

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**Diagnosticador de severidad de fallos de rodamientos
bajo velocidades variables utilizando la transformada de
Fourier y Deep learning**

Sebastián Ignacio López Norambuena

Profesor guía: **Nibaldo Rodríguez Agurto**

Profesor co-referente: **Guillermo Cabrera Guerrero**

Noviembre, 2017

Resumen

El rodamiento es uno de los componentes más utilizados en la maquinaria rotativa. Las fallas en estos elementos son muy comunes y tienen un gran impacto en la cadena de producción de una planta. Por lo tanto, es importante estudiar la tecnología de diagnóstico de fallas en rodamientos e incluirla en los planes de mantenimiento. Una señal de vibración perteneciente a un rodamiento, transporta información dinámica sobre el estado de salud de este; así que puede ser utilizada para identificar y clasificar fallas en la estructura de los rodamientos. Para automatizar este proceso, la ingeniería ha propuesto algoritmos de *machine learning* que combinados con técnicas de extracción de características; entregan un diagnóstico óptimo sobre el estado de los rodamientos. Para mejorar el rendimiento de estos modelos cuando la dimensión de los datos de entrada es muy alta, se incluyen técnicas de reducción de dimensionalidad o selección de características. En este documento se propone utilizar esta combinación de técnicas de extracción de características y algoritmos de *deep learning* que reducen la dimensionalidad y realizan un diagnóstico de la salud del rodamiento. En la etapa de extracción, se aplicará la transformada rápida de Fourier a una señal de vibración; para obtener la magnitud de los coeficientes y emplearlos como características. Posteriormente una red neuronal profunda compuesta por dos sparse autoencoders y una capa de neuronas con la función de activación softmax, se encargarán diagnosticar la salud del rodamiento.

Palabras Clave: rodamientos, diagnóstico, mantenimiento, señal de vibración, deep learning.

Índice

1	Introducción	1
2	Definición de objetivos	3
2.1	Objetivo general	3
2.2	Objetivos específicos	3
3	Plan de trabajo	3
4	Marco Teórico	3
4.1	Transformada de Fourier	4
4.1.1	Transformada de Fourier continua	5
4.1.2	Transformada de Fourier discreta	5
4.1.3	Transformada de Fourier de tiempo corto	6
4.2	Señal de vibración	6
4.3	Deep learning	7
4.3.1	Autoencoders	8
4.3.2	Sparse autoencoders	10
4.4	Método de diagnóstico basado en DNN	11
5	Experimento y resultados	13
5.1	Origen de los datos	13
5.2	Energía de los coeficientes de Fourier	13
5.3	Proceso de entrenamiento y testing	14
6	Conclusión	20

Lista de Figuras

1	4 tipos de señal de vibración de un rodamiento, con una carga de 3 hp y 0,54 mm de diámetro de falla.	7
2	Diagrama que ilustra la estructura de un autoencoder formado por la red encoder (lado izquierdo) y la red decoder (lado derecho) con sus respectivos parámetros de entrenamiento.	8
3	Diagrama que representa el entrenamiento de los <i>sparse</i> autoencoder y la asignación de sus parámetros a la DNN	12
4	Ilustración del experimento realizado por la Case Western Reserve University [29]	13
5	Resultados del diagnóstico en 30 <i>trials</i> del <i>dataset</i> E.	17
6	Exactitud promedio del diagnóstico en 30 <i>trials</i> de los 5 <i>datasets</i>	18
7	Exactitud por clase en 30 <i>trials</i> de los 5 <i>datasets</i>	18
8	F-score en 30 <i>trials</i> de los 5 <i>datasets</i>	19
9	F-score por clase en 30 <i>trials</i> de los 5 <i>datasets</i>	19
10	Diagrama de flujo de los procesos que se llevan a cabo para entrenar y probar el modelo propuesto.	22

Lista de Tablas

1	Descripción de los 5 <i>datasets</i>	14
2	10 conjuntos de hiperparámetros probados con 30 ejecuciones en el dataset E, el formato de la exactitud es: Exactitud media \pm desviación estándar.	15
3	Resultados del diagnóstico de cada <i>dataset</i>	16

1 Introducción

La maquinaria rotativa es ampliamente utilizada en muchos campos industriales y en los últimos años, el monitoreo de condiciones de estas máquinas ha cobrado especial importancia en el mantenimiento moderno. Lo anterior se ha debido, principalmente a la necesidad de reducir los tiempos de inactividad no programados y los costos asociados al mantenimiento, de manera que se pueda mantener la competitividad corporativa [1]. Así, con el monitoreo de condiciones se pueden detectar tempranamente fallas potencialmente desastrosas, reduciendo el tiempo de inactividad total de la máquina y de operaciones enteras.

Las técnicas de mantenimiento predictivo han demostrado ser estrategias eficaces para reducir fallas inesperadas en la maquinaria. El monitoreo de vibraciones es la tecnología de mantenimiento predictivo más utilizada [2], debido a la cantidad significativa de información que porta la señal sobre el estado salud de la maquinaria. Considerando que los rodamientos son un componente encontrado frecuentemente en la maquinaria rotativa y que pueden causar alrededor del 40-50% [3] de todos los desperfectos, muchas investigaciones se centran en el monitoreo de vibraciones para estos componentes.

Los rodamientos giran constantemente en condiciones ambientales severas tales como alta temperatura, velocidad de rotación variable y grandes cargas, por lo que presentan una alta frecuencia de rotura [4]. Realizar un diagnóstico efectivo para los rodamientos es complejo, porque usualmente la señal de vibración tiene un comportamiento no lineal y no estacionario [5]. Este comportamiento de la señal es consecuencia de la compleja estructura mecánica del rodamiento y sus duras condiciones de trabajo.

Los sistemas de monitoreo de condiciones son usados para obtener datos en tiempo real de las máquinas. Por lo tanto, enormes colecciones de datos son adquiridas después de un prolongado funcionamiento de estas máquinas. Como los datos generalmente se recopilan más rápido de lo que se tarda en analizarlos, existe la necesidad de un método de diagnóstico que sea capaz de analizar grandes volúmenes de datos y de realizar automáticamente un diagnóstico preciso.

Este tipo de métodos se denominan *métodos inteligentes de diagnóstico de fallas* y en general, están compuestos por dos etapas. La primera consiste en extraer características útiles desde los datos recolectados con alguna herramienta de procesamiento de señales. La segunda utiliza técnicas de inteligencia artificial para distinguir el estado de salud de la maquinaria en base a las características obtenidas en el paso anterior.

