an adaptable tool that can be explored in other industrial settings, opening possibilities for its implementation and enhancement in broader contexts within predictive maintenance.

Resumen

En la industria moderna, los fallos inesperados en equipos críticos, tales como los rodamientos industriales, pueden generar altos costos y comprometer la eficiencia operativa. El mantenimiento predictivo surge como una solución esencial al permitir anticipar fallas y mejorar la planificación del mantenimiento. Sin embargo, la eficacia de los modelos de predicción basados en *Machine Learning* (ML) está limitada por la calidad de los datos utilizados. Esta tesis busca abordar este desafío al proponer una metodología que optimiza el preprocesamiento de datos sensoriales, considerando sus características de origen y contexto.

El trabajo se desarrolla a través de cuatro etapas clave: identificación y recopilación de datos relevantes, procesamiento de la información para eliminar ruido y resaltar patrones útiles, ajuste iterativo para perfeccionar las técnicas aplicadas y validación de los modelos predictivos con algoritmos tales como: *Decision Trees, Random Forest y Neural Networks*. Los datos utilizados provienen de sensores de vibración seleccionados de un conjunto de datos reales, los que reflejan el estado de los rodamientos en diferentes muestras.

Los resultados obtenidos muestran que el enfoque propuesto mejora notablemente la precisión y fiabilidad de las predicciones, permitiendo a las empresas reducir costos operativos, minimizar tiempos de inactividad y maximizar la vida útil de sus activos. Además, esta tesis ofrece una herramienta adaptable que puede ser explorada en otros entornos industriales, abriendo posibilidades para su implementación y mejora en contextos más amplios dentro del mantenimiento predictivo.

Índice

Índice de Figuras

Índice de Tablas

Glosario de términos

1. **PLC**: Programmable Logic Controller (Controlador Lógico Programable)
2. **IoT**: Internet of Things (Internet de las Cosas)
3. **CC**: Cloud Computing (Computación en la Nube)
4. **CPS**: Cyber-Physical Systems (Sistemas Ciberfísicos)
5. **PdM**: Predictive Maintenance (Mantenimiento Predictivo)
6. **PvM**: Preventive Maintenance (Mantenimiento Preventivo)
7. **R2F**: Run to Failure (Mantenimiento Correctivo/Reparación hasta Falla)
8. **ML**: Machine Learning (Aprendizaje Automático)
9. **AI**: Artificial Intelligence (Inteligencia Artificial)
10. **RUL**: Remaining Useful Life (Vida Útil Restante)
11. **RF**: Random Forest
12. **LoR**: Logistic Regression (Regresión Logística)
13. **SVM**: Support Vector Machines (Máquinas de Vectores de Soporte)
14. **GBT**: Gradient-Boosted Tree (Árbol de Aumento de Gradiente)
15. **LR**: Linear Regression (Regresión Lineal)
16. **NN**: Neural Networks (Redes Neuronales)
17. **FL**: Fuzzy Logic (Lógica Difusa)
18. **BN**: Bayesian Networks (Redes Bayesianas)
19. **DT**: Decision Tree (Árbol de Decisión)
20. **KM**: K-Means
21. **KNN**: K Nearest Neighbor (K Vecino Más Cercano)
22. **CART**: Classification and Regression Trees (Árboles de Clasificación y Regresión)
23. **NB**: Naive Bayes
24. **IMU**: Inertial Measurement Unit (Unidad de Medición Inercial)
25. **GPS**: Global Positioning System (Sistema de Posicionamiento Global)
26. **CNN**: Convolutional Neural Networks (Redes Neuronales Convolucionales)

27. **MSE**: Mean Squared Error (Error Cuadrático Medio)
28. **KF**: Kalman Filter (Filtro de Kalman)
29. **EMD**: Empirical Mode Decomposition (Descomposición en Modos Empíricos)
30. **WT**: Wavelet Transform (Transformada Wavelet)

Capítulo I

Introducción

1. Introducción

La industria ha evolucionado a través de cuatro etapas conocidas como revoluciones industriales. La primera se caracterizó por la introducción de herramientas y la mecanización de procesos manuales de producción. La segunda revolución incorporó la electricidad como fuente de energía y dio paso a la aplicación de cadenas de producción. Posteriormente, la tercera revolución industrial trajo consigo la automatización de sistemas y el uso de tecnologías informáticas. Finalmente, en la actualidad, la industria se encuentra en la cuarta revolución industrial, la que está marcada por la integración de tecnologías como es el Internet de las cosas (IoT), el cómputo en la nube (CC) y los sistemas ciberfísicos (CPS), entre otras.

El mantenimiento predictivo (PdM) se ha consolidado como una estrategia clave en la industria moderna, aprovechando tecnologías avanzadas para anticipar fallas en los equipos y evitar interrupciones en procesos críticos. La integración de técnicas de *Machine Learning* (ML) e Inteligencia Artificial (AI), permiten al PdM mejorar la confiabilidad de los sistemas y la eficiencia operativa, reduciendo significativamente los tiempos de inactividad y prolongando la vida útil de los equipos [1].

Con el creciente uso de sistemas inteligentes y la disponibilidad de datos sensoriales, las técnicas de ML han demostrado ser un enfoque eficaz para mejorar la precisión en la predicción de fallas. Uno de los aspectos más relevantes en la implementación de ML para el mantenimiento predictivo es el preprocesamiento de datos [2]. Esta tesis se centra en el desarrollo de un modelo de mantenimiento predictivo basado en ML para rodamientos industriales, los cuáles son componentes fundamentales en los sistemas de maquinaria rotativa, proponiendo un enfoque de preprocesamiento que va más allá de la simple normalización o escalado de los datos de entrada.

Este tratamiento diferenciado incluye técnicas de filtrado de ruido, segmentación de señales y extracción de características específicas, optimizando la calidad de los datos antes de ser utilizados en los modelos de ML. A diferencia de estudios previos que se enfocan en técnicas básicas de limpieza y normalización, el enfoque desarrollado no solo mejora la precisión del modelo predictivo, sino que también permite una planificación más eficiente y una reducción considerable de los costos asociados a interrupciones imprevistas. La falta de una gestión adecuada de estos componentes puede derivar en pérdidas económicas significativas y riesgos para la seguridad [3]

El caso de estudio de esta tesis son los rodamientos industriales, estos fueron seleccionados debido a su papel crucial en las máquinas rotativas de diversas industrias, tales como la de manufactura, la minería y el transporte. Estos componentes son esenciales para soportar el movimiento rotacional y cualquier falla en ellos puede resultar en paradas no planificadas y elevados costos de mantenimiento. La capacidad de anticipar con precisión los desperfectos permite planificar el mantenimiento con antelación, minimizando el riesgo de interrupciones operativas y mejorando la eficiencia y seguridad de las operaciones [3].

El enfoque propuesto contribuye a reducir los costos asociados a paradas no programadas en la industria, las cuales pueden resultar extremadamente costosas [4].

1.1 Planteamiento del problema

La precisión y fiabilidad de los modelos predictivos en el mantenimiento de rodamientos de máquinas industriales dependen directamente de la calidad de los datos procesados. Sin embargo, las técnicas actuales de preprocesamiento suelen ignorar el origen de los datos sensoriales, lo que conlleva la pérdida de información crítica, como las características del sensor, las condiciones ambientales y el contexto operativo. Esta carencia compromete la capacidad de los modelos para predecir fallas de forma precisa y oportuna.

Por lo tanto, surge la pregunta clave: **¿Cómo mejorar el preprocesamiento de datos sensoriales para evitar que la precisión y fiabilidad de los modelos predictivos en rodamientos industriales se vean afectadas?**

Para abordar esta problemática, es necesario desarrollar métodos de preprocesamiento que integren información del origen de los datos, refinando su calidad y optimizando su representatividad antes de alimentar los algoritmos de aprendizaje automático. Este enfoque busca mejorar la detección de patrones relevantes y la predicción precisa de fallas, maximizando así la vida útil de los equipos y reduciendo los costos operativos asociados a interrupciones no planificadas.

1.2 Hipótesis

A continuación, se muestran las hipótesis planteadas.

Hipótesis 1: El preprocesamiento de datos que considera el origen de los datos (tipo de sensor, condiciones ambientales y contexto de adquisición) mejora significativamente la precisión de los algoritmos de ML en la detección de fallas, en comparación con un preprocesamiento estándar que no tiene en cuenta estas variables.

Hipótesis 2: La implementación de métodos de preprocesamiento específicos para cada origen de datos disminuye el tiempo necesario para la capacitación de modelos predictivos, optimizando el proceso de creación de modelos sin sacrificar precisión.

Hipótesis 3: Un enfoque de preprocesamiento basado en el origen de los datos reduce la complejidad computacional requerida para generar modelos de mantenimiento predictivo, mejorando la eficiencia general del sistema de mantenimiento en la industria.

1.3 Objetivos

En esta sección se muestran los objetivos de la investigación propuesta.

General

- Desarrollar un modelo de mantenimiento predictivo basado en un preprocesamiento de datos que tenga en cuenta la información sobre el origen de los datos (tipo de sensor, condiciones ambientales y contexto de adquisición), para mejorar la precisión de las predicciones de fallas y la estimación de la vida útil restante en entornos industriales.

Específicos

- Aplicar algoritmos de ML para realizar predicciones de la vida útil restante de componentes industriales, utilizando datos preprocesados con base en su origen.
- Identificar los componentes críticos en la maquinaria industrial que impactan significativamente la eficiencia operativa.
- Implementar y validar un sistema de preprocesamiento que permita mejorar la calidad de los datos antes de introducirlos en los modelos de ML.

1.4 Alcances y limitaciones

Alcances

- El modelo contará con un módulo de preprocesamiento de datos que tendrá en cuenta el origen de los datos (características del sensor, condiciones de recolección, etc.).
- El modelo será probado con datos de sensores reales de máquinas industriales.
- El modelo utilizará al menos un algoritmo de predicción basado en machine learning.
- Se realizará la validación del modelo mediante simulaciones y análisis comparativos con otros métodos de preprocesamiento.

Limitaciones

- El modelo no incluirá compatibilidad con múltiples tipos de máquinas simultáneamente.
- No se implementará autoajuste automático de algoritmos, el modelo requerirá ajustes manuales en el preprocesamiento según el origen de los datos.
- No se utilizarán tecnologías fuera del ámbito de la Industria 4.0.
- No se realizarán pruebas en ambientes industriales reales, la validación se hará mediante simulaciones controladas y datasets previamente recolectados.

1.5 Justificación

El mantenimiento predictivo se posiciona como una estrategia clave en el contexto de la Industria 4.0, impulsada por la creciente necesidad de reducir costos operativos, minimizar tiempos de inactividad no planificados y extender la vida útil de los activos. Según [5], se proyecta que el mercado global de mantenimiento predictivo crecerá de USD 10.6 mil millones en 2024 a USD 47.8 mil millones en 2029, con una tasa compuesta anual (CAGR) del 35.1%. Este crecimiento refleja la rápida adopción de tecnologías emergentes como son: el Internet de las cosas (IoT), la inteligencia artificial (IA) y el machine learning (ML), las cuales transforman los enfoques tradicionales de mantenimiento hacia modelos más inteligentes y basados en datos.

Las soluciones de mantenimiento predictivo no solo permiten a las organizaciones anticipar fallas en equipos, sino que también optimizan los programas de mantenimiento, mejorando significativamente la eficiencia operativa. Las implementaciones efectivas de estas tecnologías han demostrado reducir los costos de mantenimiento entre un 10-40% y los tiempos de inactividad no planificados hasta en un 50% [6]. Estos beneficios son especialmente relevantes en sectores industriales clave, tales como: manufactura, transporte, energía y servicios públicos, donde cualquier interrupción operativa tiene impactos económicos.

A pesar de estos avances, un desafío recurrente en el mantenimiento predictivo es la calidad de los datos empleados en los modelos de predicción. Este aspecto es crucial, ya que datos sensoriales deficientes pueden limitar la precisión de los algoritmos y afectar las decisiones basadas en evidencia. La presente investigación propone un enfoque diferenciado que aborda esta problemática al incorporar un modelo de preprocesamiento de datos sensoriales que considera aspectos como el tipo de sensor, las condiciones ambientales y el contexto de adquisición de datos. Este enfoque busca mejorar la calidad de la información utilizada por los algoritmos predictivos, permitiendo identificar patrones relevantes con mayor precisión y centrase en componentes críticos tales como los rodamientos industriales, los cuales son esenciales para maquinaria rotativa utilizada en múltiples industrias.

1.6 Estructura de la tesis

El contenido de esta tesis está organizado de la siguiente manera:

- **Capítulo II. Marco Conceptual:** Este capítulo desarrolla los fundamentos teóricos del mantenimiento predictivo y su relación con el ML, así como los conceptos clave relacionados con el preprocesamiento de datos y el análisis de rodamientos industriales como caso de estudio.
- **Capítulo III. Estado del Arte:** Se revisan los estudios más relevantes relacionados con el mantenimiento predictivo, destacando los avances en preprocesamiento de datos, los modelos de predicción empleados en la industria y su impacto en la precisión y eficacia del mantenimiento.
- **Capítulo IV. Metodología de Solución:** Aquí se describe la metodología desarrollada, incluyendo la definición y adquisición de datos, las técnicas de preprocesamiento aplicadas, el modelado predictivo y las fases de validación. Este capítulo detalla los procedimientos empleados para alcanzar los objetivos de la investigación.

- **Capítulo V. Resultados:** Se exponen los resultados obtenidos durante las pruebas experimentales, evaluando el desempeño de los modelos predictivos propuestos en comparación con enfoques existentes.
- **Capítulo VI. Conclusiones y Trabajos Futuros:** Finalmente, se presentan las conclusiones del estudio, destacando las contribuciones al campo del mantenimiento predictivo y se proponen líneas de investigación futura para continuar mejorando los enfoques planteados.

Capítulo II

Marco conceptual

2. Marco Conceptual

Este Capítulo proporciona el marco teórico que sustenta esta investigación, abarcando conceptos clave como el mantenimiento predictivo, las técnicas de ML aplicadas a la predicción de fallas, la importancia del preprocesamiento de datos y el uso de rodamientos industriales como caso de estudio.

a) Industria 4.0

Industria 4.0 es la cuarta revolución industrial caracterizada por la digitalización y la interconexión de sistemas mediante tecnologías como la inteligencia artificial, el Internet de las cosas (IoT), la robótica avanzada y la computación en la nube. Su objetivo es optimizar procesos, mejorar la eficiencia y crear fábricas inteligentes con mayor automatización.

b) Rodamiento

Un rodamiento es un componente mecánico que reduce la fricción entre dos superficies en movimiento relativo, facilitando el desplazamiento suave de piezas en máquinas y vehículos. Puede ser de distintos tipos, como rodamientos de bolas o de rodillos, y su diseño influye en la durabilidad y eficiencia del sistema mecánico.

c) Random Forest

Random Forest es un algoritmo de aprendizaje automático basado en la combinación de múltiples árboles de decisión. Funciona generando un conjunto de árboles entrenados con diferentes subconjuntos de datos y características, lo que mejora la precisión y reduce el sobreajuste. Se utiliza ampliamente en clasificación y regresión.

d) SVM

Support Vector Machine (SVM) es un algoritmo de aprendizaje supervisado utilizado para clasificación y regresión. Su objetivo es encontrar el hiperplano óptimo que mejor separa las clases en un espacio multidimensional, maximizando la distancia entre los puntos más cercanos de cada clase. Es eficaz en problemas con datos no linealmente separables

e) Árboles de decisión

Los árboles de decisión son modelos de aprendizaje automático que representan decisiones y sus posibles consecuencias en una estructura jerárquica de nodos. Cada nodo interno representa una prueba en un atributo, cada rama representa un resultado de la prueba y cada hoja una clase o valor. Son utilizados en clasificación y regresión.

f) Redes neuronales

Las redes neuronales artificiales son modelos computacionales inspirados en la estructura del cerebro humano. Están compuestas por capas de neuronas interconectadas que procesan información mediante funciones de activación. Son utilizadas en reconocimiento de imágenes, procesamiento de lenguaje natural y diversas tareas de inteligencia artificial.

g) Accuracy (exactitud)

Indica la relación proporcional entre las predicciones realizadas correctamente y el número de instancias evaluadas

h) Matriz de confusión

Se define como un método de visualización para los resultados del algoritmo clasificador. Es una tabla que desglosa el número de instancias reales de una clase específica frente al numero de instancias previstas para esa clase.

i) Recall

Representa la relación entre las instancias correctamente clasificadas contra las instancias originales

2.1 Mantenimiento industrial

En el sector industrial, las estrategias de mantenimiento son cruciales para asegurar la confiabilidad y eficiencia de maquinaria y sistemas. Estas estrategias se pueden clasificar ampliamente en mantenimiento correctivo, preventivo, predictivo y basado en condiciones. Cada tipo tiene características y aplicaciones distintas, las que son esenciales para optimizar las operaciones de mantenimiento y minimizar los costos [7].

a) Mantenimiento correctivo

El mantenimiento correctivo, también conocido como mantenimiento reactivo (R2F), implica reparar o reemplazar equipos después de que se haya producido una falla. Por lo general, se emplea en situaciones donde el costo de las paradas no programadas es menor que el costo de las medidas preventivas o cuando las fallas son poco frecuentes y no afectan significativamente las operaciones. Este enfoque se utiliza a menudo en empresas pequeñas o cuando las piezas de repuesto están fácilmente disponibles [7].

b) Mantenimiento Preventivo

El mantenimiento preventivo es un enfoque proactivo que implica inspecciones regulares y mantenimiento de equipos para evitar fallas. Se programa en función del tiempo o los intervalos de uso, independientemente de la condición actual del equipo. Este tipo de mantenimiento es adecuado para sistemas donde las fallas pueden tener impactos significativos en los calendarios de producción o la seguridad [7].

c) Mantenimiento predictivo

El PdM utiliza datos en tiempo real y análisis avanzados para predecir cuándo es probable que el equipo falle, lo que permite que el mantenimiento se realice justo a tiempo para evitar fallas. Se emplean tecnologías como IoT, Redes Neuronales Artificiales (NN) y sistemas difusos para monitorear el estado de los equipos y estimar la vida útil restante de los componentes. Este enfoque es particularmente beneficioso en sistemas complejos donde fallas inesperadas pueden conducir a interrupciones operacionales significativas [8], [9].

d) Mantenimiento basado en condición

El mantenimiento basado en condición (CBM) implica monitorear el estado real de los equipos para decidir sobre las acciones de mantenimiento. Se basa en datos recopilados de sensores y otras herramientas de monitoreo para evaluar la salud de los componentes. El CBM es efectivo en sistemas con dependencias estocásticas, donde la condición de un componente puede afectar a otros. Ayuda a optimizar los programas de mantenimiento y reducir las actividades de mantenimiento innecesarias [10], [11]

2.2 Machine Learning en el Mantenimiento Predictivo

En entornos industriales, las técnicas de ML, como Bosque Aleatorio (RF), han sido utilizadas de manera efectiva para predecir el estado de las máquinas con alta precisión, aprovechando datos de sensores y protocolos de comunicación en plataformas en la nube como Azure [1]. En la industria manufacturera, los modelos avanzados de ML, incluidos los métodos de aprendizaje profundo, han demostrado un rendimiento superior en la predicción de fallos de equipos y en la optimización de los programas de mantenimiento [12]. El sector energético también se beneficia del PdM basado en ML, este ayuda a mitigar fallos en los equipos y optimizar el mantenimiento. Sin embargo, la integración de fuentes de datos heterogéneas y la capacidad para interpretar los modelos presentan desafíos que pueden ser abordados mediante el preprocesamiento de datos y la ingeniería de características [13].

En el campo de la fabricación, el uso de redes neuronales convolucionales (CNN) y redes de memoria a corto y largo plazo (LSTM), han mostrado ser técnicas prometedoras para predecir la vida útil restante (RUL) y el estado de salud (SOH) de la maquinaria, mejorando significativamente la precisión de las predicciones y reduciendo los costos de mantenimiento. En el ámbito de los vehículos autónomos, los algoritmos de ML, incluidas técnicas de regresión, clasificación y aprendizaje profundo, son esenciales para evaluar las necesidades de mantenimiento, asegurando la confiabilidad y seguridad del sistema, además de mitigar los riesgos asociados con fallas [14].

Antes de aplicar estos algoritmos, los datos deben someterse a un preprocesamiento riguroso para garantizar su calidad y relevancia. Este paso es crucial para que los algoritmos de ML puedan trabajar de manera efectiva con la información proporcionada por los sensores. Una vez completado el preprocesamiento, algoritmos como SVM y las NN pueden aplicarse eficazmente para detectar patrones anómalos y predecir fallos incipientes.

2.3 Preprocesamiento de Datos

El preprocesamiento de datos es un paso fundamental en el ML, especialmente en aplicaciones industriales, donde los datos de los sensores suelen estar afectados por ruido, valores atípicos o ausencias de datos. Un preprocesamiento eficaz asegura que los modelos de ML trabajen con datos de alta calidad, lo cual es muy importante para realizar predicciones y análisis precisos. El tratamiento de datos que se propone incluye el filtrado de ruido, la segmentación de señales y la extracción de características, cada uno de los cuales juega un papel clave en la mejora de la calidad de los datos y el rendimiento del modelo.

a) Filtrado de ruido

El filtrado de ruido sirve para eliminar señales no deseadas que pueden distorsionar la información relevante. Este paso asegura que los datos introducidos en los algoritmos de MLS sean limpios y confiables, lo cual es fundamental para generar modelos de predicción precisos [15], [16].

b) Segmentación de señales

La segmentación de los datos en intervalos temporales permite analizar patrones específicos, lo que es especialmente útil para la identificación de fallos. Esta técnica ayuda a aislar los puntos de datos más relevantes, facilitando una detección precisa de fallos. El proceso de segmentación puede optimizarse mediante algoritmos avanzados que mejoran la selección de los segmentos de datos, aumentando así la capacidad del modelo para detectar y predecir fallos con mayor precisión [16].

c) Extracción de características

La extracción de características implica seleccionar las variables más representativas, tales como la amplitud de la vibración o la variación de temperatura, optimizando los datos de entrada para los modelos de ML. Este paso permite reducir la dimensionalidad del conjunto de datos y centrarse en las variables más relevantes.

Una selección eficaz de características puede mejorar significativamente la precisión del modelo, como lo demuestran las metodologías que seleccionan de manera meticulosa los datos óptimos dentro de los conjuntos de datos sin procesar [17].

En resumen, la calidad de los datos preprocesados es crítica para la efectividad de los modelos de ML. Un preprocesamiento adecuado no solo incrementa la precisión de las predicciones, sino que también reduce el riesgo de falsos positivos o negativos en la detección de fallos, algo esencial en entornos industriales.

2.4 Rodamientos Industriales como Caso de Estudio

Los rodamientos industriales fueron seleccionados como caso de estudio debido a su alta susceptibilidad a fallos mecánicos, los cuales pueden provocar interrupciones operativas significativas. Las señales generadas por los rodamientos, en particular las vibraciones, proporcionan patrones distintivos y detectables que son ideales para la implementación de técnicas de ML en el contexto de PdM. El análisis de vibraciones es un método ampliamente reconocido para la supervisión del estado de las máquinas rotativas, ya que ofrece datos detallados sobre su condición y puede detectar cambios que indican fallos inminentes [18], [19].

Los modelos de ML han demostrado un potencial considerable para reducir el tiempo de inactividad no planificado en los rodamientos industriales, mejorando así las estrategias de PdM. La integración del ML con la supervisión del estado, permite pronosticar con precisión el estado de las máquinas. Un ejemplo de ello es el sistema basado en *Random Forest* de [1], el cual utiliza datos de múltiples sensores y protocolos de comunicación para predecir las condiciones de las máquinas con alta exactitud.

Capítulo III

Estado del arte

2. Estado del arte

Este Capítulo presenta una revisión detallada del estado del arte en modelos de PdM, con el objetivo de identificar las prácticas actuales y los enfoques más relevantes utilizados en la industria. La rápida evolución tecnológica, impulsada por la Industria 4.0, ha dado lugar a la aparición de diversas metodologías que buscan optimizar la gestión del mantenimiento mediante el uso de herramientas avanzadas tales como el ML y los CPS. Para comprender mejor las ventajas y limitaciones de estos modelos, se plantea una serie de preguntas de investigación que guiaron la revisión de la literatura, permitiendo evaluar la eficiencia, escalabilidad y aplicabilidad de las técnicas existentes en el ámbito industrial.

Las preguntas son:

1) ¿Cuáles son los modelos de mantenimiento predictivo utilizados en la industria?
2) ¿Qué modelo es el más utilizado?
3) ¿Cuáles son las diferencias entre los modelos utilizados?
4) ¿Qué tan eficiente es el modelo más utilizado?
5) ¿Cuáles son los modelos de mantenimiento escalables?
6) ¿Cuál es la desventaja de los modelos de mantenimiento no escalables?
7) ¿Los modelos de mantenimiento pueden predecir el desgaste de distintos tipos de máquinas?
8) ¿A qué se debe la existencia de varios modelos de mantenimiento predictivo?

A partir de estas preguntas se realizó una investigación bajo los siguientes

Palabras Clave:

- Machine Learning
- Predictive Models
- Predictive Maintenance
- Predictive Maintenance Models
- Predictive System
- Industrial Maintenance
- Maintenance 4.0

Las bases de datos establecidas para hacer la búsqueda fueron las siguientes:

- Web of Science
- Science Direct

- Springerlink
- IEEExplorer

En el trabajo [20] se muestra una técnica híbrida que combina el aprendizaje supervisado y el aprendizaje semi-supervisado, se demuestra que los avances de IoT brindan una oportunidad eficiente para generar estrategias innovadoras de mantenimiento en entornos de fabricación inteligentes. Se observa que debido a los desarrollos científicos y tecnológicos las industrias deben poner en práctica políticas de PdM en lugar del mantenimiento habitual (*R2F* o *PvM*).

Se utiliza un modelo de mantenimiento que se compone de cinco fases que van desde la adquisición de datos hasta la planeación del mantenimiento, todo esto utilizando un módulo de PdM que a su vez hace uso de algoritmos de aprendizaje supervisado y semi-supervisado (Ver Figura 1).

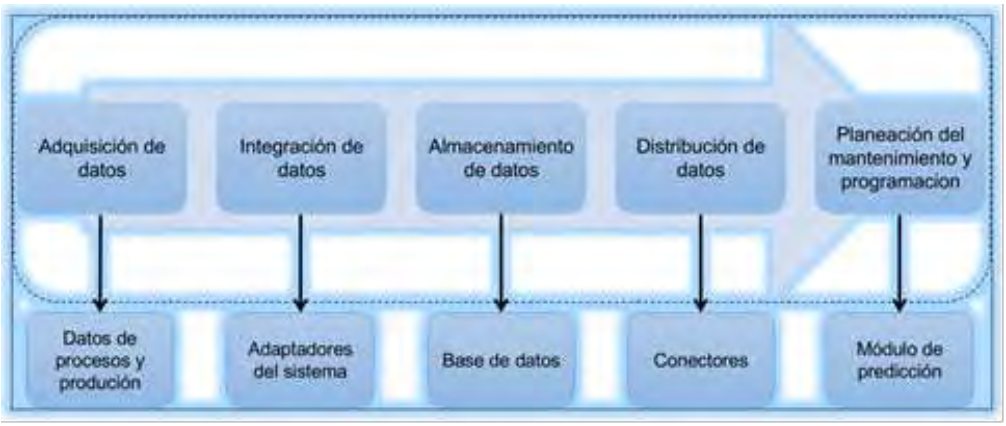


Figura 1. Metodología [20].

Este trabajo proporciona un servicio de alto valor para los usuarios del equipo al evitar tiempos de inactividad mediante la predicción de fallas. Este servicio ofrece resultados que permiten minimizar el costo total, optimizar el uso de materiales y aumentar la disponibilidad de los equipos, beneficiando así a la tecnología de fabricación en su conjunto. Como resultado, se obtiene un sistema que muestra, a través de una plataforma web, tres tipos de datos: un resumen del estado de las máquinas, la probabilidad de falla y los detalles específicos de la máquina.

La propuesta de [21] es la implementación de una arquitectura de referencia para sistemas ciberfísicos con soporte de mantenimiento basado en la condición. Además, en este trabajo se describe una propuesta para el análisis de datos orientado a la gestión del PdM en un conjunto de activos de embrague. Las técnicas utilizadas incluyen el análisis de causa raíz, potenciado por la agrupación de inducción orientada a atributos, y la estimación de la vida útil restante, mejorada con pronósticos de series de tiempo.

En la Figura 2 se observa el modelo utilizado en este trabajo.

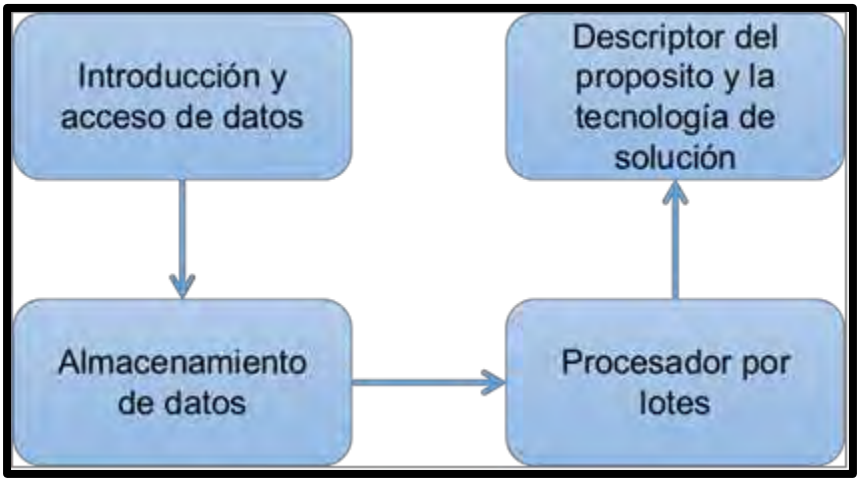


Figura 2. Metodología [21].

Este modelo se centra en los requisitos de las partes implicadas y proporciona pasos bien estructurados, los que son necesarios para construir la solución. Además, la implementación de la plataforma permite ofrecer servicios de mantenimiento basados en los datos recopilados de sensores y sistemas ciberfísicos.

Asimismo, se presenta el desarrollo de una arquitectura de plataforma para servicios de PdM, basada en sistemas ciberfísicos que facilitan la creación de ecosistemas de mantenimiento colaborativo. Esta arquitectura cumple con los requisitos establecidos por diversos casos de uso industrial que demandan soluciones de monitoreo. Dichas soluciones incluyen sensores y software a nivel de sistemas ciberfísicos que actúan como plataformas, así como herramientas para el análisis de datos.

En [22] presenta un Sistema Industrial de Agentes Múltiples diseñado para pronósticos colaborativos distribuidos en tiempo real. Este sistema cumple con las seis propiedades fundamentales de los sistemas avanzados de agentes múltiples: distribución, flexibilidad, adaptabilidad, escalabilidad, simplicidad y resistencia.

El modelo utilizado en este documento es una arquitectura jerárquica modificada, la que está compuesta por tres capas: activos virtuales, gemelos digitales y una plataforma social (Ver Figura 3)

.

