

# Aplicación de redes neuronales profundas para detección automática de roce en turbinas de gas aeroderivadas

Antonio Squicciarini<sup>1</sup>, Alejandro Zarzo Altarejos<sup>1</sup>, Carlos E. González Guillén<sup>1</sup>, Juan Manuel Muñoz Guijosa<sup>2</sup>

<sup>1</sup>Departamento de Matemática Aplicada a la Ingeniería Industrial, Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, España. Email: [a.squicciarini@alumnos.upm.es](mailto:a.squicciarini@alumnos.upm.es), [alejandro.zarzo@upm.es](mailto:alejandro.zarzo@upm.es), [carlos.gguillen@upm.es](mailto:carlos.gguillen@upm.es)

<sup>2</sup>Departamento de Ingeniería Mecánica, Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, España. Email: [juanmanuel.munoz.guijosa@upm.es](mailto:juanmanuel.munoz.guijosa@upm.es)

## Resumen

El roce del rotor contra el estátor es uno de los fallos más comunes en maquinarias rotativas. En las turbinas aeroderivadas, la detección de este mal funcionamiento se ve dificultada por condicionantes de diseño, imponiendo la utilización de sensores afectados por ruido y vibraciones de otros componentes, limitando la efectividad y aplicabilidad de los sistemas tradicionales de detección, basados en transformada de Fourier. Los algoritmos de deep learning han mostrado resultados sorprendentes en varios problemas de monitorización automática de sistemas industriales. Sin embargo, necesitan de grandes bases de datos para describir el fenómeno, por lo que los costes y riesgos asociados a la disponibilidad de estos datos limita su aplicación. Nuestra propuesta consiste en la implementación de algoritmos de deep learning que en la fase de entrenamiento utilizan únicamente datos de mal funcionamiento simulados mediante un modelo ajustado de elementos finitos. Los resultados alcanzan una sensibilidad y especificidad de hasta el 91% con datos experimentales reales, y demuestran que la metodología propuesta es una solución factible para la detección automática de roces en maquinaria rotativa.

**Palabras clave:** Detección de roces, turbinas de gas aeroderivadas, DNN, Transformada de Fourier

## Abstract

Rotor rubbing against casing is one of the most common malfunctions in rotating machinery. In the case of aeroderivative turbines, detection of this malfunction is hampered by design constraints, imposing the use of sensors that are affected by noise and vibrations of other components, limiting effectivity and applicability of traditional detection systems based in Fourier transform. Deep learning algorithms have shown surprising results in several problems of automatic monitoring of industrial systems. Nevertheless, they need a large database of malfunctions to describe the phenomenon. Thus, costs and risks associated with availability of these data limits its application. Our proposal consists of implementing deep learning algorithms that use, in training phase, only malfunction data simulated with a finite element model. Results achieve sensitivity and specificity of up to 91% with real experimental data and show that our methodology is a feasible solution for automatic detection of rub in rotating machinery.

**Keywords:** Rub Detection, Aeroderivative Gas Turbines, Deep Neural Networks, Fourier Transform

## 1 Introducción

La eficiencia neta y el rendimiento de las turbomáquinas han aumentado durante las últimas décadas gracias a la minimización de la masa, el tamaño y las tolerancias dimensionales de los componentes, al uso de materiales de vanguardia con propiedades mecánicas y térmicas de primera clase y a la introducción de tecnologías de refrigeración y lubricación de vanguardia [1, 2]. Uno de los parámetros de diseño más importantes de una turbina de gas es el

espacio libre entre el rotor y el estator. Para maximizar la eficiencia y la potencia de salida en los modernos motores aéreos, las holguras mecánicas deben mantenerse lo más pequeñas posible en toda la máquina, por ejemplo, entre las etapas del compresor o de la turbina o en la punta de las ruedas de paletas [3]. Sin embargo, el mayor rendimiento vino inevitablemente acompañado de una contrapartida en cuanto a la fiabilidad, ya que los nuevos diseños y tecnologías hicieron que las turbomáquinas fueran más vulnerables a las averías. Como consecuencia de la

reducción de la holgura, el impacto entre las palas del rotor y las juntas del estator es más probable. Debido a las enormes energías cinéticas almacenadas en estas máquinas durante su funcionamiento, los impactos entre los componentes estáticos y móviles pueden alterar el rendimiento de la máquina, socavar la integridad de la turbina e incluso provocar un fallo catastrófico [4]. Algunas de las consecuencias de este roce pueden ser el desgaste, el ruido, el sobrecalentamiento, la deformación térmica del eje y, ocasionalmente, una inestabilidad vibratoria que lleve a la destrucción de la máquina [5].

Existen varios tipos de rozamiento entre el rotor y la carcasa en las turbomáquinas: el rozamiento total y el rozamiento parcial (también llamado rozamiento fijo o de un solo punto). El roce parcial es el choque del rotor de la máquina con puntos localizados de la carcasa o el estator durante una fracción del ciclo de rotación, mientras que el roce total es un contacto continuo y circunferencial entre el rotor y el estator. Las rozaduras parciales son más comunes y mucho menos dañinas. Pero si un régimen de rozamiento parcial se prolonga en una máquina sin ser detectado durante mucho tiempo, se produce desgaste, sobrecalentamiento, inclinación del rotor y vibración auto excitada, lo que a su vez puede desencadenar un bucle de retroalimentación positiva que hace que el rozamiento sea cada vez más intenso hasta que la máquina se rompe. En turbomáquinas como las de vapor y gas en el régimen de funcionamiento nominal bajo rozamiento parcial, se ha observado que los golpes entre el rotor y el estator se producen periódicamente con una frecuencia de un impacto por ciclo de rotación ( $1\times$ ) cuando el rozamiento es más ligero, y con una frecuencia igual a un impacto por dos ciclos de rotación ( $1/2\times$ ) cuando el rozamiento es más pesado, debido a un cambio en la rigidez rotor mecánica [5]. Una supervisión constante en línea de algunos indicadores del estado de la máquina puede ayudar a garantizar un estado de funcionamiento adecuado. Las mejores fuentes de información sobre el estado de funcionamiento de una turbina de gas son las vibraciones de precesión inducidas en el eje por las cargas dinámicas del rotor. Para disponer de una medición directa de esta magnitud física es posible utilizar pares de sensores de proximidad, colocados en soportes hidrodinámicos del rotor, para tener una representación de las órbitas 2D del eje. Esta solución puede aplicarse en las turbinas de gas estándar, en las que se adoptan cojinetes hidrodinámicos como soportes, lo que permite medir el desplazamiento relativo entre las superficies del rotor y del estator. En cambio, en el caso de las turbinas de gas aeroderivadas, que utilizan otros conceptos de soporte, como los rodamientos de bolas, el uso de sensores de proximidad suele ser poco práctico, ya que los dos elementos (rotor y soportes) tienen una conexión rígida. Esta limitación obliga al operador de la máquina a buscar otras soluciones tecnológicas para la monitorización del

estado. Una de las alternativas es el uso de acelerómetros fijados a la carcasa exterior como transductores de vibración. Lamentablemente, esta solución sólo proporciona una medición indirecta de la vibración del rotor, y las señales de los acelerómetros suelen estar contaminadas por el ruido de fuentes de vibración secundarias y parásitas. Como resultado, la detección de roces es más difícil, especialmente en su fase más temprana, cuando su intensidad es muy baja. Además, los rasgos de la señal que caracterizan el roce en las señales del proximitor no coinciden con los rasgos que caracterizan el roce en las señales del acelerómetro, ya que los dos sensores miden cantidades y fenómenos físicos diferentes.