La primera de estas etapas resulta clave en el diagnóstico de fallas para maquinaria rotativa [6], porque reduce la dimensión de los datos, afecta directamente el reconocimiento de patrones y, por lo tanto, la calidad del diagnóstico. Los métodos convencionales de extracción de características incluyen técnicas en el dominio del tiempo, dominio de la frecuencia y tiempo frecuencia. Diversos autores utilizan estadísticos basados en la forma de la onda en el dominio del tiempo, tales como el valor pico, la media cuadrática, el factor de cresta, el índice de curtosis y la asimetría [7].

Los principales métodos en el dominio de la frecuencia corresponden a: el cálculo del espectro energético de Fourier [8], el espectro de potencia [5] y el análisis cepstral. Respecto a las técnicas de tiempo frecuencia, suelen ser utilizadas la transformada de Fourier de tiempo corto (STFT), la distribución Wigner-Ville (WVD) y la transformada de paquetes wavelet (WPT). También es posible encontrar otros métodos basados en procesamiento de señales como la descomposición modal empírica (EMD) [9], funciones modales intrínsecas (IMF), la transformada wavelet discreta (DWT), la transformada wavelet (WT) [10] y la transformada Hilbert-Huang (HHT) [14].

Respecto a las técnicas de inteligencia artificial, aplicadas como clasificador en este contexto, podemos encontrar redes neuronales artificiales (ANN) [12], máquinas de vectores de soporte (SVM) [22], sistemas Adaptativos de inferencia neuro difusa (ANFIS) [3], redes neuronales wavelet (WNN) [13] y otros. En la actualidad existe una tendencia en el uso de técnicas de *deep learning*.

De acuerdo a Jia et al. (2016) [8], los métodos inteligentes de diagnóstico de fallas más comunes emplean ANN's en el segundo paso, dando lugar a dos grandes desventajas. La primera es que la extracción de características depende de la experiencia en procesamiento de señales de los investigadores y se seleccionan según un problema de diagnóstico específico, por lo que probablemente no sean aptas para otros problemas. Y la segunda es que las ANN's utilizadas en estos métodos carecen de arquitecturas profundas, limitando la capacidad de estas para aprender las complejas relaciones no lineales en los problemas de diagnóstico de fallas.

Por lo tanto, es necesario extraer de forma adaptativa características ocultas en las señales de vibración que reflejen las diferentes condiciones de salud de la maquinaria, en lugar de extraer y seleccionar las características de forma manual. Las técnicas de *deep learning* tienen el potencial de superar las deficiencias previamente mencionadas que poseen los métodos inteligentes de diagnóstico de fallas. Gracias a las arquitecturas profundas, las redes neuronales profundas (DNN) pueden capturar de forma adaptativa la información representativa de datos brutos a través de múltiples transformaciones no lineales y aproximar complejas funciones no lineales con un pequeño error.

En función de lo anterior, en el presente trabajo se propone un método inteligente de diagnóstico de fallas para rodamientos con la transformada de Fourier como técnica de extracción de características, y una DNN como clasificador. La red neuronal profunda utilizará *sparse* autoencoders apilados para reducir la dimensión de las características a la vez que realizan un pre entrenamiento de las capas ocultas de la red. Además, se propone construir conjuntos de datos separados por velocidad, su respectiva carga, 3 severidades de falla para cada componente y un conjunto con todas las velocidades y cargas juntas.

2 Definición de objetivos

En esta sección se definen los objetivos general y específicos que persigue la ejecución de este proyecto.

2.1 Objetivo general

Desarrollar un diagnosticador de severidad de fallos de rodamientos, bajo velocidades variables utilizando la transformada de Fourier discreta.

2.2 Objetivos específicos

Los objetivos específicos de este proyecto son los siguientes:

1. Descomponer señales de vibración con la transformada de Fourier discreta.
2. Diseñar e implementar un algoritmo de *deep learning* con sparse autoencoders.
3. Evaluar la exactitud del diagnosticador con 4 estados de salud del rodamiento.
4. Evaluar la exactitud del diagnosticador con 10 estados de salud del rodamiento.

3 Plan de trabajo

Este plan de trabajo utiliza los objetivos específicos como una base, debido a que estos explican las tareas que se deben llevar a cabo para cumplir el objetivo general. Esta segunda entrega, contempla la implementación del modelo propuesto y la evaluación del rendimiento de este. Para la próxima entrega se habrá pulido el marco teórico y la redacción de la investigación en general, sobre todo la discusión de resultados y el estado del arte.

4 Marco Teórico

Esta sección tiene como objetivo presentar antecedentes, conceptos, definiciones y la importancia de las técnicas y herramientas que se utilizarán en el desarrollo de esta investigación. Así, comenzaremos con el empleo de enfoques en el dominio de la frecuencia, seguido del aprendizaje profundo y luego el sustento matemático de ambos. Finalmente se dará a conocer la descripción del método inteligente de diagnóstico propuesto en esta investigación y todo lo relacionado con funcionamiento de la DNN como clasificador.

En la literatura especializada, existen antecedentes que sirven como base para esta investigación, un ejemplo de estos es Mao et al. (2017) [23], quien utiliza la transformada de Fourier para extraer características de una señal y usar la energía contenida en el espectro de frecuencias como entrada de 3 clasificadores: una ANN, una DNN con autoencoders apilados y una máquina de aprendizaje extremo (ELM). Este ejemplo corresponde a un antecedente idóneo para esta investigación, debido a que implementa la misma técnica de extracción de características y clasificador que este trabajo, con la salvedad que sus autoencoders no incluyen la

dispersión de los nodos en la capa oculta, variación denominada *sparse* autoencoder y que sí es utilizada en esta investigación.

Otro ejemplo de un antecedente es Gupta et al. (2017) [24], quien postula que el análisis espectral es el enfoque más utilizado para detectar fallas en los rodamientos. El principal motivo de lo anterior es que los enfoques en el dominio del tiempo solo pueden detectar cuándo ocurre una falla en el rodamiento, mientras que el análisis espectral, además puede determinar en qué componente del rodamiento se encuentra el desperfecto. El autor también aclara que la transformada rápida de Fourier (FFT) es el algoritmo computacional más eficiente para convertir una señal en el dominio del tiempo a componentes de frecuencia discretos.

En cuanto a las redes neuronales profundas, podemos encontrar a Zhang et al. (2017) [11], este autor propone el uso de una DNN cuya función de costos es la entropía cruzada de los valores de salida y el *target*. Los autores no utilizan métodos de extracción de características y solo dividen en segmentos la señal original para entrenar a la red neuronal profunda. Su principal argumento es que se debe mantener la coherencia temporal de la señal, pero la cantidad de datos que recibe la red es enorme, resultando en grandes tiempos de entrenamiento y una gran capacidad de cómputo.

A continuación se encuentran los conceptos y definiciones matemáticas de las herramientas que se utilizarán en esta investigación, comenzando por la transformada de Fourier y sus variantes, seguido del *deep learning* (o aprendizaje profundo), los autoencoders y su variante *sparse* autoencoder.