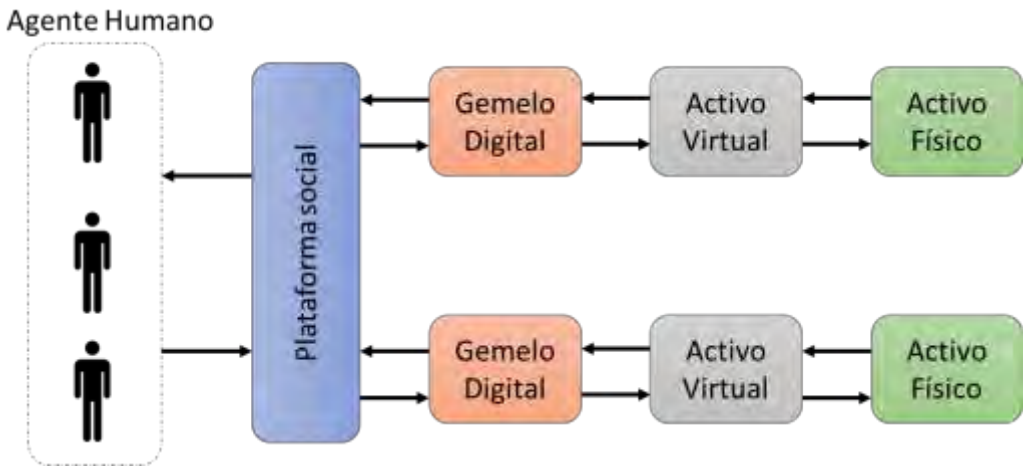


Figura 3. Modelo de [22].

En [23] se presenta un modelo de algoritmos basados en inteligencia artificial para la ejecución de actividades de PdM, aplicados al control de dos elementos críticos en el sistema de máquinas herramienta: la herramienta de corte y el motor del husillo. Además, se describe un modelo basado en datos utilizado para investigar tanto el desgaste de la herramienta como las fallas en los rodamientos.

El modelo se basa en la recopilación de datos a través de sensores conectados a los equipos. Los datos más relevantes se filtran, posteriormente se realiza la extracción de características y, finalmente se aplican a un modelo de predicción basado en ML (Ver Figura 4). El modelo proporciona resultados de predicción que oscilan entre el 78% y el 93%.

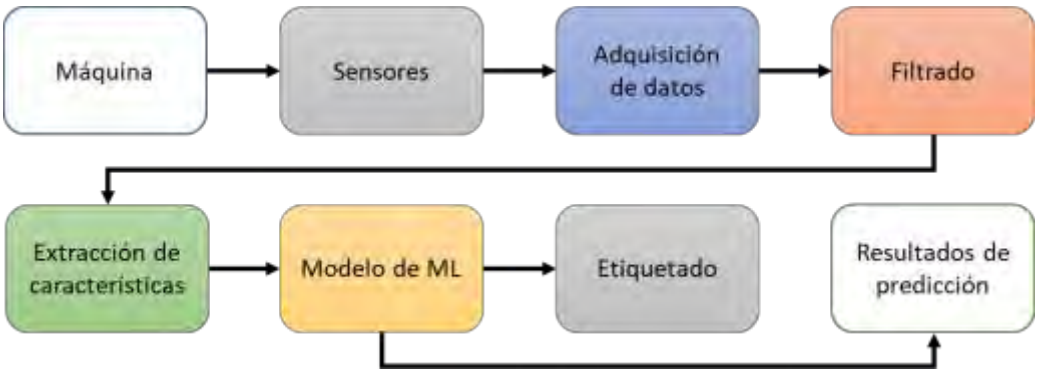


Figura 4. Procedimiento de PdM *[23].*

En la investigación realizada por [24], se presenta un modelo para estimar la probabilidad de avería de una máquina en un intervalo de tiempo específico hacia el futuro. Para este propósito, se emplean algoritmos de ML en un caso de uso basado en conjuntos de datos del mundo real. El documento describe métodos aplicados para la extracción de datos, la extracción de características y el uso de ML. Los resultados indican que las fallas de la máquina se pueden predecir de manera confiable.

La figura 5 muestra un ejemplo ilustrativo del funcionamiento del sistema.

Figura 5. Sistema de predicción [24].

Se muestra una eficiencia de hasta el 85% utilizando distintas variables en el entrenamiento del modelo. La aplicación en fresadoras demuestra que las averías importantes de estas máquinas son predecibles hasta 168 horas antes de que ocurran.

La investigación de [25] presenta un modelo de mantenimiento basado en un sistema de monitoreo ultrasónico, cuyo objetivo es extender la vida útil del eje de una turbina eólica más allá del 50% en comparación con el monitoreo manual. Este modelo utiliza electrónica ad-hoc, un firmware específico que incluye un algoritmo de detección y evaluación, y una matriz anular de transductores conectados al eje mediante un soporte mecánico especializado. El objetivo principal es realizar un control continuo sobre la evolución de las grietas en la turbina eólica, evitando que la grieta alcance un tamaño crítico.

El control del crecimiento de grietas se lleva a cabo en tres etapas operativas. En la primera etapa, el sistema detecta el inicio de una grieta y emite una primera advertencia. En la segunda etapa, se lanza una advertencia cuando la grieta identificada indica un tiempo de vida útil restante de 1 a 2 meses en condiciones nominales; en esta etapa, la grieta se considera no peligrosa. La tercera y última etapa corresponde a la advertencia final, cuando la grieta alcanza su tamaño máximo permitido, lo que representa el consumo del 96% del tiempo de vida útil restante.

Gracias al monitoreo continuo, la vida útil del eje se mejora del 50% al 96%, y las paradas de la turbina eólica se reducen considerablemente, lo que facilita la planificación eficiente de la sustitución del eje y reduce los costos asociados a estas paradas. El resultado es un modelo de diagnóstico y pronóstico que permite establecer diferentes etapas operativas y ajustar la frecuencia de pruebas en función de la criticidad de la grieta (a mayor tamaño de la grieta, mayor frecuencia de inspección).

En el trabajo de [26] se presenta un modelo de aprendizaje diseñado para detectar posibles anomalías en sistemas fotovoltaicos, permitiendo que un operador planifique intervenciones de PdM. El algoritmo de detección de anomalías compara los valores medidos con los valores esperados de producción de energía, utilizando una NN entrenada con datos previamente recopilados en la planta fotovoltaica. Los resultados experimentales muestran que el modelo tiene una tasa de detección de anomalías superior al 90%. Además, el sistema identifica el comportamiento normal de operación y genera alertas de PdM, sirviendo como un sistema de apoyo en la toma de decisiones para evitar posibles fallas.

En [27] se presenta una investigación sobre la predicción de fallas para mejorar el rendimiento en sistemas de calefacción, ventilación y aire acondicionado (HVAC). Para ello, se emplean un modelo de Regresión Lineal (LR) y una red neuronal artificial, los cuales muestran mejoras significativas cuando se complementan con Árboles de Decisión (DT). La metodología utilizada consta de un filtrado de datos, filtrado de datos faltantes, selección de atributos, normalización, multiplicación de datos, separación de datos, aplicación del modelo de predicción, validación y obtención de resultados (ver Figura 6).

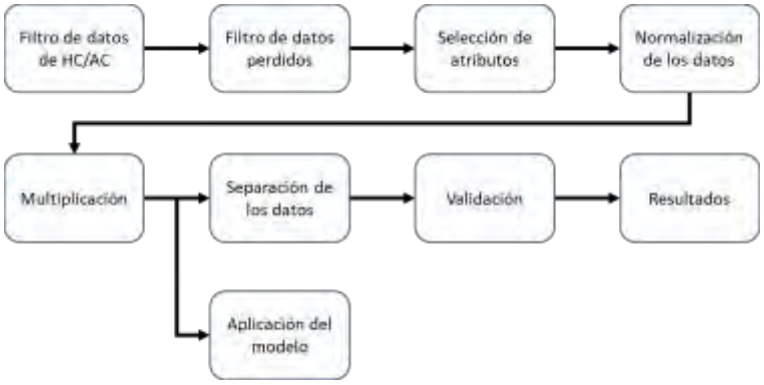


Figura 6. Metodología [27].

Como resultado, se obtuvo un 74% y 83% de precisión para los modelos de regresión y red neuronal, respectivamente. Estos datos son utilizados para predecir la condición de los elementos de un equipo.

Para reducir el número de instalaciones que experimentan fallas, los investigadores en [28], proponen un modelo de PdM basado en estadística, el que permite determinar el momento adecuado para enviar a un trabajador de mantenimiento a alguna de las instalaciones (Ver Figura 7).

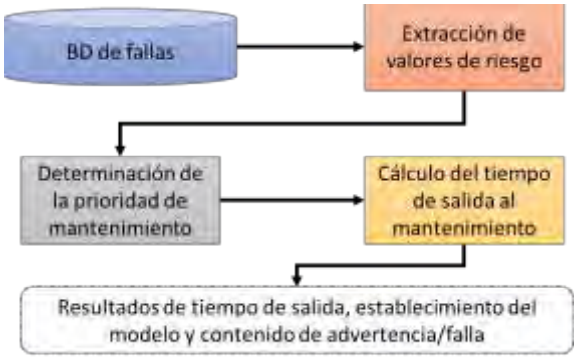


Figura 7. Metodología [28].

Como resultado del modelo propuesto, se obtienen advertencias de fallas para la programación del mantenimiento. Además, se considera el tiempo de traslado del personal de

mantenimiento hasta la ubicación de la maquinaria a reparar, lo que permite reducir la tasa de fallas en un 12% y aumenta el tiempo libre de los trabajadores de mantenimiento en 16 horas.

La investigación de [29] se centra en el PdM de robots industriales y la posibilidad de construir un modelo de monitoreo de condición basado en el análisis de datos provenientes de las mediciones de los robots. Se propone un modelo predictivo para detectar errores de precisión en un robot manipulador. Además, se lleva a cabo un procedimiento para supervisar la correlación entre los errores y un conjunto de características extraídas en tiempo real, con el fin de evaluar la efectividad del modelo propuesto.

La figura 8 muestra un esquema del modelo para la detección de fallas.

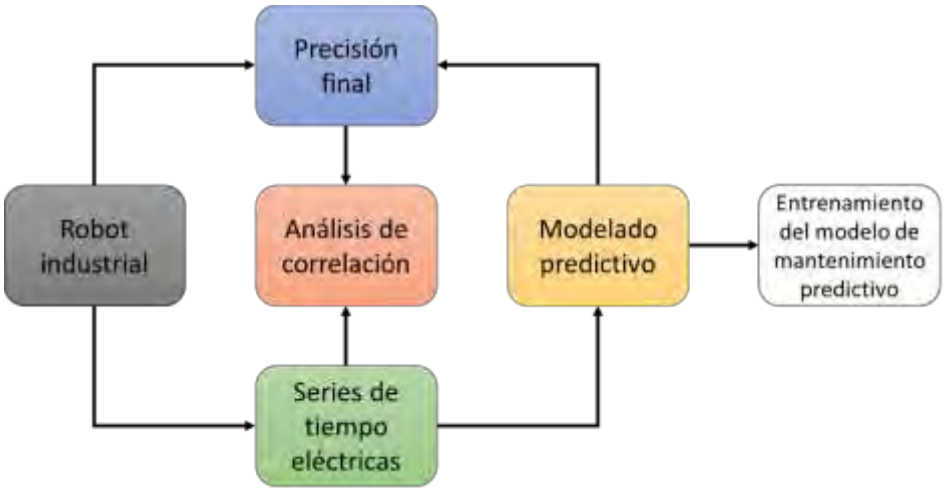


Figura 8. Detección de fallas [29].

El resultado muestra la probabilidad de que un robot necesite ser reparado, con un error de predicción de 0.0916, lo cual es muy cercano a cero y demuestra la fiabilidad del modelo propuesto.

En algunas ocasiones, las instalaciones no cuentan con una arquitectura moderna que permita el monitoreo de la maquinaria en tiempo real. Considerando esta problemática en la evolución de las técnicas de PdM, los investigadores en [30] demuestran cómo es posible habilitar los equipos mediante la adaptación de sensores de bajo costo, una arquitectura basada en el IoT y el uso de ML (ver Figura 9).

Figura 9. Arquitectura de predicción [30].

Los resultados de este trabajo muestran que la aplicación de algoritmos de ML para obtener predicciones de estado es muy eficiente. Sin embargo, el tipo de algoritmo a utilizar depende de la tarea a realizar; no obstante, los algoritmos de aprendizaje supervisado son los que mejor se adaptan a las tres detecciones realizadas.

Aunque existen investigaciones que aplican diversas técnicas de predicción de fallas, no todos los trabajos presentan sistemas completos que puedan ser utilizados en entornos reales. La investigación realizada por [31] muestra un modelo que se caracteriza por predecir fallas en turbinas eólicas, utilizando una solución basada en datos, implementada en la nube, y que está compuesta por tres módulos principales:

- Un generador de modelos predictivos que crea modelos para distintas turbinas eólicas.
- Un agente de monitoreo que realiza predicciones cada 10 minutos sobre fallas en turbinas eólicas durante la próxima hora.
- Un tablero donde se visualizan las predicciones realizadas.

La Figura 10 muestra la arquitectura del sistema utilizado.

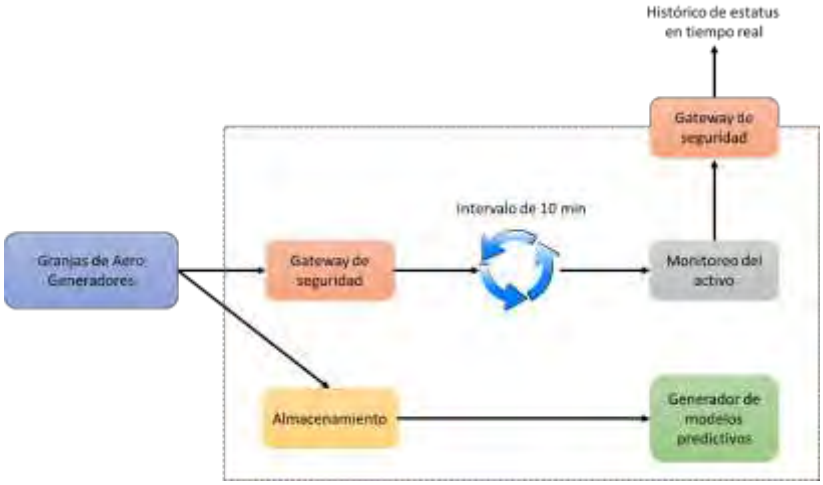


Figura 10. Arquitectura de predicción de fallas [31].

Como resultado, se obtiene una precisión del 82.04% para la detección de fallas en las turbinas eólicas. Sin embargo, se sugiere que esta precisión podría mejorarse si se realiza un balance adecuado de datos para la generación de los modelos de predicción.

Algunos modelos de PdM que determinan la vida útil restante de un equipo se basan en la simulación de su funcionamiento durante un determinado lapso de tiempo. La investigación realizada por [32] emplea técnicas de pronóstico y gestión de salud, desarrollando un sistema de control inteligente que recopila datos de la máquina para ser utilizados en el sistema de simulación.

La Figura 11 muestra el concepto principal para el cálculo de la vida útil restante.

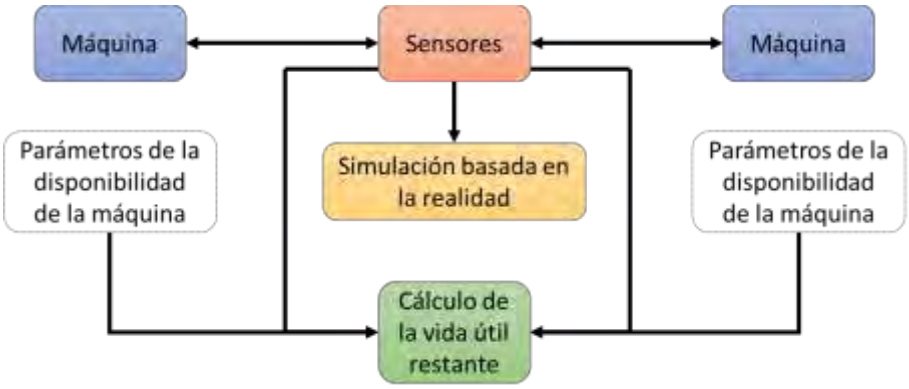


Figura 11. Cálculo de la vida útil restante [32].

No utilizar datos previos para determinar la vida útil restante es una ventaja cuando este proceso se realiza mediante simulación, ya que permite generar distintos escenarios hacia el futuro, basándose únicamente en el estado actual y el estado deseado de la máquina.

La reducción de fallas inoportunas en los sistemas de calefacción es la principal motivación para desarrollar modelos de predicción de fallas en la investigación de [33]. El objetivo principal es crear un modelo de mantenimiento capaz de identificar, predecir y notificar la ocurrencia de eventos de falla. La solución propuesta fue desarrollada en un escenario de PdM, utilizando un conjunto de datos que incluye el ciclo de vida de los sistemas de calefacción. Para verificar la efectividad de la aplicación, se evaluó el tiempo de procesamiento de datos y los resultados de predicción. Sin embargo, los resultados muestran una precisión inferior al 50%, lo que los autores atribuyen al desbalance de los datos utilizados para crear los modelos.

Por su parte, en [34] presentan un modelo de PdM para la automatización y optimización de equipos de trabajo, utilizando una red de sensores inteligentes. El sistema incluye un portal de mantenimiento en línea, un módulo de notificaciones en tiempo real, la optimización mediante la red de sensores inteligentes y la implementación de métodos de análisis de datos.

La Figura 12 muestra el proceso que se sigue para llevar a cabo las tareas de mantenimiento.

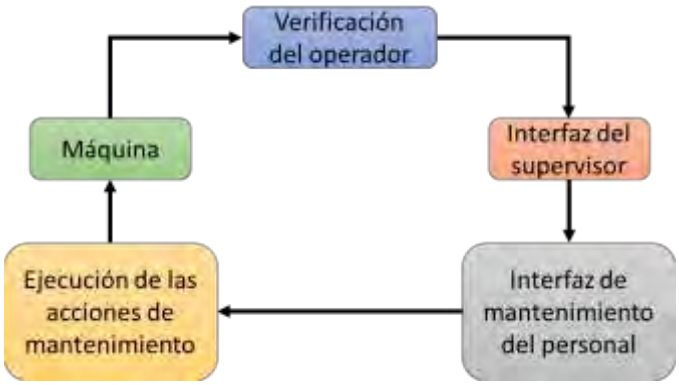


Figura 12. Proceso de mantenimiento [34].

Este trabajo proporciona un sistema que permite automatizar el mantenimiento, logrando así reducir tiempos y costos de operación.

Existen diversas herramientas que pueden ayudar a mejorar los modelos de PdM. En la investigación de [35], se utiliza la tecnología de realidad aumentada para el control de procesos en tiempo real, a través de la visualización de parámetros del proceso de fabricación.

El objetivo de este modelo es proporcionar control sobre el mantenimiento inteligente, utilizando dispositivos de realidad aumentada para visualizar diferentes tipos de información. De esta forma, el trabajador de mantenimiento puede ver los valores actuales e históricos del sistema de monitoreo en tiempo real y directamente en el lugar de trabajo. Los valores críticos de los sensores se resaltan, permitiendo un control inteligente de PdM mediante realidad aumentada.

En la figura 13 se muestra la arquitectura del sistema desarrollado.

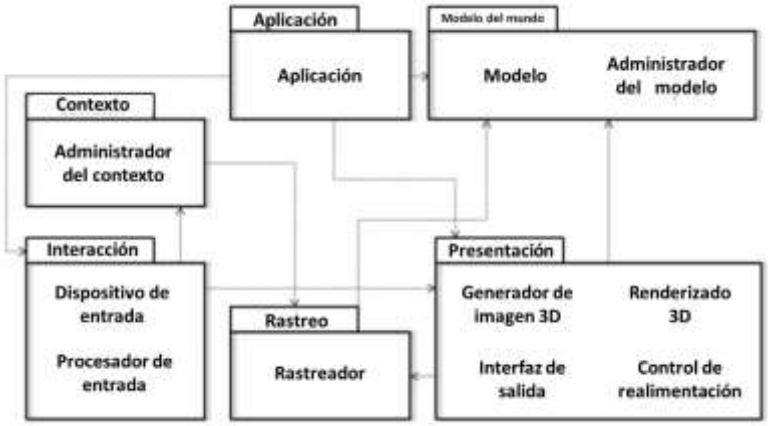


Figura 13. Arquitectura [35].

En el artículo de [36], se describe un prototipo de modelo de análisis de vibraciones cuyo propósito es servir como una alternativa de bajo costo para el PdM. El prototipo está compuesto por sensores, filtros y herramientas computacionales, y permite realizar un análisis de frecuencias detectadas en un motor.

Los resultados de este análisis se utilizan para llevar a cabo diagnósticos de mantenimiento oportuno, recomendar acciones de PdM e informar sobre el estado de salud de los motores. El sistema fue probado con motores eléctricos demostrando ser funcional.

La investigación realizada por [37] se enfoca en el estudio de diferentes algoritmos de ML para la programación de PdM, con el objetivo de determinar la vida útil restante de turbo ventiladores de aeronaves. En el estudio, se presentan comparativas de algoritmos utilizando como muestra el repositorio de pronósticos de la NASA.

La Figura 14 muestra el modelo de aprendizaje utilizado. De los algoritmos evaluados, el algoritmo de RF resultó ser el más adecuado para la predicción de la vida útil restante de los turboventiladores; sin embargo, se observa que el error del modelo es aproximadamente del 30%.

Figura 14. Diagrama bloques del modelo de aprendizaje [37].

En la investigación de [38], se diseñó un modelo de mantenimiento para predecir la probabilidad de falla de los componentes de los cajeros automáticos de diferentes marcas y modelos. El objetivo es determinar en qué periodo de tiempo ocurrirá una falla, considerando las necesidades tanto de la entidad financiera como de los clientes. En la Figura 15 se muestra el esquema de trabajo del sistema.

El modelo fue probado con tres componentes de 15 dispositivos, en una ventana de tiempo de 30 días, utilizando datos de entrenamiento recopilados entre 9 y 12 meses antes.

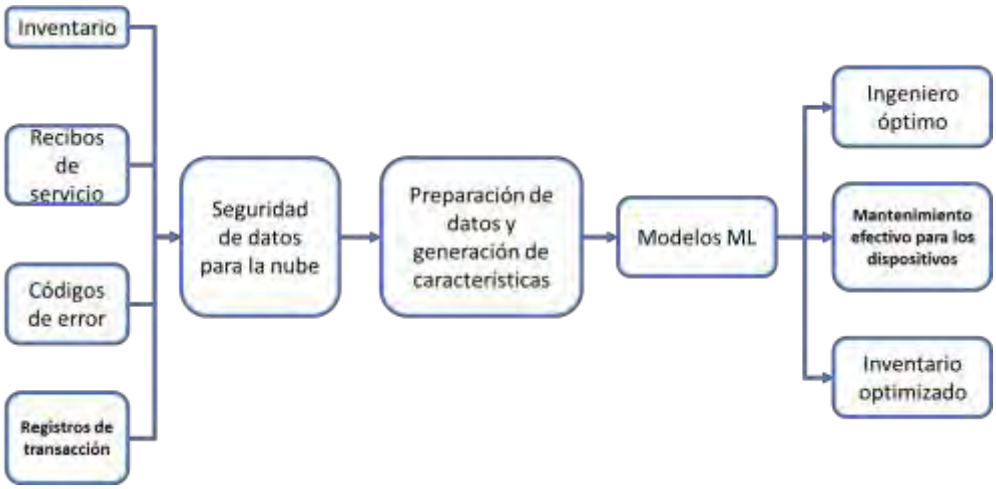


Figura 15. Esquema de trabajo [38].

El artículo de [39], muestra el uso del pronóstico del promedio móvil autorregresivo en datos de series de tiempo, recopilados de distintos sensores instalados en una máquina de corte longitudinal. Con este enfoque, se predicen fallas en el equipo y posibles defectos de corte, mejorando así el proceso general de fabricación.

El uso de ML demuestra ser un componente vital en el IoT, con aplicaciones en la gestión y control de calidad, lo que permite reducir costos de mantenimiento y optimizar el proceso de fabricación en general.

En la Figura 16 se puede observar el diagrama de bloques del funcionamiento del modelo propuesto. Este modelo se basa en la recopilación de datos a través de sensores conectados a los equipos, los que envían la información a un dispositivo de control. Dicho dispositivo está conectado a un servidor en la nube, el cual es el encargado de realizar predicciones mediante algoritmos de ML.

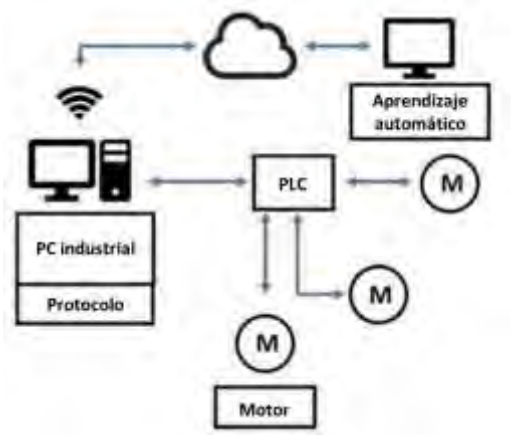


Figura 16. Diagrama de bloques [39].

Como resultado de los diferentes algoritmos utilizados, se obtiene una precisión que varía entre el 94% y el 98%, siendo el algoritmo de Redes Neuronales Profundas (DNN) el más efectivo, seguido del algoritmo Naive Bayes (NB).

En la investigación de [40] se presenta una arquitectura de sistemas ciberfísicos de bajo costo, el cual mide variables como la temperatura y la vibración en un proceso de torneado en un centro de control numérico. Estos datos se almacenan en la nube, donde se aplican técnicas de ML para predecir el rechazo de piezas mecanizadas, basándose en el umbral de calidad esperado (ver Figura 17).

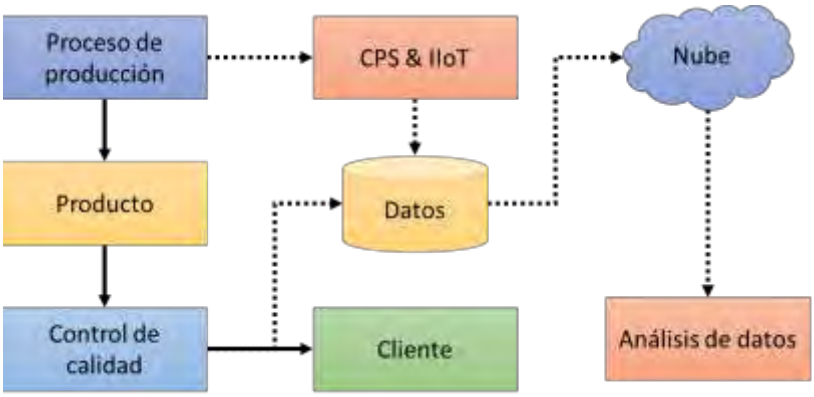


Figura 17. Esquema de funcionamiento [40].

La calidad del mecanizado se predice utilizando los datos de temperatura y vibración, evaluándose en función de la rugosidad promedio de la superficie de cada pieza mecanizada, lo que demuestra una precisión predictiva del 80% en 21 evaluaciones.

En la investigación de [15], se propone un modelo de mantenimiento basado en ML, el cual utiliza la información de vibración de un extractor de aire industrial para predecir fallas en el motor. Se sugiere realizar un agrupamiento de predicciones para minimizar el error y mejorar la programación de mantenimiento. El modelo propuesto elabora una herramienta que identifica de manera óptima la cantidad de agrupamientos necesarios para la predicción, utilizando cinco componentes de recopilación de datos.

En el documento presentado por [41], se realiza un análisis de la señal de vibración para extraer patrones de comportamiento en rodamientos. Se aplican modelos de ML para estimar la vida útil restante y clasificar el tipo de falla, además de implementar un enfoque de recomendación colaborativa para analizar la similitud de los resultados.

La figura 18 muestra la arquitectura del análisis de vibración utilizada.

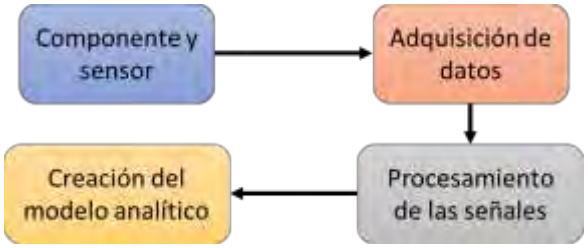


Figura 18. Arquitectura del sistema [41].

Los resultados de los algoritmos muestran una precisión en la predicción de fallas que varía entre el 78% y el 81%, siendo la distancia euclidiana el algoritmo de predicción más efectivo. Esto permite sugerir el reemplazo y la corrección de las unidades deterioradas, evitando así averías e interrupciones graves.

En la investigación de [42], se presenta un modelo de apoyo para la programación del mantenimiento basado en un enfoque predictivo/adaptativo. Este modelo considera los costos asociados a las fallas, las oportunidades de mantenimiento y los riesgos implicados

La figura 19 muestra la integración del PdM en la fábrica.

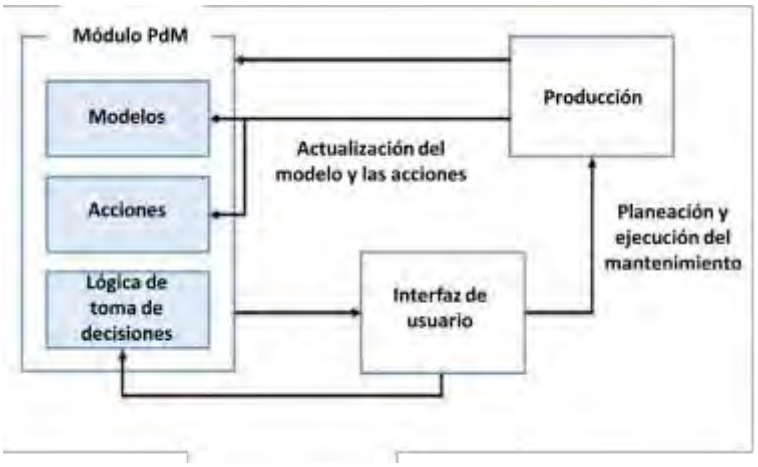


Figura 19. Integración del mantenimiento [42].

El modelo propuesto utiliza ML y métodos de regresión regularizados para aprovechar la nueva información a medida que está disponible, refinando las estimaciones de la vida útil restante, así como los costos y riesgos asociados.

El documento de [43] describe una arquitectura de ML orientada al PdM. El sistema se probó en un caso real de la industria, realizando la recopilación y análisis de datos, comparando los resultados con el análisis de una herramienta de simulación. Los datos fueron recopilados a través de varios sensores en las máquinas y utilizando diferentes protocolos de comunicación.

La Figura 20 muestra el esquema de trabajo del sistema.

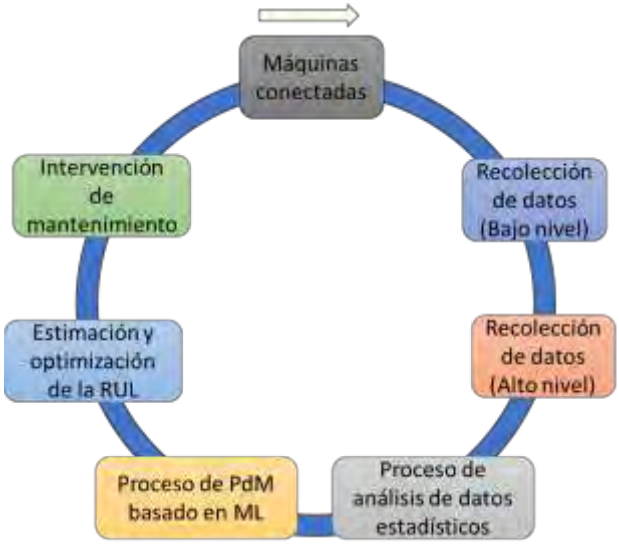


Figura 20. Modelo de mantenimiento [43].

Los resultados muestran un desempeño adecuado del enfoque para predecir diferentes estados de la máquina con alta precisión, alcanzando valores superiores al 90%.