Los algoritmos de Inteligencia Artificial (IA) han obtenido excelentes resultados en aplicaciones de mantenimiento industrial, dando lugar a toda una línea de investigación. El diagnóstico inteligente de fallos (IFD) se refiere a las aplicaciones de las teorías de la IA al diagnóstico de fallos de las máquinas. Es una forma prometedora de reconocer automáticamente los estados de salud de las máquinas, por lo que ha atraído mucha atención en las últimas décadas. Estas soluciones han sido aplicadas para el mantenimiento preventivo y predictivo automatizado de maquinarias como las máquinas rotativas [6, 7, 8], y en particular, los aerogeneradores [9], las cajas de engranajes [10], los motores de inducción [11, 12] y otras máquinas [13]. Muchos de los trabajos citados en estas publicaciones emplean redes neuronales profundas (DNN) como modelos de clasificación para la monitorización del estado. Este modelo de Machine Learning (ML) necesita, para su funcionamiento, de una gran base de datos también del fallo del sistema, para aprender a describir el comportamiento de las máquinas. Sin embargo, la recopilación de grandes cantidades de datos de malfuncionamiento reales llevaría a dañar o destruir las máquinas, lo que lo haría poco práctico desde el punto de vista económico y de seguridad. Para superar este tipo de limitaciones, en los últimos años se ha desarrollado una línea de investigación muy prometedora denominada Transfer Learning (TL) [14] [15]. Este consiste en el desarrollo de algoritmos de IA donde el dominio de origen de los datos es diferente con respecto al dominio objetivo. Los algoritmos se entrenan, por tanto, con muestras de datos relacionadas con un sistema diferente respecto al que van a operar, donde esos sistemas pueden mostrar varias e importantes diferencias. Este es el ámbito en el que se desarrolla este trabajo.

El objetivo principal de este manuscrito es mostrar la viabilidad de utilizar datos sintéticos generados con un modelo numérico de Elementos Finitos (EF) para diseñar y entrenar redes neuronales para la detección automatizada de roces en turbinas de gas aeroderivadas. La estrategia será después evaluada por medio de un test sobre datos generados por medio de una máquina experimental, que consiste en una máquina rotativa con acelerómetros sobre una carcasa

flexible. La gran ventaja de la metodología propuesta es que descartaría la necesidad de recoger grandes muestras de datos de turbinas de gas aeroderivadas defectuosas para el entrenamiento de la red. De este modo, es posible crear una herramienta para la detección automatizada de roces, que es un fallo común y potencialmente catastrófico en una turbina de gas, sin dañar ningún sistema mecánico.

Estrategias similares han sido aplicadas en diferentes contextos aplicativos con notables resultados. Es el caso de las publicaciones [16], [17], [18], [19] donde han sido desarrollados algoritmos de Structural Health Monitoring (SHM) para la detección de fallos en diferentes tipologías de estructuras. Para cumplir este objetivo se han entrenado redes neuronales profundas (DNN) y convolucionales (CNN) con datos sintéticos obtenidos simulando el comportamiento dinámico con un modelo de elementos finitos que presentaba los defectos que querías clasificar.

La metodología propuesta para la detección automatizada de averías se basa en un modelo de clasificación supervisado. Este modelo DNN debe ser capaz de detectar el estado defectuoso en una máquina real, aunque el modelo se entrena con datos del modelo numérico EF que simula el comportamiento de la máquina en cuestión. Para ello, hemos probado diferentes combinaciones de arquitecturas neuronales y técnicas de extracción de características, analizando cuáles permiten obtener los mejores resultados de clasificación. En definitiva, queremos demostrar que el uso de modelos ML entrenados con datos simulados es una estrategia factible para la detección de un fallo en una máquina rotativa como es el roce de la carcasa del rotor. El flujo de trabajo propuesto también podría aplicarse a la clasificación de otros tipos de averías aerodinámicas.

## 2 Metodología

### 2.1 Resumen

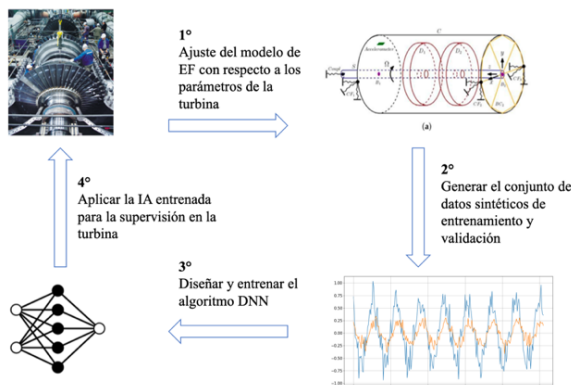


Figura 1 Resumen de la estrategia

La estrategia adoptada en este trabajo consiste en crear una DNN de clasificación binaria, entrenada con datos sintéticos que provienen de un modelo de EF, que

permita detectar si una máquina rotativa está funcionando correctamente. Para ello se dispone de datos de funcionamiento correcto de la máquina rotativa, se ajusta un modelo de EF ajustado al funcionamiento correcto de esta máquina. Este modelo de EF se usa para generar datos sintéticos de funcionamiento correcto y con roce con los que entrenar una DNN de clasificación binaria. Una vez entrenada, la DNN se aplica sobre la máquina rotativa real. Este método permite obtener un software de mantenimiento preventivo sin necesidad de pruebas experimentales de roce. Los pasos que componen la metodología propuesta para obtener un algoritmo de mantenimiento preventivo están ilustrados en la Figura 1.

### 2.2 Ajuste del modelo de elementos finitos y generación de la base de datos

#### 2.2.1 Descripción del modelo de elementos finitos

En este trabajo se ha considerado el modelo de EF de máquina rotativa descrito en [20], ver figura 2. Todos los componentes se modelan bajo las hipótesis de elasticidad lineal (a excepción de los discos sólidos rígidos), homogeneidad y pequeños desplazamientos y deformaciones. Las dimensiones y propiedades mecánicas más destacables del modelo se muestran en los artículos [20] y [21].