4.1 Transformada de Fourier

La transformada de Fourier es una herramienta matemática poderosa, por lo que es utilizada en diversos campos de la ciencia y la ingeniería. Puede ser vista como la transformación de una señal en un dominio (típicamente tiempo o espacio) a otro dominio, el dominio de la frecuencia. Al aplicar la transformada de Fourier a una señal, se obtienen sus componentes fundamentales en frecuencia que a menudo revelan estructuras ocultas, que en el dominio del tiempo, no son posibles de observar.

Como consecuencia de lo anterior, el análisis de Fourier está en la base de muchas teorías de la ciencia y juega un papel fundamental en el diseño práctico de la ingeniería. El alcance actual de la influencia del análisis de Fourier está indicado por una lista parcial de científicos e ingenieros que la utilizan [16]:

- Los ingenieros de audio utilizan técnicas de Fourier, en parte porque el oído parece ser sensible al comportamiento del dominio de frecuencia.
- Los estadísticos y probabilistas caracterizan y calculan las distribuciones de probabilidad usando transformadas de Fourier. Las transformadas de Fourier de las funciones de covarianza se utilizan para caracterizar y estimar las propiedades de los procesos aleatorios.

- Los radioastrónomos utilizan la transformada de Fourier para formar imágenes a partir de datos interferométricos recogidos por arreglos de antenas.
- Los cristalógrafos encuentran las estructuras cristalinas usando transformadas de Fourier de los patrones de difracción de rayos X.
- Los ingenieros biomédicos recogen datos en el dominio de la frecuencia y luego los transforman inversamente con Fourier para obtener imágenes de resonancia magnética.

4.1.1 Transformada de Fourier continua

La transformada de Fourier continua es resultado de la generalización de las series de Fourier. Puede aplicarse prácticamente a cualquier función, no requiere que la función sea periódica, y para datos discretos, puede ser evaluada rápidamente usando una técnica moderna llamada transformada de Fourier Rápida (FFT). La transformada de Fourier de una función $f(t)$ se define [15] como:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (1)$$

Donde ω es igual a la frecuencia angular $2\pi f$, y $F(\omega)$ es una función continua de valores complejos. Una vez que se ha determinado una transformada de Fourier, la función original $f(t)$ puede ser recuperada a partir de la transformada de Fourier inversa:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{i\omega t} d\omega \quad (2)$$

4.1.2 Transformada de Fourier discreta

La transformada discreta de Fourier (DFT) (Ecc. 3), como lo indica su nombre, opera sobre un número finito de datos y es discreta en ambos dominios, el del tiempo y el de la frecuencia. La DFT trabaja sobre un bloque de N muestras de la secuencia $x(n)$ y entrega N componentes de frecuencia indexadas mediante la variable entera k .

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)kn} \quad , 0 \leq k \leq N-1 \quad (3)$$

La transformada discreta de Fourier inversa, o IDFT está dada por la Ecc. 4, la cual recibe N componentes de frecuencia y entrega N muestras de la secuencia $x(n)$. La fórmula es muy similar a la de la DFT, se distingue en el signo del exponente de e y en la presencia del factor de normalización $1/N$. Ambas diferencias son necesarias para que se cumpla la condición de la Ecc. 5.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j(2\pi/N)kn} \quad , 0 \leq n \leq N-1 \quad (4)$$

$$x(n) = IDFT[DFT[x(n)]] \quad (5)$$

4.1.3 Transformada de Fourier de tiempo corto

Consideremos una señal $x(t)$ y supongamos que es estacionaria cuando se ve a través de una ventana $w(t)$, cuya duración es T , centrada en la posición de tiempo τ , la transformada de Fourier de la señal en ventana $x(t)$ es la transformada de Fourier de corto tiempo [19] (STFT):

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-j\omega t}dt \quad (6)$$

Esta transformada mapea la señal desde el dominio del tiempo en un plano de tiempo-frecuencia conjunto (τ, f) . Para la implementación, la versión discreta de la Ecc. 6 debe utilizarse como:

$$X_n(e^{jw_k}) = \sum_{m=-\infty}^{\infty} w(n - m)x(m)e^{-jw_k m} \quad (7)$$

Donde n y k ($1 \leq k \leq N$) son el tiempo discreto y la frecuencia respectivamente, y N es la longitud de la ventana

4.2 Señal de vibración

Los rodamientos pueden comenzar a fallar por muchos motivos: un mal diseño o instalación, corrosión, poca lubricación y sobrecarga son algunos de ellos. Durante la operación del rodamiento se generan impulsos de banda ancha cuando la bola pasa por el defecto a una frecuencia determinada por la velocidad del eje, la geometría del rodamiento y la ubicación del defecto. La dificultad de diagnosticar la salud de los rodamientos reside en el hecho de que los patrones de un rodamiento defectuoso se extienden a través de una amplia banda de frecuencias, por lo que pueden ser fácilmente enmascarados por ruido y efectos de baja frecuencia.

La transformada rápida de Fourier (FFT) fue seleccionada como técnica de extracción de características por ser una herramienta de baja complejidad computacional y porque puede trasladar la señal original al dominio de la frecuencia. Sin embargo, tiene una serie de desventajas en este tipo de problemas en que la señal de vibración (Figura 1) es tremendamente compleja. Podríamos decir que la desventaja más importante es que, esta técnica es menos eficiente cuando las fallas son distribuidas y la señal tiene un comportamiento no lineal, el espectro de frecuencias varía en el tiempo y la técnica no es capaz de realzar las componentes impulsivas características de la falla.

En la Fig. 10 se encuentra el diagrama de flujo de los pasos descritos en esta sección, que van desde la carga de las señales, hasta el entrenamiento de la red neuronal profunda y las gráficas de las métricas para el diagnóstico. En el primer bloque del diagrama se encuentra la sucesión de pasos para obtener las características que alimentarán a la capa de entrada del clasificador.