Existen trabajos que buscan automatizar el mantenimiento en la mayor medida posible. Un ejemplo es el estudio de [44], en el cual se presenta un motor de autoajuste integrado para PdM,

basado en tecnologías Big Data y diseñado para aplicaciones en la Industria 4.0. El enfoque propuesto incluye un bloque de ingeniería de características específicas que extrae automáticamente métricas de series de tiempo en la producción, mejorando el rendimiento predictivo en un 77% en promedio. Además, cuenta con un enfoque de autoajuste que selecciona dinámicamente el mejor algoritmo de predicción, incrementando el rendimiento predictivo hasta en un 60%.

Se ofrecen modelos predictivos transparentes, capaces de proporcionar a los usuarios finales información detallada sobre el conocimiento adquirido, y se evaluaron experimentalmente en un conjunto de datos públicos sobre fallas. Los resultados de predicción van desde el 35% con SVM hasta el 63% con el algoritmo de RF.

El estudio de [45] tiene como objetivo presentar y describir un modelo de mantenimiento basado en la web, el cual permite la simulación de un sistema de pronóstico y gestión de salud, estando enfocado en predecir la vida útil restante de componentes específicos de una aeronave mediante diferentes técnicas de ML. Para acceder a esta herramienta, se realiza una conexión web que permite la creación de escenarios, en los cuales se puede seleccionar un conjunto de datos de trabajo. Posteriormente, el sistema web simula el comportamiento de la aeronave "digital" y, a través de algoritmos de ML se predice el tiempo de vida útil restante de los componentes seleccionados.

Los conjuntos de datos sugeridos están compuestos por datos sintéticos obtenidos de sensores de aeronaves, y los métodos propuestos representan distintas metodologías para el cálculo de la vida útil restante.

El trabajo realizado por [46] presenta un enfoque para la extracción de los datos crónicos más relevantes de un conjunto de datos industriales, los cuales se utilizan para predecir el tiempo de falla de una máquina determinada.

La Figura 21 muestra el proceso para la realización del PdM.

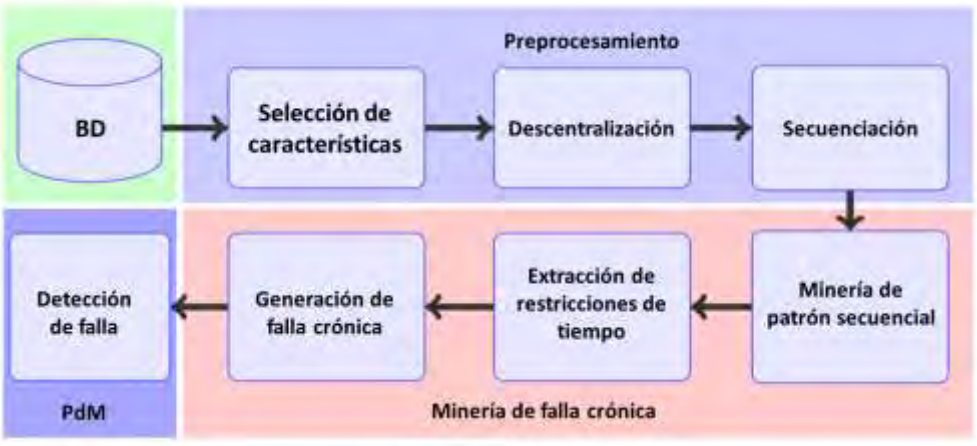


Figura 21. Mantenimiento predictivo de [46].

Este enfoque se valida a través de varios experimentos realizados en un conjunto de datos de referencia, los que arrojan resultados de predicción de hasta el 90%.

El trabajo de [47] presenta un modelo que utiliza autómatas híbridos temporizados para el PdM de plantas industriales. Este modelo híbrido simplifica las señales discretas y continuas, reduciéndolas a estados individuales que representan la condición actual de las máquinas.

El enfoque permite una detección efectiva de fallas mediante la implementación de adquisición de datos y detección de anomalías, además del pronóstico para otras aplicaciones, como la planificación del PdM.

En la Figura 22 se muestra el procedimiento de identificación de fallas.

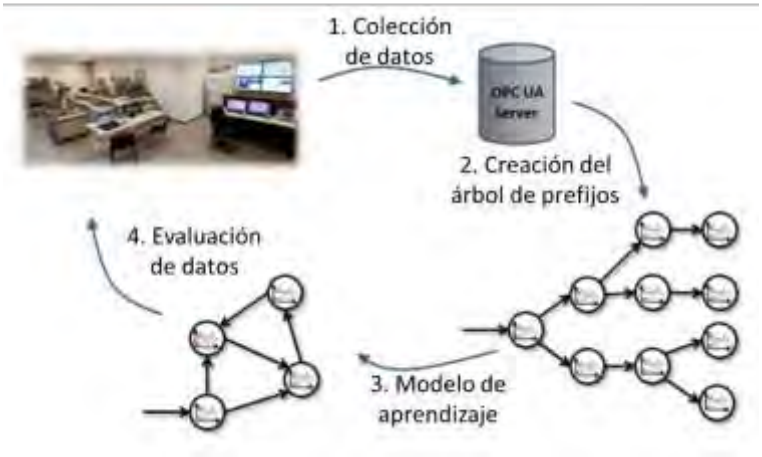


Figura 22. Proceso de detección de fallas [47].

En las pruebas realizadas, se obtuvieron resultados de predicción que oscilan entre el 80% y el 90%.

El artículo de [48] presenta un modelo para el PdM para transformadores de energía. Se proponen técnicas para predecir cuáles transformadores tienen mayor probabilidad de fallar y cuál podría ser la solución adecuada, utilizando datos como las especificaciones del transformador, la carga, la ubicación y el clima. Al final, se genera una lista de los transformadores con riesgo de fallo para facilitar la programación del mantenimiento.

La Figura 23 muestra el flujo de trabajo para la predicción de fallas.

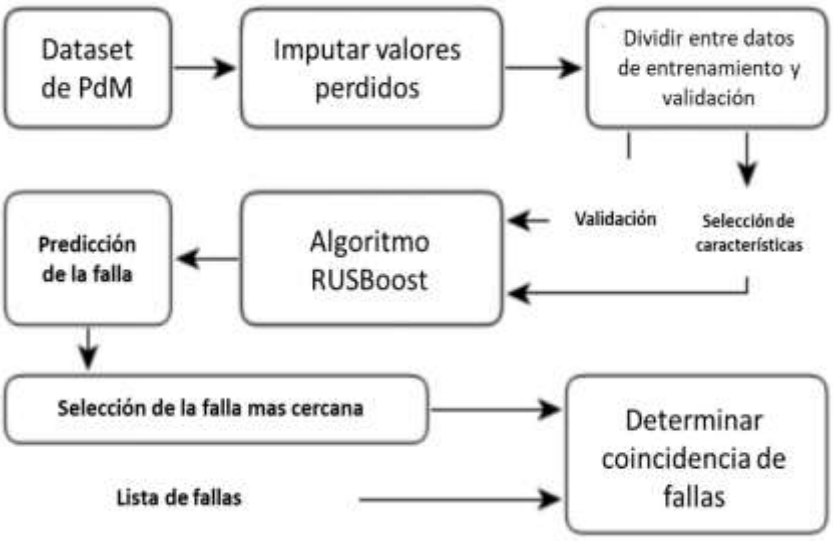


Figura 23. Flujo de trabajo del sistema [48].

Los algoritmos utilizados fueron RUSBoost y RF, los cuales se probaron en más de 700,000 transformadores de distribución en el sur de California. Los resultados de estas pruebas demuestran que ambos algoritmos superan el estado actual de la práctica.

El trabajo de [49] presenta un modelo de PdM inteligente para la industria moderna, el que consiste en un análisis de datos para la detección temprana de posibles fallas en las máquinas. Este modelo apoya a los técnicos proporcionando sugerencias de mantenimiento.

La Figura 24 representa la arquitectura propuesta.

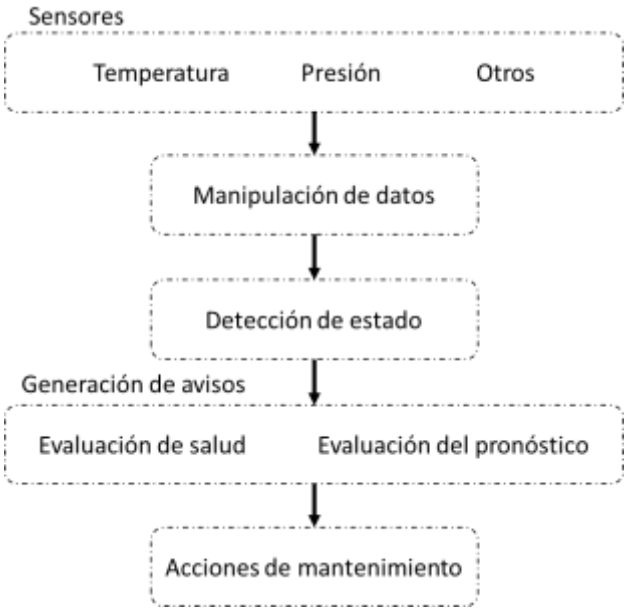


Figura 24. Arquitectura del sistema [49].

El modelo propuesto amplía las funcionalidades básicas del monitoreo basado en condición mediante la integración de sistemas de soporte de decisiones. Esto mejora la interacción entre humanos y máquinas, optimizando el rendimiento en la ejecución de las intervenciones de mantenimiento. La arquitectura del sistema considera análisis avanzado, así como la línea de datos recolectados para la detección temprana de posibles fallas en la máquina.

A continuación, se presenta una comparativa de las investigaciones. En donde podemos ver si el en el trabajo se usa aprendizaje automático/aprendizaje automático embebido (ML/EML), si se aplica a una o diferentes máquinas, si lleva un registro del historial de mantenimientos realizados, si es basado en sistemas ciberfísicos (CPS), el origen de sus datos de entrenamiento, en donde se ejecuta, si es escalable, si se ejecuta de forma local y si hace que las máquinas sean auto predecibles.

Tabla 1. Comparativa de los trabajos consultados parte 1.

Trabajo	ML/ EML	Diferentes máquinas	Historial de mantenimiento	Basado en CPS	Datos de entrenamiento	Ejecutable en	Escalable	Centralizado	Máquinas Auto predecibles
A Hybrid Machine Learning Approach for Predictive Maintenance in Smart Factories of the Future [20].	**ML**	X	X	✓	Genéricos	CPS	X	✓	X
A Big Data implementation of the MANTIS reference architecture for predictive maintenance [21].	X	X	X	✓	Dataset	Nube Local	X	✓	X
An Industrial Multi Agent System for real-time distributed collaborative prognostics [22].	**ML**	✓	✓	X	Genéricos	PC Local	✓	Local	X
Predictive Maintenance of Machine Tool Systems Using Artificial Intelligence Techniques Applied to Machine Condition Data [23].	**ML**	X	X	X	Dataset	PC Local	X	Local	X
Log-based predictive maintenance in discrete parts manufacturing [24].	**ML**	✓	X	X	Genéricos	PC Local	X	Local	X
Predictive maintenance of wind turbine low-speed shafts based on an autonomous ultrasonic system [25].	**ML**	X	X	X	Dataset	PC Local	X	Local	X
Anomaly detection and predictive maintenance for photovoltaic systems [26].	**ML**	✓	X	X	Dataset	CPS	X	✓	X
A Predictive Preference Model for Maintenance of a Heating Ventilating and Air Conditioning System [27]	**ML**	X	X	X	Genéricos	CPS	X	✓	X
A statistical approach to reduce failure facilities based on predictive maintenance [28]	X	X	X	X	Dataset	PC Local	X	Local	X
Data analytics for predictive maintenance of industrial robots [29].	**ML**	X	X	X	Dataset	PC Local	X	Local	X
Enabling of Predictive Maintenance in the Brownfield through Low-Cost Sensors, an IIoT-Architecture and Machine Learning [30].	**ML**	✓	X	X	Dataset	Nube	X	✓	X

Tabla 2. Comparativa de los trabajos consultados parte 2.

Trabajo	ML/ EML	Diferentes máquinas	Historial de mantenimiento	Basado en CPS	Datos de entrenamiento	Ejecutable en	Escalable	Centralizado	Auto predecible
A RUL calculation approach based on physical-based simulation models for predictive maintenance [32].	**ML**	✓	X	X	Genéricos	PC Local Simulación	✓	Local	X
Predictive Maintenance System for Efficiency Improvement of Heating Equipment [33].	**ML**	X	X	X	Dataset	PC Local	✓	Local	X
Predictive analysis for industrial maintenance automation and optimization using a smart sensor network [34].	X	✓	✓	X	NA	Supervisor	✓	Local	X
Intelligent predictive maintenance control using augmented reality [35].	X	✓	✓	✓	Genéricos	Nube	X	✓	X
Predictive maintenance for motors based on vibration analysis with compact rio [36].	X	X	✓	X	Genéricos	PC Local	X	Local	X
Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning [37].	**ML**	X	X	X	Dataset	PC Local	X	Local	X
Failure Prediction Model for Predictive Maintenance [38].	**ML**	✓	X	✓	Genéricos	CSP	X	✓	X
Machine learning for predictive maintenance of industrial machines using IoT sensor data [39].	**ML**	✓	X	X	Genéricos	Nube	✓	✓	X
An Industry 4.0-Enabled Low-Cost Predictive Maintenance Approach for SMEs [40] (Sezer et al., 2018).	**ML**	X	X	✓	Genéricos	CPS/Nube	X	✓	X
A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance [15].	**ML**	X	X	X	Genéricos	PC Local	X	Local	X
Predictive maintenance for wind turbine diagnostics using vibration signal analysis based on collaborative recommendation approach [41].	**ML**	X	X	X	Dataset	PC Local	✓	Local	X

Tabla 3. Comparativa de los trabajos consultados parte 3.

Trabajo	ML/ EML	Diferentes máquinas	Historial de mantenimiento	Basado en CPS	Datos de entrenamiento	Ejecutable en	Escalable	Centralizado	Auto predecible
Real-time predictive maintenance for wind turbines using Big Data frameworks [31].	**ML**	X	✓	X	Dataset	PC Local	✓	Local	X
An adaptive machine learning decision system for flexible predictive maintenance [42].	**ML**	X	X	✓	Dataset	CPS	X	✓	X
Machine Learning approach for Predictive Maintenance in Industry 4.0 [43].	**ML**	X	X	X	Dataset	PC Local	X	Local	X
iSTEP, an Integrated Self-Tuning Engine for Predictive Maintenance in Industry 4.0 [44]	X	X	X	✓	Dataset	CPS/PC Local	X	✓	X
Web-based tool for predicting the Remaining Useful Lifetime of Aircraft Components [45].	**ML**	✓	X	✓	Genéricos	CPS	X	✓	X
Frequent Chronicle Mining: Application on Predictive Maintenance [46].	**ML**	✓	X	X	Dataset	PC Local	X	Local	X
System modeling based on machine learning for anomaly detection and predictive maintenance in industrial plants [47]	**ML**	✓	X	✓	Genéricos	CPS/Nube	X	✓	X
Data Driven Predictive Maintenance of Distribution Transformers [48].	**ML**	✓	✓	X	Dataset	PC Local	X	Local	X
Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture [49].	**ML**	✓	X	✓	Dataset	CPS/Nube	X	✓	X

3.1 Discusión

El análisis del estado del arte revela que el mantenimiento predictivo (PdM) ha evolucionado significativamente con la integración de técnicas de Machine Learning (ML). La literatura revisada muestra diversas metodologías aplicadas en distintos sectores industriales, con un énfasis creciente en la calidad del preprocesamiento de datos como un factor determinante para la precisión de los modelos predictivos.

Uno de los hallazgos más relevantes es que la mayoría de los estudios coinciden en que el preprocesamiento de datos juega un papel clave en la precisión de las predicciones. Sin embargo, se observa que muchas investigaciones utilizan técnicas estándar de limpieza y normalización, sin considerar el origen de los datos, la influencia del tipo de sensor o las condiciones operativas en la adquisición de datos. Este aspecto representa una brecha en la investigación actual, ya que la calidad de los datos impacta directamente en la capacidad del modelo para detectar patrones relevantes y anticipar fallas.

En cuanto a los algoritmos utilizados, se identificó que Random Forest (RF), Support Vector Machines (SVM) y Redes Neuronales son los enfoques más empleados en el PdM. Estos algoritmos han demostrado ser eficaces en distintos entornos, pero su rendimiento varía según la calidad de los datos de entrada. Investigaciones previas han resaltado que el uso de modelos complejos como redes neuronales profundas (DNN) puede mejorar la precisión en ciertos escenarios, pero también aumenta la complejidad computacional y los requisitos de entrenamiento. Esto refuerza la necesidad de contar con un preprocesamiento eficiente que optimice el uso de recursos y mejore la interpretabilidad de los modelos.

Otro aspecto importante es la adopción de Sistemas Ciberfísicos (CPS) e Internet de las Cosas (IoT) en la industria. La combinación de sensores inteligentes con plataformas en la nube ha permitido mejorar la recolección y análisis de datos en tiempo real. No obstante, los estudios revisados muestran que la integración de estas tecnologías todavía enfrenta desafíos, como la heterogeneidad de los datos y la falta de estándares unificados para su procesamiento. Esto sugiere que futuras investigaciones deben enfocarse en desarrollar estrategias de preprocesamiento más robustas y adaptables a diferentes tipos de sensores y entornos industriales.

En términos de escalabilidad, los estudios muestran diferencias en la aplicación de los modelos en distintos escenarios. Algunos enfoques han logrado automatizar el mantenimiento predictivo en fábricas inteligentes, mientras que otros aún dependen de ajustes manuales para mejorar su precisión. Esto indica que, aunque el PdM basado en ML es prometedor, todavía existen limitaciones en términos de aplicabilidad y costos de implementación en la industria real.

Finalmente, es importante mencionar que la mayoría de las investigaciones revisadas se han basado en conjuntos de datos previamente recopilados o simulaciones controladas, sin pruebas en entornos industriales reales. Esto plantea una limitación importante en la validez de los resultados, ya que las condiciones de operación pueden influir en la efectividad de los modelos de predicción. La falta de estudios aplicados directamente en la industria refuerza la necesidad de validar los enfoques en escenarios reales para asegurar su efectividad.

Si bien los trabajos recientes han destacado el impacto del preprocesamiento en la detección de fallas, la mayoría se limita a técnicas genéricas de normalización y reducción de ruido sin considerar las particularidades del entorno de los datos. Estas limitaciones incluyen la falta de personalización de las técnicas al tipo de sensor, las condiciones ambientales y el contexto operativo y aspectos críticos en entornos industriales complejos. La metodología propuesta en esta investigación aborda estas carencias mediante la personalización del preprocesamiento, optimizando características clave como las amplitudes de vibración y las frecuencias dominantes, lo que resultó en una mejora del 96% en precisión frente al promedio de 93% reportado en estudios previos [78, 79, 80]. Este enfoque subraya la importancia de un tratamiento diferenciado para maximizar la confiabilidad de los modelos predictivos. Sin embargo, hay margen para explorar nuevas técnicas y métodos de preprocesamiento que puedan optimizar aún más la calidad de las predicciones en entornos industriales complejos.

Capítulo IV

Metodología de solución

3. Metodología de solución

Este Capítulo muestra la metodología de solución empleada en esta investigación, se organiza en cuatro fases principales para abordar de manera estructurada el desarrollo de un modelo de PdM basado en ML. Cada fase incluye actividades específicas para garantizar la correcta adquisición, preprocesamiento, ajuste y validación de los datos.

Fase 1: Definición y Adquisición de Datos

Esta fase se enfoca en la identificación y recolección de datos relevantes para el análisis. Las actividades realizadas incluyen:

- Selección de sensores adecuados (vibración, temperatura y acústicos) en función de las características del sistema analizado.
- Configuración de parámetros de adquisición, como frecuencia de muestreo y condiciones operativas.
- Recolección de datos experimentales provenientes de maquinaria en funcionamiento bajo diversas condiciones.
- Revisión y selección de conjuntos de datos disponibles en repositorios en línea para complementar la información.

Fase 2: Preprocesamiento de Datos

El objetivo de esta fase es preparar los datos recolectados para su uso en los modelos predictivos. Las actividades clave incluyen:

- Filtrado de ruido en las señales recolectadas para garantizar la calidad de los datos.
- Segmentación de las señales en intervalos específicos que faciliten el análisis de patrones.
- Extracción de características relevantes, como amplitudes de vibración y frecuencias dominantes.
- Evaluación preliminar de los datos para verificar su calidad y representatividad.

Fase 3: Ajuste del Preprocesamiento

En esta fase se busca optimizar las técnicas aplicadas durante el preprocesamiento para mejorar la calidad de los datos y su impacto en los modelos. Las actividades realizadas incluyen:

- Validación de las técnicas de filtrado y segmentación utilizadas en la fase anterior.
- Ajustes iterativos en los parámetros del preprocesamiento, adaptándolos a las condiciones específicas de los sensores y datos recolectados.
- Comparación de los datos ajustados con los originales para identificar mejoras significativas.

Fase 4: Validación de Modelos Predictivos

La última fase está enfocada en el desarrollo, entrenamiento y evaluación de los modelos predictivos utilizando los datos preprocesados. Las actividades realizadas incluyen:

- Entrenamiento de algoritmos predictivos como Árboles de decisión y Redes neuronales con los datos procesados.
- Análisis del rendimiento de los modelos utilizando métricas como *precision y recall*.
- Comparación de los resultados obtenidos con modelos existentes para identificar ventajas y limitaciones del enfoque propuesto.

4.1 Definición y Adquisición de Datos

En la primera etapa del desarrollo de esta tesis, fue fundamental identificar los datos relevantes y definir su origen. Para ello se realizó una investigación exhaustiva sobre las máquinas comúnmente empleadas en la industria y sus características mecánicas.

La revisión de la literatura muestra una amplia variedad de sensores utilizados para la detección de anomalías o fallas en máquinas industriales. Estos incluyen sensores de medición térmica, dispositivos piezoeléctricos para la captura de señales acústicas y unidades de medición inercial (IMU). Dada esta variedad de dispositivos de medición, que generan distintos tipos de datos, es esencial seleccionar las características más relevantes y representativas para la predicción de fallas, a fin de optimizar la precisión y eficiencia del análisis.

La figura 25 muestra tres de los sensores que se utilizan comúnmente para obtener datos de máquinas.

Figura 25. Sensores utilizados en la detección de fallas.

En el estudio de [50] se proponen diversas técnicas para la predicción de fallas, tales como el análisis de vibración, análisis de aceite, análisis térmico y análisis acústico. Cada uno de estos métodos emplea uno o varios sensores que proporcionan información crítica sobre el estado actual de los componentes de una máquina, permitiendo así llevar un registro histórico de su desgaste.

La investigación de [51] presenta un enfoque de análisis de vibración utilizando una placa de desarrollo equipada con un sensor de vibración, conectada a una matriz de puertas lógicas programable. En este sistema, los datos de vibración generados al encender un motor eléctrico de corriente continua se procesan para detectar patrones indicativos de falla.

El trabajo de [52] explora la predicción de fallas en una banda transportadora de una planta embotelladora, destacando el análisis de vibración como una técnica adecuada para anticipar fallas en distintos tipos de rodamientos.

Por su parte, [53] proponen el uso de sensores de temperatura y vibración para detectar anomalías en sistemas de producción de energía hidroeléctrica. En esta investigación, se instalaron múltiples sensores de cada tipo en las bombas, según su importancia, y los datos recolectados fueron empleados para la predicción de fallas en el equipo.

En [54] se describen las inversiones realizadas en la industria petrolera, donde los equipos mecánicos rotativos, como motores de inducción, compresores y bombas, son esenciales para la operación. En su trabajo, implementan algoritmos de ML para predecir fallas en un compresor de refuerzo y destacan que el uso de estos algoritmos puede reducir riesgos y costos operativos.

En el ámbito de la fabricación de piezas mecánicas, el uso de equipos de fresado es fundamental; sin embargo, estos equipos son sensibles y una calibración incorrecta puede causar accidentes o afectar la calidad de las piezas producidas. En el estudio de [55], se propone el uso de sensores acústicos y de vibración para detectar fallas o desajustes en las fresas, así como el uso de algoritmos de ML para la predicción del desgaste de los componentes.

La Tabla 4 compara los distintos tipos de industria, sensores, maquinaria evaluada y técnicas utilizadas en los estudios mencionados.

La información recopilada de estas investigaciones hace evidente que los equipos y componentes rotativos están presentes en una variedad de industrias, haciendo de la predicción de fallas en motores un objetivo clave en la Industria 4.0. Un modelo de predicción de fallas en máquinas rotativas beneficiará a numerosos sectores industriales; por lo tanto, el enfoque de esta tesis se centra en un motor eléctrico y sus rodamientos.

Tabla 4. Tabla comparativa de las predicciones de fallas en las industrias.

Artículo	Tipo de industria	Sensores utilizados	Máquina o componente evaluado	Técnica utilizada
Predictive maintenance, its implementation and latest trends [50].	N/A	-IMU -Térmicos -Acústicos	N/A	-Análisis de vibración -Análisis térmico -Análisis acústico
Live Demonstration: Autoencoder-Based Predictive Maintenance for IoT [51].	N/A	-IMU	-Motor DC	-Análisis de vibración
Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts [52].	-Embotelladora	-IMU	-Banda transportadora	-Análisis de vibración
Industrial Internet of Things monitoring solution for advanced predictive maintenance applications [53].	-Hidroeléctrica	-IMU -Temperatura	-Motor AC	-Análisis de vibración -Análisis térmico
Data Driven Predictive Maintenance of Distribution Transformers [48].	-Eléctrica	-Temperatura -Humedad	-Motor AC -Condensador	-Análisis térmico -Análisis de humedad
Predictive Maintenance of Oil and Gas Equipment using Recurrent Neural Network [54].	-Petrolera	N/A	-Motor AC	N/A
Machine Learning Framework for Predictive Maintenance in Milling [55].	-Mecanizado	-Acústicos -Térmicos	-Fresas -Rodamientos	-Análisis de vibración -Análisis acústico
A RUL Calculation Approach based on Physical-based Simulation Models [32].	-Mecanizado	-Fotoeléctricos	-Fresas -Rodamientos	-Análisis de frecuencias
Predictive maintenance for motors based on vibration with compact rio [36].	N/A	-IMU	-Motor AC	-Análisis de vibración
Predictive maintenance for wind turbine diagnostics using vibration signal based [41].	-Eolo eléctrica	-IMU	-Generador	-Análisis de vibración

Dado que el motor eléctrico es el equipo objetivo del estudio, es fundamental comprender las partes que lo componen y los dispositivos que pueden emplearse para la recolección de datos. A continuación, se describen los principales componentes de un motor eléctrico:

1. Rotor: Componente móvil responsable del giro.
2. Rodamientos: Elementos que minimizan las vibraciones y facilitan un movimiento estable.
3. Eje: Estructura que guía el movimiento de rotación.
4. Bobinado: Devanado o inductor que genera los campos magnéticos necesarios para el funcionamiento del motor.
5. Carcasa: Estructura externa que protege los componentes internos.
6. Ventilador: Mecanismo que evacua el calor generado hacia el exterior.
7. Estator: Parte fija que alberga el núcleo magnético del motor.

Figura 26. Motor eléctrico [56].

Para determinar los datos a recolectar, fue esencial contar con información detallada sobre los sensores utilizados en la industria, lo cual permitió definir las características de los datos en función de las capacidades de detección de cada sensor.

Se investigaron tres tipos de sensores: IMU, acústico y térmico, los cuales son ampliamente empleados para la captura de datos en sistemas industriales, incluyendo aplicaciones de predicción de fallas.

La IMU (Figura 27) es un dispositivo electrónico que destaca por su capacidad de medir velocidad, orientación y fuerzas gravitacionales mediante una combinación de acelerómetros y giroscopios. Este dispositivo detecta la tasa de aceleración mediante uno o varios acelerómetros y registra cambios en las características de rotación utilizando giroscopios.

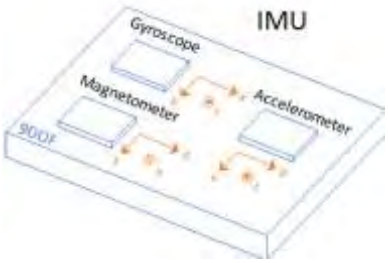


Figura 27. Unidad de medición inercial [57].

Para este propósito, se utilizó una IMU con el objetivo de recolectar datos de aceleración en relación con un eje específico, en este caso el eje Y, donde se obtuvieron valores tanto negativos como positivos que representan las vibraciones generadas por el motor en funcionamiento.

Por otro lado, se investigó el sensor acústico o micrófono (Figura 28), un dispositivo utilizado para transformar ondas sonoras en energía eléctrica. Este sensor cuenta con un diafragma, atraído por un electroimán, que al vibrar altera la corriente transmitida en el circuito en respuesta a las distintas presiones sonoras.

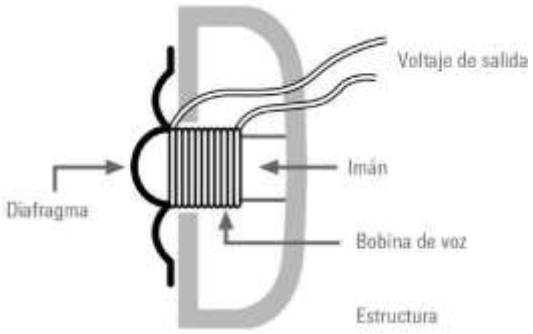


Figura 28. Sensor acústico [58].

Asimismo, los sensores térmicos o detectores de calor (Figura 29) son dispositivos que transforman los cambios de temperatura, mediante convección, en señales eléctricas que pueden ser procesadas por equipos eléctricos o electrónicos.

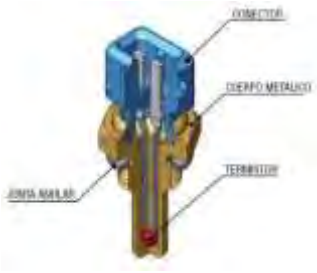


Figura 29. Sensor Térmico [59].

Considerando la información obtenida sobre la máquina objetivo, sus componentes y los dispositivos disponibles para monitorear su funcionamiento, fue necesario identificar conjuntos de datos que pudieran contener información útil para evaluar los componentes de un motor eléctrico.

Se realizó una búsqueda exhaustiva de repositorios de datos de equipos industriales relacionados con el PdM. Como resultado, se seleccionaron tres fuentes de datos en línea que concentran grandes volúmenes de información con diversas características:

1. **Kaggle**: Kaggle es una comunidad en línea para científicos de datos y profesionales del ML. Permite a los usuarios encontrar y publicar conjuntos de datos, explorar y construir modelos en un entorno web, colaborar con otros científicos de datos y participar en concursos de ciencia de datos. La plataforma alberga miles de conjuntos de datos públicos y fragmentos de código (denominados "kernels") [60].

2. **Google Dataset Search**: Este motor de búsqueda de Google, lanzado el 5 de septiembre de 2018, facilita a los investigadores la localización de conjuntos de datos en línea disponibles para uso público. Google *Dataset Search* está dirigido especialmente a científicos de datos y periodistas [61].

3. ***Prognostics Center of Excellence*** **(NASA)**: Este repositorio reúne conjuntos de datos donados por universidades, agencias y empresas, enfocados exclusivamente en series temporales para pronósticos, desde un estado nominal hasta un estado de falla. La recopilación de datos en este repositorio es un proceso continuo [62]

Debido a la naturaleza industrial de estos datos, existen dificultades para acceder a conjuntos de datos reales de mantenimiento; la mayoría de los datos disponibles han sido generados por investigadores para fines específicos. Cabe mencionar que la fabricación de datos puede comprometer la fiabilidad de los resultados obtenidos en las predicciones de los algoritmos.

