

Diagnóstico de falla de rodamientos con Machine Learning

Cesar Vicuña H., Johan Callomamani B., Adolfo Ramon P., Paul Cusi H., Jairo Pinedo T.
 Universidad Nacional de Ingeniería
 (Maestría de Inteligencia Artificial, Machine Learning)

Resumen.

El objetivo principal fue la implementación de modelos supervisados de machine learning para poder determinar si un rodamiento presenta falla de pista exterior o no, también evaluación del mejor tratamiento de datos para optimizar resultados, para ello se monitoreó vibraciones en un rodamiento el cual presenta un inicio normal y en el tiempo es llevado a la falla, con un equipo acelerómetros ICP de cuarzo de alta sensibilidad PCB 353B33.

Se aplicaron 03 modelos supervisados que fueron Decision tree Logistic regression, Random forest y Extreme gradient boosting (XGBoost) y se han utilizado 3 datasets distintos:

Dataset dominio en el tiempo (20,480 variables + 1 variables objetivo, 982 registros)

Dataset dominio en frecuencias (10,240 variables + 1 variables objetivo, 982 registros)

Dataset características estadísticas de señales (9 variables + 1 variables objetivo, 982 registros)

Abstract

The main objective is the implementation of supervised machine learning models to determine whether a bearing has an outer race failure or not, as well as the evaluation of the best data treatment to optimize results. To do this, vibration data is used in a bearing which has a normal start and is brought to failure over time.

The following supervised models are applied:

Decision tree

Logistic regression

Random forest

Extreme gradient boosting (XGBoost)

Three different datasets have been used:

Time domain dataset (20,480 variables + 1 target variable, 982 records)

Frequency domain dataset (10,240 variables + 1 target variable, 982 records)

Statistical characteristics of signals dataset (9 variables + 1 target variable, 982 records)

supervisados para el diagnóstico de fallas en rodamientos: Decision Tree, Logistic Regression, Random Forest y Extreme Gradient Boosting (XGBoost). Cada uno de estos modelos ofrece ventajas únicas en términos de interpretabilidad, capacidad de manejo de datos y precisión predictiva.

El modelo Decision Tree es conocido por su simplicidad e interpretabilidad, facilitando la comprensión de las decisiones del modelo y la identificación de las características más relevantes para la predicción [2]. Este modelo divide el espacio de datos en regiones homogéneas utilizando reglas de decisión jerárquicas, lo que permite una visualización clara de cómo se toman las decisiones. Además, su capacidad para manejar tanto datos numéricos como categóricos lo hace versátil en diversas aplicaciones.

La regresión logística, por otro lado, es una técnica estadística que permite la clasificación binaria y ha sido ampliamente utilizada en diagnósticos médicos y otras aplicaciones [3]. Su fundamento matemático se basa en la modelación de la probabilidad de un evento mediante una función logística, lo que permite una interpretación probabilística de los resultados. Aunque tradicionalmente se ha utilizado para problemas de clasificación binaria, sus extensiones permiten abordar problemas multicategoricos, ampliando su aplicabilidad.

Random Forest, una extensión del Decision Tree, mejora la precisión del modelo al reducir el riesgo de sobreajuste mediante la creación de múltiples árboles de decisión y la combinación de sus resultados [4]. Este enfoque de ensemble learning no solo aumenta la robustez del modelo, sino que también mejora la generalización a datos no vistos. Cada árbol en el bosque se entrena con un subconjunto diferente de los datos y, al final, se realiza un voto mayoritario para la clasificación final, lo que reduce el sesgo y la varianza del modelo.

Finalmente, XGBoost es un algoritmo de boosting altamente eficiente y preciso que ha ganado popularidad en competiciones de machine learning y aplicaciones industriales debido a su capacidad para manejar grandes volúmenes de datos y su excelente rendimiento predictivo [5]. Este modelo se basa en la construcción secuencial de árboles de decisión donde cada nuevo árbol intenta corregir los errores cometidos por los árboles anteriores. La implementación eficiente de XGBoost permite una rápida convergencia y una excelente capacidad de manejo de características ausentes y desbalanceo de clases.

El uso de estos modelos en el diagnóstico de fallas en rodamientos ha sido investigado en varios estudios previos. Por ejemplo, Li et al. [6] demostraron que el uso de Random Forest en la detección de fallas mejora significativamente la precisión comparado con modelos más simples. Zhang et al. [7] exploraron el uso de XGBoost en diagnósticos industriales y encontraron que su capacidad para manejar datos desbalanceados y su eficiencia computacional lo hacen ideal para aplicaciones en tiempo real. Además, el estudio de Wuest et al. [8] resaltó la importancia de la selección adecuada de características y su impacto en la precisión de

I. INTRODUCCIÓN

El diagnóstico de fallas en rodamientos es un aspecto crítico en el mantenimiento predictivo de sistemas rotativos, ya que estas fallas pueden llevar a costosas paradas de maquinaria y a fallos catastróficos si no se detectan a tiempo. Los rodamientos son componentes esenciales en diversas aplicaciones industriales y su desempeño confiable es vital para la operación eficiente de equipos como motores, generadores y sistemas de transmisión [1]. Con el avance de las tecnologías de machine learning, se han desarrollado diversos modelos para mejorar la precisión y eficiencia en la detección de estas fallas. Entre estos modelos, los supervisados han demostrado un desempeño destacado debido a su capacidad para aprender patrones complejos a partir de datos etiquetados.