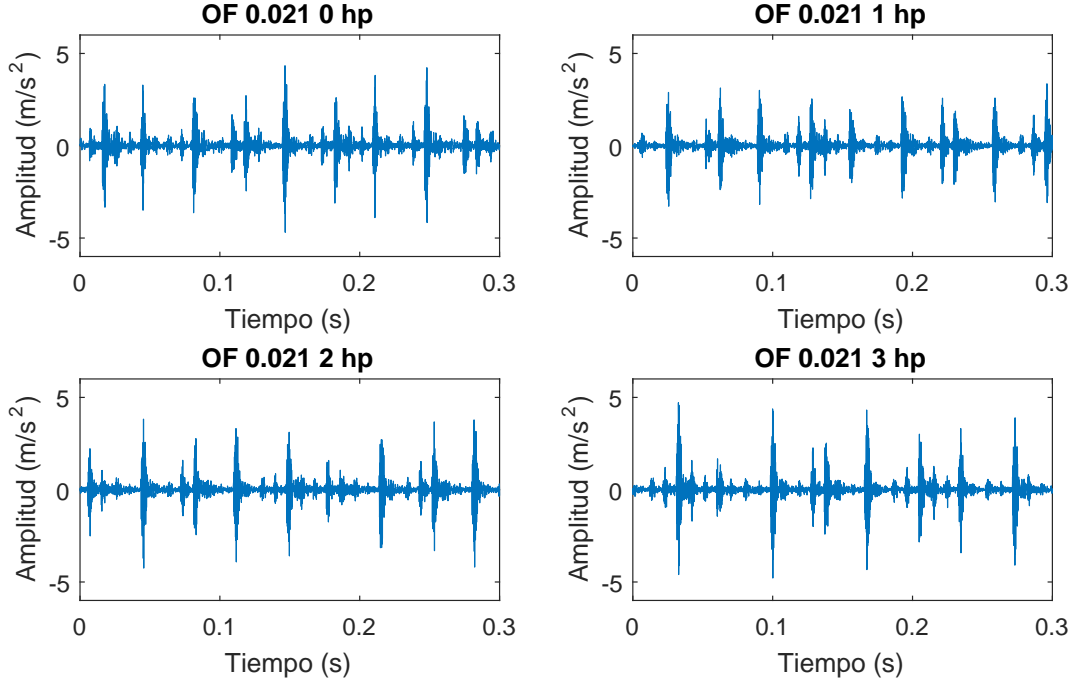


Figura 1: 4 tipos de señal de vibración de un rodamiento, con una carga de 3 hp y 0,54 mm de diámetro de falla.

4.3 Deep learning

En los años recientes, un nuevo enfoque de *machine learning*, denominado *deep learning*, ha llamado la atención de investigadores de múltiples áreas debido a una serie de ventajas respecto a otros enfoques. *Deep learning* ha tenido un gran impacto en trabajos actuales de reconocimiento de voz, visión artificial y procesamiento del lenguaje natural, a causa de la efectividad empírica de este respecto a enfoques alternativos en estos ámbitos. El desarrollo de esta tecnología fue potenciado, en gran medida, por la creciente capacidad de cómputo de los equipos, y por el uso de los inmensos volúmenes de datos que son generados todos los días por la humanidad.

Esta ola de interés por el *deep learning*, se centra principalmente en los perceptrón multicapa profundo (deep multilayer perceptron), las redes convolucionales profundas (deep convolutional network) y las redes neuronales recurrentes (recurrent neuronal network) [25]. Sin embargo, la mayoría de los métodos de *deep learning* se basan en el perceptrón multicapa profundo, porque lo utilizan como el bloque esencial de construcción. Este nuevo enfoque formula el problema subyacente como una arquitectura en forma de red, en la que su capa de salida define una función de costos (también llamada función de pérdida) que es necesaria para aprender.

El desarrollo del *deep learning* está relacionado con el uso de técnicas de aprendizaje no supervisado como los diferentes tipos de autoencoder que existen para realizar un pre entrenamiento de las capas ocultas. De acuerdo a Ranzato et al. (2008) [27], uno de los principales propósitos del aprendizaje no supervisado es producir buenas representaciones de los datos

que puedan ser usadas para la detección, reconocimiento, predicción o visualización. Las buenas representaciones eliminan variaciones irrelevantes de los datos de entrada, mientras que se preserva la información que es útil para la tarea a realizar.

El aprendizaje no supervisado tiene la habilidad de crear jerarquías de características profundas (deep features hierarchies) mediante el apilado de módulos no supervisados uno encima de otro. Los módulos que se encuentran apilados se alimentan con los vectores de representación producidos por aquellos que se encuentran en un nivel inferior de la pila. Las representaciones producidas en el nivel superior captan las dependencias de alto nivel entre las variables de entrada, mejorando la capacidad de la red para capturar las regularidades subyacentes en los datos de entrada.

4.3.1 Autoencoders

Un autoencoder (AE) es una red neuronal artificial no supervisada de tres capas que es entrenada para replicar los datos de entrada en la salida. El entrenamiento de los AE es no supervisado en el sentido de que no requiere datos etiquetados para aprender a replicarlos en la salida. Este tipo especial de red neuronal es utilizada para pre entrenar las capas ocultas de las redes neuronales profundas, de manera que encuentran los mejores pesos sinápticos posibles que agilizan la búsqueda de mínimos globales para la función de costos.

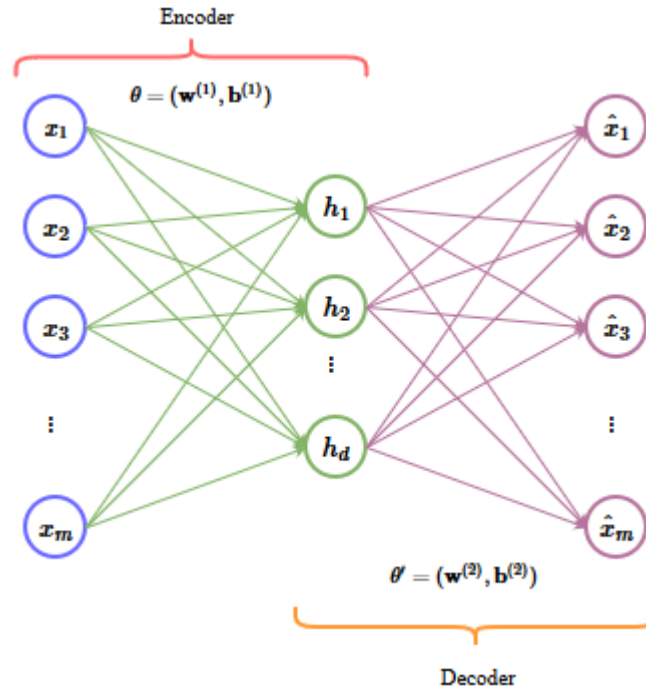


Figura 2: Diagrama que ilustra la estructura de un autoencoder formado por la red encoder (lado izquierdo) y la red decoder (lado derecho) con sus respectivos parámetros de entrenamiento.

Es posible identificar dos estructuras en la red autoencoder, por una parte el codificador en el lado izquierdo de la Figura 2, que mapea los datos de entrada desde un espacio de alta dimensión en códigos de un espacio de baja dimensión. Por otra parte, que se encuentra al lado derecho de la Figura 2, reconstruye los datos de entrada a partir de los códigos obtenidos por la capa oculta llevándolos a su dimensión original [17]. Ahora que está claro cuál es el encoder y el decoder, podemos definir matemáticamente ambas estructuras.