En el sitio del *Prognostics Center of Excellence* de la NASA, se encontró un conjunto de datos que contiene información de sensores IMU instalados en chumaceras de ejes de motores eléctricos, específicamente para la detección de vibraciones. Este conjunto de datos incluye un artículo de referencia [63] que describe el proceso de captura y documenta los momentos en que ocurrieron fallas en los equipos monitorizados. Los datos incluyen múltiples muestras tomadas por cuatro acelerómetros ubicados en un motor eléctrico de una fábrica, capturadas mientras el motor estaba en operación, y finalizan en el momento de una falla.

4.2 Fase de Preprocesamiento de Datos y Modelado Predictivo

Para comprender mejor la estructura de los datos y las técnicas empleadas en la predicción de fallas, se realizó una revisión de investigaciones relacionadas con el mantenimiento industrial. En este proceso se identificaron modelos relevantes para la predicción de fallas, definidos como:

"Un modelo de PdM *es un prototipo o guía que sirve de referencia para realizar la predicción de fallas o estimar la vida útil restante de una máquina o de sus componentes* [42]*."*

El modelo propuesto en el trabajo de [64] consta de seis fases (Figura 30) y describe un flujo integral para el análisis de datos de sensores conectados a maquinaria industrial. En la primera fase, se obtienen las mediciones de los sensores, los cuales reciben señales de la máquina y las transforman en voltajes que pueden ser transmitidos a una puerta de enlace de Internet de las Cosas Industrial (IIoT).

Posteriormente, las señales de datos son enviadas a un centro de procesamiento local, donde se concentran las señales provenientes de diversos sensores que operan con diferentes protocolos de comunicación. En este centro, las señales se procesan para permitir una interpretación precisa de las condiciones de la máquina.

A continuación, los datos son transmitidos a un centro de preparación donde se desarrolla un modelo de predicción de fallas. Este modelo, creado con algoritmos de ML, calcula la Vida Útil Restante (RUL) de los componentes de la máquina. Finalmente, la información de RUL se envía a un módulo de simulación, donde se realiza una predicción de falla; los resultados obtenidos son interpretados para planificar y ejecutar el PdM a cargo del personal técnico.

Figura 30. Modelo de mantenimiento de [64].

Otro modelo de PdM es el presentado por [31], el cual consta de seis bloques (Figura 31). En el Bloque 1, se conectan sensores a una serie de aerogeneradores en una granja, utilizando un sistema de Supervisión, Control y Adquisición de Datos (SCADA), el cual permite el monitoreo remoto y la interconexión de múltiples granjas de aerogeneradores a una única central de procesamiento.

En el Bloque 2 se reciben mensajes con los datos recopilados por los sensores en las granjas, los cuales son enviados al Bloque 3, donde un operador supervisa y compara estos datos en tiempo real. A continuación, en el Bloque 4, se encuentran los datos históricos, que sirven de referencia para evaluar las condiciones actuales de los aerogeneradores.

El Bloque 5 recibe y almacena los datos en un repositorio central, manteniendo un historial que resulta fundamental para el Bloque 6, donde se encuentra el generador de modelos de predicción. Este último bloque utiliza el historial de datos para entrenar los algoritmos de predicción, mejorando la precisión y efectividad en la anticipación de fallas.

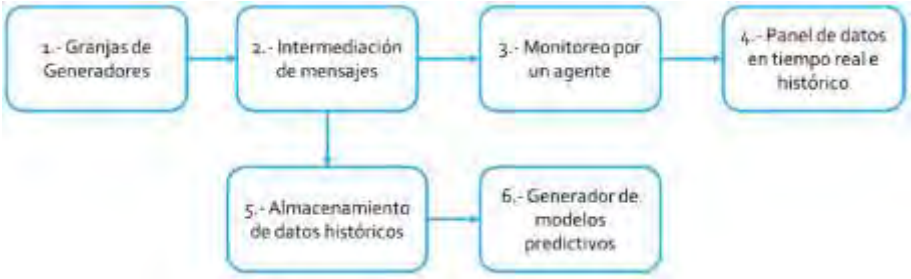


Figura 31. Modelo de mantenimiento de [31].

El modelo de PdM presentado por [65] se organiza en cuatro capas (Figura 32). La capa de persistencia se encarga del almacenamiento de datos de mantenimiento, proporcionando una base de datos histórica para el análisis. La capa de procesamiento por lotes analiza la información proveniente de los sistemas heredados y transfiere los resultados a la siguiente capa.

La capa de procesamiento en tiempo real recibe información directamente de los sensores conectados a una fábrica digital equipada con sistemas de IIoT. En esta capa se ejecutan las funciones de sensado, detección, predicción y toma de decisiones sobre las tareas necesarias de mantenimiento. Finalmente, la capa de interacción permite al usuario configurar, monitorear y visualizar la información generada en las capas de análisis y predicción, facilitando el control y supervisión del proceso de mantenimiento.

Figura 32. Modelo de mantenimiento de [65].

Otro modelo de PdM es el propuesto por [39]. En este modelo, los sensores integrados en una serie de motores están conectados a un controlador lógico programable (PLC) que captura los datos sobre la condición de la máquina. Estos datos se transmiten a través de una conexión de red hacia una computadora local, que a su vez los envía a un servidor en la nube.

En el servidor se ejecutan algoritmos de ML, los que procesan los datos para predecir las fallas. Posteriormente, los resultados de la predicción son devueltos a la computadora local, donde un usuario los interpreta para programar las intervenciones de mantenimiento necesarias.

Figura 33. Modelo de mantenimiento de [39].

Algunos modelos de mantenimiento combinan sistemas ciberfísicos con servidores en la nube, como es el caso del modelo presentado por [40] (Figura 34). Este modelo controla los procesos de producción para asegurar la calidad de los productos al momento de ser entregados al cliente.

En este enfoque se monitorean las máquinas industriales mediante sensores; los datos capturados son transmitidos de forma inalámbrica al sistema ciberfísico donde se organizan para su posterior almacenamiento en un repositorio de datos. Una vez almacenados estos datos, son enviados a la nube donde se analizan para identificar signos de desgaste en las máquinas y, programar intervenciones de mantenimiento que garanticen la calidad del producto final y la satisfacción del cliente.

Figura 34. Modelo de mantenimiento de [40].

Se llevó a cabo una búsqueda de estudios que proporcionan información sobre los algoritmos empleados en la predicción de fallas en la industria. Se seleccionaron artículos en los que se menciona el uso de uno o varios algoritmos para la predicción de fallas o para la estimación de la Vida Útil Restante (RUL). En estos trabajos se observa el uso de algoritmos como SVM, RF, LR, entre otros.

La Tabla 5 presenta una lista de los estudios revisados y los algoritmos de predicción empleados en cada uno de ellos.

Tabla 5. Algoritmos utilizados para la predicción de fallas.

Trabajo	RF	LoR	SVM	GBT	LR	NN	FL	BN	DT	KM	KNN	CART	NB
iSTEP, an Integrated Self-Tuning Engine for Predictive Maintenance in Industry 4.0 [44].	x	x	x	x	--	--	--	--	--	--	--	--	--
Data analytics for predictive maintenance of industrial robots [29].	--	--	--	--	x	--	--	--	--	--	--	--	--
Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture [49].	--	--	--	--	--	x	x	--	--	--	--	--	--
A Big Data implementation of the MANTIS reference architecture for predictive maintenance [21].	--	--	x	--	--	x	--	x	--	--	--	--	--
Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning [37].	x	--	x	--	x	--	--	--	x	x	x	--	--
Predictive Maintenance System for Efficiency Improvement of Heating Equipment (Santiago et al., 2019).	x	--	--	--	--	--	--	--	--	--	--	--	--
An Industry 4.0-Enabled Low Cost Predictive Maintenance Approach for SMEs [40].	--	--	--	--	x	x	--	--	--	--	--	--	--
Approach for a Holistic Predictive Maintenance Strategy by Incorporating a Digital Twin [64].	--	--	x	--	x	--	--	--	--	--	x	--	--
Real-time predictive maintenance for wind turbines using Big Data frameworks [31].	x	--	--	--	--	x	--	--	--	--	--	--	--
Machine learning for predictive maintenance of industrial machines using IoT sensor data [39].	--	--	x	--	--	x	--	--	--	--	--	x	x
Total	4	1	5	1	4	5	1	1	1	1	2	1	1

Las abreviaciones utilizadas en la tabla incluyen los siguientes significados: **RF** para *Random Forest*, **LoR** para *Logistic Regression*, **SVM** para *Support Vector Machines*, **GBT** para *Gradient-Boosted Tree*, **LR** para *Linear Regression*, **NN** para *Neural Networks*, **FL** para *Fuzzy Logic*, **BN** para *Bayesian Networks*, **DT** para *Decision Tree*, **KM** para *K-Means*, **KNN** para *K Nearest Neighbor*, **CART** para *Classification and Regression Trees*, y **NB** para *Naive Bayes*.

Dado que existen una gran variedad de algoritmos para realizar predicciones, en esta comparación se identificó que los algoritmos más comúnmente utilizados son RF, SVM, LR y NN. Por consiguiente, estos fueron los algoritmos seleccionados para su implementación en los modelos de predicción de fallas.

Antes de llevar a cabo un preprocesamiento específico se realizó una prueba preliminar utilizando varios algoritmos de predicción. El conjunto de datos consta de tres series de registros, cada una compuesta por lecturas tomadas en un período de tiempo determinado. En estos conjuntos, los datos iniciales reflejan un funcionamiento adecuado de los rodamientos del motor, mientras que los datos finales indican el punto en el que se requiere mantenimiento correctivo. A continuación, se describe la información de cada conjunto de datos:

- **Set de datos 1**:
    - Duración del registro: 22 de octubre de 2003, 12:06:24 a 25 de noviembre de 2003, 23:39:56.
    - Número de archivos: 2156.
    - Número de canales: 8.
    - Disposición de canales: Balero 1 – Ch1 & 2; Balero 2 – Ch3 & 4; Balero 3 – Ch5 & 6; Balero 4 – Ch7 & 8.
    - Intervalo de recolección de datos: Cada 10 minutos.
    - Formato del archivo: ASCII.
    - Descripción: Al finalizar la prueba, se detectó un defecto en el balero 3 y una falla en el balero 4.
- **Set de datos 2**:
    - Duración del registro: 12 de febrero de 2004, 10:32:39 a 19 de febrero de 2004, 06:22:39.
    - Número de archivos: 984.
    - Número de canales: 4.
    - Disposición de canales: Balero 1 – Ch1; Balero 2 – Ch2; Balero 3 – Ch3; Balero 4 – Ch4.
    - Intervalo de recolección de datos: Cada 10 minutos.
    - Formato del archivo: ASCII.
    - Descripción: Al finalizar la prueba, se detectó una falla en el balero 1.

- **Set de datos 3**:
    - Duración del registro: 4 de marzo de 2004, 09:27:46 al 17 abril de 2004, 19:01:57.
    - Número de archivos: 6324.
    - Número de canales: 4.
    - Disposición de canales: Balero 1 – Ch1; Balero 2 – Ch2; Balero 3 – Ch3; Balero 4 – Ch4.
    - Intervalo de recolección de datos: Cada 10 minutos.
    - Formato del archivo: ASCII.
    - Descripción: Al finalizar la prueba, se detectó una falla en el balero 3.

Para las primeras pruebas, se eligió el set de datos 1, del cual se extrajo una muestra para formar el conjunto de datos A. Este conjunto está compuesto por 204,800 filas correspondientes a las lecturas realizadas en los cuatro rodamientos, dividido en 10 clases.

La Figura 35 muestra la cabecera del conjunto de datos A.

```
In [3]: db.head(204800) #24800 filas en total
```

Out[3]:

	0	1	2	3	4
0	-0.049	-0.071	-0.132	-0.010	0
1	-0.042	-0.073	-0.007	-0.105	0
2	0.015	0.000	0.007	0.000	0
3	-0.051	0.020	-0.002	0.100	0
4	-0.107	0.010	0.127	0.054	0
...	...	...	...	...	...
204795	0.000	0.005	0.002	0.000	9
204796	0.002	0.002	0.005	0.000	9
204797	0.002	0.002	0.002	0.000	9
204798	0.002	0.002	0.005	0.000	9
204799	0.000	0.005	0.002	0.000	9

204800 rows × 5 columns

Figura 35. Cabecera de datos del conjunto A.

La figura 36 muestra la distribución de los datos

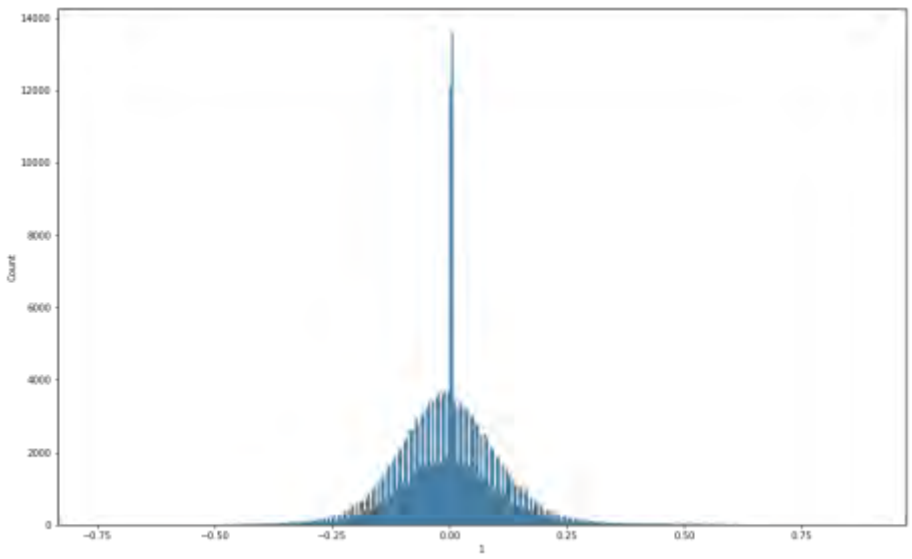


Figura 36. Distribución de datos del conjunto A.

El primer algoritmo puesto a prueba fue el de DT, una técnica de ML que permite construir modelos predictivos de análisis de datos basados en ciertas características o propiedades. Un árbol de decisión es un grafo acíclico dirigido que facilita la toma de decisiones mediante una estructura jerárquica.

En cada nodo de bifurcación del grafo, se evalúa una característica específica del vector de características. Si el valor de la característica está por debajo de un umbral determinado, se sigue la rama izquierda; en caso contrario se sigue la rama derecha. Cuando se alcanza un nodo hoja, el modelo toma una decisión sobre la clase a la que pertenece el ejemplo evaluado.

La elección del mejor atributo en un árbol de decisión se establece mediante la entropía, seleccionando aquel que ofrezca la mayor ganancia de información. La función empleada puede variar, pero en su forma más básica se expresa como:

$$E = -\left(\frac{|p|}{|d|}\right) log_2 \left(\frac{|p|}{|d|}\right) - \left(\frac{|n|}{|d|}\right) log_2 \left(\frac{|n|}{|d|}\right), \qquad (1)$$

donde ***p*** representa el conjunto de ejemplos positivos, ***n*** el de ejemplos negativos, y ***d***, el total de ejemplos. Para obtener una descripción más detallada, véase [66].

Para evaluar el desempeño de este algoritmo, el conjunto de datos se dividió en un 70% para entrenamiento y un 30% para pruebas. Posteriormente se llevó a cabo el entrenamiento de un modelo de DT.

Finalmente se realizó una prueba de predicciones cuyos resultados se presentan en la Figura 37.

```
In [19]: comp = pd.DataFrame({'Reales': y_test, 'Predicciones': y_prds})

In [20]: comp.head(20)
```

Out[20]:

	Reales	Predicciones
21751	1	7
48071	2	8
92969	4	5
115011	5	4

Figura 37. Comparación de resultados con DT *en el conjunto A.*

Se utilizó la métrica *exactitud* para evaluar la precisión del modelo, obteniéndose un valor de 0.2625.

Para observar la variación de los datos se creó una matriz de confusión (Figura 38).

```
Out[26]: array([[ 978,  841,  912,  843,  903,  227,  817,  497,  196,    2],
                [ 888,  825,  840,  846,  876,  256,  898,  578,  190,    0],
                [ 857,  844,  845,  866,  897,  248,  784,  546,  180,    0],
                [ 921,  874,  826,  910,  837,  272,  850,  573,  208,    0],
                [ 248,  243,  276,  261,  260, 4311,  247,  179,   72,    0],
                [ 247,  243,  276,  259, 4597,   94,  247,  179,   72,    0],
                [ 731,  775,  812,  836,  764,  214,  862,  688,  359,    0],
                [ 516,  541,  518,  551,  505,  167,  751, 1479, 1019,    0],
                [ 176,  186,  220,  205,  163,   51,  376, 1035, 3662,    0],
                [   0,    1,    0,    0,    1,    0,    0,    0,    0, 6214]],
               dtype=int64)
```

Figura 38. Matriz de confusión con DT *en el conjunto A.*

En la figura 39 se puede observar la dispersión de los datos reales con los datos de la predicción.

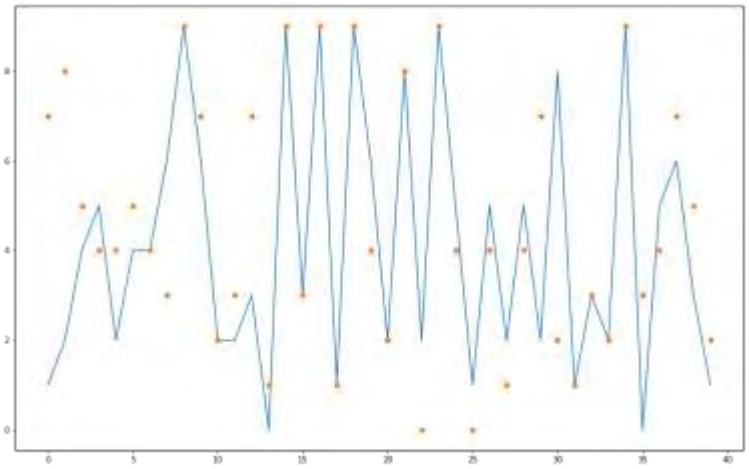


Figura 39. Dispersión de datos con DT *en el conjunto A.*

Se realizó una validación cruzada para identificar el mejor conjunto de características y mejorar los resultados de la predicción. Los mejores parámetros obtenidos fueron:

DecisionTreeClassifier(max_depth=70, max_features=0.8,min_samples_split=0.1)

Con estos parámetros se entrenó nuevamente el modelo, obteniendo una precisión de 0.2964. Aunque se observó una mejora, los resultados de este algoritmo con el conjunto de datos siguen siendo bajos. En la Figura 40 se muestra la dispersión de los datos de predicción en comparación con los valores reales.

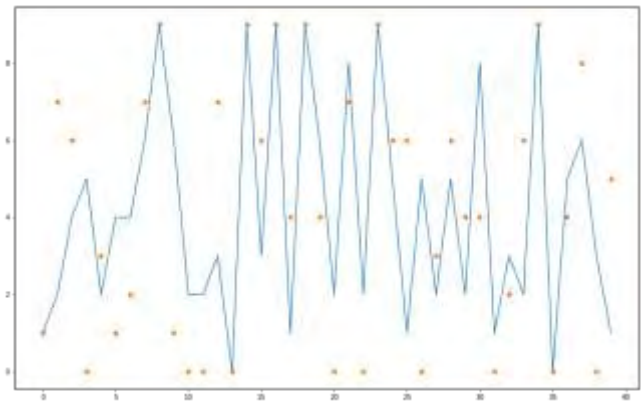


Figura 40. Dispersión de datos con DT *y validación cruzada en el conjunto A.*

Posteriormente, se seleccionó el algoritmo de RF. Esta técnica de aprendizaje supervisado genera múltiples DT sobre el conjunto de datos de entrenamiento, y sus resultados se combinan para formar un modelo único y robusto, mejorando la precisión en comparación con los resultados de cada árbol de manera individual.

Cada árbol se construye mediante un proceso en dos etapas:

- Se genera un número considerable de DT con el conjunto de datos. Cada árbol contiene un subconjunto aleatorio de variables m (predictores) de forma que m < M (donde M = total de predictores).
- Cada árbol crece hasta su máxima extensión.

Cada árbol generado en el algoritmo *Random Forest* utiliza un grupo de observaciones aleatorias. Las observaciones no empleadas en la construcción de un árbol son utilizadas para validar el modelo (para más detalles, ver [67]).

Para el entrenamiento, se empleó el 70% de los datos para entrenamiento y el 30% para pruebas. El modelo se entrenó con 500 árboles usando el conjunto de datos A, obteniendo una precisión de 0.2886. Este valor es superior al obtenido con DT sin validación cruzada, aunque inferior al caso de DT con validación cruzada. En la Figura 41 se muestra la dispersión de los datos de predicción en comparación con los valores reales.

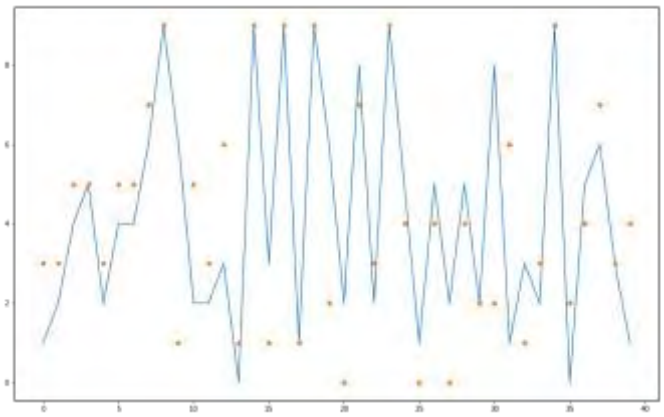


Figura 41. Dispersión de datos con RF *en el conjunto A.*

A continuación, se realizó el entrenamiento utilizando el algoritmo de **Regresión Lineal**. Este algoritmo es una técnica paramétrica usada para predecir variables continuas dependientes en función de un conjunto de variables independientes. Es un método popular que aprende un modelo basado en una combinación lineal de las características del ejemplo de entrada.

Si se desea construir un modelo $f_{w,b}(X)$ como una combinación lineal de características del ejemplo X:

$$f_{w,b}(X) = wx + b, \qquad (2)$$

Donde ***w*** es un vector de parámetros con *D-dimensiones* y ***b*** es un número real. La notación $\boldsymbol{f_{w,b}}$ significa que el modelo ***f*** está parametrizado por dos valores ***w*** y ***b,*** para mayor detalle consultar [66].

Para evaluar la precisión del modelo, se utilizó la métrica de error cuadrático medio (MSE), la cual arrojó un valor de 8.2603. Cabe mencionar que un modelo con alta precisión debería tener un valor de MSE cercano a 0. En la Figura 42 se muestra la dispersión de los datos de predicción en comparación con los valores reales.

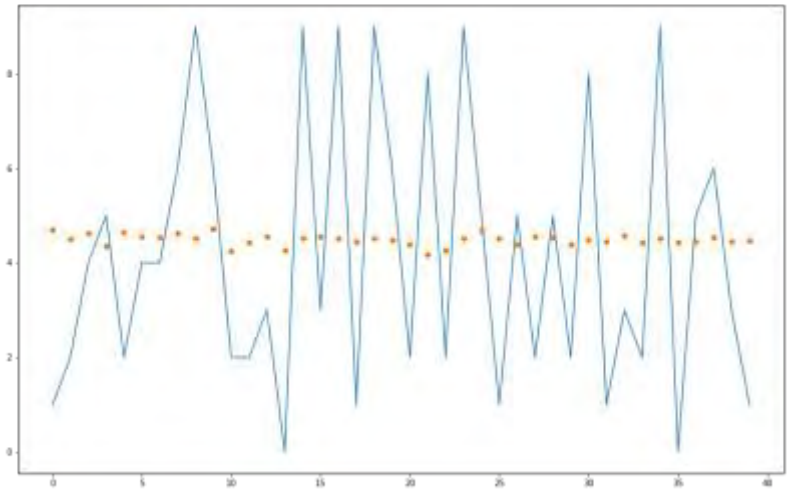


Figura 42. Dispersión de datos con LR *en el conjunto A.*

Adaptación de los Repositorios

Debido a los bajos niveles de precisión en las predicciones, se realizó una adaptación del repositorio para mejorar el rendimiento en los entrenamientos. Partiendo del conjunto de datos A, se generaron cinco conjuntos adicionales (Ver Tabla 6).

Los repositorios fueron adaptados en función de las características necesarias para la predicción. A continuación, se describen los conjuntos creados:

- **Conjunto B**: Se establecieron cuatro clases, categorizando el funcionamiento de los baleros en estado normal, estado moderado, estado de advertencia y estado crítico.
- **Conjunto C**: Este conjunto cuenta con tres clases: estado normal, estado de advertencia y estado crítico.
- **Conjunto C+**: Los datos fueron convertidos a valores numéricos positivos para facilitar su visualización en un plano de dos ejes, manteniendo las clases de estado normal, advertencia y crítico.
- **Conjunto D**: Este conjunto también cuenta con tres clases (estado normal, estado de advertencia y estado crítico), pero se distingue de los anteriores al incluir únicamente los datos de un solo balero, específicamente aquel en el que ocurrió la falla.
- **Conjunto D+**: Es idéntico al conjunto D, excepto que, al igual que en el conjunto C+, los datos fueron convertidos a valores numéricos positivos para su representación en dos ejes.

Tabla 6. Descripción de los conjuntos de datos.

Conjunto	No. de clases	No. de baleros	Registros por clase
B	4	4	0(61440), 1(81920), 2(40960), 3(20480)
C	3	4	0(81920), 1(81920), 2(40960)
C+	3	4	0(81920), 1(81920), 2(40960)
D	3	1	0(81920), 1(81920), 2(40960)
D+	3	1	0(81920), 1(81920), 2(40960)

Implementación y Evaluación de los Algoritmos de Predicción con los Nuevos Repositorios

Con los nuevos conjuntos de datos se procedió a aplicar los mismos algoritmos de predicción para evaluar el rendimiento y observar el comportamiento de los modelos. Comenzando con el conjunto B, se realizó el entrenamiento utilizando el algoritmo de DT.

La Figura 43 muestra la cabecera de los datos del conjunto B.

```
In [3]: db.head(204800) #204800 filas en total
```

Out[3]:

	0	1	2	3	4
0	-0.049	-0.071	-0.132	-0.010	0
1	-0.042	-0.073	-0.007	-0.105	0
2	0.015	0.000	0.007	0.000	0
3	-0.051	0.020	-0.002	0.100	0
4	-0.107	0.010	0.127	0.054	0
...	...	...	...	...	...

Figura 43. Cabecera de datos del conjunto B.

La figura 44 muestra la distribución de los datos y la figura 45 la distribución de las clases en donde se observa que los datos de falla están presentes al centro debido a la alta variabilidad en el punto crítico de vida de los rodamientos.

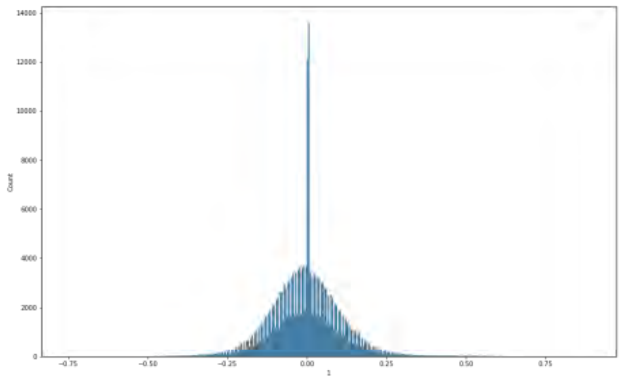


Figura 44. Distribución de datos del conjunto B.

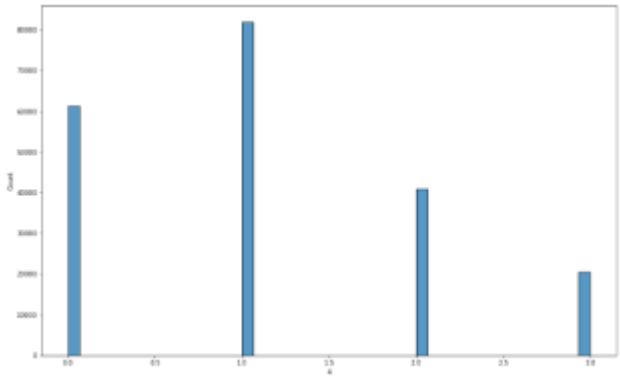


Figura 45. Distribución de clases del conjunto B

El conjunto de datos se dividió en un 70% para entrenamiento y un 30% para pruebas. Posteriormente se realizó el entrenamiento de un modelo de DT. Los resultados de la prueba de predicciones se presentan en la Figura 46.

```
In [20]: comp.head(20)
```

Out[20]:

	Reales	Predicciones
21751	0	3
48071	1	1
92969	1	1
115011	1	2
61087	1	1

Figura 46. Comparación de resultados con DT *en el conjunto B.*

Para evaluar la precisión del modelo, se utilizó la métrica *exactitud*, obteniendo un valor de 0.5377. Para observar la variación de los datos, se generó una matriz de confusión (Figura 47).

```
Out[26]: array([[ 9835,  5608,  2768,   418],
                [ 4649, 15831,  3644,   525],
                [ 2689,  4259,  3711,  1429],
                [  375,   609,  1428,  3662]], dtype=int64)
```

Figura 47. Matriz de confusión con DT *en el conjunto B.*

La Figura 48 muestra la dispersión entre los datos reales (línea azul) y los datos predichos (puntos amarillos).

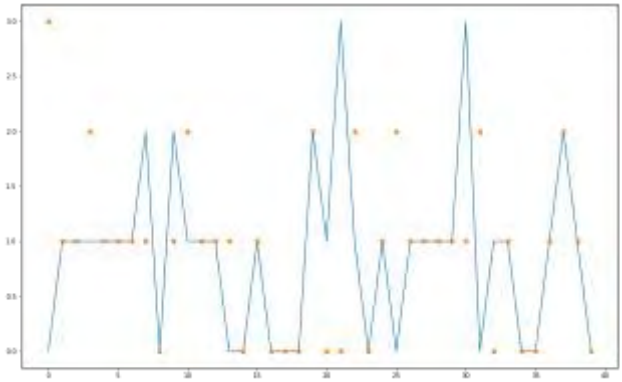


Figura 48. Dispersión de datos con DT *en el conjunto B.*

Se realizó una validación cruzada para optimizar el conjunto de características y mejorar los resultados de la predicción. Los mejores parámetros obtenidos fueron:

DecisionTreeClassifier(max_depth=100, max_features=0.8, min_samples_split=0.1)

Con estos parámetros, se entrenó nuevamente el modelo, logrando una precisión de 0.5595. Aunque hubo una mejora, los resultados de este algoritmo en el conjunto de datos B aún permanecen cercanos al 50%. La dispersión entre los datos se observa en la Figura 49.

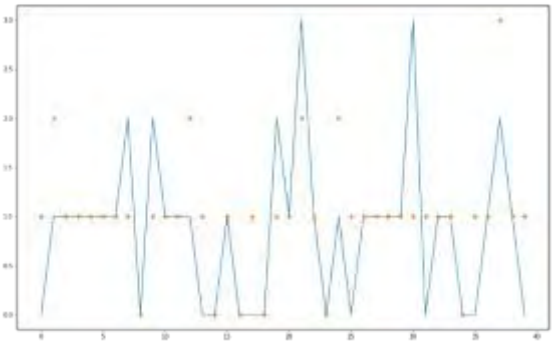


Figura 49. Dispersión de datos con DT *y validación cruzada en el conjunto B.*

Posteriormente se seleccionó el algoritmo de RF para realizar el entrenamiento. Se utilizó el 70% de los datos para el entrenamiento y el 30% para pruebas, y se entrenó el modelo con 500 árboles en el conjunto B, obteniendo una precisión de 0.5863, superior a la obtenida con DT.

La Figura 50 muestra la dispersión de este algoritmo.