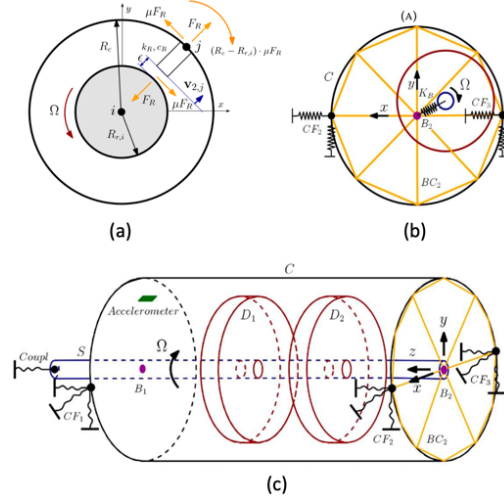
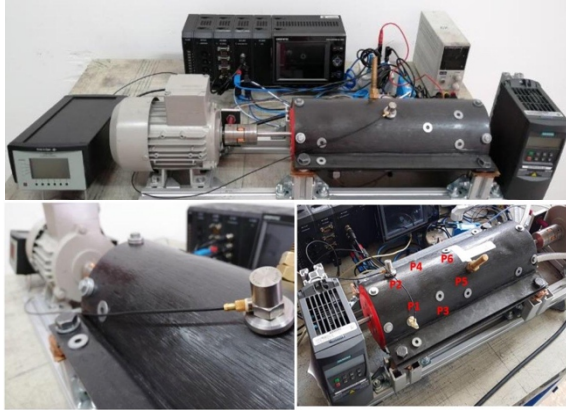


Figura 2 Esquemas del modelo flexible rotor-carcasa con roce en [20]. (a) Modelo de rozamiento y fuerzas en los nodos del eje y la carcasa. (b)(c) Vista frontal y lateral del modelo.

La simulación del modelo requiere el sistema dinámico de ecuaciones diferenciales ordinarias (SODE), considerando el eje como una viga de Euler-Bernoulli y la carcasa como una cáscara curva de Mindlin-Reissner. El sistema tiene un amortiguamiento de tipo Rayleigh. Para la integración numérica del sistema reducido de ecuaciones, se elige el método implícito Newmark- $\beta$ . El sistema permite reproducir un estado de desequilibrio aplicando una fuerza centrífuga

rotante síncrona sobre el eje. Para reproducir el mal funcionamiento del roce se coloca un obstáculo con coeficientes de rigidez y amortiguación  $k_R$  y  $c_R$  en el nodo  $j$  de la carcasa, como se puede ver en la Figura 2 (c). La intensidad del contacto se controla a través de una interferencia impuesta entre los dos elementos. Todas las hipótesis del modelo están explicadas extensamente en el artículo de referencia [20].

### 2.2.2 Ajustar el modelo con respecto al experimento real de referencia



**Figura 3** Máquina rotativa experimental para simular el mal funcionamiento de tipo roce

Para obtener datos reales y ajustar el modelo de EF se ha desarrollado un experimento de máquina rotativa en el banco experimental que se muestra en la figura 3. El experimento ha sido diseñado y construido en el laboratorio de la División de Ingeniería de Maquinaria del Departamento de Ingeniería Mecánica de la Universidad Politécnica de Madrid [21]. Este modelo es capaz de reproducir la dinámica de una máquina rotativa, y puede ser operado bajo desequilibrio controlable y simular avería de tipo roce. En la imagen se pueden observar los lugares donde se colocan los acelerómetros en la parte exterior de la carcasa. Para poder describir el comportamiento de la máquina real, el modelo de EF tiene que ser ajustado con respecto a las especificaciones y comportamiento de la máquina real en correcto funcionamiento. El ajuste se hace por medio de pruebas de respuestas modales de las máquinas. Las pruebas modales han sido realizadas en el rotor, en una de las mitades de la carcasa y en el conjunto completo con un martillo modal, tomando mediciones con acelerómetros en una cuadrícula de puntos. En [21] se tiene la comparación de las frecuencias naturales medidas en el equipo experimental y sus partes con los valores propios del modelo, y los parámetros finales obtenidos del modelo EF con respecto a los del sistema real original. Nótese que, aunque se usan los datos de la máquina para ajustar el modelo, este proceso no necesita de los datos reales de roce. De hecho, en este trabajo únicamente se cuenta con los datos reales de roce para poder evaluar

el desempeño de la red neuronal entrenada con datos simulados, es decir, para validar la metodología.

### 2.3 Generación de la base de datos de entrenamiento

La señal utilizada para describir el comportamiento del sistema se produce midiendo la aceleración en dos puntos P1 y P2 de la carcasa. La simulación de EF consiste en variar el parámetro de interferencia entre el eje y el cilindro de contacto (que simula el tornillo del contacto real), aumentando la intensidad del contacto, mientras se mantiene fija la velocidad de rotación del eje. Esta es fija e igual a 95 [Hz] (o 5700 [rpm]), lo que permite obtener con suficiente intensidad en el montaje experimental tanto el roce 1x como el 1/2x, pero dentro de su rango de velocidad de funcionamiento seguro. El parámetro de interferencia  $\epsilon$  varía linealmente desde 0 hasta  $24 \cdot 10^{-6}$  [m], valores encontrados experimentalmente que permiten obtener las dos tipologías de roce a esta velocidad de rotación [21]. La señal tiene una tasa de muestreo de 10 kHz en ambos casos (simulado y real).

A esta señal se le aplica un remuestreo síncrono con respecto a la velocidad de rotación, de modo que cada rotación de la máquina este descrita por 32 puntos. Esto equivale aproximadamente a 3100 [Hz], considerando la velocidad del rotor igual a 95 [Hz], una frecuencia de muestreo lo suficientemente alta como para recoger la información armónica fundamental que caracteriza el roce. Cada muestra de entrada estará compuesta por un número fijo de ciclos de rotación del eje. En el modelo de EF una simulación completa tiene una duración de 5 [s]. Para balancear las muestras de fallo y de normalidad, y también para mejorar el desempeño de los algoritmos de redes neuronales profundas se ha aplicado la técnica de aumento de datos basada en introducir a los datos un ruido blanco gaussiano de diferentes intensidades. Este mecanismo de aumento de datos puede justificarse e interpretarse como una forma de incluir algorítmicamente conocimiento previo en los datos generando versiones modificadas artificialmente de estos, que capturan los tipos de variaciones que esperamos ver en los datos reales, manteniendo las etiquetas originales [14]. El ruido blanco gaussiano se añade de forma aditiva (3). La intensidad  $\sigma_N^{(SNR)}$  se estima por medio de la formulación (4), fijando los niveles de *Signal To Noise Ratio* (SNR), calculando la potencia de la señal original sin roce, pero bajo desequilibrio  $P(\mathbf{a}_D)$ .

$$\mathbf{a}_{D+N}(t) = \mathbf{a}_D(t) + \mathcal{N}(0, \sigma_N^2) \quad (3)$$

$$SNR = 10 \cdot \log_{10} \left( \frac{P(\mathbf{a}_D)}{\sigma_N^2} \right)$$

$$\sigma_N = \sqrt{\frac{P(\mathbf{a}_D)}{\frac{SNR}{10}}} \quad (4)$$

$$\mathbf{a}_{R+N}(t) = \mathbf{a}_R(t) + \mathcal{N}(0, \sigma_N^2) \quad (3)$$

La potencia de la señal  $P(\mathbf{a})$  se define como la suma de los cuadrados absolutos de los valores de la señal dividida por el número de puntos de la muestra o, alternativamente, el cuadrado de su raíz de la media cuadrática (RMS).