Dadas las muestras de entrenamiento $\mathbf{x} = (x_1, x_2, \dots, x_n)$, donde la muestra $x_n \in \mathbb{R}^m$, se define la red encoder como una función codificadora f_θ (Ecc. 9) cuya tarea es representar estas muestras en un vector con códigos de menor dimensión denominado $\mathbf{h} = (h_1, h_2, \dots, h_n)$, en donde $h_n \in \mathbb{R}^d$ y $d < m$. La matriz de pesos sinápticos de la función encoder corresponde a $\mathbf{w}^{(1)} = (w_1^{(1)}, w_2^{(1)}, \dots, w_m^{(1)})$ con $w_m^{(1)} \in \mathbb{R}^d$, mientras que $\mathbf{b}^{(1)} = (b_1^{(1)}, b_2^{(1)}, \dots, b_d^{(1)})^T$ representa el arreglo de *biases* pertenecientes a las neuronas de la capa oculta. Los pesos sinápticos y los *biases* conforman θ , el conjunto de parámetros a minimizar en la función de costos que será detallada más adelante.

$$\mathbf{h} = f_\theta(\mathbf{x}) \quad (8)$$

$$f_\theta(\mathbf{x}) = s_f(\mathbf{w}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) \quad (9)$$

$$\theta = (\mathbf{w}^{(1)}, \mathbf{b}^{(1)}) \quad (10)$$

La red decoder se define como una función de reconstrucción denotada como $g_{\theta'}$ (Ecc. 12), dicha función mapea el vector \mathbf{h} desde el espacio de baja dimensión en el que se encuentra, de vuelta al espacio de alta dimensión original realizando así la reconstrucción. Los datos reconstruidos se denotan mediante $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$ (Ecc. 11), en donde $\hat{x}_n \in \mathbb{R}^m$. El primer parámetro que compone a θ' (Ecc. 13) perteneciente al decoder, es la matriz de pesos sinápticos de salida $\mathbf{w}^{(2)} = (w_1^{(2)}, w_2^{(2)}, \dots, w_m^{(2)})$ con $w_m^{(2)} \in \mathbb{R}^d$. Mientras que el segundo, es el arreglo de *biases* $\mathbf{b}^{(2)} = (b_1^{(2)}, b_2^{(2)}, \dots, b_d^{(2)})^T$.

$$\hat{\mathbf{x}} = g_{\theta'}(\mathbf{h}) \quad (11)$$

$$g_{\theta'}(\mathbf{h}) = s_g(\mathbf{w}^{(2)}\mathbf{h} + \mathbf{b}^{(2)}) \quad (12)$$

$$\theta' = (\mathbf{w}^{(2)}, \mathbf{b}^{(2)}) \quad (13)$$

Los conjuntos de parámetros θ y θ' de las redes encoder y decoder respectivamente, son aprendidos simultáneamente en la tarea de reconstruir en la mayor medida posible los datos originales de entrada. En otras palabras, es deseable obtener el menor error de reconstrucción (Ecc. 14) posible a través de las n muestras de entrenamiento, utilizando el error cuadrático medio (MSE). Este estimador es la función de costo estándar para los autoencoder y muchos otros algoritmos de *machine learning* [25].

$$J_{mse}(\theta, \theta') = \frac{1}{2n} \sum_{i=1}^n \|x_i - \hat{x}_i\|^2 \quad (14)$$

El proceso de aprendizaje del autoencoder puede ser visto como un problema de optimización, porque tiene como objetivo encontrar valores que se aproximan al mínimo global de la función de costos. La función que debemos optimizar (Ecc. 15), también llamada función de pérdida, se compone por el error de reconstrucción (Ecc. 14) y otro término llamado *weight decay* (Ecc. 16). De acuerdo a [26], el término *weight decay* fuerza a algunos de los pesos sinápticos que tienen una pobre capacidad de generalización a tomar valores cercanos a 0, evitando el sobreentrenamiento de la red.

$$\phi_{AE}(\theta, \theta') = J_{mse}(\theta, \theta') + J_{weight}(\theta, \theta') \quad (15)$$

$$J_{weight}(\theta, \theta') = \frac{\lambda}{2} \sum_{l=1}^2 \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (w_{ji}^{(l)})^2 \quad s_1 = m, s_2 = d, s_3 = m. \quad (16)$$

Resulta común aplicar el algoritmo *backpropagation*, ampliamente utilizado en *machine learning*, de la función de costos de los autoencoder estándar para encontrar valores mínimos. Este algoritmo calcula el gradiente a la función de costos para encontrar direcciones, las cuales permiten actualizar los pesos y *biases* iniciales, de manera que con cada iteración se acerquen más al mínimo global de la función. Obtenidos los mejores valores posibles, la capa oculta \mathbf{h} del autoencoder es capaz de representar a los datos de entrada en una dimensión menor que la original, sin perder información relevante para el reconocimiento de patrones.

4.3.2 Sparse autoencoders

Podemos definir al *sparse* autoencoder, desde ahora denominado SAE, como una extensión del autoencoder estándar que obliga a sus unidades de la capa oculta a mantener una baja activación media. Creando así, características dispersas que logran una mejor generalización en la tarea de clasificar[28]. El SAE impone un término de penalización (Ecc. 18) a las unidades de la capa oculta, forzando a que la activación media de cada unidad sea cercana al parámetro de dispersión ρ que generalmente toma un valor muy pequeño cercano a 0.

$$\phi_{SAE}(\theta, \theta') = \phi_{AE}(\theta, \theta') + J_{sparse}(\theta, \theta') \quad (17)$$

La restricción se hace efectiva agregando el término de penalización a la función de costos del autoencoder estándar ϕ_{AE} (Ecc. 15), resultando en una nueva función de costos ϕ_{SAE} (Ecc. 17).

$$J_{sparse}(\theta, \theta') = \beta \sum_{j=1}^d KL(\rho || \hat{\rho}_j) \quad (18)$$

$$KL(\rho || \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j} \quad (19)$$

$$\hat{\rho}_j = \frac{1}{n} \sum_{k=1}^n h_{j,k} \quad (20)$$

J_{sparse} impone la restricción de dispersión, β controla el peso de esta restricción y $KL(\rho||\hat{\rho}_j)$ (Ecc. 19) es la divergencia Kullback-Leibler entre una variable aleatoria Bernoulli con media ρ y una variable aleatoria Bernoulli media $\hat{\rho}_j$. La divergencia KL es una función estándar para medir la diferencia entre dos distribuciones de probabilidad diferentes. Definiremos $\hat{\rho}_j$ (Ecc. 20) como la probabilidad media de activación en la j -ésima unidad de la capa oculta \mathbf{h} a través las n muestras, y ρ la probabilidad de activación deseada ($\rho \ll 1$) [18].