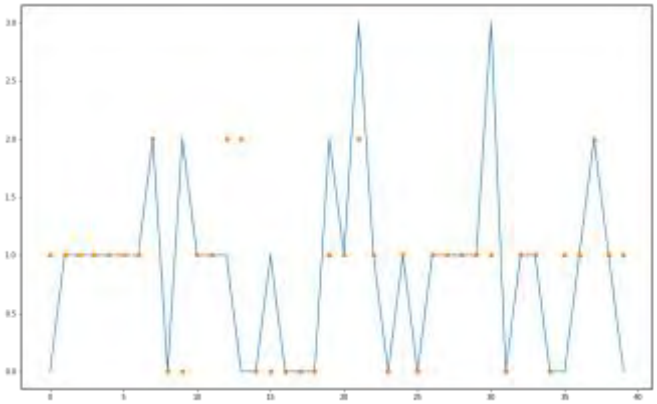


Figura 50. Dispersión de datos con RF *en el conjunto B.*

Utilizando el conjunto de datos B, también se entrenó el modelo con el algoritmo de LR. El error cuadrático medio (MSE) obtenido fue de 0.8872. Aunque este valor es menor que el obtenido con el conjunto A, sigue siendo considerablemente alto.

La Figura 51 se muestra la dispersión de los datos de predicciones con respecto a los datos reales.

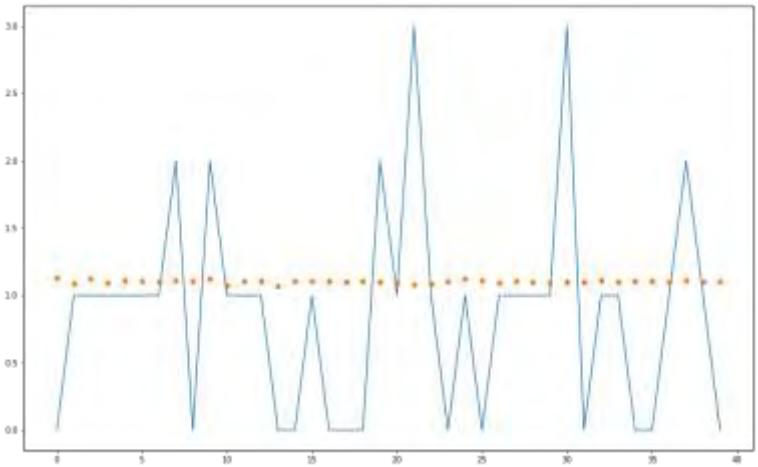


Figura 51. Dispersión de datos con LR *en el conjunto B.*

Continuando con el conjunto C se realizó el entrenamiento para DT

La Figura 52 muestra la cabecera de los datos del conjunto C

```
In [3]: db.head(204800) #204800 filas en total
```

Out[3]:

	0	1	2	3	4
0	-0.049	-0.071	-0.132	-0.010	0
1	-0.042	-0.073	-0.007	-0.105	0
2	0.015	0.000	0.007	0.000	0
3	-0.051	0.020	-0.002	0.100	0
4	-0.107	0.010	0.127	0.054	0
...	...	...	...	...	...

Figura 52. Cabecera de datos del conjunto C.

La Figura 53 muestra la distribución de los datos y la Figura 54 la distribución de las clases.

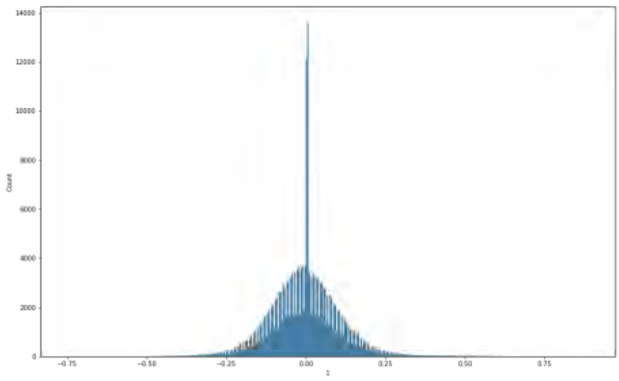


Figura 53. Distribución de datos del conjunto C.

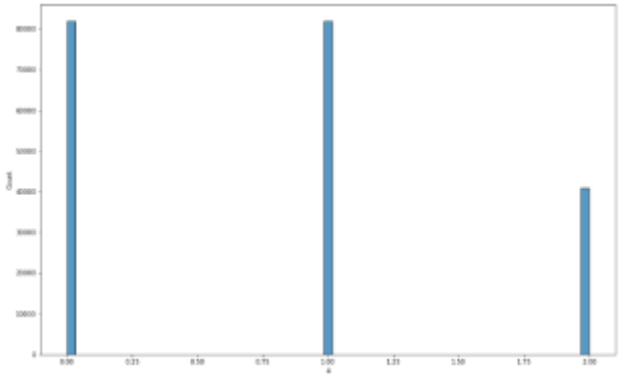


Figura 54. Distribución de clases del conjunto C.

Se dividió el conjunto de datos en un 70% para realizar el entrenamiento y un 30% para realizar las pruebas, posteriormente se realizó el entrenamiento de un modelo de DT.

Posteriormente se realizó una prueba de predicciones la cual obtuvo los siguientes resultados (Figura 55).

```
In [20]: comp.head(20)
```

Out[20]:

	Reales	Predicciones
21751	0	2
48071	0	2
92969	1	1
115011	1	0
61087	0	1
95299	1	1

Figura 55. Comparación de resultados con DT *en el conjunto C.*

Se utilizó la métrica exactitud para determinar la precisión del modelo, con ella se obtuvo un valor de **0.6042**.

Para observar la variación de los datos se creó una matriz de confusión (Figura 56).

```
Out[26]: array([[14142,  8223,  2331],
                [ 6446, 15744,  2433],
                [ 2190,  2691,  7240]], dtype=int64)
```

Figura 56. Matriz de confusión con DT *en el conjunto C.*

En la figura 57 se puede observar la dispersión de los datos reales con los datos de la predicción.

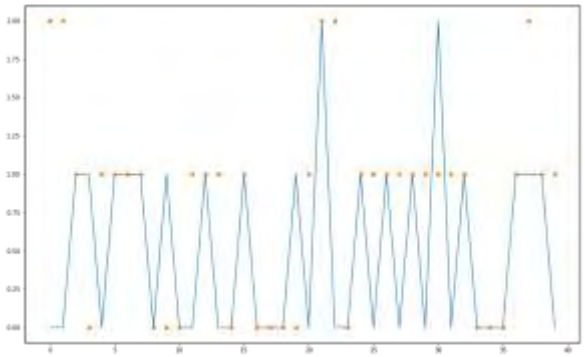


Figura 57. Dispersión de datos con DT *en el conjunto C.*

Se realizó una validación cruzada para obtener el posible mejor conjunto de características y tratar de mejorar los resultados de la predicción, los mejores parámetros obtenidos fueron:

DecisionTreeClassifier(max_depth=50, max_features=0.9, min_samples_split=0.1)

Utilizando estos parámetros se realizó un nuevo entrenamiento del modelo, obteniendo una precisión de 0.5859. Sin embargo, los resultados de este algoritmo con el conjunto de datos C no fueron mejores que los obtenidos sin validación cruzada. La Figura 58 muestra la dispersión entre los datos de predicción y los valores reales.

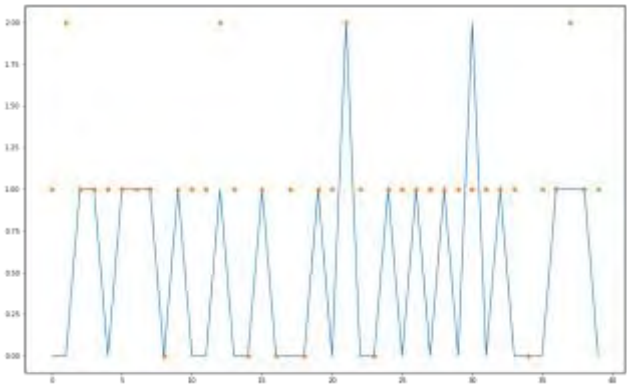


Figura 58. Dispersión de datos con DT *y validación cruzada en el conjunto C.*

Posteriormente, se seleccionó el algoritmo de RF para realizar el entrenamiento. Se utilizó el 70% de los datos para entrenamiento y el 30% para pruebas, y el modelo se entrenó con 500 árboles en el conjunto C, obteniendo una precisión de 0.6307.

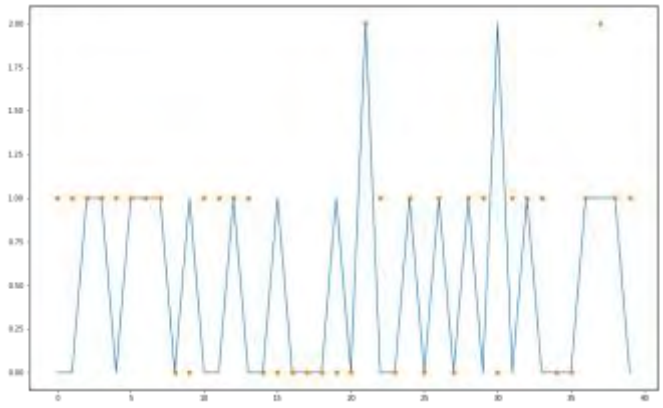


Figura 59. Dispersión de datos con RF *en el conjunto C.*

Utilizando el conjunto de datos C, también se entrenó el modelo con el algoritmo de LR. El error cuadrático medio (MSE) obtenido fue de 0.5574.

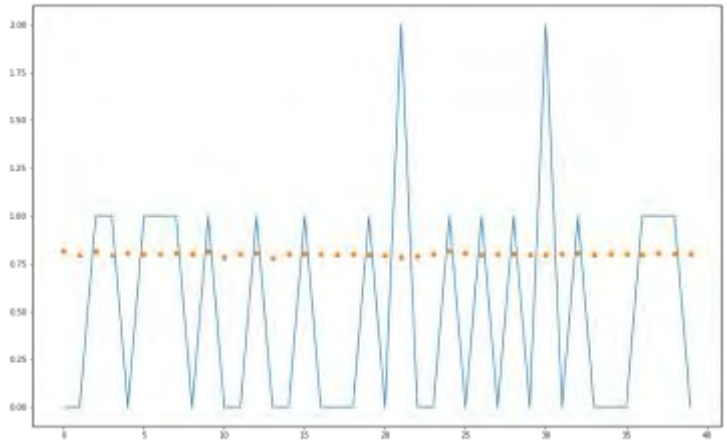


Figura 60. Dispersión de datos con LR *en el conjunto C.*

A continuación, se continuó con el conjunto C+ para realizar el entrenamiento utilizando el algoritmo de DT. La Figura 61 muestra la cabecera de los datos del conjunto C+.

```
In [3]: db.head(204800) #204800 filas
```

Out[3]:

	0	1	2	3	4
0	0.049	0.071	0.132	0.010	0
1	0.042	0.073	0.007	0.105	0
2	0.015	0.000	0.007	0.000	0
3	0.051	0.020	0.002	0.100	0
4	0.107	0.010	0.127	0.054	0

Figura 61. Cabecera de datos del conjunto C+.

La Figura 62 muestra la distribución de los datos y la Figura 63 presenta la distribución de las clases.

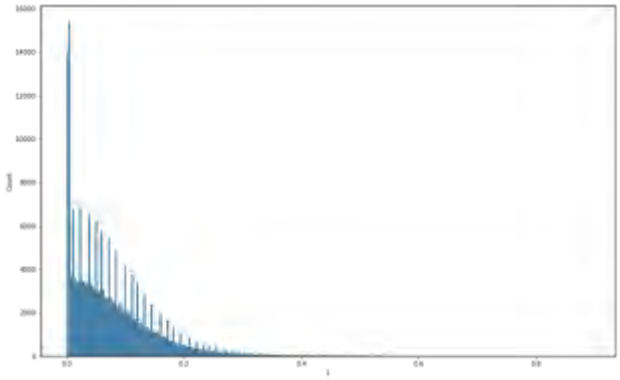


Figura 62. Distribución de datos del conjunto C+.

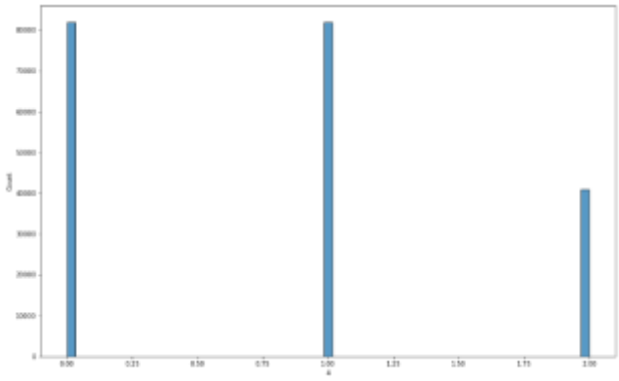


Figura 63. Distribución de clases del conjunto C+.

El conjunto de datos se dividió en un 70% para entrenamiento y un 30% para pruebas. Posteriormente, se realizó el entrenamiento de un modelo de DT. A continuación, se llevó a cabo una prueba de predicciones cuyos resultados se presentan en la Figura 64.

```
In [20]: comp.head(20)
Out[20]:
```

	Reales	Predicciones
21751	0	2
48071	0	2
92969	1	1
115011	1	0
61087	0	1
95299	1	1
92013	1	1

Figura 64. Comparación de resultados con DT *en el conjunto C+.*

Para determinar la precisión del modelo, se utilizó la métrica *exactitud*, con la cual se obtuvo un valor de 0.5961. Para observar la variación de los datos, se creó una matriz de confusión (Figura 65).

```
Out[26]: array([[13980,  8332,  2384],
                [ 6597, 15608,  2418],
                [ 2258,  2821,  7042]], dtype=int64)
```

Figura 65. Matriz de confusión con DT *en el conjunto C+.*

En la Figura 66 se puede observar la dispersión de los datos reales en comparación con los datos de la predicción.

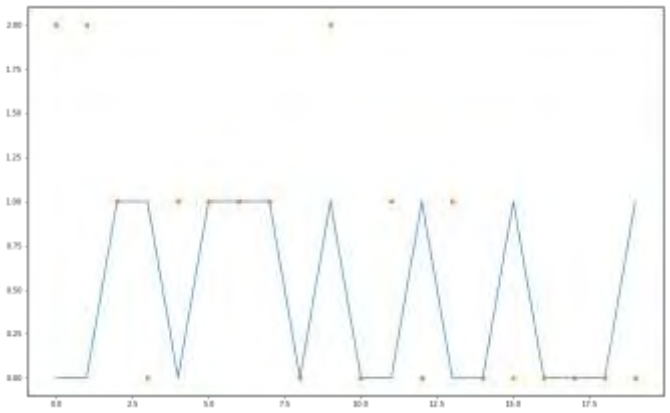


Figura 66. Dispersión de datos con DT *en el conjunto C+.*

Se realizó una validación cruzada para identificar el mejor conjunto de características y tratar de mejorar los resultados de la predicción. Los mejores parámetros obtenidos fueron:

DecisionTreeClassifier(max_depth=9, max_features=0.9, min_samples_split=0.1)

Utilizando estos parámetros se realizó un nuevo entrenamiento del modelo, obteniendo una precisión de 0.5839. Los resultados de este algoritmo con el conjunto de datos C+ fueron inferiores a los obtenidos sin validación cruzada. En la Figura 67 se muestra la dispersión de los datos de predicción frente a los valores reales.

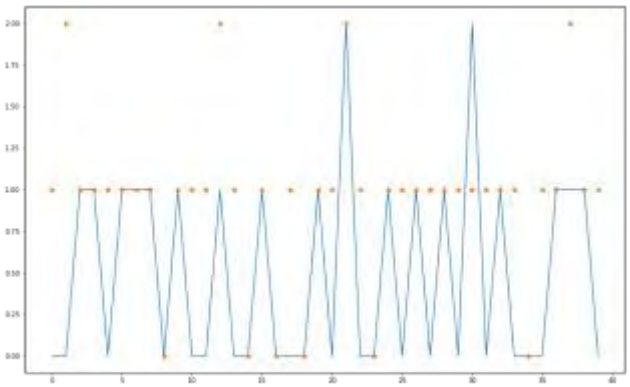


Figura 67. Dispersión de datos con DT *y validación cruzada en el conjunto C+.*

Posteriormente, se seleccionó el algoritmo de RF para realizar el entrenamiento. Se utilizó el 70% de los datos para entrenamiento y el 30% para pruebas, y el modelo se entrenó con 500 árboles en el conjunto C+, obteniendo una precisión de 0.6249. La Figura 68 muestra la dispersión de los datos de predicción con este algoritmo.

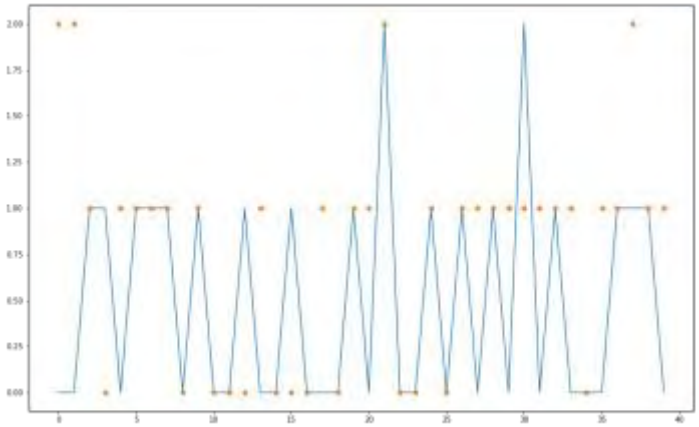


Figura 68. Dispersión de datos con RF *en el conjunto C+.*

Finalmente, utilizando el conjunto de datos C+, se realizó el entrenamiento con el algoritmo de LR, obteniéndose un error cuadrático medio (MSE) de 0.3854. En la Figura 69 se muestra la dispersión de los datos de predicción en comparación con los datos reales.

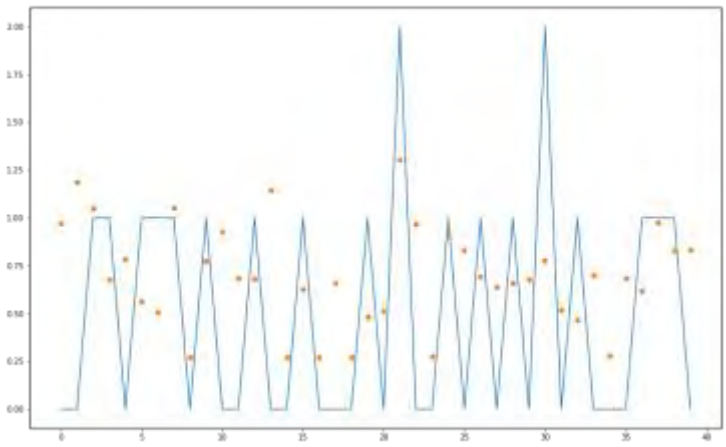


Figura 69. Dispersión de datos con LR *en el conjunto C+.*

Continuando con el conjunto D se realizó el entrenamiento utilizando el algoritmo de DT. La Figura 70 muestra la cabecera de los datos del conjunto D.

```
In [3]: db.head() #24800 filas en total
```

Out[3]:

	0	1
0	-0.049	0
1	-0.042	0
2	0.015	0
3	-0.051	0
4	-0.107	0

Figura 70. Cabecera de datos del conjunto D.

La Figura 71 muestra la distribución de los datos y la Figura 72 la distribución de las clases.

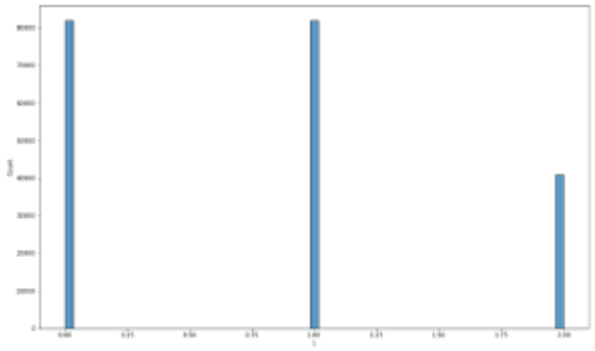


Figura 71. Distribución de datos del conjunto D.

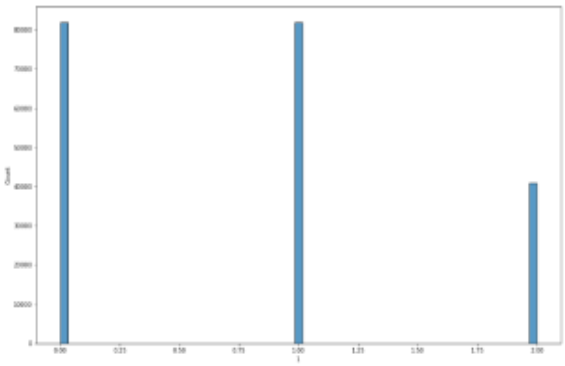


Figura 72. Distribución de clases del conjunto D.

El conjunto de datos se dividió en un 70% para entrenamiento y un 30% para pruebas. Posteriormente, se entrenó un modelo de DT. Los resultados de la prueba de predicciones se presentan en la Figura 73.

```
In [20]: comp.head(20)
```

Out[20]:

	Reales	Predicciones
21751	0	1
48071	0	1
92969	1	1
115011	1	1
61087	0	1
95299	1	1
92013	1	1

Figura 73. Comparación de resultados con DT *en el conjunto D.*

Para determinar la precisión del modelo, se utilizó la métrica *exactitud*, obteniéndose un valor de 0.5703. Para observar la variación de los datos, se generó una matriz de confusión (Figura 74).

```
Out[26]: array([[ 6932, 17107,   657],
                [  941, 22193,  1489],
                [  171,  6032,  5918]], dtype=int64)
```

Figura 74. Matriz de confusión con DT *en el conjunto D.*

En la Figura 75 se puede observar la dispersión entre los datos reales y los datos predichos.

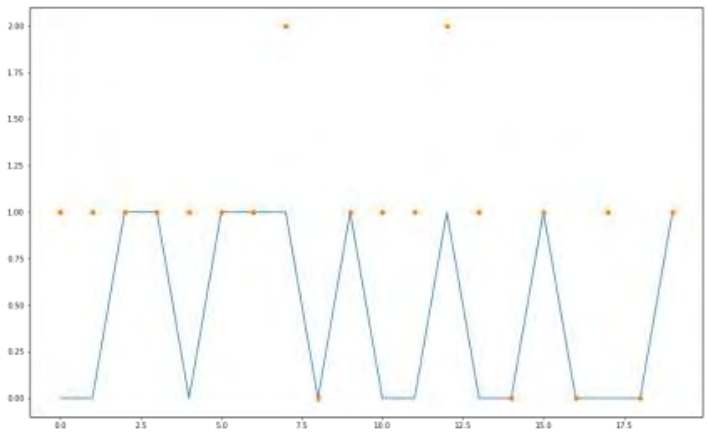


Figura 75. Dispersión de datos con DT *en el conjunto D.*

Se realizó una validación cruzada para identificar el mejor conjunto de características y tratar de mejorar los resultados de la predicción. Los mejores parámetros obtenidos fueron:

DecisionTreeClassifier(max_features=0.1, min_samples_split=0.1)

Con estos parámetros se realizó un nuevo entrenamiento del modelo, obteniendo una precisión de 0.5708. Los resultados de este algoritmo con el conjunto de datos D fueron prácticamente idénticos a los obtenidos sin validación cruzada. En la Figura 76 se muestra la dispersión entre los datos de predicción y los valores reales.

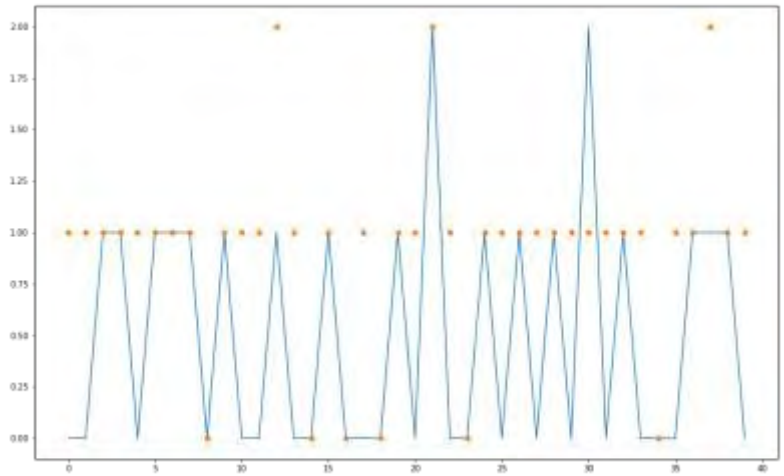


Figura 76. Dispersión de datos con DT *y validación cruzada en el conjunto D.*

Posteriormente se seleccionó el algoritmo de RF para realizar el entrenamiento. Se utilizó el 70% de los datos para entrenamiento y el 30% para pruebas, y se entrenó el modelo con 500 árboles en el conjunto D, obteniendo una precisión de 0.57. La Figura 77 muestra la dispersión de los datos de predicción con este algoritmo.

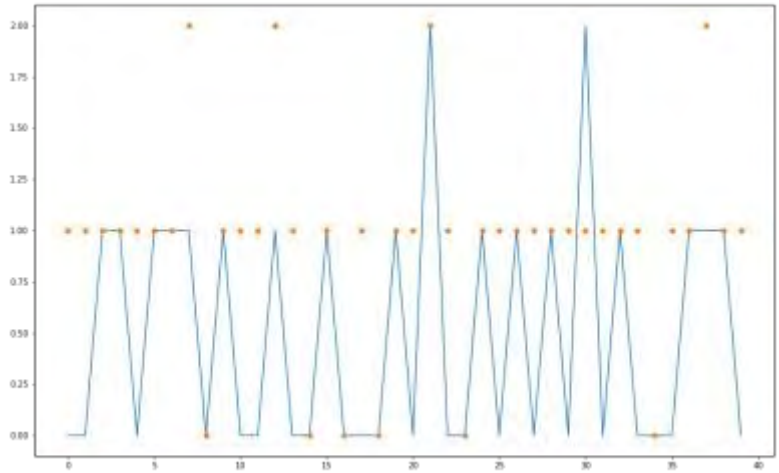


Figura 77. Dispersión de datos con RF *en el conjunto D.*

Utilizando el conjunto de datos D, también se entrenó el modelo con el algoritmo de LR, obteniéndose un error cuadrático medio (MSE) de 0.55739. En la Figura 78 se muestra la dispersión de los datos de predicción en comparación con los datos reales.

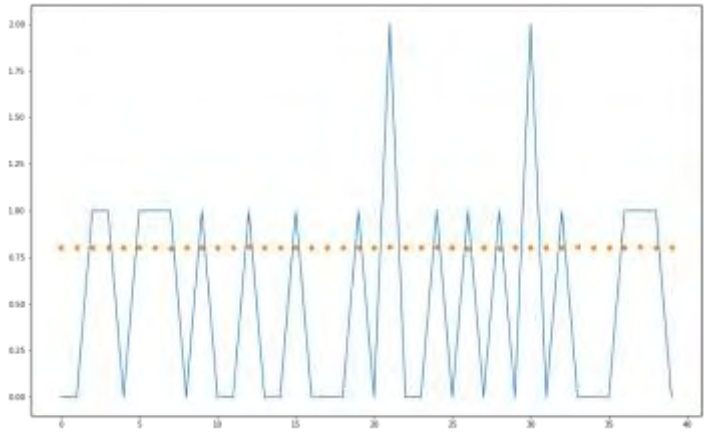


Figura 78. Dispersión de datos con LR *en el conjunto D.*

Continuando con el conjunto D+, se realizó el entrenamiento utilizando el algoritmo de DT. La Figura 79 muestra la cabecera de los datos del conjunto D+.

```
In [3]: db.head()
```

Out[3]:

	0	1
0	0.049	0
1	0.042	0
2	0.015	0
3	0.051	0
4	0.107	0

Figura 79. Cabecera de datos del conjunto D+.

La Figura 80 muestra la distribución de los datos y la figura 81 la distribución de las clases.

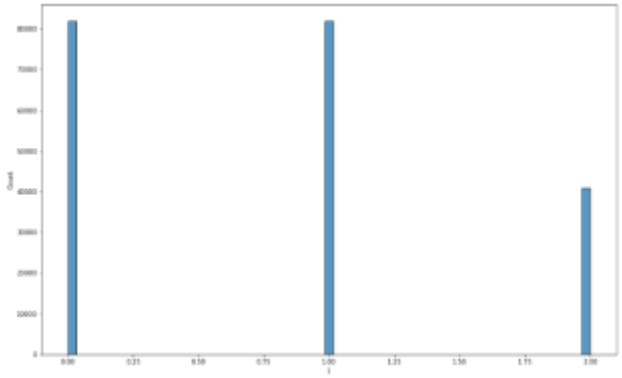


Figura 80. Distribución de datos del conjunto D+.

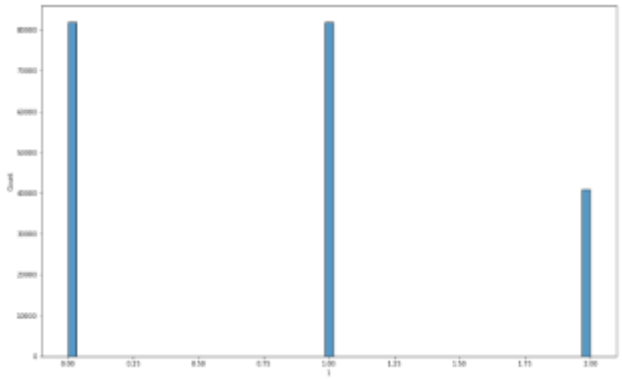


Figura 81. Distribución de clases del conjunto D+.

Se dividió el conjunto de datos en un 70% para entrenamiento y un 30% para pruebas. Posteriormente, se realizó el entrenamiento de un modelo de DT, seguido de una prueba de predicciones cuyos resultados se presentan en la Figura 82.

```
In [20]: comp.head(20)
```

Out[20]:

	Reales	Predicciones
21751	0	1
48071	0	1
92969	1	1
115011	1	1
61087	0	1
95299	1	1
92013	1	1

Figura 82. Comparación de resultados con DT *en el conjunto D+.*

Para evaluar la precisión del modelo se utilizó la métrica *exactitud*, obteniéndose un valor de 0.5701. Para observar la variación de los datos, se generó una matriz de confusión (Figura 83).

```
Out[26]: array([[ 7430, 16687,   579],
                [ 1513, 21801,  1309],
                [  274,  6050,  5797]], dtype=int64)
```

Figura 83. Matriz de confusión con DT *en el conjunto D+.*

En la Figura 84 se muestra la dispersión de los datos reales frente a los datos predichos.

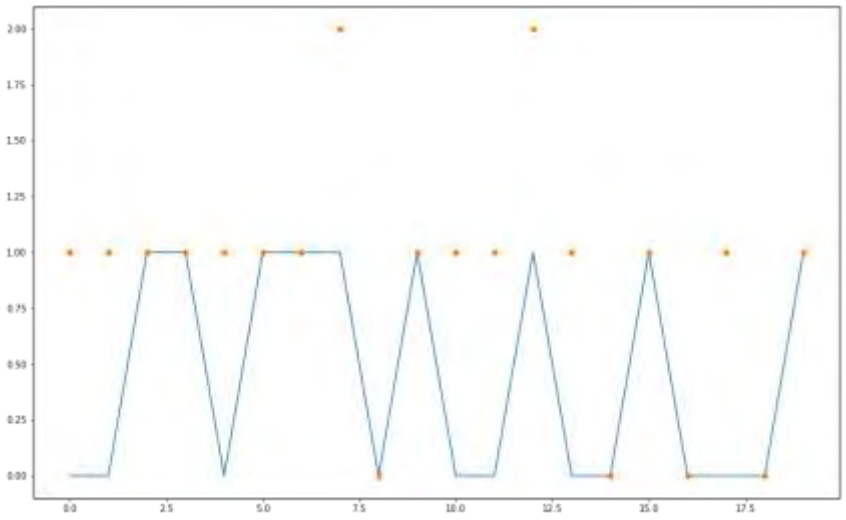


Figura 84. Dispersión de datos con DT *en el conjunto D+.*

Se realizó una validación cruzada para identificar el mejor conjunto de características y mejorar los resultados de la predicción. Los mejores parámetros obtenidos fueron:

DecisionTreeClassifier(max_features=0.1, min_samples_split=0.1)

Con estos parámetros se realizó un nuevo entrenamiento del modelo obteniendo una precisión de 0.5681. Los resultados de este algoritmo con el conjunto de datos D+ fueron

prácticamente idénticos a los obtenidos sin validación cruzada. En la Figura 85 se muestra la dispersión de los datos de predicción frente a los valores reales.