En este trabajo, se investiga el uso de cuatro modelos

los modelos de machine learning aplicados a la detección de fallas. Otros estudios han explorado la combinación de estos modelos con técnicas de preprocesamiento de señales y extracción de características para mejorar aún más la precisión del diagnóstico [9], [10].

En las siguientes secciones, se describen en detalle la metodología de implementación de estos modelos, los conjuntos de datos utilizados para el entrenamiento y prueba, y se comparan los resultados obtenidos para determinar el modelo más adecuado para el diagnóstico de fallas en rodamientos.

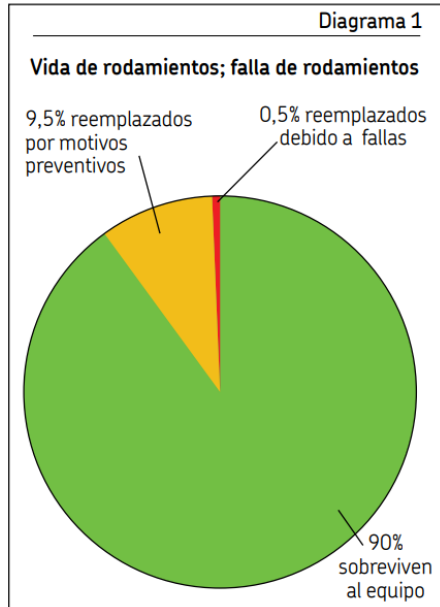


Figura 1. Referencias Daños de rodamientos [SKF]

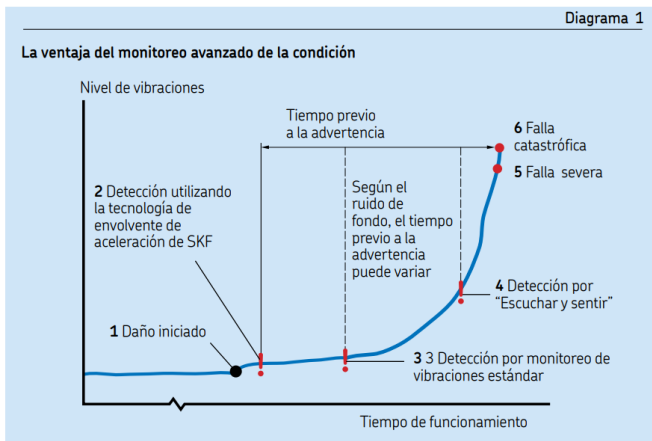


Figura 2. Referencias Daños de rodamientos [SKF]

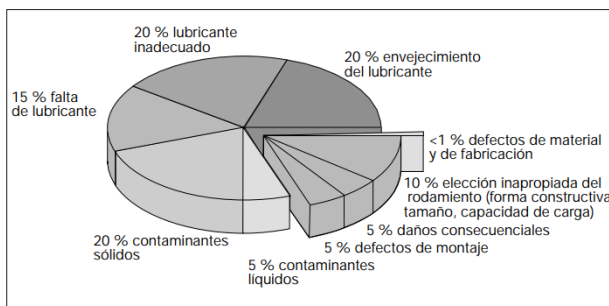


Figura 3. Referencia de Averías de los rodamientos [FAG]

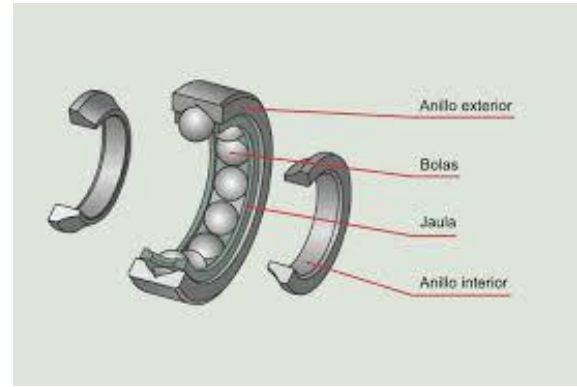


Figura 4. Elemento rodante.



Figura 5. Cojinete con rodamiento

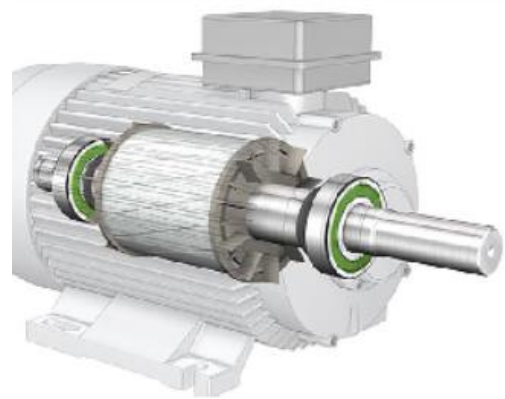


Figura 6. Uso del elemento rodante.