Después, sobre cada muestra se aplica la transformada de Fourier, obteniendo el espectro  $\mathbf{F}_{O+N}(\omega)$ .

$$\mathbf{a}_{D+N}(t) \rightarrow \mathcal{F}(\mathbf{a}_{D+N}(t))(\omega) = \mathbf{F}_{D+N}(\omega) \quad (5)$$

$$\mathbf{a}_{R+N}(t) \rightarrow \mathcal{F}(\mathbf{a}_{R+N}(t))(\omega) = \mathbf{F}_{R+N}(\omega)$$

Como valor de entrada a la red se toma el módulo de los armónicos  $|\mathbf{F}_{D+N}(\omega)|$ .

## 2.4 Diseño y entrenamiento de la red

Considerando los éxitos obtenidos en la última década de los algoritmos de redes profundas, para el proyecto hemos elegido de utilizar un algoritmo DNN como modelo de clasificación binaria de tipo supervisado [14][17][26]. La principal ventaja de estos algoritmos con respecto a otros de ML importantes es la extracción automática de características jerárquicas que compensa parcialmente la necesidad de diseñar métodos manuales de extracción de características [26]. El algoritmo de aprendizaje profundo más extendido son las Redes Neuronales Profundas (DNNs). Estos modelos se pueden resumir en la aplicación sucesiva (en capas) de varias combinaciones lineales con diferentes funciones de activación no lineales y diferenciables de tal forma que las salidas de una capa son las entradas de la siguiente. Cada una de las combinaciones lineales seguidas de una función de activación es una unidad de procesamiento o neurona, de tal forma que aquellas que tienen las mismas entradas y por tanto pueden procesar al mismo tiempo están en la misma capa. Estas funciones de activación no lineales permiten representar problemas no lineales, mientras que la diferenciabilidad permite la aplicación de los métodos de cálculo y estimación eficiente del gradiente de la función de coste basados en el algoritmo de optimización *back propagation*. Una red se define profunda cuando hay al menos una capa oculta entre la entrada y la salida de la red [14]. La ecuación (6) es una formulación general de una DNN de  $L$  capas.

$$\begin{aligned} \tilde{y}_n &= h_w(\mathbf{x}_n) \\ &= \phi^{(L)}\left(\mathbf{w}^{(L)}\phi^{(L-1)}\dots\phi^{(2)}\left(\mathbf{w}^{(2)}\phi^{(1)}(\mathbf{w}^{(1)}\mathbf{x}_n)\right)\right) \quad (6) \end{aligned}$$

Cada capa  $l \in \{1, \dots, L\}$  tiene su matriz de pesos  $\mathbf{w}^{(l)} \in \mathbb{R}^{k_l \times k_{l-1}}$  y su función de activación  $\phi^{(l)}: \mathbb{R}^{k_l} \rightarrow \mathbb{R}^{k_l}$  dada por  $\phi^{(l)}(\mathbf{z}) = (\phi_1^{(l)}(z_1), \dots, \phi_{k_l}^{(l)}(z_{k_l}))$ , donde  $k_l$  es el número de entradas de la capa  $l$  (salidas de la capa anterior) y de la red neuronal. Con esta notación, la

operación que ocurre en la neurona  $i$ -ésima de la capa  $l$  viene dada por  $\phi_i^{(l)}(\mathbf{w}_i^{(l)}\mathbf{u})$ , donde  $\mathbf{w}_i^{(l)}$  es la fila  $i$ -ésima de la matriz  $\mathbf{w}^{(l)}$  y  $\mathbf{u} = \phi^{(l-1)}\left(\mathbf{w}^{(l-1)}\phi^{(l-2)}\left(\dots\left(\mathbf{w}^{(1)}\mathbf{x}_n\right)\right)\right)$  es la salida de la capa  $l$  y entrada de cada una de las neuronas de la capa  $l$ .

En el modelo aquí considerado las funciones de activación de las capas ocultas son *Rectified Linear Unit* o ReLU (7), que son funciones de activación con derivadas constantes, evitando así el problema del desvanecimiento del gradiente. Dado que se tiene un problema de clasificación binaria, la capa de salida está compuesta por una única neurona y tiene como función de activación la sigmoide (8), que permite interpretar la salida como una probabilidad de pertenencia a una de las clases, ya que  $\tilde{y}_n = \text{Sig}(x) \in [0,1] \subset \mathbb{R}$ .

$$\text{ReLU}(x) = \max(x, 0) \quad (7)$$

$$\text{Sig}(x) = \frac{1}{1 + e^{-x}} \quad (8)$$

Con el objetivo de reducir el riesgo de sobreajuste a los datos de entrenamiento, se han implementado varias técnicas de regularización en la estructura de la red y en la selección del modelo. Estas son el *early stopping*, *dropout*, regularización  $l_2$  o  $l_1$  [14]. En nuestro problema es fundamental minimizar el riesgo de sobreajuste de la red, ya que los datos de prueba difieren de los de entrenamiento debido a las aproximaciones implementadas en el modelo de EF. El *early stopping* consiste en definir como criterio de parada de la fase de entrenamiento el valor de pérdida obtenido por el grupo de validación, evitando así una discrepancia entre los resultados del conjunto de datos de entrenamiento y el de validación.

El *dropout*, en cambio, consiste en desconectar aleatoriamente un porcentaje determinado de neuronas y sus conexiones asociadas en las capas seleccionadas. Teóricamente consiste en definir una versión modificada del peso  $w_{ij}^{(l)}$  que conecte la neurona  $i$  de la capa  $l-1$  con la neurona  $j$  de la siguiente capa  $l$ , donde  $\epsilon_i^{(l)}$  es un término de ruido de Bernulli que cambia en cada proceso de actualización de pesos.

$$\begin{aligned} \tilde{w}_{ij}^{(l)} &= w_{ij}^{(l)} \epsilon_i^{(l-1)} \epsilon_j^{(l)} \\ \epsilon^{(l)} &\sim \text{Ber}(\{0,1\} | 1 - p^{(l)}) \quad (9) \end{aligned}$$

Si  $\epsilon_i^{(l)} = 0$ , en la etapa de entrenamiento actual el modelo no utiliza la neurona  $i$ -ésima de la capa  $l$  esa neurona. El *dropout* permite prevenir el sobreajuste evitando complejas adaptaciones entre neuronas ocultas.