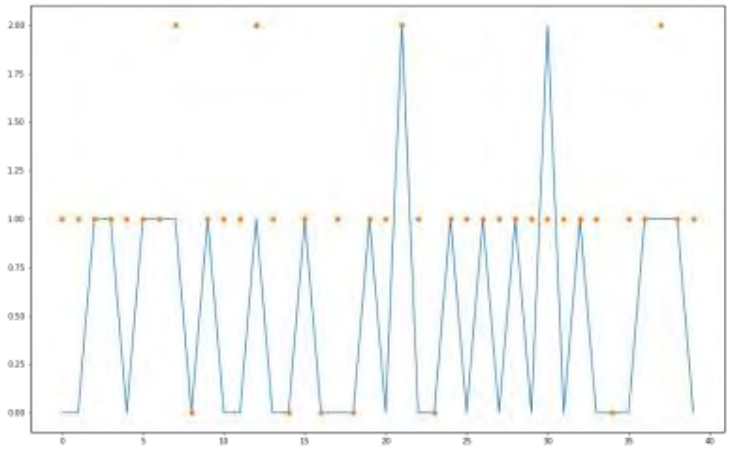


Figura 85. Dispersión de datos con DT *y validación cruzada en el conjunto D+.*

Posteriormente se seleccionó el algoritmo de RF para realizar el entrenamiento. Se utilizó el 70% de los datos para entrenamiento y el 30% para pruebas, y el modelo se entrenó con 500 árboles en el conjunto D+, obteniendo una precisión de 0.5701.

La Figura 86 muestra la dispersión de este algoritmo.

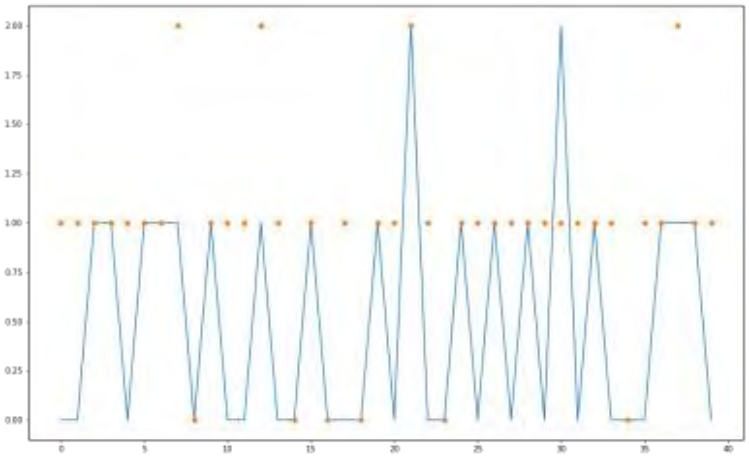


Figura 86. Dispersión de datos con RF *en el conjunto D+.*

Utilizando el conjunto de datos D+ se realizó el entrenamiento con el algoritmo de LR, el error cuadrático medio fue de **0.4523**.

En la figura 87 se muestra la dispersión de los datos de predicciones con respecto a los datos reales.

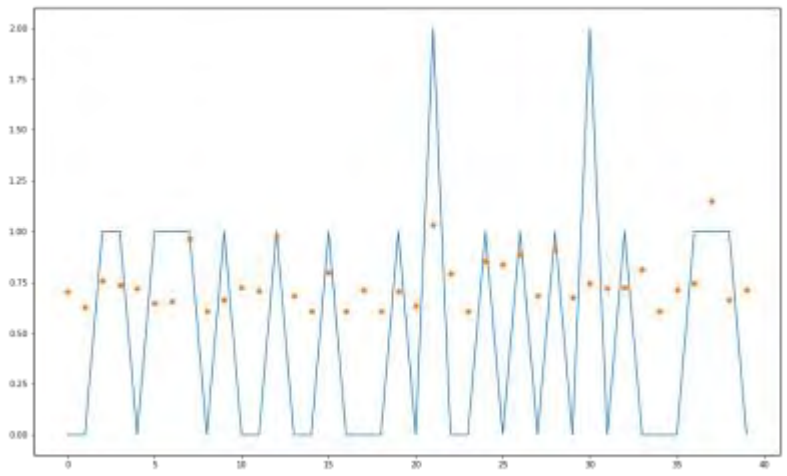


Figura 87. Dispersión de datos con LR *en el conjunto D+.*

Comparación de los Modelos

En la Tabla **7** se presentan los resultados de los algoritmos de DT y RF.

Tabla 7. Comparación de algoritmos de clasificación.

Conjunto	Validación cruzada	Árboles de decisión	Bosque aleatorio
A	------	0.2625162760416666	0.2886881510416667
A	✓	0.2901041666666666	------
B	------	0.537744140625	0.586376953125
B	✓	0.5595052083333333	------
C	------	0.6042643229166667	0.6307779947916666
C	✓	0.5859049479166667	------
C+	------	0.59619140625	0.62490234375
C+	✓	0.5839680989583333	------
D	------	0.570361328125	0.570361328125
D	✓	0.5708658854166667	------
D+	------	0.5701171875	0.5701171875
D+	✓	0.5681315104166667	------

En la Tabla 8 se muestran los resultados obtenidos con el algoritmo de LR.

Tabla 8. Comparación del algoritmo de regresión.

Conjunto	Error Cuadrático Medio
A	8.26037042110339
B	0.8872698022875637
C	0.557404920853931
C+	0.38543296971712626
D	0.5573988038005401
D+	0.45234239592767883

Identificación de las Técnicas de Normalización y Preprocesamiento de Datos

La estandarización de conjuntos de datos es una práctica común, ya que muchos algoritmos de ML requieren que los datos estén distribuidos de manera adecuada (que no se interpongan con otros datos) para resaltar características importantes. En la práctica frecuentemente se ignora la distribución de los datos y se transforman para centrarlos, eliminando el valor medio de cada característica y escalándolos dividiendo por su desviación estándar.

Algunos algoritmos requieren un orden específico para interpretar correctamente las características, mejorando así los resultados. A continuación, se describen algunas técnicas de normalización de datos:

Escalado: La función de escalado permite estandarizar características eliminando la media y escalando a la varianza unitaria. La puntuación estándar de una muestra X se calcula como:

$$z = (x - u) / s , \qquad (3)$$

donde *u* es la media de las muestras de entrenamiento o cero si with_mean = False, y *s* es la desviación estándar de las muestras de entrenamiento o uno si with_std = False. El escalado se realiza de forma independiente en cada característica al calcular las estadísticas en las muestras de entrenamiento [68], [69].

Normalización: Este proceso escala las muestras individuales para tener una norma unitaria. Es útil al emplear métricas cuadráticas como el producto punto para cuantificar la similitud entre pares de muestras [70].

Codificación de Características Categóricas: Cuando las características son categóricas, como género o región, pueden codificarse eficientemente como números enteros. La función *OrdinalEncoder* transforma cada característica categórica en códigos enteros que van de 0 a *n_categorías* - 1 [71].

Discretización: También conocida como cuantificación o agrupamiento, convierte características continuas en valores discretos. Esta técnica puede transformar conjuntos de datos de atributos continuos en conjuntos con solo atributos nominales.

Normalización y Procesamiento de Datos de los Repositorios

Al aplicar estas técnicas, se observó que los datos transformados perdían características importantes, haciéndolos indistinguibles entre sí y dificultando su uso en los algoritmos de predicción. En la Figura 88 se observa un gráfico del conjunto de datos original; la Figura 89 muestra los datos escalados, y la Figura 90 los datos normalizados.

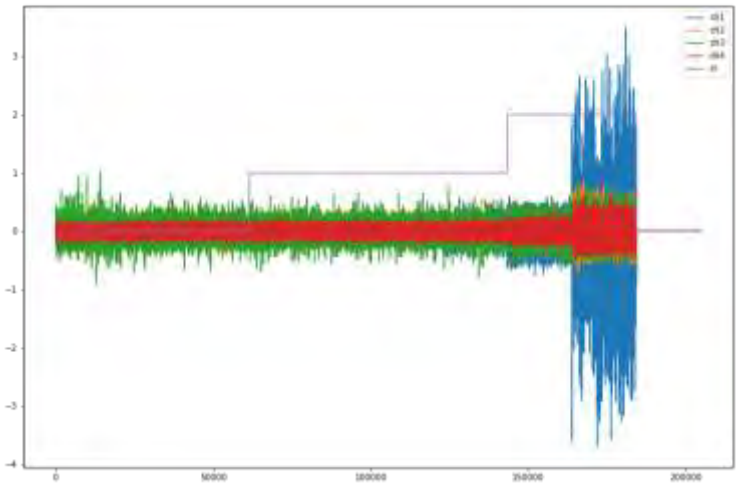


Figura 88. Gráfico de datos originales.

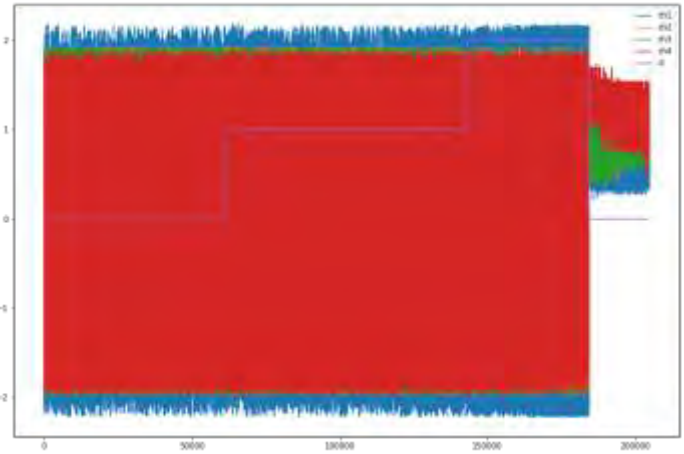


Figura 89. Gráfico de datos escalados.

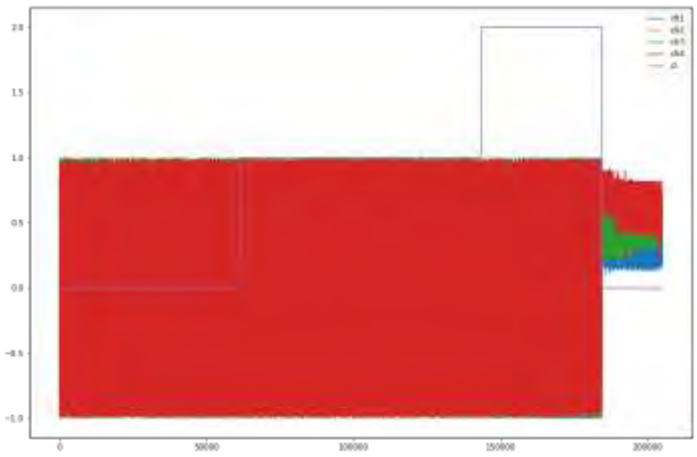


Figura 90. Gráfico de datos normalizado.

Debido a la variabilidad inherente de los datos provenientes de los sensores, se decidió dividir el conjunto de datos en valores positivos y negativos para mejorar la apreciación de los picos que determinan las probabilidades de falla.

Considerando el funcionamiento de los baleros y las flechas, existen movimientos definidos como ascendentes y descendentes, de los cuales se obtienen datos positivos y negativos (Figura 91).

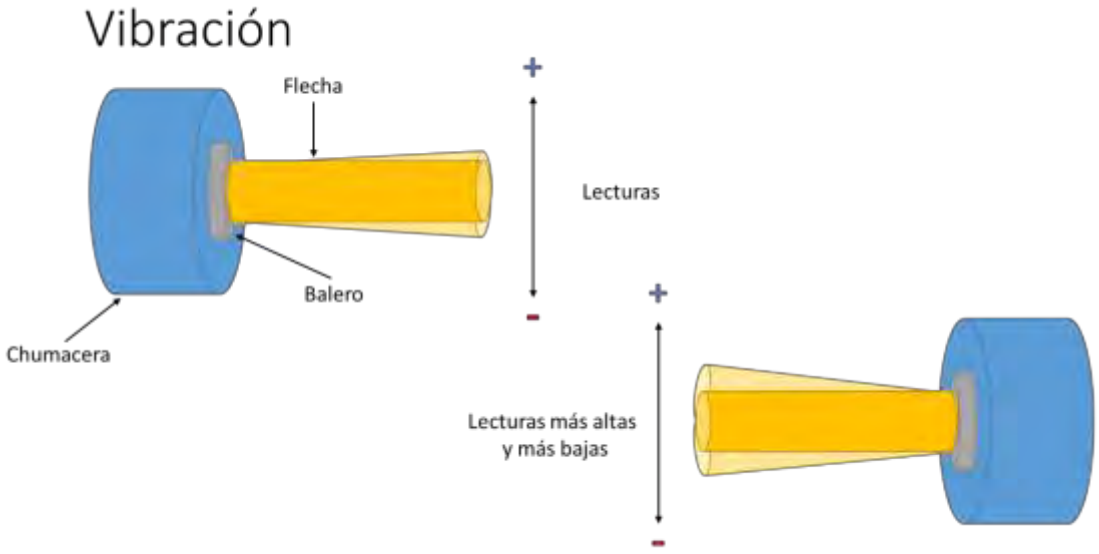


Figura 91. Baleros y flechas.

En este contexto se determinó que los picos ascendentes y descendentes, en relación con la dirección del movimiento, corresponden a mayores niveles de vibración. Sin embargo, los datos alejados del punto de origen 0 pueden indicar una futura falla, independientemente de si son positivos o negativos.

Para optimizar los datos se realizó una discretización, dividiendo el conjunto en valores más representativos orientados en un solo sentido (positivo o negativo). Esto permite a los algoritmos

identificar grupos de datos de forma más efectiva y realizar mejores predicciones. De este proceso se obtuvieron dos nuevos conjuntos de datos: E+ y E-.

Implementación y Evaluación de los Algoritmos de Predicción con los Repositorios Normalizados

El primer conjunto de datos (E+) se compone de los valores mayores a 0, con las siguientes especificaciones:

- Media: 0.0820
- Desviación estándar: 0.1381
- Valor máximo: 3.501
- Valor mínimo: 0.0

La Figura 92 es una representación gráfica de los datos.

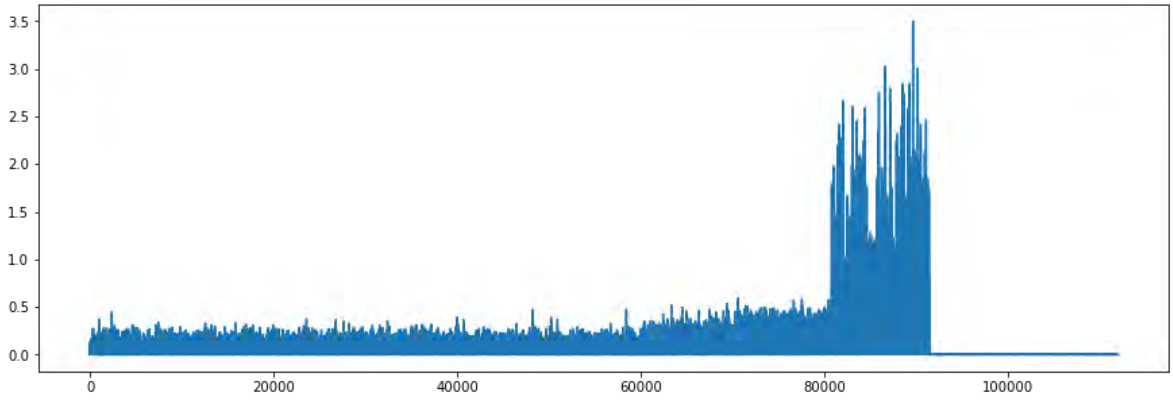


Figura 92. Conjunto de datos E+.

Aplicando el algoritmo de DT, se obtuvo una precisión de 72.92%. La Figura 93 muestra la matriz de confusión obtenida.

```
Out[36]: array([[ 5856,   277,     0,     0],
                [ 1010, 16238,   506,    75],
                [  227,  5020,   827,   459],
                [   34,   980,   500,  1560]], dtype=int64)
```

Figura 93. Matriz de confusión con DT *con el conjunto E+.*

En la Figura 94 se observa la dispersión de los datos en relación con las predicciones.

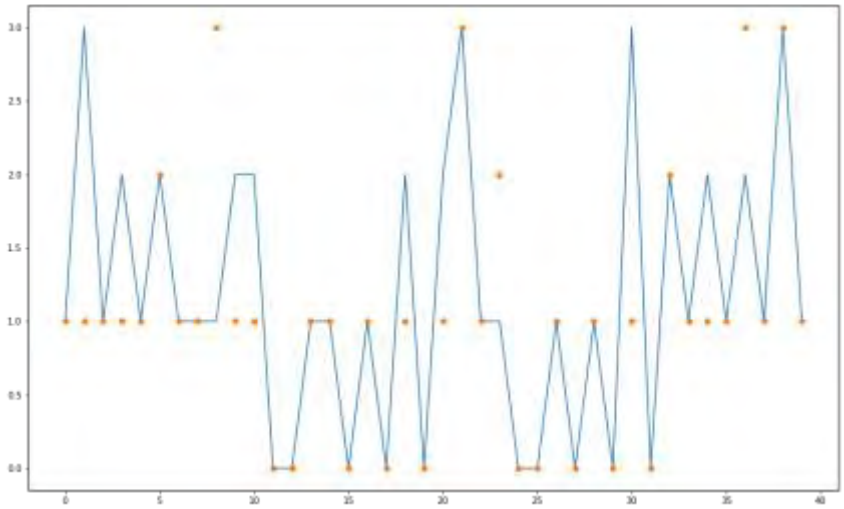


Figura 94. Matriz de dispersión con DT *en el conjunto E+.*

Se observa que, en la mayoría de las predicciones, hay aciertos significativos, mostrando una mejora respecto a los conjuntos de datos anteriores.

Posteriormente se aplicó el algoritmo de RF, obteniéndose una precisión de 72.96%. La Figura 95 muestra la matriz de confusión correspondiente.

```
Out[60]: array([[ 5856,   277,     0,     0],
                [ 1010, 16238,   503,    78],
                [  227,  5020,   806,   480],
                [   34,   980,   466,  1594]], dtype=int64)
```

Figura 95. Matriz de confusión del algoritmo RF *en el conjunto E+.*

En la Figura 96 se presenta la matriz de dispersión de los datos con respecto a las predicciones.

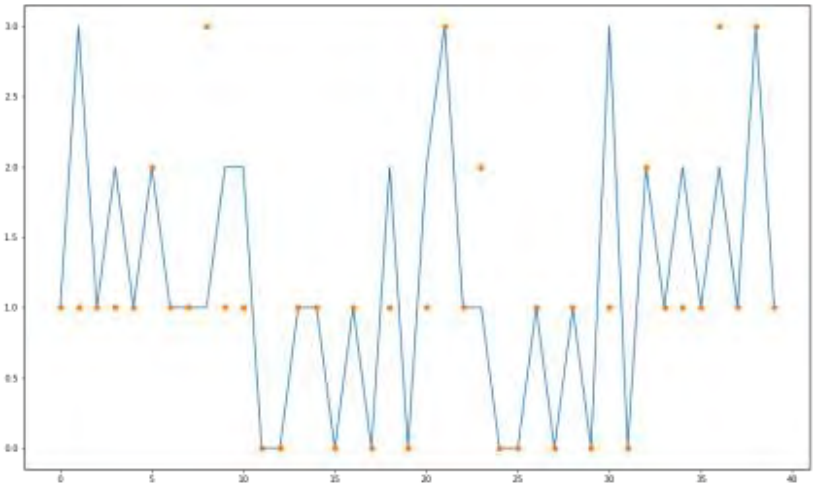


Figura 96. Matriz de dispersión con RF *en el conjunto E+.*

Se observa nuevamente que la mayoría de las predicciones son acertadas, lo cual indica buenos resultados de predicción.

El segundo conjunto de datos (E-) se compone de los valores menores a 0, con las siguientes especificaciones:

- Media: -0.1039
- Desviación estándar: 0.1601
- Valor máximo: -0.002
- Valor mínimo: -3.696

La Figura 97 es una representación gráfica de estos datos.

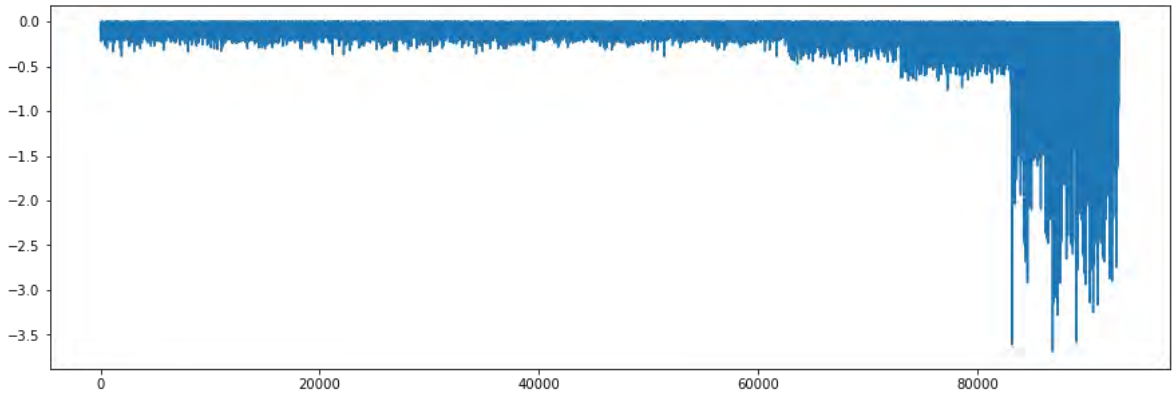


Figura 97. Conjunto de datos E-.

Aplicando el algoritmo de DT se obtuvo una precisión de 70.28%. La Figura 98 muestra la matriz de confusión obtenida.

```
Out[30]: array([[17831,     0,    82,   117],
                [ 3367,     0,    48,   122],
                [ 2748,     0,   146,   510],
                [ 1123,     0,   164,  1614]], dtype=int64)
```

Figura 98. Matriz de confusión con DT con el conjunto E-.

En la Figura 99 se observa la dispersión de los datos en relación con las predicciones.

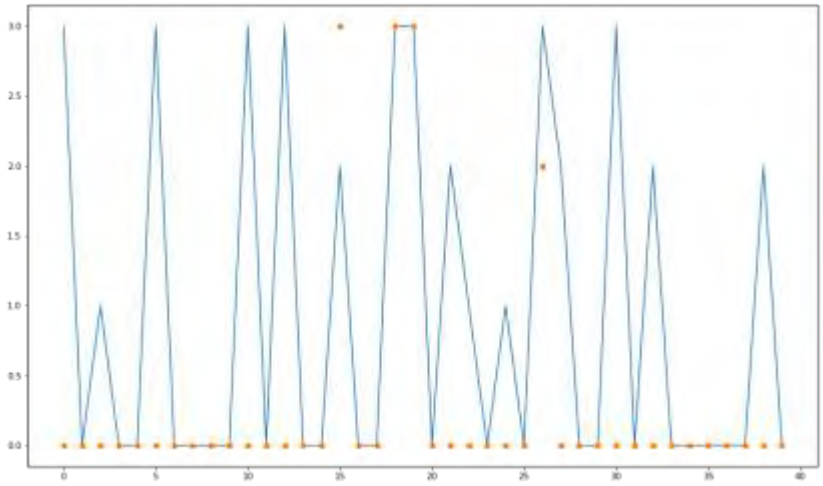


Figura 99. Matriz de dispersión con DT *en el conjunto E+.*

En esta imagen se aprecia una mayor dispersión en comparación con el conjunto E+, aunque los resultados aún son mejores que en los conjuntos anteriores.

Posteriormente se aplicó el algoritmo de RF, obteniéndose una precisión de 72.96%. La Figura 100 muestra la matriz de confusión correspondiente.

```
Out[60]: array([[ 5856,   277,     0,     0],
                [ 1010, 16238,   503,    78],
                [  227,  5020,   806,   480],
                [   34,   980,   466,  1594]], dtype=int64)
```

Figura 100. Matriz de confusión del algoritmo RF *en el conjunto E+.*

En la Figura 101 se presenta la dispersión de los datos en relación con las predicciones.

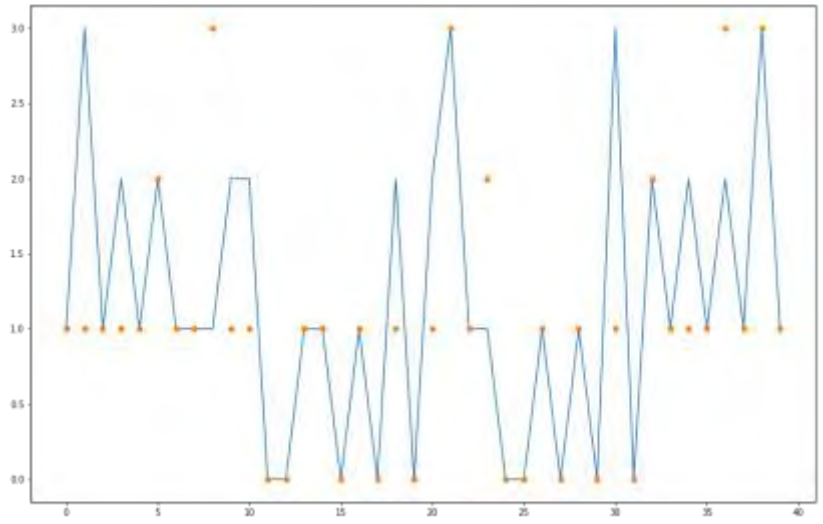


Figura 101. Matriz de dispersión con RF *en el conjunto E+.*

También en este caso se observa que la mayoría de las predicciones son correctas, lo que indica buenos resultados de predicción.

En la Tabla 9 se presenta una comparación de los datos y resultados de los conjuntos E+ y E-.

Tabla 9. Comparación de datos y resultados de los conjuntos E.

Conjunto	E+	É-
Media	0.08204	-0.10389
Desviación estándar	0.13814	0.16012
Valor máximo	3.50100	-0.002
Valor mínimo	0.0	-3.696
Árboles de decisión	0.72927	0.70289
Bosque aleatorio	0.72966	0.72966

Hasta este punto de la investigación es evidente que la adaptación de los datos mejora significativamente el desempeño de los algoritmos. Por lo tanto, se decidió enfocar el análisis en un conjunto de datos y un algoritmo específicos para obtener resultados aún más precisos.

4.3 Fase de Ajuste de Preprocesamiento

Para llevar a cabo esta etapa, fue necesario investigar diversas técnicas de preprocesamiento de datos, especialmente enfocadas en métodos de filtrado que permitieran obtener datos más representativos y característicos de los conjuntos originales. Los datos representativos son aquellos que facilitan la identificación de diferencias entre características, optimizando así el rendimiento de los algoritmos de predicción al alimentarse de información específica, en lugar de datos similares con objetivos diferentes.

Un dato es una información concreta sobre hechos, mediciones u otros elementos que permite su análisis y estudio. Estos datos poseen distintas propiedades tales como: tipo, disponibilidad, formato y naturaleza, entre otras, los que pueden influir en su tratamiento, especialmente cuando se presentan alteraciones debido a la forma en que son capturados o recopilados.

Usualmente, los conjuntos de datos iniciales no están procesados, lo cual es beneficioso ya que permite aplicar diversos procesos de ajuste según las necesidades de cada proyecto. Sin embargo, el preprocesamiento debe adaptarse a la naturaleza de los datos y sus características para obtener resultados óptimos. En este proyecto, se exploraron técnicas de filtrado adecuadas para el tratamiento de datos de vibración en rodamientos.

A continuación, se describen las principales técnicas de procesamiento de datos investigadas:

1. **Transformada de Fourier**

La Transformada de Fourier, nombrada en honor a Joseph Fourier, es una herramienta matemática que permite transformar señales del dominio del tiempo al dominio de la frecuencia, y tiene aplicaciones en múltiples campos como física, ingeniería, teoría de números y procesamiento de

señales. En el contexto de procesamiento de señales, permite descomponer una señal en componentes de diferentes frecuencias, lo que facilita el análisis de patrones no evidentes en el dominio temporal.

La transformada de Fourier corresponde a la aplicación de la función **ƒ** con otra función **g** y se define por la siguiente ecuación:

$$g(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(x) e^{-i\xi x} dx, \qquad (4)$$

donde las variables **x** y **ξ** suelen estar asociadas a dimensiones como el tiempo y la frecuencia respectivamente. Esta técnica es especialmente útil para visualizar en qué punto se concentran los datos al analizarlos en términos de frecuencia [72].

2. **Transformada de Hilbert**

La Transformada de Hilbert-Huang es una técnica compuesta por un proceso de filtrado llamado descomposición en modos empíricos, el cual permite estimar el espectro de señales no lineales y no estacionarias. A menudo, es utilizada en el procesamiento de señales y telecomunicaciones. Su ventaja radica en que no requiere preprocesamiento, entregando resultados de fácil interpretación física, lo cual la hace útil para caracterizar dispositivos de sonido, identificar bordes en imágenes y en sistemas de modulación.

La transformada de Hilbert de una función real se expresa como $s(t)$ y se obtiene mediante la convolución de las señales $s(t)$ y $1/(\pi t)$ donde se obtiene $\hat{S}(t)$, se puede interpretar como la salida de un sistema lineal e invariante en el tiempo con entrada $s(t)$y respuesta al impulso $1/(\pi t)$.

En [73] se muestra una actualización de la transformada de Hilbert, su comparación con la transformada de Fourier y una visión de las capacidades de la herramienta de tratamiento de señales, no sólo de vibraciones.

3. **Transformada de Wavelet**

Esta técnica matemática es ampliamente utilizada en el procesamiento de señales, diagnóstico de defectos y detección de anomalías, siendo aplicable tanto a señales estacionarias como no estacionarias. Su capacidad para proporcionar un análisis de resolución múltiple en el tiempo y la frecuencia, a diferentes escalas y resoluciones, la hace especialmente valiosa para el diagnóstico de defectos.

Las wavelets son funciones que se encuentran definidas en el espacio y se emplean como herramientas de análisis para examinar las características periódicas y no periódicas de una señal en el plano tiempo-frecuencia. Una familia de wavelets se define por la siguiente ecuación:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right) \qquad a, b \in R \qquad a > 0 \qquad (5)$$

donde a representa el escalamiento y b el desplazamiento. Para más información, consultar [74].

4. **Filtro de Kalman**

El filtro de Kalman tiene numerosas aplicaciones en tecnología. Una aplicación común es la guía, navegación y control de vehículos, especialmente naves espaciales. Además, se utiliza en campos como el procesamiento de señales y la econometría.

En otras palabras, el filtro de Kalman busca minimizar el Error Cuadrático Medio, y para ello el objetivo será encontrar el valor de la ganancia que nos garantiza esta propiedad. Así pues, el filtro de Kalman acabará por dar, en el instante k, un estimador lineal que será la combinación de los estimadores hasta el instante k − 1 y de la nueva observación del instante k.