Una vez que hemos definido el término de dispersión y lo hemos incorporado a la función de costos del autoencoder (Ecc. 21), aplicaremos el método del gradiente conjugado para actualizar los pesos y *biases*. El método del gradiente conjugado pertenece a una clase de métodos de optimización de segundo orden conocidos colectivamente como métodos de dirección conjugada [26]. La motivación para utilizar este método es que converge más rápido hacia un mínimo que el descenso del gradiente, y a la vez se evitan los requisitos de computación y almacenamiento del hessiano, como lo requiere el método de Newton.

$$\phi_{SAE}(\theta, \theta') = \frac{1}{2n} \sum_{i=1}^n \|x_i - \hat{x}_i\|^2 + \frac{\lambda}{2} \sum_{l=1}^2 \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (w_{ji}^{(l)})^2 + \beta \sum_{j=1}^d KL(\rho||\hat{\rho}_j) \quad (21)$$

Para simplificar la notación del método, definiremos Θ (Ecc. 22) como un arreglo que contiene los parámetros de la función de costos. Según la Ecc. 23, este método toma los parámetros iniciales y los actualiza sumándoles una dirección multiplicada por una tasa de aprendizaje denominada η . El arreglo que contiene las direcciones (Ecc. 24) se obtiene utilizando la dirección previa multiplicada por el parámetro conjugado γ más el residuo (Ecc. 25). El arreglo gradiente (Ecc. 26) está compuesto por las derivadas parciales de la función de costos, a la vez que la derivada parcial $\frac{\partial \phi_{SAE}}{\partial \theta}$ se compone de las derivadas parciales de pesos y *biases* (Ecc. 27).

$$\Theta = (\theta, \theta') \quad (22)$$

$$\Theta(n+1) = \Theta(n) + \eta s(n) \quad (23)$$

$$s(n+1) = r(n+1) + \gamma s(n) \quad (24)$$

$$r(n+1) = -\nabla \phi_{SAE}(\Theta(n+1)) \quad (25)$$

$$\nabla \phi_{SAE} = \left(\frac{\partial \phi_{SAE}}{\partial \theta}, \frac{\partial \phi_{SAE}}{\partial \theta'} \right) \quad (26)$$

$$\frac{\partial \phi_{SAE}}{\partial \theta} = \left(\frac{\partial \phi_{SAE}}{\partial \mathbf{w}^{(1)}}, \frac{\partial \phi_{SAE}}{\partial \mathbf{b}^{(1)}} \right) \quad (27)$$

4.4 Método de diagnóstico basado en DNN

En esta sección, se dará detalle del proceso de entrenamiento de la red neuronal profunda (DNN) y los elementos que la componen. La red utilizada en este modelo, posee una capa de entrada, dos capas ocultas y una capa de salida con la función de transferencia softmax. Las

dos capas ocultas son pre entrenadas con dos *sparse* autoencoder (SAE) respectivamente. Estas redes de tres capas permiten encontrar los parámetros de entrenamiento a la vez que reducen la dimensión de los datos de entrada codificandolos en un espacio de menor dimensión.

Una vez que el primer SAE termina de aprender su representación de los datos a través del algoritmo del gradiente conjugado (Ecc. 26), inicializa con sus parámetros óptimos θ_1 a la primera capa oculta de la DNN como se muestra en la parte izquierda A de la Fig. 3. Luego el segundo SAE, constituido por la primera y la segunda capa oculta de la DNN, aprende un conjunto de parámetros que inicializa a la segunda capa oculta y los códigos de la segunda capa oculta alimentan a la regresión softmax para que sea entrenada.

Una vez obtenidos los parámetros de las capas ocultas, la DNN realiza un *fine tuning* que básicamente es aplicar el algoritmo *backpropagation* a la red entera para mejorar los pesos y *biases* en conjunto. Se utiliza el descenso del gradiente en vez del gradiente conjugado porque los parámetros se encuentran cerca del mínimo global de la función de costos, por lo que no son necesarias demasiadas iteraciones en el proceso.

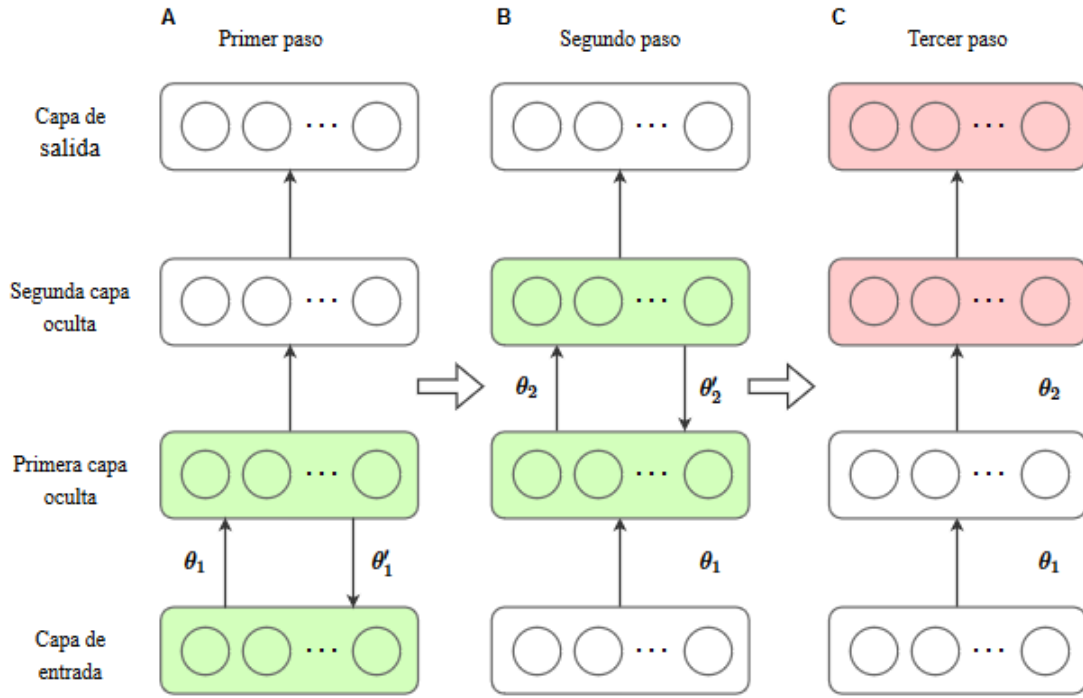


Figura 3: Diagrama que representa el entrenamiento de los *sparse* autoencoder y la asignación de sus parámetros a la DNN

5 Experimento y resultados

5.1 Origen de los datos

Los datos utilizados en este trabajo corresponden a señales de vibración de un rodamiento de bolas, obtenidos de un experimento realizado por el Bearing Data Center of Case Western Reserve University [29]. Los aparatos empleados en este laboratorio de vibraciones corresponden a: un motor de 2 hp a la izquierda del banco de pruebas (como se aprecia en la Figura 4), un transductor/codificador de torque en el centro y un dinamómetro a la derecha. Un acelerómetro es el encargado de registrar las señales de vibración emitidas por el rodamiento que soporta el eje del motor.