Aplicando regularización  $l_2$  se impone una penalización proporcional al cuadrado de la norma 2 del vector de pesos en la función de coste

$$\mathcal{L}(\mathbf{w}) = - \sum_{n=1}^N \log p(\mathbf{y}_n | \mathbf{x}_n; \mathbf{w}) + \lambda \|\mathbf{w}\|_2^2$$

Esto penaliza pesos grandes, simplificando el modelo y reduciendo el riesgo de sobreajuste de la red. Esta técnica de regularización se puede interpretar como una “Maximum Posterior” (MAP) imponiendo un sesgo gaussiano a priori en los valores de las ponderaciones.

Utilizar la norma 1 en lugar de la norma 2 daría lugar a diferencias significativas en el resultado final. La regularización  $l_1$  produce un modelo disperso, con muchas conexiones apagadas ( $w_{ij}^{(l)} = 0$ ), simplificando aún más la red.

En la fase experimental hemos utilizado las tres técnicas enunciadas, optimizado los hiperparámetros que aparecen en estas (porcentaje de desactivación en *dropout* y constantes de proporcionalidad en la función de coste en la regularización  $l_2$  o  $l_1$ ) y comparado los resultados.

Al estar ante un problema de clasificación binaria supervisada e interpretar la salida de la red neuronal como una probabilidad, aplicando máxima verosimilitud [25] se obtiene como función de pérdida natural la *Binary Cross Entropy*  $\mathbb{H}(y_n, \tilde{y}_n)$ , es decir, la función de pérdida es

$$\mathcal{L}(\mathbf{w}) = \mathcal{L}(\mathbf{w}, \mathbf{X}) = \frac{1}{N} \sum_{n=1}^N \mathbb{H}(y_n, \tilde{y}_n) + R(\mathbf{w}) \quad (10)$$

$$\mathbb{H}(y_n, \tilde{y}_n) = y_n \log \tilde{y}_n + (1 - y_n) \log(1 - \tilde{y}_n),$$

donde  $N$  es el número de datos en la muestra  $\mathbf{X}$  y  $R(\mathbf{w})$  es el término de regularización  $l_2$  o  $l_1$  si lo hubiera.

Como optimizador o proceso de entrenamiento se ha seleccionado el descenso del gradiente estocástico, en el que la actualización de los pesos en el paso  $t$  viene dado por

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathcal{L}(\mathbf{w}_t, \mathbf{X}_t) \quad (11)$$

donde  $\mathbf{X}_t$  es el lote de datos seleccionado en el paso  $t$  para realizar la actualización y  $\eta_t$  es la tasa de aprendizaje del entrenamiento. Para la elección de este parámetro se ha seguido un enfoque heurístico, empezando por valores bajos ( $\eta_0 \sim 10^{-5}$ ) y aumentándolo gradualmente mediante la observación de los valores de pérdida.

Para el diseño de la red se ha seguido un procedimiento de incremento de la complejidad de la red aumentando progresivamente el número de capas ocultas (varios estudios, experimentales y teóricos han demostrado que las redes profundas funcionan mejor que las redes con una sola capa y muchas neuronas [14]) y ajustando los hiperparámetros y analizando los resultados por medio de la técnica de *k-fold Cross Validation* (CV), con  $k = 10$ . Esta técnica consiste en dividir el conjunto de datos  $\mathcal{D}$  en  $k$  subconjuntos aleatorios de iguales

dimensiones, utilizar  $k - 1$  subconjuntos para el entrenamiento y utilizando el último dejado fuera para la validación. Se repite el entrenamiento  $k$  veces, de forma que cada subconjunto se utiliza en la validación una única vez. Para comparar resultados se usa la media de las métricas en cada uno de los  $k$  experimentos. Una vez elegido el modelo mejor mirando los resultados en la fase de CV, el modelo final se entrena usando toda la base de datos  $\mathcal{D}$  generada con el modelo de EF. Combinando la estrategia de *early stopping* con la validación cruzada, el número de épocas se elige igual al valor máximo alcanzado en los varios entrenamientos con todos los 10 subconjuntos.

## 2.5 Métricas de evaluación

Teniendo en cuenta que la base de datos de test está desequilibrada, la lectura de los resultados podría ser engañosa si no se tienen en cuenta las métricas adecuadas. Esto no afecta a la fase de entrenamiento del modelo, ya que fue posible equilibrar la base de datos de EF con el aumento de datos. Denotamos una muestra de roce como una instancia positiva P, y por tanto la muestra sin roce como una instancia negativa N. Las métricas aplicadas para analizar los resultados y evaluar las prestaciones del algoritmo son la precisión, sensibilidad, especificidad, G-Mean y curva ROC.

$$\text{Precisión} = \frac{TP + TN}{\text{Número total de muestras}} \quad (12)$$

$$\text{Sensibilidad} = \frac{TP}{TP + FN} = TPR \quad (13)$$

$$\text{Especificidad} = \frac{TN}{TN + FP} = 1 - FPR \quad (14)$$

$$G_{\text{mean}} = \sqrt{\text{Sensibilidad} \cdot \text{Especificidad}} \quad (15)$$

Para evaluar el rendimiento de la clasificación en el conjunto de datos real, la métrica de evaluación debe devolver resultados no sesgados también dentro de una base de datos desequilibrada. De hecho, el número de muestras de roce (positivas) en la base de datos experimental es cinco veces mayor que el número de muestras sin roce en la misma base de datos. De ahí que introduzcamos la G Mean, que es la media geométrica entre la sensibilidad y la especificidad. Estos últimos parámetros permiten medir el porcentaje de muestras de la base de datos que han sido clasificadas correctamente. La sensibilidad se refiere al porcentaje de muestras positivas P (roce) detectadas sobre el total de datos de roce. La especificidad es el mismo con respecto a las muestras negativas N (sin roce). Tanto la especificidad como la sensibilidad no se ven influidos por las diferentes proporciones de dos tipos de muestras. Para evaluar la capacidad de

discriminación de las redes, se ha calculado la curva ROC. La Característica Operativa del Receptor (ROC) es una curva que traza la relación entre la Tasa de Verdaderos Positivos (TPR) y la Tasa de Falsos Negativos (FPR) para un conjunto de predicciones del modelo bajo diferentes umbrales. Esta métrica calcula el rendimiento de la clasificación binaria variando el nivel de umbral de 0 a 1. El área bajo la curva (AUC) se calcula para resumir el rendimiento de la predicción del modelo.

### 3 Resultados y Conclusiones

#### 3.1 Generación de datos sintéticos y reales

Como primera parte de la metodología propuesta se ha producido la base de datos sintéticas por medio del modelo EF. Se utilizará solo esta base de datos para entrenar las redes neuronales y elegir los mejores modelos. Para probar la validez de la metodología propuesta, posteriormente el modelo final entrenado se ha evaluado en la base de datos experimentales obtenida con el modelo rotativo real, ilustrado en el apartado 2.2.2.