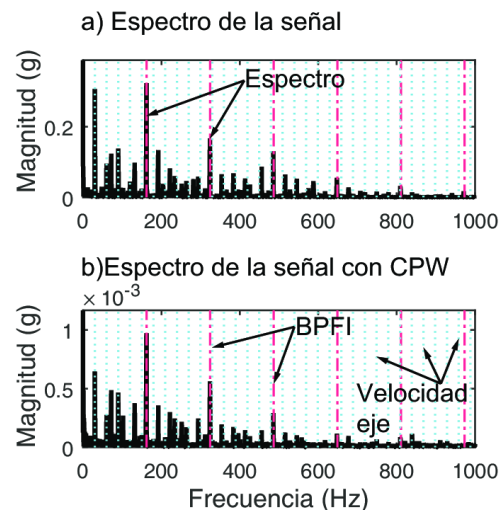


Figura 7. Fallas de rodamientos.

II. MARCO TEÓRICO

a. Diagnóstico de Fallas en Rodamientos

El diagnóstico de fallas en rodamientos es esencial en el mantenimiento predictivo de equipos industriales. Los rodamientos son componentes críticos en diversas máquinas rotativas, y su fallo puede causar paradas imprevistas y costosas reparaciones. Tradicionalmente, se han utilizado técnicas basadas en el análisis de vibraciones y señales acústicas para detectar fallas en rodamientos [11]. Sin embargo, la precisión y eficiencia de estas técnicas pueden ser limitadas por factores como el ruido ambiental y la complejidad de las señales.

b. Machine Learning en el Diagnóstico de Fallas

La aplicación de machine learning en el diagnóstico de fallas ha ganado considerable atención debido a su capacidad para manejar grandes volúmenes de datos y descubrir patrones complejos. Los algoritmos de machine learning pueden ser clasificados en supervisados y no supervisados. En el contexto de diagnóstico de fallas, los modelos supervisados son ampliamente utilizados debido a su capacidad para aprender a partir de datos etiquetados [12]. Estos modelos incluyen técnicas como árboles de decisión, regresión logística, bosques aleatorios y XGBoost.

c. Árboles de Decisión

Los árboles de decisión son algoritmos de clasificación que dividen iterativamente el espacio de características mediante reglas de decisión basadas en los valores de los atributos. Este enfoque permite una visualización clara y una fácil interpretación de los resultados [13]. Los árboles de decisión son eficaces para manejar tanto datos numéricos como categóricos, y son particularmente útiles en aplicaciones donde la interpretabilidad es crucial. La función de decisión en un árbol de decisión puede ser representada como:

$$f(x) = \sum_{i=1}^N w_i \cdot I(x \in R_i)$$

donde w_i es el peso asignado a la región R_i , y $I(\cdot)$ es una función indicadora que es igual a 1 si x pertenece a R_i y 0 en caso contrario.

d. Regresión Logística

La regresión logística es una técnica estadística utilizada para la clasificación binaria. Modela la probabilidad de un evento en función de una o más variables independientes utilizando la función logística [14]. La regresión logística es apreciada por su simplicidad y por proporcionar una salida probabilística, lo que permite una interpretación directa de las predicciones. La función de probabilidad en la regresión logística se define como:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n)}}$$

donde $\beta_0, \beta_1, \dots, \beta_n$ son los coeficientes del modelo que se estiman a partir de los datos.

e. Bosques Aleatorios

El algoritmo de bosques aleatorios es una extensión del árbol de decisión que mejora la precisión al reducir el riesgo de sobreajuste. Este modelo de ensemble learning crea múltiples árboles de decisión utilizando diferentes subconjuntos de datos y características, y combina sus resultados mediante un voto mayoritario [15]. Los bosques aleatorios son robustos y eficaces en la reducción de la varianza del modelo, lo que mejora su capacidad de generalización. El proceso de combinación de resultados en bosques aleatorios puede ser expresado como:

$$\hat{y} = \text{majority_vote}\{h_1(x), h_2(x), \dots, h_k(x)\}$$

donde $h_i(x)$ representa el resultado del i -ésimo árbol de decisión.

f. XGBoost

XGBoost es un algoritmo de boosting altamente eficiente y preciso que ha demostrado un rendimiento superior en varias competiciones de machine learning. Este algoritmo construye secuencialmente árboles de decisión, donde cada nuevo árbol intenta corregir los errores de los árboles anteriores [16]. XGBoost es conocido por su capacidad para manejar grandes volúmenes de datos y por su eficiencia computacional. El objetivo en XGBoost se define como la minimización de una función de pérdida regularizada:

$$\mathcal{L}(t) = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^t \Omega(f_k)$$

donde l es la función de pérdida, $\hat{y}_i^{(t)}$ es la predicción del modelo en la iteración t , y Ω es el término de regularización que penaliza la complejidad del modelo.