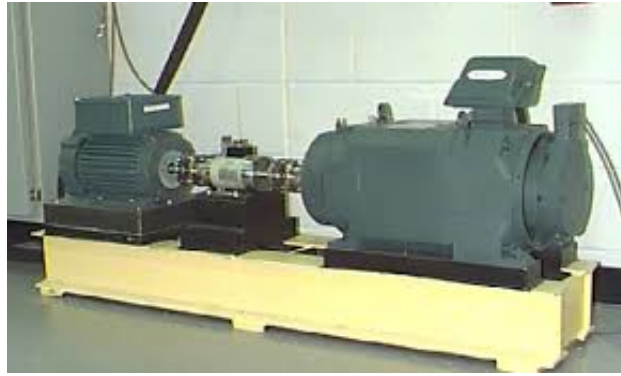


Figura 4: Ilustración del experimento realizado por la Case Western Reserve University [29]

El rodamiento ubicado en el *drive end* fue sometido a descargas eléctricas que generaron defectos en un punto único, con tres diámetros de falla: 0,007, 0,014 y 0,021 pulgadas. Estas imperfecciones son inducidas bajo las velocidades de rotación 1.730, 1.750 1.772 y 1.792 rpm y 4 diferentes cargas de 0, 1, 2 y 3 hp respectivamente. Por lo anterior es que los datos pueden ser organizados en conjuntos separados por velocidad/carga como se detallará en la siguiente sección. La frecuencia de muestreo es de 12 kHz para las cuatro condiciones de salud: normal (N), falla en el anillo externo (OF), falla en el anillo interno (IF) y falla en la bola (RF).

El principal motivo para utilizar esta base de datos es que sirve como punto de comparación para este trabajo, permitiendo efectuar el contraste con investigaciones similares que realizan diagnóstico de fallas inteligente en rodamientos. De esta manera, es posible desarrollar conclusiones a cerca del desempeño de la configuración propuesta y de la aplicación del *deep learning* a este tipo de problemas. Además, al tener tipos de fallas con severidades, es posible elevar el grado de complejidad del problema aparte de probar la capacidad de reconocimiento de más estados de salud con del modelo propuesto.

5.2 Energía de los coeficientes de Fourier

Cinco conjuntos de datos, ilustrados en la Tabla 1, serán utilizados en este trabajo para probar el rendimiento del método propuesto. Los *datasets* poseen 10 etiquetas de clasificación que corresponden a los 10 estados de salud del rodamiento, separados bajo las velocidades de

Dataset	Carga (hp)	Cantidad de muestras	Tipo de falla	Diámetro de falla (in)	Etiqueta
A/B/C/D/E	0/1/2/3/0-3	59/59/59/59/236	RF	0,007	1
		59/59/59/59/236	RF	0,014	2
		59/59/59/59/236	RF	0,021	3
		59/59/59/59/236	IF	0,007	4
		59/59/59/59/236	IF	0,014	5
		59/59/59/59/236	IF	0,021	6
		59/59/59/59/236	OF	0,007	7
		59/59/59/59/236	OF	0,014	8
		59/59/59/59/236	OF	0,021	9
		59/59/59/59/236	N	N/A	10

Tabla 1: Descripción de los 5 *datasets*.

eje de 1.797, 1.772, 1.750 y 1.720 rpm y las cargas 0, 1, 2 y 3 hp correspondientemente. El conjunto E es el de mayor tamaño, porque reúne las 4 velocidades y es, por lo tanto, el que más se aproxima a una señal de vibración de un rodamiento sometido a carga y velocidad variables.

El largo de la señal de vibración bruta es de 120.832 puntos, resultado de 10,06 segundos de medición y el tamaño de segmentación de las subseñales es de 2.048 puntos (unos 170,6 ms). Este tamaño de segmentación corresponde entre 5,12 a 4,9 veces el periodo de rotación, considerando que este fluctúa entre 400 y 418 puntos de la señal. Con dicho tamaño de segmentación, tal como lo indica la Tabla 1, la cantidad de muestras totales que tienen los 4 primeros *datasets* es de 590 y 2.360 en el caso del E.

Luego de obtener las subseñales de 2.048 puntos, se les aplica la transformada rápida de Fourier (FFT) obteniendo la misma cantidad de componentes de frecuencia. Dado que la transformada tiene la propiedad de simetría, es posible tomar los primeros 1.024 coeficientes de Fourier para cada muestra. A continuación, se les aplica valor absoluto y se elevan al cuadrado, adquiriendo el espectro energético de las subseñales.

5.3 Proceso de entrenamiento y testing

Obtenida la energía de los coeficientes de Fourier, se reordenan de manera aleatoria las nuevas muestras contenidas en los conjuntos de datos y se divide cada uno de estos en dos partes: un conjunto de entrenamiento con 413 muestras (70%) y uno de *testing* con 177 muestras (30%). El conjunto de entrenamiento tiene como objetivo encontrar los mejores parámetros posibles de las capas ocultas de la DNN (pesos y *biases*). El de *testing* es usado para medir la exactitud (Ecc. 31) del modelo propuesto en este trabajo.

$$Precisión = \frac{TP}{TP + FP} \quad (28)$$

Nro.	λ_1	β_1	ρ_1	λ_2	β_2	ρ_2	Exactitud (%)
1	0,0179	0,9226	0,2623	0,0155	0,9450	0,1104	99,7363 \pm 0,3723
2	0,0199	0,9876	0,0143	0,0008	0,9405	0,2718	99,9294 \pm 0,0964
3	0,0029	0,9865	0,3396	0,0062	0,9712	0,2620	99,9718 \pm 0,0938
4	0,0201	0,9488	0,3735	0,0012	0,9736	0,0650	99,9153 \pm 0,1263
5	0,0139	0,9740	0,2715	0,0023	0,9249	0,0476	99,9153 \pm 0,1088
6	0,0023	0,9213	0,3031	0,0181	0,9491	0,1993	99,8729 \pm 0,1868
7	0,0062	0,9437	0,2972	0,0153	0,9456	0,3838	99,9859 \pm 0,0431
8	0,0121	0,9832	0,1569	0,0071	0,9617	0,1361	99,9670 \pm 0,0712
9	0,0210	0,9733	0,2621	0,0209	0,9667	0,2341	99,8635 \pm 0,2178
10	0,0212	0,9867	0,0684	0,0009	0,9703	0,0895	99,9718 \pm 0,0684

Tabla 2: 10 conjuntos de hiperparámetros probados con 30 ejecuciones en el dataset E, el formato de la exactitud es: Exactitud media \pm desviación estándar.

$$Sensibilidad = \frac{TP}{TP + FN} \quad (29)$$

$$F_1score = \frac{2 \cdot Precisión \cdot Sensibilidad}{Precisión + Sensibilidad} \quad (30)$$

$$Exactitud = \frac{VP + VN}{VP + FP + VN + FN} \quad (31)$$

Además de la exactitud, se emplean métricas que son habitualmente usadas en problemas de clasificación multiclase. Resulta común usar la validación cruzada para evaluar el aprendizaje de técnicas de inteligencia artificial, sin embargo, en este trabajo solo se corren 30 ejecuciones aleatorias por *dataset* sin este método de prueba. A los resultados de estas 30 pruebas, se les calcula la exactitud media, la exactitud promedio por clase, el F_1score (Ecc. 30) o media armónica por clase y el F_1score medio.