Para la generación de la base de datos simulada con el modelo de EF, se han elegido 5 coeficientes de SNR entre  $-7.5$  y  $0$ . Para cada nivel de ruido se han generado 5 ensayos con la técnica del aumento de datos, para un total de 25 ensayos con roce con el parámetro de penetración del tornillo  $\epsilon$  que varía entre  $0$  y  $24 \cdot 10^{-6}$  [m]. Aplicando el aumento de datos también al ensayo afectado solo con desequilibrio, ha sido posible balancear la base de datos de entrenamiento.

La base de datos reales consta de 4 ensayos experimentales utilizando el banco de prueba de la Figura 3. Cada ensayo consta de una primera parte donde, pasada la fase de arranque y llegada a la velocidad de rotación de  $95$  [Hz], se graba el comportamiento vibratorio de la máquina solo con desequilibrio. Después, apretando el tornillo montado en la carcasa, se induce el malfuncionamiento de tipo roce. En cada ensayo se han producido dos niveles de intensidad de contacto. Esta variación ha sido inducida manualmente, sin un ajuste de la intensidad del contacto. Para poder comparar las prestaciones de clasificación de las DNNs tanto sobre los datos del modelo EF como sobre los datos del modelo experimental real se necesita disponer de un cuantificador no sesgado de la intensidad del rozamiento correspondiente a cada muestra de datos. En este trabajo se considera como métrica de la intensidad del roce los desplazamientos relativos entre el rotor y el estator medidos mediante sensores de proximidad en el caso real (aunque estos sensores no pueden montarse en las turbinas de gas aeroderivadas, no sería necesario ya que en este trabajo solo se usan para validar la metodología). Esta es una métrica que se correlaciona con la energía liberada durante cada impacto rotor-estator. Comparando los rangos de

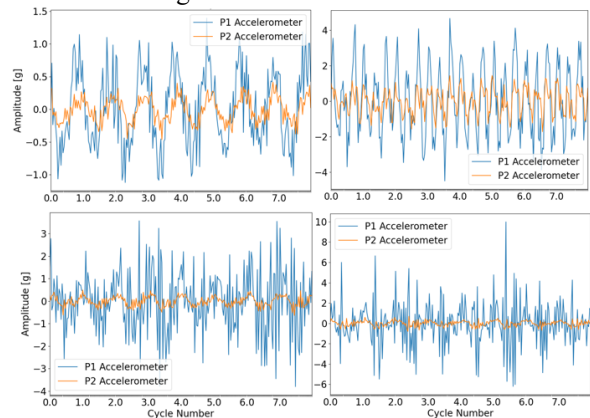
valores, el modelo experimental como el simulado de EF presentan un roce de intensidad comparable, entre  $0$  y  $85$  [ $\mu\text{m}$ ], ya que las muestras de ambas bases de datos se encuentran entre estos dos valores de penetración del tornillo. Por lo tanto, se puede observar que la magnitud de los fenómenos de malfuncionamiento, tanto en el modelo real como en el modelo sintético, son similares. En la figura 3 se representan muestras del modelo experimental y real.

**Tabla 1. Composición de las bases de datos generada por el modelo EF y Experimental**

Modelo	EF	Experimental
Total	21402	4740
No Roce	10701	720
Roce	10701	4020

#### 3.2 Estructura de los datos

Tanto en el modelo de EF como en el experimental, la señal es producida por dos sensores situados en dos posiciones externas diferentes de la carcasa. Cada acelerómetro produce una señal diferente como se puede ver en la Figura 4, en la que se representan diferentes muestras, del modelo de EF y del experimento real, con y sin roce. Se puede observar como cuando se produce una avería, ésta actúa modificando la rigidez del sistema y alterando el modo de vibración de la máquina. Debido a que los sensores tienen comportamientos distintos se decide incluir la información de ambos en la entrada con la que se alimentará al algoritmo.



**Figura 4** Gráfica de la señal proporcionada por los dos sensores P1 (azul) y P2 (amarillo). En la columna izquierda muestras sin roce, a la derecha con roce. En la fila inferior muestras experimentales reales, en la superior muestras del modelo de EF.

Como se ha descrito en 2.3 los datos de cada uno de estos sensores provienen de un remuestreo sincrónico, tras el cual la frecuencia angular de los datos de un acelerómetro es de 32 muestras por revolución del rotor. La longitud de cada muestra es un parámetro importante en la fase del diseño de la red, medida como el número de revoluciones del rotor que abarca cada

muestra. Teniendo en cuenta que los roces puntuales muy fuertes suelen manifestarse con una periodicidad de  $1/2 \times$  en condiciones estacionarias a la velocidad nominal de trabajo de las turbinas de gas reales en tierra, se necesita una longitud mínima de dos ciclos de rotación del eje para describir la respuesta del sistema a este fallo. Por esta razón, para estudiar la influencia del número de vueltas considerado en el rendimiento de la clasificación, se han probado varias opciones con muestras compuestas por 2 rotaciones o múltiplos (2, 4, 6, 8 rotaciones por instancia). Las distintas pruebas mostraron resultados similares con la validación cruzada en comparación con los resultados experimentales. Por lo tanto, se eligió la opción más conservadora entre las probadas, con un tamaño de muestra de  $R=8$  rotaciones. A estas muestras se les aplica la transformada de Fourier pasando al dominio de la frecuencia. De esta señal sólo se utiliza el módulo de los armónicos. La consideración de la fase en la muestra ha dejado los resultados obtenidos durante los experimentos inalterados.

En la tabla 1 se puede observar la composición de las bases de datos. En particular, la base de datos experimentales no está balanceada. Esta condición no es un problema en la fase de entrenamiento, ya que utilizamos solamente datos de EF, pero es importante utilizar métricas no sensibles a esta condición a la hora de analizar los resultados.

### 3.3 Estructura de la Red

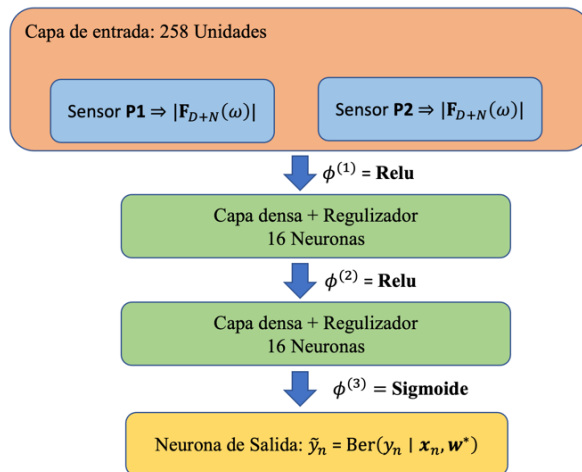


Figura 5 Estructura de la red neuronal

El input de la red es el resultado de la combinación de las informaciones proporcionadas por los sensores P1 y P2, en concreto cada dato de entrada consiste en 258 módulos de los armónicos de la señal correspondiente a 8 rotaciones (32 puntos por rotación), 129 armónicos para cada uno de los sensores. Como es habitual para acelerar y facilitar la convergencia del algoritmo de optimización [14][24][25] cada una de las características (módulos de los armónicos) introducidas a la red neuronal se han estandarizado de forma independiente. Esta operación se ha ejecutado en

la base de datos sintética de entrenamiento y en la real de prueba por separado. La estructura de la red es el resultado de varios experimentos ejecutados siguiendo lo que se ha enunciado en el capítulo 2.3, en todos estos experimentos se ha considerado la inicialización habitual de Glorot [14][24][25]. En la figura 5 es posible observar la estructura final del modelo utilizado. Se han aplicado por separado las diferentes estrategias de regularización.