g. Extracción de Características

La precisión de los modelos de machine learning depende en gran medida de la calidad de las características extraídas de los datos. En el contexto del diagnóstico de fallas en rodamientos, las características comúnmente extraídas incluyen medidas estadísticas de señales de vibración, transformadas de Fourier, y análisis de tiempo-frecuencia [17]. Técnicas avanzadas como la Transformada Wavelet y la Descomposición de Modos Empíricos también se utilizan para mejorar la calidad de las características [18].

h. Preprocesamiento de Datos

El preprocesamiento de datos es un paso crucial en la implementación de modelos de machine learning. Incluye la normalización de datos, la eliminación de ruido, y el manejo de datos faltantes. La normalización asegura que las características tengan escalas comparables, mientras que la eliminación de ruido mejora la relación señal-ruido [19]. El manejo de datos faltantes puede realizarse mediante imputación o eliminación, dependiendo de la cantidad de datos faltantes y su importancia.

i. Evaluación de Modelos

La evaluación de modelos de machine learning se realiza utilizando métricas como precisión, recall, F1-score y área bajo la curva ROC (AUC-ROC). Estas métricas proporcionan una visión completa del rendimiento del modelo en términos de su

capacidad para predecir correctamente las clases positivas y negativas [20]. La validación cruzada es una técnica comúnmente utilizada para evaluar la robustez del modelo y evitar el sobreajuste.

j. Aplicaciones Industriales

Los modelos de machine learning para el diagnóstico de fallas en rodamientos han sido implementados en diversas aplicaciones industriales, incluyendo la manufactura, la energía eólica, y el transporte. Estos modelos han demostrado mejorar significativamente la eficiencia del mantenimiento predictivo y reducir los costos operativos [21]. La integración de estos modelos con sistemas de monitoreo en tiempo real permite la detección temprana de fallas y la planificación de intervenciones de mantenimiento más efectivas [22].

III. IDENTIFICACIÓN DEL PROBLEMA

La falla de rodamientos es una de las principales causas de fallos en maquinaria industrial. Un rodamiento defectuoso puede causar paradas imprevistas, lo que resulta en costosos tiempos de inactividad y posibles daños adicionales a la maquinaria.

Se requiere un sistema eficiente para monitorear y controlar la evolución de las fallas en los rodamientos. Esto permitirá programar el reemplazo de los rodamientos de manera planificada, minimizando el impacto en la producción y evitando paradas imprevistas.

Para la presente investigación se instalaron acelerómetros ICP de cuarzo de alta sensibilidad PCB 353B33 en la carcasa del cojinete (dos acelerómetros para cada cojinete [ejes x e y] para el conjunto de datos).



Figura 8. Sensor de vibración .

IV. SOLUCIÓN

Para determinar el tipo de falla de un rodamiento vamos a realizar un modelo de clasificación, con lo cual vamos a poder determinar el tipo de falla que presenta el rodamiento, los modelos que se van utilizar son:

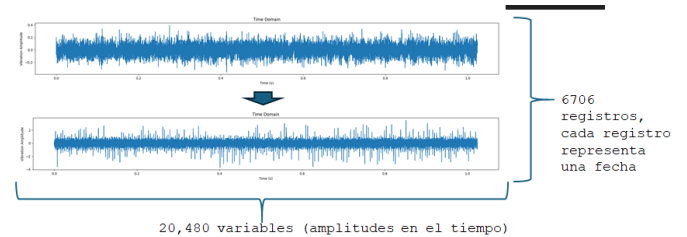
- 1) Árboles de decisión.
- 2) Regresión logística
- 3) Random forest
- 4) Clasificador XGBoost

V. DESCRIPCIÓN DEL CONJUNTO DE DATOS

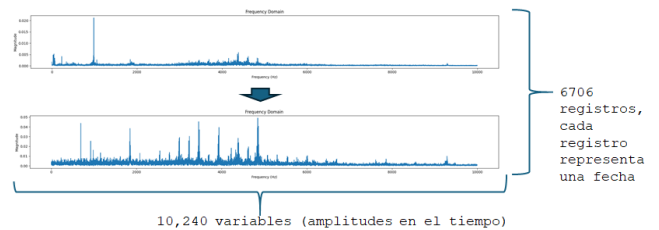
La información utilizada en el presente estudio abarca datos obtenidos de la toma de vibración de 3 datasets distintos:

- Dataset dominio en el tiempo (20,480 variables + 1 variables objetivo, 982 registros)
- Dataset dominio en frecuencias (10,240 variables + 1 variables objetivo, 982 registros)
- Dataset características estadísticas de señales(9 variables + 1 variables objetivo, 982 registros)

Para la presente investigación se utilizaron Dataset en dominio del tiempo (20,480 variables + 1 variables objetivo, 6706 registros)

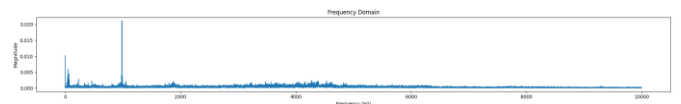


Dataset dominio en el tiempo (20,480 variables + 1 variables objetivo, 6706 registros).