El filtro nos dará un estimador que minimiza el error cuadrático medio a partir de unas observaciones. Si se considera un vector aleatorio X de dimensión l. Siendo $\hat{X}k|j$ el estimador de X en el instante k a partir de las observaciones tomadas hasta el instante $j, Z1, \ldots, Zj, con\ j\ \leq\ k$ (Raga, 2018).

$$\hat{X}_{k|j} = min_{\hat{X}_{k|j} \in R^n} traza \left\{ E\left[(X_k - \hat{X}_{k|j})(X_k - \hat{X}_{k|j})^t \right] \right\} \qquad (6)$$

En (Somonte Martín, 2010) se realiza una comparativa entre técnicas de filtrado donde se compara la transformada de Fourier, la transformada de Wavelet y la transformada de Hilbert. Se menciona que al aplicar la transformada de Fourier los datos resultantes no aportan información útil de las señales, esto se debe a que existe una gran cantidad de ruido en los datos y por lo tanto no es posible identificar las frecuencias de los defectos.

Al hacer uso de la transformada de Wavelet se observó que al ser un filtro que trabaja sobre altas frecuencias, no resulta útil para el tratamiento de señales de vibración en los rodamientos. Por otro lado, la transformada de Hilbert resulta ser de utilidad, sin embargo, para lograr estos resultados es necesario que esta se aplique de manera anterior a la aplicación de la transformada de Fourier.

Comparación de Técnicas de Filtrado

En [75] se realizó una comparativa entre varias técnicas de filtrado, incluyendo la transformada de Fourier, la transformada de Wavelet y la transformada de Hilbert. Los resultados demostraron que:

- La **transformada de Fourier** no aportaba información útil sobre las señales debido a la alta cantidad de ruido en los datos, lo cual dificultaba la identificación de frecuencias específicas de los defectos.
- La **transformada de Wavelet**, al estar enfocada en altas frecuencias, no resultó útil para el tratamiento de señales de vibración en rodamientos.
- La **transformada de Hilbert** mostró ser útil, pero requería una aplicación preliminar de la transformada de Fourier para obtener los resultados deseados.

Dadas las características y el origen de los datos en este proyecto, el preprocesamiento se volvió necesario, ya que el ruido inherente de los acelerómetros afecta la precisión de los algoritmos de ML debido a los datos con de lecturas erróneas. Se decidió emplear el Filtro de Kalman para el

tratamiento de estos datos, debido a que, a diferencia de la combinación de la transformada de Hilbert y la de Fourier, el Filtro de Kalman requiere menos procesamiento.

Aplicación del Filtro de Kalman

Se desarrolló un programa en Python para procesar los datos utilizando el Filtro de Kalman. Como resultado, las inconsistencias bruscas en los datos, causadas por el ruido, se redujeron significativamente, generando datos más representativos y cercanos a la señal sin ruido.

La Figura 102 muestra un gráfico del conjunto de datos completo sin procesar.

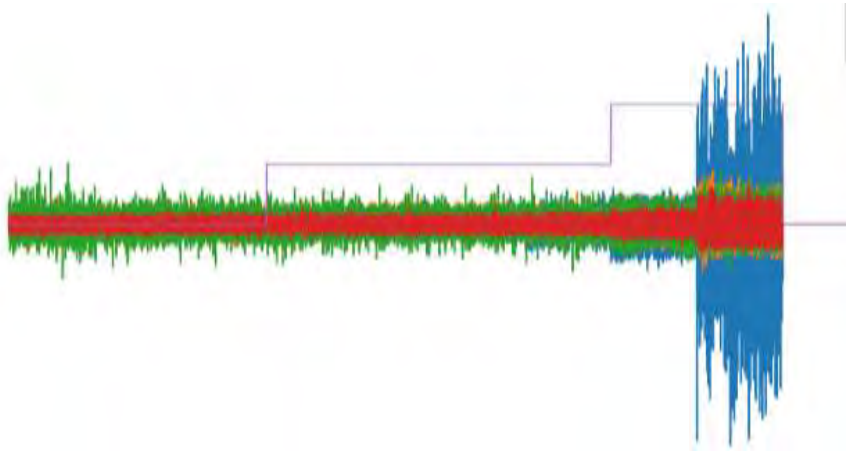


Figura 102. Conjunto de datos general.

En la figura 102, los datos de los cuatro canales de los baleros se representan con colores distintos, mostrando las lecturas de los acelerómetros en el rango de -4 a +4 en el eje Y, mientras que el eje X indica el número de cada lectura.

La Figura 103 presenta una sección segmentada de los datos del canal 1.

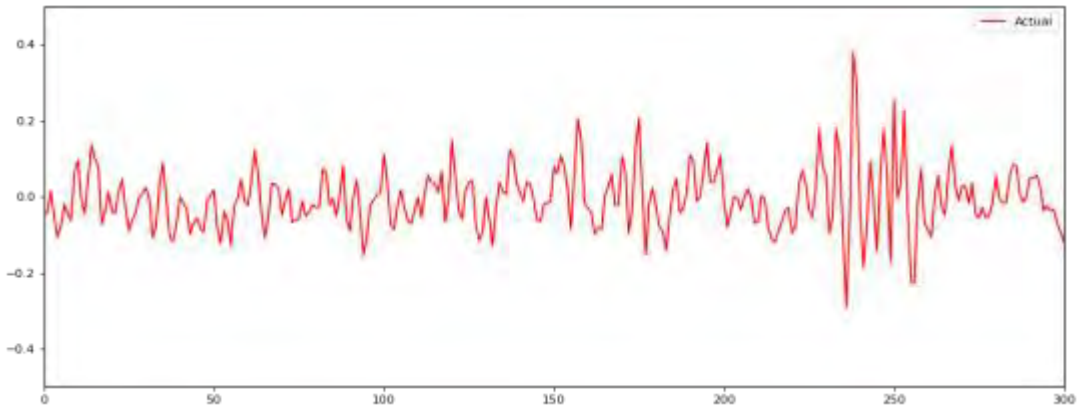


Figura 103. Muestra segmentada del canal 1.

Como se observa, en la zona de los datos entre 230 y 260 existe una variación no representativa, ya que la falla ocurre realmente alrededor de las lecturas próximas a 180,000 (ver el canal azul en la Figura 102).

En la Figura 104, se muestra la misma muestra segmentada pero ya procesada con el Filtro de Kalman, con un Error Cuadrático Medio (MSE) de 0.06 en la estimación. En esta imagen, las señales se han suavizado minimizando el ruido de la señal.

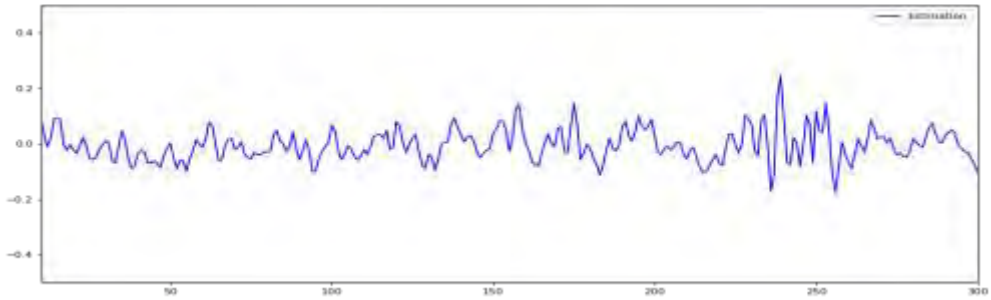


Figura 104. Filtro de Kalman aplicado a la muestra de datos.

La Figura 105 presenta los datos procesados superpuestos.

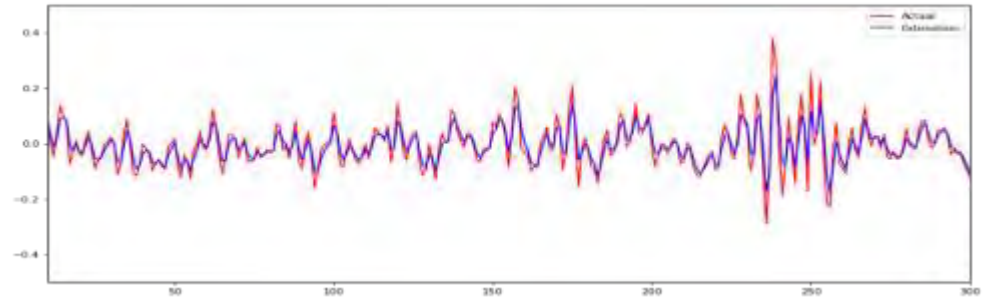


Figura 105. Gráfico de datos procesados.

Esta figura muestra el resultado del filtrado en el conjunto de datos completo del canal 1, permitiendo distinguir mejor el momento en que ocurre la falla del rodamiento y su posterior reparación (Figura 106).

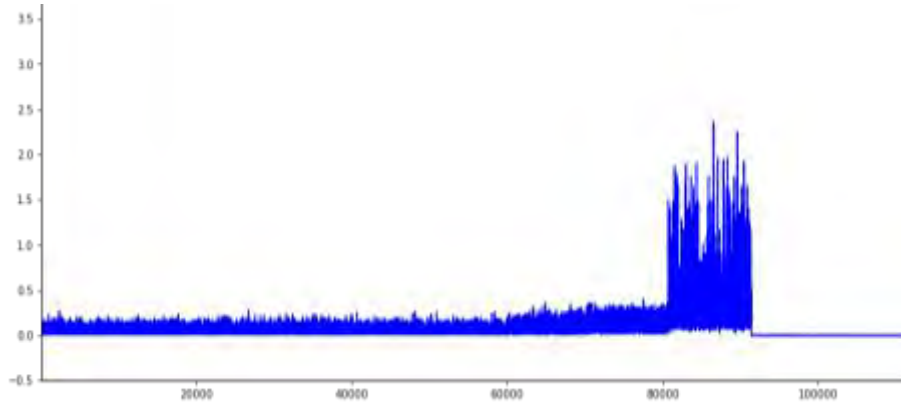


Figura 106. Conjunto de datos del canal 1 completo y filtrado.

Aplicación de Algoritmos

Se aplicaron los mismos algoritmos utilizados en los conjuntos de datos previos, y los resultados obtenidos se presentan en la Tabla 10.

Tabla 10. Tabla comparativa de los resultados.

Conjunto de datos	Árboles de decisión	Bosque Aleatorio	Precisión máxima	Error cuadrático medio
A	0.262516	0.2886	28%	8.2603
B	0.5595	0.5863	58%	0.8872
C	0.6042	0.6307	63%	0.5574
C+	0.5961	0.6249	62%	0.3854
D	0.5708	0.5703	57%	0.5573
D+	0.5701	0.5701	57%	0.4523
E-	0.7028	0.7296	72%	X
E+	0.7292	0.7296	72%	X
FS	**0.9203**	**0.9265**	**92%**	**X**
F	**0.7385**	**0.7387**	**73%**	**X**

4.4 Fase de Validación y Evaluación de Resultados

En esta fase se utilizaron los datos del experimento 3 del repositorio encontrado en [62]. Este contiene lecturas de acelerómetros ubicados en cuatro rodamientos. Este conjunto incluye un total de 6,324 archivos con señales de vibración capturadas a intervalos de 10 minutos, cada una con 20,480 registros correspondientes a un segundo de captura. Las lecturas abarcan un periodo desde el 4 de marzo de 2004 a las 09:27:46 hasta el 18 de abril de 2004 a las 02:42:55.

- Origen de los datos: Acelerómetros.
- Canales de datos: Balero 1 – ch1, Balero 2 – ch2, Balero 3 – ch3, Balero 4 – ch4.
- Total de registros por canal: 129,515,520.
- Evento de falla: Ocurrió una falla en el balero número 3 al finalizar el experimento

Dado que el conjunto de datos original contenía 129,515,520 registros, era necesario reducir la cantidad de datos redundantes sin perder la representatividad de los eventos de falla. Para lograr esto, se decidió seleccionar el 10% de los registros de cada archivo, extrayendo una muestra aleatoria que capturara las principales características del conjunto original.

Este porcentaje se eligió con el objetivo de equilibrar la representatividad de los datos y la eficiencia del procesamiento. En el contexto de este estudio, la selección del 10% de los datos responde a la necesidad de procesar la información en sistemas con capacidades limitadas, tales como los sistemas embebidos, los cuales se usan en aplicaciones industriales donde se requiere que las máquinas realicen autoanálisis y predicciones en tiempo real.

El uso de un mayor porcentaje de datos incrementa el tiempo de procesamiento, lo que reduce la ventana de tiempo disponible (Aumenta si las capacidades de procesamiento son

limitadas) para planificar el mantenimiento. En aplicaciones de PdM, disponer de una ventana de predicción más amplia es crucial para coordinar las intervenciones de mantenimiento sin interrumpir las operaciones. En ensayos preliminares se consideraron otros porcentajes de muestra, pero el 10% demostró ser óptimo al ofrecer un equilibrio entre precisión y eficiencia de procesamiento, permitiendo la implementación en sistemas de menor capacidad sin comprometer la precisión del análisis.

Este tratamiento permitió reducir significativamente el tamaño del dataset sin comprometer la variabilidad de las señales, obteniendo un total de 25,296 registros que representan fielmente las condiciones de vibración a lo largo del experimento. Este proceso de reducción de datos se ilustra en la figura 107, donde se muestra una representación del nuevo conjunto de datos.

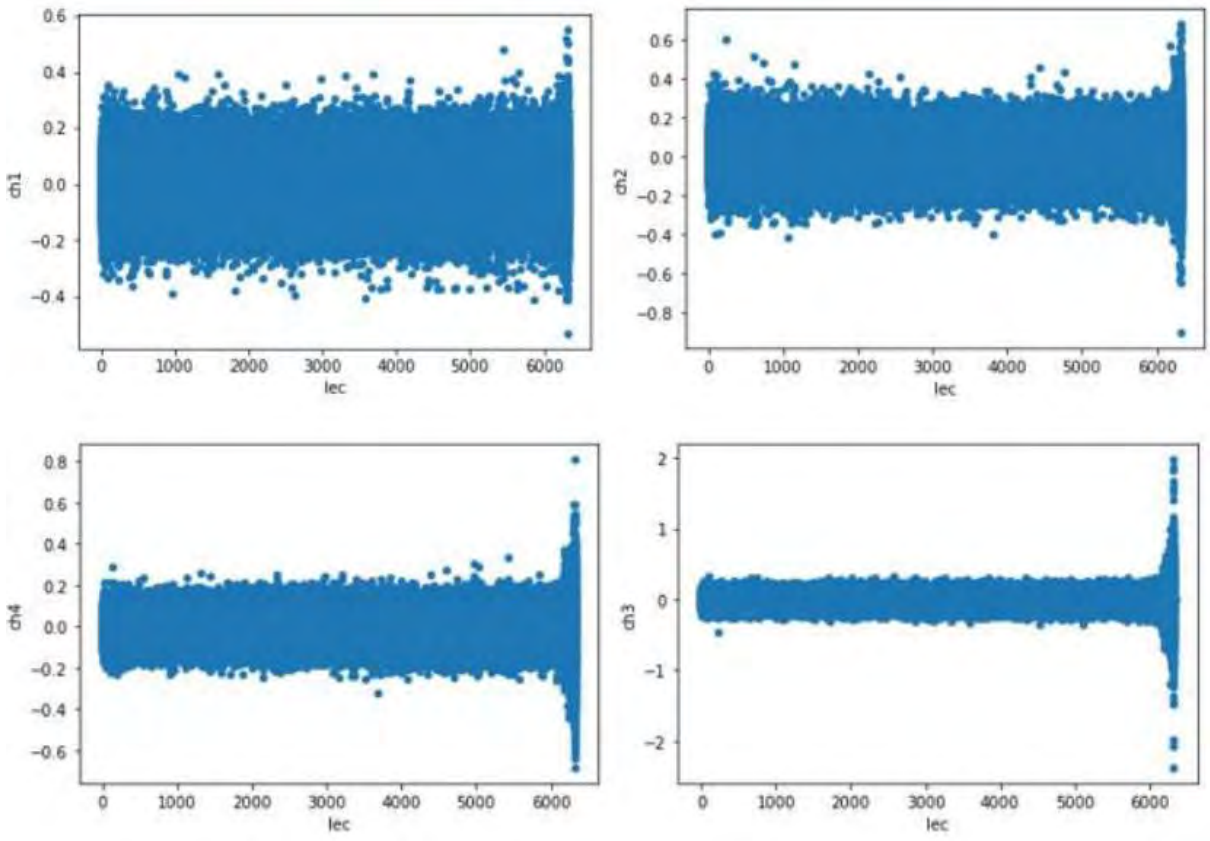


Figura 107. Conjunto de datos segmentado.

Al transformar los datos a valores absolutos, el conjunto de datos se vuelve más representativo del fenómeno de falla, facilitando la detección de patrones críticos y evitando la creación de clases innecesarias o curvas de predicción difíciles de ajustar. Esta simplificación del dataset mejora directamente la precisión del modelo, ya que los algoritmos pueden identificar de forma más clara los patrones de falla relevantes, y aumenta el recall, al reducir la probabilidad de pasar por alto eventos de falla debido a interpretaciones erróneas de los signos de las lecturas.

Como siguiente paso, para reducir aún más la redundancia en los datos, se calculó la media de los registros para cada archivo. Al promediar las lecturas, se eliminó información no representativa, como fluctuaciones pequeñas que no aportan valor a la predicción de fallas, y se conservó únicamente la información más relevante.

Este proceso permitió generar un conjunto de datos más compacto, compuesto por 6,324 registros, que se utilizó para crear los modelos de clasificación y predicción. En este nuevo dataset, los registros corresponden a un total de 1,054 horas de operación, en donde la falla del rodamiento se detectó en la hora 1,033. La figura 108 muestra el comportamiento de los datos del rodamiento 3, donde ocurrió la falla.

Figura 108. Datos del rodamiento 3.

Tras completar el preprocesamiento, los datos quedaron listos para la siguiente fase, en la cual se generaron los modelos de ML y se llevó a cabo la experimentación.

Se crearon modelos utilizando algoritmos de clasificación como DT, RF y SVM, y para los modelos de regresión se emplearon NN y LR. Se utilizaron muestras del 70% del dataset generado después del preprocesamiento. La experimentación se llevó a cabo en lenguaje Python en la plataforma Google Colab, y los resultados obtenidos se evaluaron en la fase 4 del estudio.

Fase de Validación y Evaluación

En esta fase, se evaluó el desempeño de los modelos de clasificación y regresión a través de métricas clave, permitiendo una validación de la eficacia de los modelos empleados. La precisión de los modelos se evaluó utilizando métricas como la *accuracy, precision, recall* y el error cuadrático medio (MSE), fundamentales para medir la eficacia del entrenamiento y determinar la aplicabilidad del modelo en un contexto industrial [76], [77].

Las métricas son definidas como:

Accuracy: Indica la relación proporcional entre las predicciones realizadas correctamente y el número de instancias evaluadas. Está determinada por la ecuación 7.

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \qquad (7)$$

Donde tp son verdaderos positivos, tn verdaderos negativos, fp falsos positivos y fn falsos negativos.

Precision: Representa la relación de elementos clasificados correctamente con todos los elementos clasificados, alcanza su mejor valor en 1 y la peor puntuación en 0 y está determinada por la ecuación 8.

$$Precision = \frac{tp}{tp + fp} \tag{8}$$

Donde tp son verdaderos positivos y fp falsos positivos.

Recall: Representa la relación entre las instancias correctamente clasificadas contra las instancias originales, alcanza su mejor valor en 1 y la peor puntuación en 0 y está determinada por la ecuación 9.

$$Recall = \frac{tp}{tp + fn} \tag{9}$$

Donde tp son verdaderos positivos y fn falsos negativos.

MSE: En los algoritmos de regresión el MSE promedia el error de predicción al cuadrado para todas las predicciones y encapsula tanto la exactitud como la precisión, alcanza su mejor valor teórico en 0 y está definida por la ecuación 10.

$$MSE = \frac{1}{N} \cdot \sum_{t=1}^{N} (y_t - f_t)^2 \tag{10}$$

Donde N es el número total de observaciones en el conjunto de datos, y_t es el valor real de la t-ésima observación, f_t es el valor predicho por el modelo para la i-ésima observación, $(y_t - f_t)^2$ es la diferencia al cuadrado entre el valor real y el valor predicho, que mide el error individual de cada observación Σ es la suma de todos los errores cuadrados para cada observación en el conjunto de datos y 1/n es el promedio de los errores cuadrados, lo que permite obtener una medida general del error del modelo.

El preprocesamiento de datos implementado en esta tesis, resultó en mejoras significativas en la precisión y capacidad de predicción en comparación con estudios previos. En el contexto industrial, contar con una mayor ventana de predicción de fallas es esencial para planificar intervenciones de mantenimiento con antelación, minimizando el impacto de las fallas no programadas. Este marco predictivo, apoyado en modelos de ML, permitió anticipar con mayor precisión el estado de los componentes, facilitando la programación de mantenimiento basada en datos históricos y lecturas de sensores [78]. Este enfoque proactivo optimizó los recursos, redujo el impacto de fallas imprevistas y aportó información valiosa sobre la fiabilidad y riesgo de los equipos [79].

Capítulo V

Resultados

5. Resultados

Este Capítulo presenta los resultados obtenidos tras la aplicación de la metodología de preprocesamiento y modelado predictivo en el diagnóstico de fallas en rodamientos de máquinas rotativas. En esta Sección, se analizan las métricas de evaluación de los modelos de clasificación y regresión, destacando la eficacia del método de preprocesamiento propuesto en esta tesis en comparación con enfoques previos de la literatura.

En la Tabla 11 se presentan los resultados de las métricas de evaluación obtenidos para los modelos de clasificación y regresión. Al utilizar los datos originales (sin preprocesamiento), la exactitud alcanzada fue del 26%, en contraste con estudios previos donde la precisión se situó entre el 93% y el 96% mediante diferentes técnicas de preprocesamiento [76], [77], [80]. Tras aplicar el método de preprocesamiento propuesto en esta tesis, la precisión en clasificación se incrementó hasta un **96%,** mientras que en los modelos de regresión se logró extender la ventana de predicción de fallas a **8,240 minutos**, en comparación con los 4,725 minutos obtenidos en estudios anteriores [80], representando un incremento del **74.4%** en el tiempo disponible para planificar el mantenimiento.

Además, con este enfoque se estiman beneficios económicos debido al aumento en la anticipación en las paradas no programadas. Según [4], la implementación de un sistema predictivo optimizado podría resultar en ahorros de hasta $5,356,000 USD en fábricas de bienes de consumo y $274,666,639 USD en la industria automotriz, evidenciando el impacto del modelo en la reducción de costos de mantenimiento y la mejora de la eficiencia operativa.

La literatura propone diversos métodos de preprocesamiento para abordar problemas de diagnóstico de fallas mediante técnicas de Machine Learning (ML). Entre los enfoques más comunes se incluyen la limpieza, normalización y escalamiento, pero estos procesos pueden comprometer información esencial para el modelo [79], [80]. Algunos estudios han optado por transformar señales de vibración en imágenes para aplicar algoritmos de análisis de imágenes; sin embargo, aunque esta estrategia puede facilitar el uso de ciertos modelos, también presenta riesgos. Forzar la adaptación de los datos sin considerar las características específicas de las señales de vibración puede degradar la calidad de las predicciones. Como destaca [80], es crucial ajustar los datos de manera que optimicen los resultados de los modelos sin comprometer su capacidad para capturar patrones relevantes, garantizando así la integridad de la información procesada.

Un aspecto clave en este estudio es que el conocimiento específico sobre el comportamiento de los rodamientos al presentar fallas ha sido utilizado para optimizar el tratamiento de los datos, evitando procesos innecesarios o costosos en términos computacionales. Este enfoque no solo mejora la precisión de las predicciones, sino que también permite optimizar los recursos de la empresa, lo que es un factor esencial en el PdM. Los resultados obtenidos demuestran que un preprocesamiento basado en entendimiento del sistema y los sensores puede generar una mejora significativa en comparación con técnicas más genéricas o automatizadas.

Tabla 11. Resultados

Dataset	Algoritmo	Mse	Accuracy	Precision	Recall
Original al 10%					
	LR	8.26	-	-	-
	NN	7.5	-	-	-
	DT	-	0.26	0.26	0.26
	RF	-	0.28	0.29	0.29
	SVM	-	0.31	0.31	0.31
[77]	CNN	-	0.95	-	-
	NN	0.0172	-	-	-
[76]	NN	-	0.93	-	-
[80]	NN	-	0.96	-	-
Método propuesto	LR	0.0325	-	-	-
	NN	0.0019	-	-	-
	DT	-	0.92	0.93	0.92
	RF	-	0.92	0.93	0.92
	SVM	-	0.96	0.96	0.94

Esta tesis emplea un enfoque de preprocesamiento basado en el conocimiento específico del comportamiento de los rodamientos en presencia de fallas, evitando así procesos innecesarios y computacionalmente costosos. Este enfoque, además de mejorar la precisión de las predicciones, optimiza los recursos, lo cual es fundamental en el PdM. Los resultados obtenidos demuestran que un preprocesamiento que aproveche la comprensión del sistema y de los sensores puede generar una mejora considerable en comparación con métodos genéricos o automatizados.

Capítulo VI

Conclusiones y trabajos futuros

6. Conclusiones y trabajos futuros

Conclusiones

En esta tesis se desarrolló un modelo de Mantenimiento Predictivo (PdM) para rodamientos en máquinas rotativas, optimizando tanto las técnicas de preprocesamiento como los modelos de Machine Learning (ML).

El preprocesamiento propuesto, que incluye la elección de muestras representativas y la transformación de datos basada en características clave, permitió incrementar la precisión de los modelos de clasificación hasta un 96% y ampliar la ventana de predicción de fallas en un 74.4% en comparación con estudios previos [78].

Estas mejoras destacan la importancia de personalizar el tratamiento de los datos de acuerdo con el origen, tipo de sensor y contexto operativo, elementos que son críticos para maximizar la precisión en escenarios industriales complejos. Además, este estudio evaluó tres hipótesis clave relacionadas con la mejora del preprocesamiento de datos en el mantenimiento predictivo:

Hipótesis 1: Se confirmó que el preprocesamiento de datos basado en su origen mejora significativamente la precisión de los modelos de ML en la detección de fallas, evidenciado por el aumento en la exactitud del modelo predictivo.

Hipótesis 2: Se logró una reducción parcial en el tiempo de entrenamiento de los modelos, dependiendo del algoritmo utilizado. Mientras que técnicas como Random Forest y Árboles de Decisión se beneficiaron directamente de la optimización del preprocesamiento, modelos más complejos como Redes Neuronales Recurrentes (RNN) aún requieren ajustes adicionales.

Hipótesis 3: Se verificó parcialmente que un enfoque de preprocesamiento basado en el origen de los datos reduce la complejidad computacional sin sacrificar precisión. Si bien el uso de datos mejor estructurados optimizó la eficiencia de los modelos, en algunos casos el ajuste y la validación cruzada siguieron requiriendo recursos computacionales significativos.

Los beneficios prácticos del modelo predictivo optimizado son significativos para la industria, con estimaciones de ahorro en costos de mantenimiento que alcanzan hasta $5,356,000 USD en bienes de consumo y $274,666,639 USD en el sector automotriz [4]. Estos resultados no solo reflejan un impacto económico directo, sino también una mejora en la planificación del mantenimiento, al permitir una intervención anticipada que minimiza las interrupciones no programadas, optimiza los recursos disponibles y mejora la continuidad operativa de los procesos industriales.

Además de los resultados cuantitativos, esta tesis enfatiza el valor de un enfoque basado en datos adaptados al entorno específico de operación, lo que refuerza su relevancia para aplicaciones prácticas en la Industria 4.0.

Trabajos Futuros

Se propone extender el modelo integrando datos de sensores adicionales, como los de temperatura, presión y humedad, con el objetivo de ampliar su aplicabilidad a otros tipos de máquinas y entornos industriales. Asimismo, se recomienda explorar algoritmos avanzados, incluyendo redes neuronales convolucionales (CNN) y redes neuronales recurrentes (RNN), que podrían permitir una mejor identificación de patrones complejos y una mayor robustez en las predicciones.

Un aspecto clave para futuros desarrollos será la validación del modelo en entornos industriales reales, donde se pueda evaluar su efectividad bajo condiciones operativas específicas. También se sugiere trabajar en la automatización del preprocesamiento de datos para simplificar la implementación del modelo y garantizar su adaptabilidad a diferentes configuraciones industriales. Finalmente, es importante considerar aspectos éticos y de seguridad en el manejo y uso de los datos, especialmente en entornos donde el PdM tiene implicaciones directas en la seguridad de las operaciones. Abordar estas líneas permitirá escalar la metodología propuesta a diversas industrias, fomentando un mantenimiento predictivo más eficiente, seguro y adaptable a las necesidades de la industria moderna.

Productos derivados

Como parte de los productos académicos generados en esta investigación, se publicó el artículo titulado "Optimización del Preprocesamiento de Datos para el Mantenimiento Predictivo Basado en Machine Learning" en la revista Computación y Sistemas, volumen 28, número 4. Este trabajo detalla los hallazgos clave de la metodología desarrollada en esta tesis, destacando las mejoras en la precisión de los modelos predictivos y su aplicabilidad en entornos industriales.

La publicación puede consultarse en el siguiente enlace:

- https://www.cys.cic.ipn.mx/ojs/index.php/CyS/article/view/4913

Capítulo VII

Bibliografía

7. Bibliografía

[1] Dhruv Patel y Prathmesh Kalgutkar, «Predictive Maintenance for Industrial Equipment Using Machine Learning», *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 618-625, ago. 2024, doi: 10.48175/IJARSCT-19379.

[2] A. Canito *et al.*, «Flexible Architecture for Data-Driven Predictive Maintenance with Support for Offline and Online Machine Learning Techniques», en *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, oct. 2021, pp. 1-7. doi: 10.1109/IECON48115.2021.9589230.

[3] H. Mahmoud, «Industrial Scalable Rolling Element Bearing Diagnostic and Prognostic Modelling».

[4] E. Johnson, «The True Cost of Downtime | Siemens», 2023. Accedido: 10 de diciembre de 2023. [En línea]. Disponible en: https://assets.new.siemens.com/siemens/assets/api/uuid:3d606495-dbe0-43e4-80b1-d04e27ada920/dics-b10153-00-7600truecostofdowntime2022-144.pdf

[5] «Predictive Maintenance Market Share, Global Industry Size Forecast», MarketsandMarkets. Accedido: 26 de noviembre de 2024. [En línea]. Disponible en: https://www.marketsandmarkets.com/Market-Reports/operational-predictive-maintenance-market-8656856.html

[6] M. R. F. https://www.marketresearchfuture.com, «Predictive Maintenance Market Size & Share Report, 2030». Accedido: 26 de noviembre de 2024. [En línea]. Disponible en: https://www.marketresearchfuture.com/reports/predictive-maintenance-market-2377

[7] M. M. Hamasha *et al.*, «Strategical selection of maintenance type under different conditions», *Sci. Rep.*, vol. 13, n.º 1, p. 15560, sep. 2023, doi: 10.1038/s41598-023-42751-5.

[8] A. Aboshosha, A. Haggag, N. George, y H. A. Hamad, «IoT-based data-driven predictive maintenance relying on fuzzy system and artificial neural networks», *Sci. Rep.*, vol. 13, n.º 1, p. 12186, jul. 2023, doi: 10.1038/s41598-023-38887-z.

[9] B. Einabadi, M. Mahmoodjanloo, A. Baboli, y E. Rother, «Dynamic predictive and preventive maintenance planning with failure risk and opportunistic grouping considerations: A case study in the automotive industry», *J. Manuf. Syst.*, vol. 69, pp. 292-310, ago. 2023, doi: 10.1016/j.jmsy.2023.06.012.