### 3.4 Resultados

Como es posible observar en la tabla 2, los resultados obtenidos con las distintas técnicas son comparables y ninguna opción destaca sobre las otras. Todos los modelos muestran una gran capacidad para clasificar correctamente tanto las muestras con roce como las que no tienen roce, con valores de precisión hasta el 97% en el entrenamiento final. La tabla 3 muestra los resultados de las pruebas en la base de datos experimental, para observar cuál generaliza mejor la información de las señales del acelerómetro. Dado que la base de datos experimental está muy desequilibrada, es mejor comparar la sensibilidad, la especificidad y la métrica G-mean con respecto a la precisión. Observando los resultados en los conjuntos de prueba y de entrenamiento, podemos observar que, mientras que en los datos de EF todos los modelos han logrado un rendimiento similar, en este caso se observan algunas diferencias más marcadas.

Tabla 2. Resultados de la validación cruzada sobre los datos de entrenamiento sintéticos proporcionados con el modelo de EF

Regularizador	$l2 (\lambda = 0.01)$	$l1 (\lambda = 0.01)$	Dropout 50%
Precision [%] CV	96	96	94
Perdida CV	0.23	0.24	0.18
Precision [%] Final	97	97	96
Perdida Final	0.19	0.21	0.15

Tabla 3. Resultados sobre los datos de prueba proporcionado por medio del modelo experimental

Regularizador	$l2 (\lambda = 0.01)$	$l1 (\lambda = 0.01)$	Dropout 50%
Sensibilidad [%]	99	99	84
Especificidad [%]	79	83	74
G Mean	89	91	79
AUC ROC	98	99	85

Sin embargo, es importante señalar que todos los modelos mantuvieron un rendimiento que, aunque ligeramente peor, muestra que el modelo es capaz de detectar las muestras de mal funcionamiento también en el caso real. Todos los modelos entrenados funcionan correctamente con los datos generados por una máquina real, lo que demuestra la viabilidad de



nuestra metodología. Las redes neuronales profundas han sido capaces de generalizar la información de los datos sintéticos de entrenamiento construidos a partir de un modelo para predecir correctamente el roce en datos experimentales reales de una máquina rotativa. Las redes que utilizaban regularizaciones  $l1$  o  $l2$  han tenido ligeramente mejores resultados que las que aplicaban *dropout*. Esto se puede observar también en la figura 6, donde se representan las curvas ROC de las redes entrenadas sobre el conjunto de pruebas experimentales reales, con el fin de comparar los resultados de clasificación de las diferentes soluciones del modelo. En este gráfico se analiza también el caso de un modelo sin regularización, el cual ha obtenido mejores resultados en los datos reales respecto a los modelos con *dropout*, pero no ha alcanzado los resultados obtenidos con los regularizadores  $l1$  y  $l2$ . En la figura 7 se analiza el valor de la predicción con respecto a la intensidad del roce medida con la deformación del eje de la máquina, podemos ver que la predicción depende de la magnitud del contacto. Esta tendencia se observa en todos los modelos, con datos reales y sintéticos. En el caso experimental hay más incertidumbres en los valores bajos de deformación.

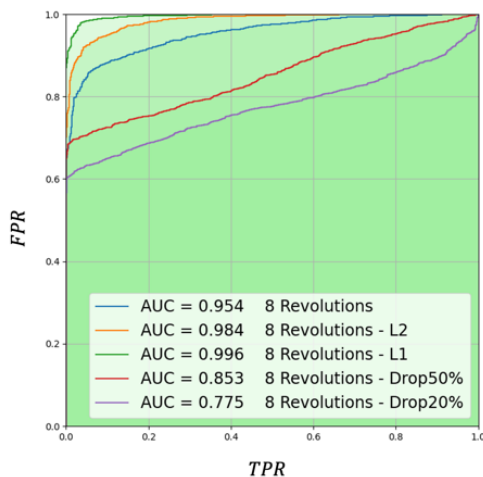


Figura 6 Curvas ROC de los diferentes modelos

### 3.5 Conclusiones

Los resultados de este trabajo demuestran que el entrenamiento de un modelo DNN para la detección automatizada de roces en máquinas rotativas utilizando datos simulados (en particular, en turbinas de gas aeroderivadas monitorizadas con acelerómetros) conduce a un alto rendimiento y precisión en la detección de roces en las máquinas reales.

El entrenamiento de las redes con datos generados por un modelo numérico ajustado con respecto al sistema real permite de ahorrar tiempo y recursos, ya que no necesita de la recolección de datos reales de máquinas defectuosas lo cual sería técnica y económicamente inviable.

Usar los datos del dominio de la frecuencia para entrenar a la red neuronal permite que esta generalice los patrones presentes en los datos del modelo numérico sintético de EF, lo que conduce a un rendimiento muy bueno dentro de los datos reales y experimentales. Se han obtenido modelos que alcanzan una precisión de hasta el 97% en los datos simulados, y de G-Mean hasta el 91% en datos reales. Aunque el estudio se ha centrado únicamente en la detección de un tipo de avería en máquinas rotativas, debería ser posible generalizar la metodología para la clasificación múltiple de un conjunto de averías en máquinas, siempre que afecten a la dinámica del sistema mecánico, dando lugar a una alteración de los modos de vibración de la máquina.

En cuanto a las líneas futuras, se podrían utilizar técnicas más avanzadas de análisis de tiempo-frecuencia, como la transformada de Fourier con ventana y la transformada wavelet. Además, aunque las redes neuronales tienen la desventaja de ser de difícil interpretación, actualmente, para compensar este problema de interpretabilidad, se están desarrollando nuevas técnicas, llamadas métodos de atribución, los cuales podrían implementarse.

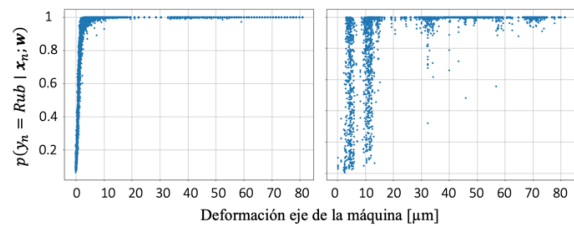


Figura 7 Predicción de roce frente a la intensidad del contacto en los casos con roce. En la izquierda datos sintéticos de entrenamiento, en la derecha datos reales de test.