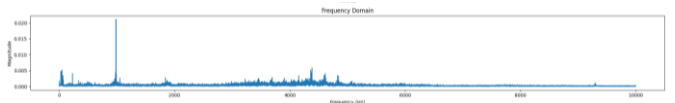


Variable objetivo se asigna estudiando el dominio de frecuencias, se caracteriza la falla y se agrega la columna clase al dataset.

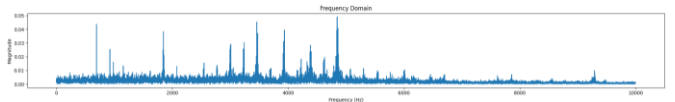
Rodamiento Normal:



Rodamiento Inicio de falla.



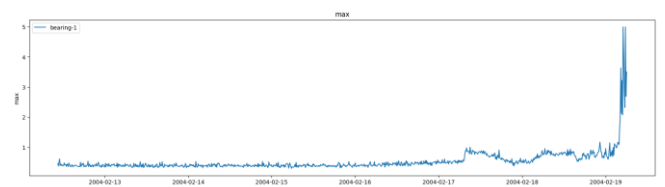
Rodamiento con falla catastrófica



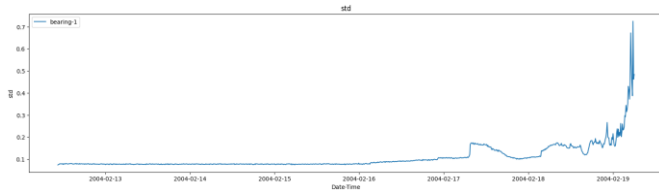
Dataset características estadísticas de señales(9 variables + 1 variables objetivo, 6706 registros)

Características de las variables

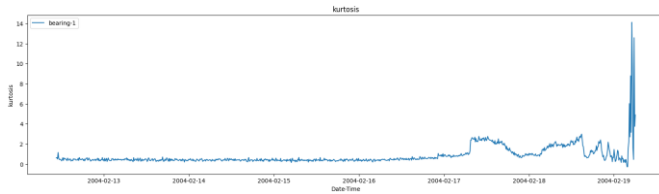
. Maximos



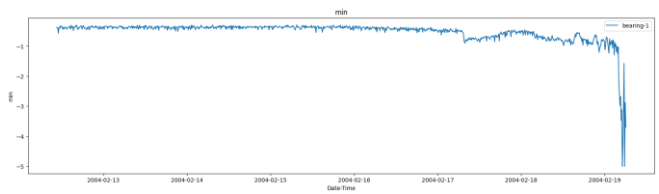
. Desviacion estándar



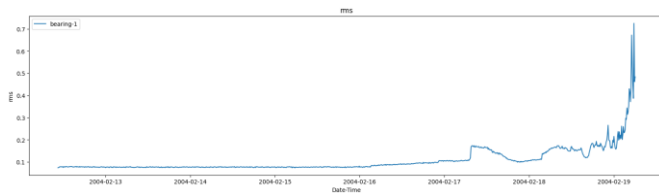
. Kurtosis



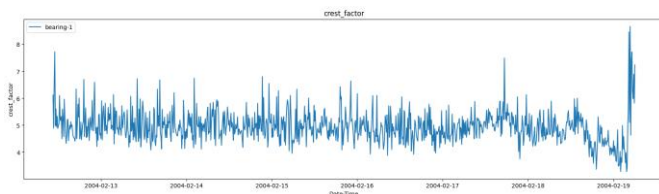
. Minimo



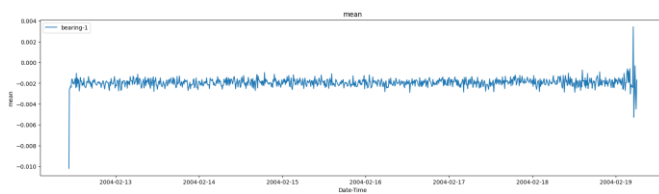
. Media cuadratica



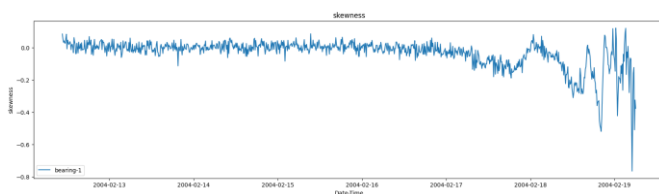
. Factor de cresta



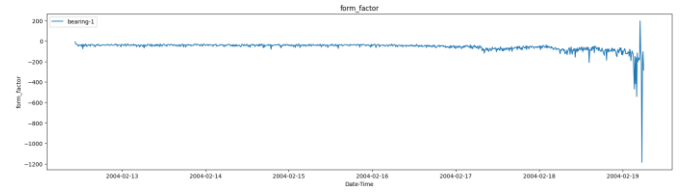
. Media



. Skewness



. Factor de forma



VI. PREPARACION DE LA INFORMACION

Se realizó con el programa Python

1. Transformación a dominio de frecuencias
2. Búsqueda del inicio de falla
3. Asignación de la variable objetivo
4. Transformación características de señales

VII. DESCRIPCIÓN DE TÉCNICAS UTILIZADAS

Algoritmos de machine learning:

Para los 3 datasets utilizamos los siguientes modelos de machine learning supervisados, decision tree, logistic regression, random forest, extreme gradient boosting (XGBoost).