[10] H. Li, W. Zhu, L. Dieulle, y E. Deloux, «Condition-based maintenance strategies for stochastically dependent systems using Nested Lévy copulas», *Reliab. Eng. Syst. Saf.*, vol. 217, p. 108038, ene. 2022, doi: 10.1016/j.ress.2021.108038.

[11] W. Li, D. Xie, Y. Qian, X. Zhao, S. Cai, y Y. Hou, «Calculation of Rotor's Torsional Vibration Characteristics Based on Equivalent Diameter of Stiffness», en *2010 Asia-Pacific Power and Energy Engineering Conference*, mar. 2010, pp. 1-4. doi: 10.1109/APPEEC.2010.5449178.

[12] J. Kairo, «Machine Learning Algorithms for Predictive Maintenance in Manufacturing», *J. Technol. Syst.*, vol. 6, n.º 4, Art. n.º 4, ago. 2024, doi: 10.47941/jts.2144.

[13] D. Ghelani, «Harnessing machine learning for predictive maintenance in energy infrastructure: A review of challenges and solutions», *Int. J. Sci. Res. Arch.*, vol. 12, n.º 2, Art. n.º 2, 2024, doi: 10.30574/ijsra.2024.12.2.0525.
[14] C. V. Shah, «Machine Learning Algorithms for Predictive Maintenance in Autonomous Vehicles», *Int. J. Eng. Comput. Sci.*, vol. 13, n.º 01, Art. n.º 01, ene. 2024, doi: 10.18535/ijecs/v13i01.4786.
[15] T. Gupta, «A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance». Accedido: 18 de enero de 2022. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/8387124/
[16] H. Seo, D. Kim, y J.-H. Byun, «Data Pre-processing for Manufacturing Quality Improvement», 대한산업공학회지, vol. 49, n.º 3, pp. 248-257, jun. 2023, doi: 10.7232/JKIIE.2023.49.3.248.
[17] M. Ramkumar, K. Malathi, y K. Pavithra, «Optimizing Machine Learning Model Accuracy via OBNT Algorithm: Advanced Data Preprocessing Technique», en *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, dic. 2023, pp. 1-6. doi: 10.1109/ICSES60034.2023.10465344.
[18] L. C. Brito, G. A. Susto, J. N. Brito, y M. A. V. Duarte, «Fault Detection of Bearing: An Unsupervised Machine Learning Approach Exploiting Feature Extraction and Dimensionality Reduction», *Informatics*, vol. 8, n.º 4, Art. n.º 4, dic. 2021, doi: 10.3390/informatics8040085.
[19] G. Öcalan y İ. Türkoğlu, «Fault Diagnosis of Rotating Machines Using Raw Vibration Signals and Deep Learning», en *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*, oct. 2021, pp. 1-7. doi: 10.1109/ASYU52992.2021.9599062.
[20] S. Cho *et al.*, «A Hybrid Machine Learning Approach for Predictive Maintenance in Smart Factories of the Future», en *Advances in Production Management Systems. Smart Manufacturing for Industry 4.0*, I. Moon, G. M. Lee, J. Park, D. Kiritsis, y G. von Cieminski, Eds., Cham: Springer International Publishing, 2018, pp. 311-317. doi: 10.1007/978-3-319-99707-0_39.
[21] F. Larrinaga *et al.*, «A Big Data implementation of the MANTIS reference architecture for predictive maintenance», *Proc. Inst. Mech. Eng. Part J. Syst. Control Eng.*, vol. 233, n.º 10, Art. n.º 10, nov. 2019, doi: 10.1177/0959651819835362.
[22] A. Salvador Palau, M. H. Dhada, K. Bakliwal, y A. K. Parlikad, «An Industrial Multi Agent System for real-time distributed collaborative prognostics», *Eng. Appl. Artif. Intell.*, vol. 85, pp. 590-606, oct. 2019, doi: 10.1016/j.engappai.2019.07.013.
[23] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. G. Jun, y J. W. Sutherland, «Predictive Maintenance of Machine Tool Systems Using Artificial Intelligence Techniques Applied to Machine Condition Data», *Procedia CIRP*, vol. 80, pp. 506-511, ene. 2019, doi: 10.1016/j.procir.2018.12.019.
[24] C. Gutschi, N. Furian, J. Suschnigg, D. Neubacher, y S. Voessner, «Log-based predictive maintenance in discrete parts manufacturing», *Procedia CIRP*, vol. 79, pp. 528-533, ene. 2019, doi: 10.1016/j.procir.2019.02.098.

[25] N. Galarza-Urigoitia *et al.*, «Predictive maintenance of wind turbine low-speed shafts based on an autonomous ultrasonic system», *Eng. Fail. Anal.*, vol. 103, pp. 481-504, sep. 2019, doi: 10.1016/j.engfailanal.2019.04.048.
[26] M. De Benedetti, F. Leonardi, F. Messina, C. Santoro, y A. Vasilakos, «Anomaly detection and predictive maintenance for photovoltaic systems», *Neurocomputing*, vol. 310, pp. 59-68, oct. 2018, doi: 10.1016/j.neucom.2018.05.017.
[27] M. M. Tehrani, Y. Beauregard, M. Rioux, J. P. Kenne, y R. Ouellet, «A Predictive Preference Model for Maintenance of a Heating Ventilating and Air Conditioning System», *IFAC-Pap.*, vol. 48, n.º 3, Art. n.º 3, ene. 2015, doi: 10.1016/j.ifacol.2015.06.070.
[28] J. Shimada y S. Sakajo, «A statistical approach to reduce failure facilities based on predictive maintenance», en *2016 International Joint Conference on Neural Networks (IJCNN)*, jul. 2016, pp. 5156-5160. doi: 10.1109/IJCNN.2016.7727880.
[29] T. Borgi, A. Hidri, B. Neef, y M. S. Naceur, «Data analytics for predictive maintenance of industrial robots», en *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, ene. 2017, pp. 412-417. doi: 10.1109/ASET.2017.7983729.
[30] P. Strauß, M. Schmitz, R. Wöstmann, y J. Deuse, «Enabling of Predictive Maintenance in the Brownfield through Low-Cost Sensors, an IIoT-Architecture and Machine Learning», en *2018 IEEE International Conference on Big Data (Big Data)*, dic. 2018, pp. 1474-1483. doi: 10.1109/BigData.2018.8622076.
[31] M. Canizo, E. Onieva, A. Conde, S. Charramendieta, y S. Trujillo, «Real-time predictive maintenance for wind turbines using Big Data frameworks», en *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, jun. 2017, pp. 70-77. doi: 10.1109/ICPHM.2017.7998308.
[32] P. Aivaliotis, K. Georgoulias, y G. Chryssolouris, «A RUL calculation approach based on physical-based simulation models for predictive maintenance», en *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, jun. 2017, pp. 1243-1246. doi: 10.1109/ICE.2017.8280022.
[33] A. R. Santiago, M. Antunes, J. P. Barraca, D. Gomes, y R. L. Aguiar, «Predictive Maintenance System for Efficiency Improvement of Heating Equipment», en *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, abr. 2019, pp. 93-98. doi: 10.1109/BigDataService.2019.00019.
[34] B. V. Ramani, C. A. Amith, J. M. Oommen, J. Babu, T. Paul, y V. Sankar, «Predictive analysis for industrial maintenance automation and optimization using a smart sensor network», en *2016 International Conference on Next Generation Intelligent Systems (ICNGIS)*, sep. 2016, pp. 1-5. doi: 10.1109/ICNGIS.2016.7854004.
[35] M. Kostoláni, J. Murín, y Š. Kozák, «Intelligent predictive maintenance control using augmented reality», en *2019 22nd International Conference on Process Control (PC19)*, jun. 2019, pp. 131-135. doi: 10.1109/PC.2019.8815042.
[36] C. G. Novoa, G. A. G. Berríos, y R. A. Söderberg, «Predictive maintenance for motors based on vibration analysis with compact rio», en *2017 IEEE Central*

America and Panama Student Conference (CONESCAPAN), sep. 2017, pp. 1-6. doi: 10.1109/CONESCAPAN.2017.8277603.
[37] V. Mathew, T. Toby, V. Singh, B. M. Rao, y M. G. Kumar, «Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning», en *2017 IEEE International Conference on Circuits and Systems (ICCS)*, dic. 2017, pp. 306-311. doi: 10.1109/ICCS1.2017.8326010.
[38] K. Sachin, «Failure Prediction Model for Predictive Maintenance». Accedido: 18 de enero de 2022. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/8648634/authors
[39] A. Kanawaday y A. Sane, «Machine learning for predictive maintenance of industrial machines using IoT sensor data», en *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, nov. 2017, pp. 87-90. doi: 10.1109/ICSESS.2017.8342870.
[40] E. Sezer, D. Romero, F. Guedea, M. Macchi, y C. Emmanouilidis, «An Industry 4.0-Enabled Low Cost Predictive Maintenance Approach for SMEs», en *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, jun. 2018, pp. 1-8. doi: 10.1109/ICE.2018.8436307.
[41] G. K. Durbhaka y B. Selvaraj, «Predictive maintenance for wind turbine diagnostics using vibration signal analysis based on collaborative recommendation approach», en *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, sep. 2016, pp. 1839-1842. doi: 10.1109/ICACCI.2016.7732316.
[42] G. A. Susto, «An adaptive machine learning decision system for flexible predictive maintenance». Accedido: 18 de enero de 2022. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/6899418/
[43] M. Paolanti, «Machine Learning approach for Predictive Maintenance in Industry 4.0». Accedido: 18 de enero de 2022. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/8449150/
[44] D. Apiletti *et al.*, «iSTEP, an Integrated Self-Tuning Engine for Predictive Maintenance in Industry 4.0», en *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, dic. 2018, pp. 924-931. doi: 10.1109/BDCloud.2018.00136.
[45] D. Azevedo, B. Ribeiro, y A. Cardoso, «Web-based tool for predicting the Remaining Useful Lifetime of Aircraft Components», en *2019 5th Experiment International Conference (exp.at'19)*, jun. 2019, pp. 231-232. doi: 10.1109/EXPAT.2019.8876503.
[46] C. Sellami, A. Samet, y M. A. Bach Tobji, «Frequent Chronicle Mining: Application on Predictive Maintenance», en *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, dic. 2018, pp. 1388-1393. doi: 10.1109/ICMLA.2018.00226.
[47] B. Kroll, «System modeling based on machine learning for anomaly detection and predictive maintenance in industrial plants». Accedido: 18 de enero de 2022. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/7005202/
[48] F. Kabir, B. Foggo, y N. Yu, «Data Driven Predictive Maintenance of Distribution Transformers», en *2018 China International Conference on Electricity*

Distribution (CICED), sep. 2018, pp. 312-316. doi: 10.1109/CICED.2018.8592417.
[49] A. Cachada *et al.*, «Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture», en *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, sep. 2018, pp. 139-146. doi: 10.1109/ETFA.2018.8502489.
[50] Selcuk, «Predictive maintenance, its implementation and latest trends - Sule Selcuk, 2017». Accedido: 2 de octubre de 2024. [En línea]. Disponible en: https://journals.sagepub.com/doi/abs/10.1177/0954405415601640
[51] P. K. Gopalakrishnan, B. Kar, S. K. Bose, M. Roy, y A. Basu, «Live Demonstration: Autoencoder-Based Predictive Maintenance for IoT», en *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, may 2019, pp. 1-1. doi: 10.1109/ISCAS.2019.8702230.
[52] K. S. Kiangala y Z. Wang, «Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts», *Int. J. Adv. Manuf. Technol.*, vol. 97, n.º 9, pp. 3251-3271, ago. 2018, doi: 10.1007/s00170-018-2093-8.
[53] F. Civerchia, S. Bocchino, C. Salvadori, E. Rossi, L. Maggiani, y M. Petracca, «Industrial Internet of Things monitoring solution for advanced predictive maintenance applications», *J. Ind. Inf. Integr.*, vol. 7, pp. 4-12, sep. 2017, doi: 10.1016/j.jii.2017.02.003.
[54] T. Abbasi, K. H. Lim, y K. S. Yam, «Predictive Maintenance of Oil and Gas Equipment using Recurrent Neural Network», *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 495, n.º 1, p. 012067, abr. 2019, doi: 10.1088/1757-899X/495/1/012067.
[55] E. Traini, G. Bruno, G. D'Antonio, y F. Lombardi, «Machine Learning Framework for Predictive Maintenance in Milling», *IFAC-Pap.*, vol. 52, n.º 13, pp. 177-182, ene. 2019, doi: 10.1016/j.ifacol.2019.11.172.
[56] Miguel, «TIPOS DE MOTORES ELÉCTRICOS INTEGRANTES: VALERIA MARTINEZ MATIAS WILBERT VAZQUEZ DIAZ JAZMIN CITLALLI SEGURA LOPEZ MARTIN JESUS SALGADO GABRIELA JULISSA. - ppt descargar». Accedido: 17 de enero de 2022. [En línea]. Disponible en: https://slideplayer.es/slide/15714960/
[57] MATLAB, «Model IMU, GPS, and INS/GPS - MATLAB & Simulink - MathWorks España». Accedido: 17 de enero de 2022. [En línea]. Disponible en: https://es.mathworks.com/help/nav/ug/model-imu-gps-and-insgps.html
[58] Audiotechnica, «Una Breve Guía de Micrófonos - ¿Qué Hace un Micrófono?» Accedido: 17 de enero de 2022. [En línea]. Disponible en: https://www.audio-technica.com/es-ar/support/una-breve-guia-de-microfonos-que-hace-un-microfono/
[59] Ingeniería, «Sensores Automotrices archivos», INGENIERÍA Y MECÁNICA AUTOMOTRIZ. Accedido: 17 de enero de 2022. [En línea]. Disponible en: https://www.ingenieriaymecanicaautomotriz.com/category/mecanica/sensores-automotrices/
[60] «Kaggle: Your Machine Learning and Data Science Community». Accedido: 17 de enero de 2022. [En línea]. Disponible en: https://www.kaggle.com/
[61] Google, «Dataset Search». Accedido: 17 de enero de 2022. [En línea]. Disponible en: https://datasetsearch.research.google.com/

[62] NASA, «Prognostics Center of Excellence Data Set Repository - NASA». Accedido: 2 de octubre de 2024. [En línea]. Disponible en: https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/
[63] H. Qiu, J. Lee, J. Lin, y G. Yu, «Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics», *J. Sound Vib.*, vol. 289, n.º 4, pp. 1066-1090, feb. 2006, doi: 10.1016/j.jsv.2005.03.007.
[64] A. Werner, N. Zimmermann, y J. Lentes, «Approach for a Holistic Predictive Maintenance Strategy by Incorporating a Digital Twin», *Procedia Manuf.*, vol. 39, pp. 1743-1751, ene. 2019, doi: 10.1016/j.promfg.2020.01.265.
[65] K. Hribernik, M. von Stietencron, A. Bousdekis, B. Bredehorst, G. Mentzas, y K.-D. Thoben, «Towards a Unified Predictive Maintenance System - A Use Case in Production Logistics in Aeronautics», *Procedia Manuf.*, vol. 16, pp. 131-138, ene. 2018, doi: 10.1016/j.promfg.2018.10.168.
[66] A. Burkov, *The Hundred-Page Machine Learning Book en español*. 2019.
[67] F. Cánovas-García, F. Alonso-Sarría, F. Gomariz-Castillo, y F. Oñate-Valdivieso, «Modification of the random forest algorithm to avoid statistical dependence problems when classifying remote sensing imagery», *Comput. Geosci.*, vol. 103, pp. 1-11, jun. 2017, doi: 10.1016/j.cageo.2017.02.012.
[68] scale, «sklearn.preprocessing.scale — scikit-learn 0.23.2 documentation». Accedido: 6 de noviembre de 2020. [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.scale.html#sklearn.preprocessing.scale
[69] standarscale, «sklearn.preprocessing.StandardScaler — scikit-learn 0.23.2 documentation». Accedido: 6 de noviembre de 2020. [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[70] normalize, «sklearn.preprocessing.normalize — scikit-learn 0.23.2 documentation». Accedido: 6 de noviembre de 2020. [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html#sklearn.preprocessing.normalize
[71] «OrdinalEncoder», scikit-learn. Accedido: 17 de enero de 2022. [En línea]. Disponible en: https://scikit-learn/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html
[72] J. Bobadilla, «LA TRANSFORMADA DE FOURIER. UNA VISIÓN PEDAGÓGICA», p. 33, 1999.
[73] C. Gavilán Moreno, «El uso de la transformada de Hilbert para el diagnostico de vibraciones. Análisis de transitorios.», *Dyna Bilbao*, ene. 2007.
[74] UNAM, «Transformada Wavelet – acervo para el mejoramiento del aprendizaje de alumnos de ingeniería, en Inteligencia Artificial». Accedido: 18 de enero de 2022. [En línea]. Disponible en: http://virtual.cuautitlan.unam.mx/intar/?page_id=1108
[75] R. Somonte Martín, «Análisis de filtros para la detección de fallos en rodamientos industriales a partir de su señal de vibración», 2010, Accedido: 2

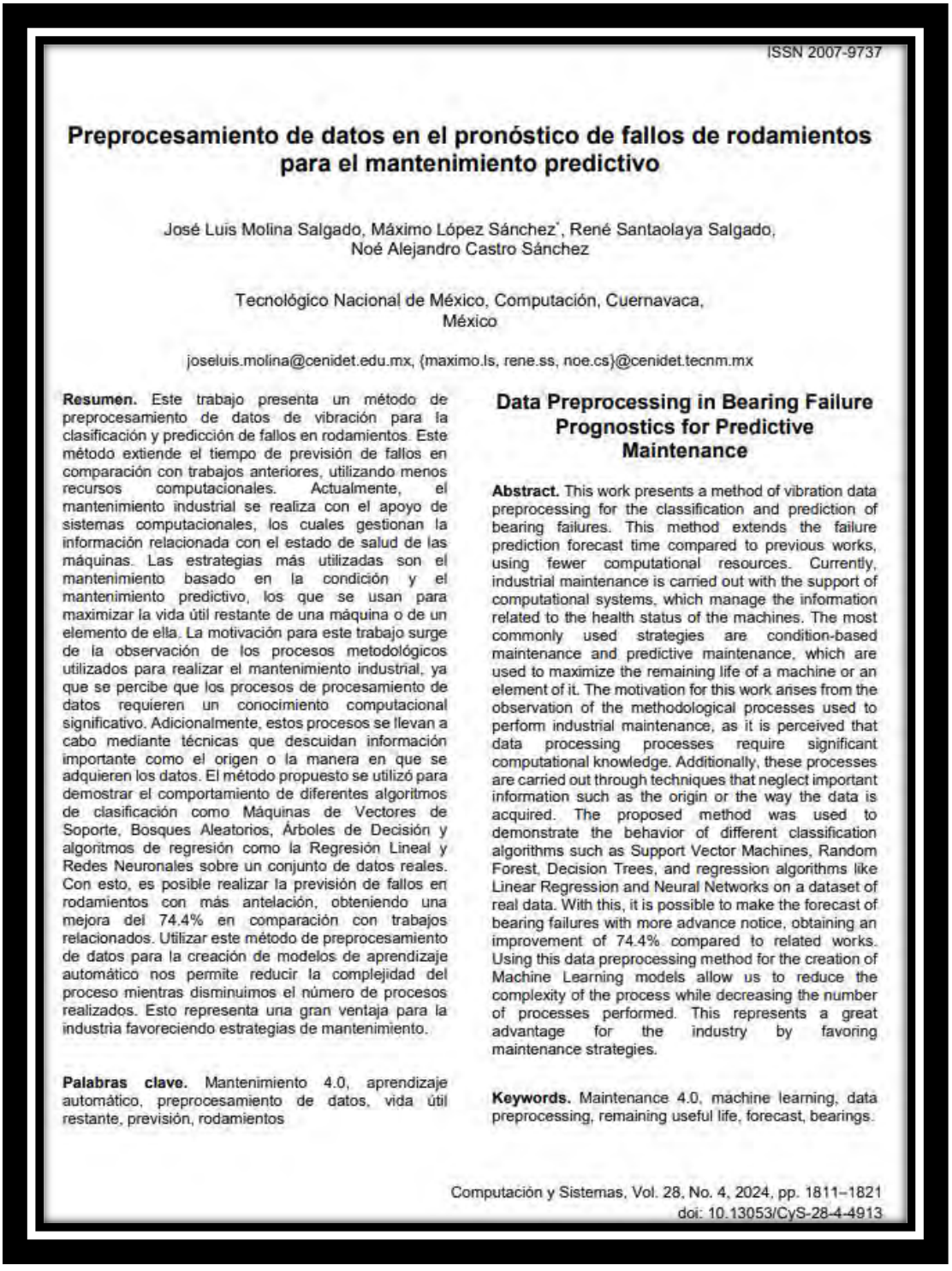
de julio de 2021. [En línea]. Disponible en: https://e-archivo.uc3m.es/handle/10016/10783
[76] J. Ben Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello, y F. Fnaiech, «Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals», *Appl. Acoust.*, vol. 89, pp. 16-27, mar. 2015, doi: 10.1016/j.apacoust.2014.08.016.
[77] J. Jiao, M. Zhao, J. Lin, y K. Liang, «A comprehensive review on convolutional neural network in machine fault diagnosis», *Neurocomputing*, vol. 417, pp. 36-63, dic. 2020, doi: 10.1016/j.neucom.2020.07.088.
[78] F. A. Khairy, G. Chetan, y R. Kosta, «A system for maintenance recommendation based on failure prediction», 2019
[79] P. C. Pereira, «Hidden Value in Maintenance System Data: Using Machine Learning to Correlate and Predict the Risk of Asset Failures», presentado en Offshore Technology Conference, OnePetro, may 2020. doi: 10.4043/30730-MS.
[80] W. Ahmad, S. A. Khan, y J.-M. Kim, «A Hybrid Prognostics Technique for Rolling Element Bearings Using Adaptive Predictive Models», *IEEE Trans. Ind. Electron.*, vol. 65, n.º 2, pp. 1577-1584, feb. 2018, doi: 10.1109/TIE.2017.2733487.

Anexos

Anexos

En esta sección se presentan las evidencias que documentan las actividades realizadas a lo largo del desarrollo del doctorado.

Artículo JCR, ponencias de publicación de artículos en congreso, Patentes otorgadas.



ISSN 2007-9737

Preprocesamiento de datos en el pronóstico de fallos de rodamientos para el mantenimiento predictivo

José Luis Molina Salgado, Máximo López Sánchez*, René Santaolaya Salgado, Noé Alejandro Castro Sánchez

Tecnológico Nacional de México, Computación, Cuernavaca, México

joseluis.molina@cenidet.edu.mx, {maximo.ls, rene.ss, noe.cs}@cenidet.tecnm.mx

Resumen. Este trabajo presenta un método de preprocesamiento de datos de vibración para la clasificación y predicción de fallos en rodamientos. Este método extiende el tiempo de previsión de fallos en comparación con trabajos anteriores, utilizando menos recursos computacionales. Actualmente, el mantenimiento industrial se realiza con el apoyo de sistemas computacionales, los cuales gestionan la información relacionada con el estado de salud de las máquinas. Las estrategias más utilizadas son el mantenimiento basado en la condición y el mantenimiento predictivo, los que se usan para maximizar la vida útil restante de una máquina o de un elemento de ella. La motivación para este trabajo surge de la observación de los procesos metodológicos utilizados para realizar el mantenimiento industrial, ya que se percibe que los procesos de procesamiento de datos requieren un conocimiento computacional significativo. Adicionalmente, estos procesos se llevan a cabo mediante técnicas que descuidan información importante como el origen o la manera en que se adquieren los datos. El método propuesto se utilizó para demostrar el comportamiento de diferentes algoritmos de clasificación como Máquinas de Vectores de Soporte, Bosques Aleatorios, Árboles de Decisión y algoritmos de regresión como la Regresión Lineal y Redes Neuronales sobre un conjunto de datos reales. Con esto, es posible realizar la previsión de fallos en rodamientos con más antelación, obteniendo una mejora del 74.4% en comparación con trabajos relacionados. Utilizar este método de preprocesamiento de datos para la creación de modelos de aprendizaje automático nos permite reducir la complejidad del proceso mientras disminuimos el número de procesos realizados. Esto representa una gran ventaja para la industria favoreciendo estrategias de mantenimiento.

Palabras clave. Mantenimiento 4.0, aprendizaje automático, preprocesamiento de datos, vida útil restante, previsión, rodamientos

Data Preprocessing in Bearing Failure Prognostics for Predictive Maintenance

Abstract. This work presents a method of vibration data preprocessing for the classification and prediction of bearing failures. This method extends the failure prediction forecast time compared to previous works, using fewer computational resources. Currently, industrial maintenance is carried out with the support of computational systems, which manage the information related to the health status of the machines. The most commonly used strategies are condition-based maintenance and predictive maintenance, which are used to maximize the remaining life of a machine or an element of it. The motivation for this work arises from the observation of the methodological processes used to perform industrial maintenance, as it is perceived that data processing processes require significant computational knowledge. Additionally, these processes are carried out through techniques that neglect important information such as the origin or the way the data is acquired. The proposed method was used to demonstrate the behavior of different classification algorithms such as Support Vector Machines, Random Forest, Decision Trees, and regression algorithms like Linear Regression and Neural Networks on a dataset of real data. With this, it is possible to make the forecast of bearing failures with more advance notice, obtaining an improvement of 74.4% compared to related works. Using this data preprocessing method for the creation of Machine Learning models allow us to reduce the complexity of the process while decreasing the number of processes performed. This represents a great advantage for the industry by favoring maintenance strategies.

Keywords. Maintenance 4.0, machine learning, data preprocessing, remaining useful life, forecast, bearings.

Computación y Sistemas, Vol. 28, No. 4, 2024, pp. 1811–1821
doi: 10.13053/CyS-28-4-4913


IXCICOM
Congreso Internacional de Computación
CERTIFICA QUE:
JOSE LUIS MOLINA SALGADO
Universidad
Ocaña - Colombia
UAGro
VIRTUAL


IXCICOM
Congreso Internacional de Computacion
CERTIFICA QUE:
JOSE LUIS MOLINA SALGADO
Presento la ponencia titulada:
Universidad
UAGro
VIRTUAL

ECONOMÍA IMPI

TÍTULO DE PATENTE No. 401178

Titular(es):	SECRETARIA DE EDUCACIÓN PÚBLICA - TECNOLÓGICO NACIONAL DE MÉXICO
Domicilio:	Arcos de Belén No. 79, Piso 3, Centro, 06010, Cuauhtémoc, Ciudad de México, MÉXICO
Denominación:	DISPOSTIVO, MÉTODO Y SISTEMA PARA LA EMISIÓN DE SEÑALES AUDIBLES Y VISUALES EN EQUIPOS REMOTOS.
Clasificación:	CIP: G06F3/01; A61B5/00; G06F3/03; H01L31/00; H03M7/40 CPC: G06F3/013; A61B5/1103; A61B5/6821; H03M7/4025
Inventor(es):	MÁXIMO LÓPEZ SÁNCHEZ; JOSÉ LUIS MOLINA SALGADO; MELISA HERNÁNDEZ SALINAS; NIMROD GONZÁLEZ FRANCO; JUAN GABRIEL GONZÁLEZ SERNA; JULIA YAZMÍN ARANA LLANES

SOLICITUD

Número:	**Fecha de Presentación:**	**Hora:**
MX/a/2017/008985	3 de julio de 2017	09:40

Vigencia: Veinte años

Fecha de Vencimiento: 3 de julio de 2037

Fecha de Expedición: 9 de marzo de 2023

La patente de referencia se otorga con fundamento en los artículos 1°, 2° fracción V, 6° fracción III, y 59 de la Ley de la Propiedad Industrial.

De conformidad con el artículo 23 de la Ley de la Propiedad Industrial, la presente patente tiene una vigencia de veinte años improrrogables, contada a partir de la fecha de presentación de la solicitud y estará sujeta al pago de la tarifa para mantener vigentes los derechos.

Quien suscribe el presente título lo hace con fundamento en lo dispuesto por los artículos 9° fracción I, 9, 10 y 119 de la Ley Federal de Protección a la Propiedad Industrial; artículos 1°, 3° fracción V, inciso a), 4° y 12° fracciones I y III del Reglamento del Instituto Mexicano de la Propiedad Industrial; artículos 1°, 3°, 4°, 5° fracción V, inciso a), 16 fracciones I y III y 30 del Estatuto Orgánico del Instituto Mexicano de la Propiedad Industrial; 1°, 3° y 5° fracción I Acuerdo Delegatorio de Facultades del Instituto Mexicano de la Propiedad Industrial.

El presente documento electrónico ha sido firmado mediante el uso de la firma electrónica avanzada por el servidor público competente, amparada por un certificado digital vigente a la fecha de su elaboración, y es válido de conformidad con lo dispuesto en los artículos 7 y 9 fracción I de la Ley de Firma Electrónica Avanzada y artículo 12 de su Reglamento. Su integridad y autoría, se podrá comprobar en [illegible]

Asimismo, se emitió conforme lo previsto por los artículos 1° fracción III; 2° fracción VI; 37, 38 y 39 del Acuerdo por el que se establecen lineamientos en materia de Servicios Electrónicos del Instituto Mexicano de la Propiedad Industrial.

SUBDIRECTORA DIVISIONAL DE EXAMEN DE FONDO DE PATENTES ÁREAS MECÁNICA, ELÉCTRICA Y DE DISEÑOS INDUSTRIALES Y MODELOS DE UTILIDAD

MARINA OLIMPIA CASTRO ALVEAR

Cadena Original:

MARINA OLIMPIA CASTRO ALVEAR|00001000000510173863|SERVICIO DE ADMINISTRACION TRIBUTARIA|[illegible]|MX/2023/30802|MX/a/2017/008985|Título de patente normal|[illegible]

Sello Digital:

[illegible]

MX/2023/30802

ECONOMÍA IMPI

TÍTULO DE PATENTE No. 406900

Titular(es):	SECRETARÍA DE EDUCACIÓN PÚBLICA - TECNOLÓGICO NACIONAL DE MÉXICO
Domicilio:	Arcos de Belén 79, Piso 3, Col. Centro, Cuauhtémoc, Ciudad de México, 06010, MÉXICO
Denominación:	DISPOSITIVO Y MÉTODO PARA LA MANIPULACIÓN DEL CURSOR EN EQUIPOS ELECTRÓNICOS A PARTIR DEL MOVIMIENTO REALIZADO EN LOS EJES LATERAL, LONGITUDINAL Y VERTICAL
Clasificación:	CIP: G08C17/06; A61B5/00; A61B5/11 CPC: G08C17/06; A61B5/721; G08C2201/00; G08C2201/30; G08C2201/31; G08C2201/32; G08C2201/50
Inventor(es):	FERNANDO PATIÑO REYES; MÁXIMO LOPEZ SÁNCHEZ; GUILLERMO ROJAS SOLIS; JUAN GABRIEL GONZÁLEZ SERNA; JORGE MISAEL RUIZ MARTÍNEZ; JOSE LUIS MOLINA SALGADO; NIMROD GONZÁLEZ FRANCO

SOLICITUD

Número:	Fecha de Presentación:	Hora:
MX/a/2019/000675	18 de Diciembre de 2018	12:45

Vigencia: Veinte años

Fecha de Vencimiento: 18 de diciembre de 2038

Fecha de Expedición: 29 de septiembre de 2023

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

SUBDIRECTORA DIVISIONAL DE EXAMEN DE FONDO DE PATENTES ÁREAS MECÁNICA, ELÉCTRICA Y DE DISEÑOS INDUSTRIALES Y MODELOS DE UTILIDAD

MARINA OLIMPIA CASTRO ALVEAR

Cadena Original:
[illegible]

Sello Digital:
[illegible]

MX/2023/92184