### 4 Referencias

- [1] E. Benini, *Advances in Gas Turbine Technology*. BoD – Books on Demand, 2011.
- [2] E. Benini, *Progress in Gas Turbine Performance*. 2013. doi: 10.5772/2797.
- [3] R. E. Chupp, R. C. Hendricks, S. B. Lattime, and B. M. Steinetz, “Sealing in Turbomachinery,” *Journal of Propulsion and Power*, vol. 22, no. 2, pp. 313–349, Mar. 2006, doi: [10.2514/1.17778](https://doi.org/10.2514/1.17778).
- [4] H. Ma, F. Yin, Y. Guo, X. Tai, and B. Wen, “A review on dynamic characteristics of blade–casing rubbing,” *Nonlinear Dyn*, vol. 84, no. 2, pp. 437–472, Apr. 2016, doi: 10.1007/s11071-015-2535-x.
- [5] D. E. Bently, Hatch&apos, and C. T, “Fundamentals of Rotating Machinery Diagnostics,” *Mechanical Engineering-CIME*, vol. 125, no. 12, pp. 53–54, Dec. 2003.

- [6] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mechanical Systems and Signal Processing*, vol. 108, pp. 33–47, Aug. 2018, doi: [10.1016/j.ymssp.2018.02.016](https://doi.org/10.1016/j.ymssp.2018.02.016).
- [7] K. Adamsab, "Machine learning algorithms for rotating machinery bearing fault diagnostics," *Materials Today: Proceedings*, vol. 44, pp. 4931–4933, Jan. 2021, doi: [10.1016/j.matpr.2020.12.050](https://doi.org/10.1016/j.matpr.2020.12.050).
- [8] D. Kolar, D. Lisjak, M. Pająk, and D. Pavković, "Fault Diagnosis of Rotary Machines Using Deep Convolutional Neural Network with Wide Three Axis Vibration Signal Input," *Sensors*, vol. 20, no. 14, Art. no. 14, Jan. 2020, doi: [10.3390/s20144017](https://doi.org/10.3390/s20144017).
- [9] G. Helbing and M. Ritter, "Deep Learning for fault detection in wind turbines," *Renewable and Sustainable Energy Reviews*, vol. 98, pp. 189–198, Dec. 2018, doi: [10.1016/j.rser.2018.09.012](https://doi.org/10.1016/j.rser.2018.09.012).
- [10] S. Basangar and B. N. Tripathi, "Literature Review on Fault Detection of Equipment using Machine Learning Techniques," in *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, Jan. 2020, pp. 62–67. doi: [10.1109/ICCAKM46823.2020.9051543](https://doi.org/10.1109/ICCAKM46823.2020.9051543).
- [11] P. Gangsar and R. Tiwari, "Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review," *Mechanical Systems and Signal Processing*, vol. 144, p. 106908, Oct. 2020, doi: [10.1016/j.ymssp.2020.106908](https://doi.org/10.1016/j.ymssp.2020.106908).
- [12] P. Kumar and A. S. Hati, "Review on Machine Learning Algorithm Based Fault Detection in Induction Motors," *Arch Computat Methods Eng*, vol. 28, no. 3, pp. 1929–1940, May 2021, doi: [10.1007/s11831-020-09446-w](https://doi.org/10.1007/s11831-020-09446-w).
- [13] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, Jan. 2019, doi: [10.1016/j.ymssp.2018.05.050](https://doi.org/10.1016/j.ymssp.2018.05.050).
- [14] K. P. Murphy, *Probabilistic machine learning: an introduction*. Cambridge, Massachusetts: The MIT Press, 2022.
- [15] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, p. 106587, Apr. 2020, doi: [10.1016/j.ymssp.2019.106587](https://doi.org/10.1016/j.ymssp.2019.106587).
- [16] P. Seventekidis, D. Giagopoulos, A. Arailopoulos, and O. Markogiannaki, "Structural Health Monitoring using deep learning with optimal finite element model generated data," *Mechanical Systems and Signal Processing*, vol. 145, p. 106972, Nov. 2020, doi: [10.1016/j.ymssp.2020.106972](https://doi.org/10.1016/j.ymssp.2020.106972).
- [17] P. Seventekidis and D. Giagopoulos, "A combined finite element and hierarchical Deep learning approach for structural health monitoring: Test on a pin-joint composite truss structure," *Mechanical Systems and Signal Processing*, vol. 157, p. 107735, Aug. 2021, doi: [10.1016/j.ymssp.2021.107735](https://doi.org/10.1016/j.ymssp.2021.107735).
- [18] Z. Mousavi, M. M. Etefagh, M. H. Sadeghi, and S. N. Razavi, "Developing deep neural network for damage detection of beam-like structures using dynamic response based on FE model and real healthy state," *Applied Acoustics*, vol. 168, p. 107402, Nov. 2020, doi: [10.1016/j.apacoust.2020.107402](https://doi.org/10.1016/j.apacoust.2020.107402).
- [19] Z. Mousavi, S. Varahram, M. M. Etefagh, M. H. Sadeghi, and S. N. Razavi, "Deep neural networks–based damage detection using vibration signals of finite element model and real intact state: An evaluation via a lab-scale offshore jacket structure," *Structural Health Monitoring*, vol. 20, no. 1, pp. 379–405, Jan. 2021, doi: [10.1177/1475921720932614](https://doi.org/10.1177/1475921720932614).
- [20] A. Silva, A. Zarzo, J. Munoz-Guijosa, and F. Miniello, "Evaluation of the Continuous Wavelet Transform for Detection of Single-Point Rub in Aeroderivative Gas Turbines with Accelerometers," *Sensors*, vol. 18, no. 6, p. 1931, Jun. 2018, doi: [10.3390/s18061931](https://doi.org/10.3390/s18061931).
- [21] A. Silva, A. Zarzo, J. M. Machuca González, and J. M. Munoz-Guijosa, "Early fault detection of single-point rub in gas turbines with accelerometers on the casing based on continuous wavelet transform," *Journal of Sound and Vibration*, vol. 487, p. 115628, Nov. 2020, doi: [10.1016/j.jsv.2020.115628](https://doi.org/10.1016/j.jsv.2020.115628).
- [22] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine Learning for industrial applications: A comprehensive literature review," *Expert Systems with Applications*, vol. 175, p. 114820, Aug. 2021, doi: [10.1016/j.eswa.2021.114820](https://doi.org/10.1016/j.eswa.2021.114820).
- [23] M. Z. Alom et al., "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," *arXiv*, arXiv:1803.01164, Sep. 2018. doi: [10.48550/arXiv.1803.01164](https://doi.org/10.48550/arXiv.1803.01164).
- [24] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc, 2019.
- [25] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.