No aplicamos escalamiento, para las dataset del dominio del tiempo y frecuencia, si para el dataset de características.

Dividimos la data de entrenamiento y de prueba en 80% y 20% respectivamente.

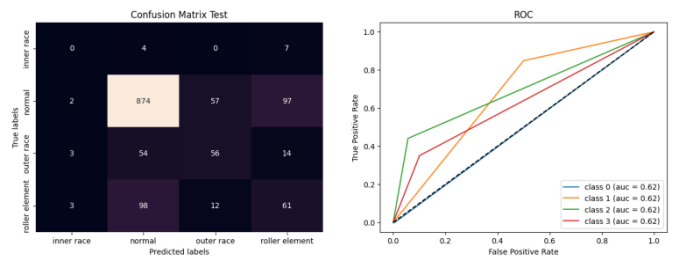
Hacemos uso de la matriz de confusion para analizar el resultado de los modelos.

VIII. RESULTADOS

- Decision tree:

1. Dataset dominio en el tiempo

	precision	recall	f1-score	support
0	0.00	0.00	0.00	11
1	0.85	0.85	0.85	1030
2	0.45	0.44	0.44	127
3	0.34	0.35	0.35	174
accuracy			0.74	1342
macro avg	0.41	0.41	0.41	1342
weighted avg	0.74	0.74	0.74	1342
AUC ROC :	0.6220294055911942			

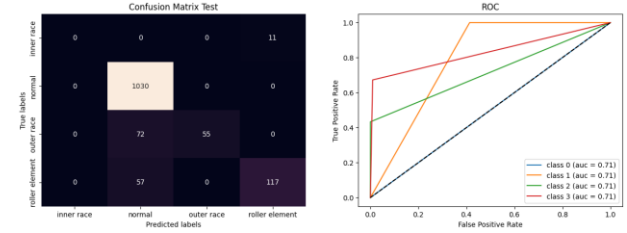


2. Dataset dominio en la frecuencia

2. Dataset dominio en la frecuencia

	precision	recall	f1-score	support
0	0.00	0.00	0.00	11
1	0.89	1.00	0.94	1030
2	1.00	0.43	0.60	127
3	0.91	0.67	0.77	174
accuracy			0.90	1342
macro avg	0.70	0.53	0.58	1342
weighted avg	0.90	0.90	0.88	1342

AUC ROC : 0.7103256640705581



3. Dataset de características

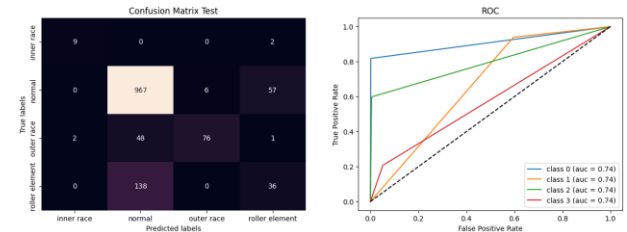
	precision	recall	f1-score	support
0	0.92	1.00	0.96	11
1	0.99	0.99	0.99	1030
2	1.00	0.98	0.99	127
3	0.92	0.95	0.94	174
accuracy			0.98	1342
macro avg	0.96	0.98	0.97	1342
weighted avg	0.98	0.98	0.98	1342

AUC ROC : 0.9853873692478821

3. Dataset de características

	precision	recall	f1-score	support
0	0.82	0.82	0.82	11
1	0.84	0.94	0.89	1030
2	0.93	0.60	0.73	127
3	0.38	0.21	0.27	174
accuracy			0.81	1342
macro avg	0.74	0.64	0.67	1342
weighted avg	0.79	0.81	0.79	1342

AUC ROC : 0.7385467384797968

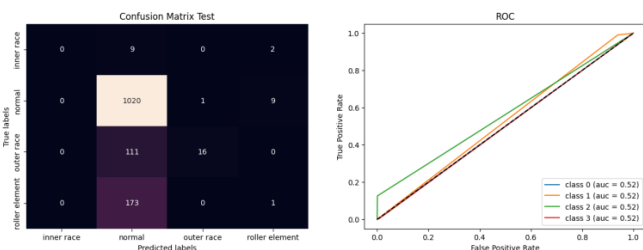


• Logistic regression:

1. Dataset dominio en el tiempo

	precision	recall	f1-score	support
0	0.00	0.00	0.00	11
1	0.78	0.99	0.87	1030
2	0.94	0.13	0.22	127
3	0.08	0.01	0.01	174
accuracy			0.77	1342
macro avg	0.45	0.28	0.28	1342
weighted avg	0.70	0.77	0.69	1342

AUC ROC : 0.521584902868997

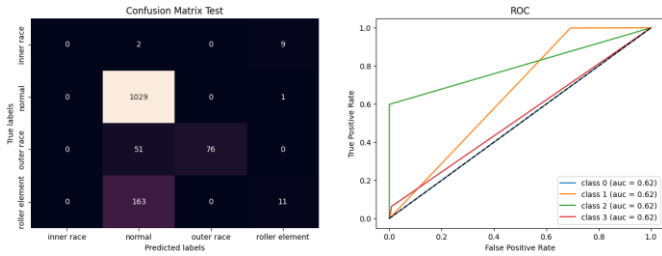


• Random forest:

1. Dataset dominio en el tiempo

	precision	recall	f1-score	support
0	0.00	0.00	0.00	11
1	0.83	1.00	0.90	1030
2	1.00	0.60	0.75	127
3	0.52	0.06	0.11	174
accuracy			0.83	1342
macro avg	0.59	0.42	0.44	1342
weighted avg	0.80	0.83	0.78	1342

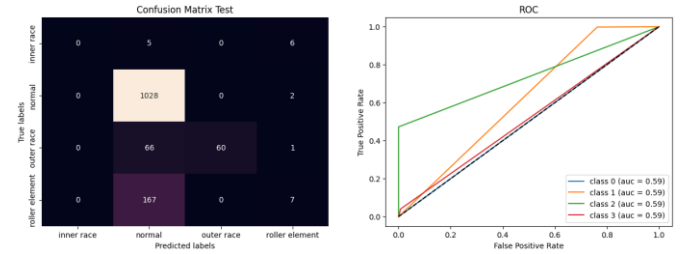
AUC ROC : 0.6199754222156594



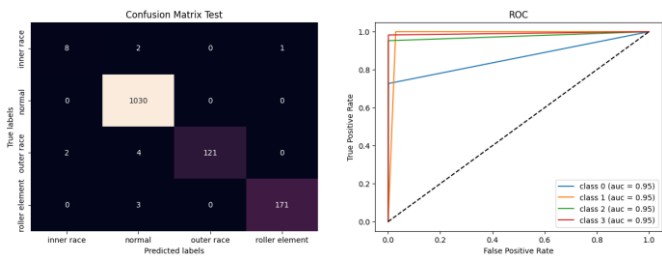
2. Dataset dominio en la frecuencia

	precision	recall	f1-score	support
0	0.80	0.73	0.76	11
1	0.99	1.00	1.00	1030
2	1.00	0.95	0.98	127
3	0.99	0.98	0.99	174
accuracy			0.99	1342
macro avg	0.95	0.92	0.93	1342
weighted avg	0.99	0.99	0.99	1342
AUC ROC : 0.9539477882053344				

	precision	recall	f1-score	support
0	0.00	0.00	0.00	11
1	0.81	1.00	0.90	1030
2	1.00	0.47	0.64	127
3	0.44	0.04	0.07	174
accuracy			0.82	1342
macro avg	0.56	0.38	0.40	1342
weighted avg	0.77	0.82	0.76	1342
AUC ROC : 0.5925253862617472				



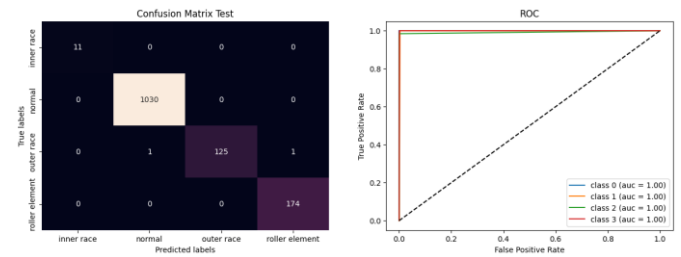
2. Dataset dominio en la frecuencia



3. Dataset de características

	precision	recall	f1-score	support
0	0.92	1.00	0.96	11
1	0.99	0.99	0.99	1030
2	1.00	0.99	1.00	127
3	0.95	0.96	0.96	174
accuracy			0.99	1342
macro avg	0.97	0.99	0.98	1342
weighted avg	0.99	0.99	0.99	1342
AUC ROC : 0.9892615726997428				

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	1.00	1.00	1030
2	1.00	0.98	0.99	127
3	0.99	1.00	1.00	174
accuracy			1.00	1342
macro avg	1.00	1.00	1.00	1342
weighted avg	1.00	1.00	1.00	1342
AUC ROC : 0.997523834489406				

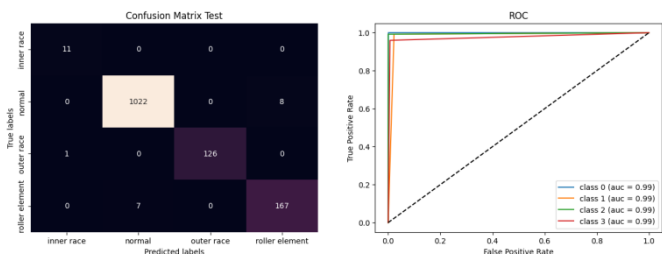


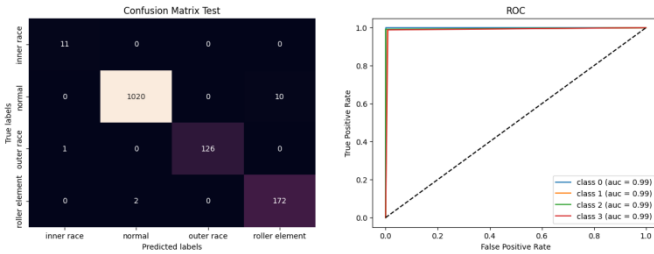
3. Dataset de Características

	precision	recall	f1-score	support
0	0.92	1.00	0.96	11
1	1.00	0.99	0.99	1030
2	1.00	0.99	1.00	127
3	0.95	0.99	0.97	174
accuracy			0.99	1342
macro avg	0.96	0.99	0.98	1342
weighted avg	0.99	0.99	0.99	1342
AUC ROC : 0.9943999723084441				

- Extreme gradient boosting (XGBoost):

1. Dataset dominio en el tiempo





IX. CONCLUSIONES

- 4.1. Se concluye que el dataset de vibración en el tiempo es el que menos desempeño tiene con todos los modelos y comparándolo con los demás datasets. (Mejor desempeño con precisión 0.93 con Random forest y peor desempeño con 0.64 con Regresión logística).
- 4.2. Se concluye que el dataset de características tuvo mejor desempeño y cercano al dataset de frecuencia (Mejor desempeño con precisión 0.99 con Random forest y peor desempeño con 0.93 con Regresión logística) mientras que dataset de frecuencia tuvo un desempeño aceptable (Mejor desempeño con precisión 0.99 con Random forest y peor desempeño con 0.86 con Regresión logística)
- 4.3. El modelo de regresión logística brinda menor precisión.
- 4.4. Se concluye que el dataset que toma más tiempo en ser ejecutado es el del tiempo 10.7 segundos en promedio entre los modelos, seguido por el de frecuencias con 3.8 segundos y el más rápido es el de características con 0.2 segundos en promedio.
- 4.5. Se concluye que el dataset que cuesta más almacenar es el del tiempo con 218 MB seguido del de frecuencias con 125 MB y el más óptimo para almacenar es el de características con 0.1 MB.
- 4.6. Se concluye que, analizando todas las variables como desempeño, tiempo de ejecución y especie del dataset, el mejor dataset para la predicción de falla es el de características con el modelo de decision tree (precisión de 0.99, con 0.1 segundos de ejecución y 0.1MB de espacio de almacenamiento del dataset).

X. RECOMENDACIONES

- ✓ Se recomienda realizar el estudio con mayores tipos de fallas de rodamientos, como son en elemento rodante y pista interior.
- ✓ Se recomienda categorizar cada falla en normal, alerta, crítico por cada tipo de falla, para poder tener mayor control de las fallas de los rodamientos.

XI. REFERENCIAS

- [1] R. K. Mobley, *An Introduction to Predictive Maintenance*, 2nd ed., Butterworth-Heinemann, 2002.
- [2] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81-106, 1986.
- [3] D. W. Hosmer, S. Lemeshow, *Applied Logistic Regression*, 2nd ed., Wiley-Interscience, 2000.
- [4] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.
- [6] X. Li, Q. Ding, y J. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1-11, 2018.
- [7] J. Zhang, L. Zhang, X. Si, y M. Zhang, "Machine Learning Algorithms in the Application of Aerospace Bearing Fault Diagnosis: A Review," *Sensors*, vol. 21, no. 9, p. 3114, 2021.
- [8] T. Wuest, D. Weimer, C. Irgens, y K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23-45, 2016.
- [9] S. M. Salih, A. J. Abdulrahman, y A. M. Shareef, "Vibration-Based Bearing Fault Diagnosis Using Machine Learning Approaches," *IEEE Access*, vol. 8, pp. 66233-66245, 2020.
- [10] M. E. Torres, E. Hernández, y F. A. Fernández, "Feature extraction methods for machine learning in condition monitoring and fault diagnosis of industrial processes," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1541-1551, 2017.
- [11] R. K. Mobley, *An Introduction to Predictive Maintenance*, 2nd ed., Butterworth-Heinemann, 2002.
- [12] T. Hastie, R. Tibshirani, y J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer, 2009.
- [13] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81-106, 1986.
- [14] D. W. Hosmer, S. Lemeshow, y R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed., Wiley, 2013.
- [15] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [16] T. Chen y C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.
- [17] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed., Academic Press, 2008.
- [18] N. E. Huang et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903-995, 1998.
- [19] I. Guyon y A. Elisseeff, "An introduction to feature extraction," en *Feature Extraction: Foundations and Applications*, Springer, 2006, pp. 1-25.
- [20] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861-874, 2006.
- [21] S. M. Salih, A. J. Abdulrahman, y A. M. Shareef, "Vibration-Based Bearing Fault Diagnosis Using Machine Learning Approaches," *IEEE Access*, vol. 8, pp. 66233-66245, 2020.
- [22] M. E. Torres, E. Hernández, y F. A. Fernández, "Feature extraction methods for machine learning in condition monitoring and fault diagnosis of industrial processes," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1541-1551, 2017.