La DNN diseñada está compuesta por 4 capas, el número de unidades de la primera es 1.024, mientras que el de la cuarta capa es de 10, equivalente a la cantidad de clases. La segunda y la tercera corresponden a las capas ocultas, cuyas cantidades óptimas de nodos es complejo de encontrar, por lo que en este trabajo fueron determinadas a través de prueba y error. Las mejores cantidades de nodos ocultos encontradas son: 105 (aproximadamente un décimo de los nodos de la capa de entrada) y 25 (aproximadamente un cuarto de la primera capa oculta) unidades.

Durante el proceso de entrenamiento del primer SAE, los pesos y *biases* son iniciados de manera aleatoria, mientras que los del segundo SAE son inicializados con los mejores parámetros del primero. La función de transferencia de los nodos de la capa oculta de los SAE es la logística sigmoide y la cantidad de iteraciones en el proceso de entrenamiento es de 200. Por su parte,

Dataset	Carga (hp)	Exactitud m. (%)	D. estándar	Mínimo (%)	Máximo (%)	Tiempo e. (s)
A	0	99,8117	0,3089	98,8701	100	64,860
B	1	99,9623	0,1433	99,4350	100	59,106
C	2	99,9829	0,0936	99,4872	100	62,127
D	3	99,9058	0,2142	99,4350	100	64,101
E	0-3	99,9718	0,0862	99,5763	100	196,268

Tabla 3: Resultados del diagnóstico de cada *dataset*.

la capa de salida tiene una función softmax y su cantidad de iteraciones en el entrenamiento es el mismo valor mencionado en el SAE.

Los hiperparámetros penalizador de pesos (λ), penalizador de dispersión (β) y proporción de dispersión (ρ) de los *sparse* autoencoder son complejos de hallar, convirtiendo la búsqueda manual de estos en una tarea difícil. Para encontrar los mejores hiperparámetros, fue necesario construir una grilla con valores aleatorios normalmente distribuidos para los dos SAE y probar el desempeño de estos. La Tabla 2 muestra las 10 agrupaciones con valores que fueron probados en la DNN con el *dataset* E, y también los resultados de cada agrupación luego de 30 ejecuciones.

Los intervalos para construir la grilla de la Tabla 2 son los siguientes: $[0,0002;0,022]$, $[0,91;0,99]$ y $[0,00002;0,4]$ para λ , β y ρ respectivamente. Como es posible observar en aquella Tabla, ninguna agrupación de valores obtuvo una exactitud media menor a 99,86% y la mayoría registró una baja desviación estándar. Como consecuencia de esto, es posible afirmar que la variación de los valores para estos hiperparámetros, dentro los intervalos presentados, no genera grandes diferencias en los resultados.

La séptima agrupación de valores obtuvo la mayor exactitud promedio y la menor desviación estándar, lo que hace pensar que son los mejores candidatos por homogeneidad de resultados y exactitud. Estos valores son empleados para entrenar el modelo con los 5 *datasets*, obteniendo así los resultados de 30 procesos de entrenamiento y *testing* para cada uno de ellos. Dichos resultados se encuentran en la Tabla 3, incluyendo también la exactitud mínima y máxima alcanzada por el modelo en cada *dataset*.

Como es posible observar en la Tabla 3, la exactitud promedio supera el 99,80% en todos los *datasets* y todos alcanzan el 100% en la mayoría de las ejecuciones. El *dataset* A es el que menor homogeneidad presenta en sus resultados, con 21 oportunidades en las que alcanzó el 100%. El resto de los *datasets* sobrepasa el 99,90% de exactitud promedio y sus resultados son bastante homogéneos, sobretodo en el C y el E.

Si se presta atención a los resultados del *dataset* E en las Tablas 2 y 3, se puede apreciar la exactitud media cambia desde un 99,9859% a un 99,9718%. Este cambio se debe a que los valores de los pesos y *biases* se inician de manera aleatoria en el primer SAE, haciendo variar

la exactitud de cada ejecución. Este efecto de variación influye en todas las pruebas de todos los *datasets* y es responsable en parte, de la heterogeneidad de los resultados del diagnóstico.

En la Figura 5 es posible distinguir que en 30 procesos de entrenamiento y *testing*, el modelo alcanza el 100% de exactitud en 26 oportunidades (86,6%). De aquellas ejecuciones en las que no se consigue el 100%, la mas baja es de un 99,5763%, la cual provoca un impacto en la exactitud media y en la desviación estándar. Sin embargo, las 4 caídas (13,4%) en la exactitud son de menor magnitud respecto a las bajas que presentan los demás *dataset*, lo cual se puede percibir en la Figura 6.

Respecto a los resultados del diagnóstico de los 5 *datasets*, la Figura 6 presenta las curvas de exactitud a través de las ejecuciones. La Figura deja en evidencia que las caídas de exactitud en los *datasets* con excepción del A, están en torno del 99,4%, lo que sugiere que los resultados fueron bastante parecidos y que no hay una gran variabilidad. De no ser por la baja de exactitud del *dataset* E en la ejecución número 14, los resultados serían aun más parejos.

La exactitud por etiqueta de clasificación se encuentra contemplada en la Figura 5, la cual nos advierte que el modelo propuesto tiene problemas en ciertas clases. Aquellas son las 1, 2 y 3 correspondientes a fallas en las bolas del rodamiento (0,007, 0,014 y 0,021 pulgadas respectivamente), clases que registran sus rendimientos más bajos en el *dataset* A. En este *dataset* el rodamiento no soporta una carga, haciendo que la señal que se genera con el contacto de la falla con otras superficies sea tenue, complicando la tarea de clasificación.

La gráfica del F-score (o media armónica) de las 5 pruebas, materializada en la Figura 8, es bastante parecida a la de la exactitud media, solo que las bajas de exactitud son más acentuadas. En la Figura 9 se encuentra el F-score por etiqueta obtenido de las pruebas, en ella se da cuenta de las grandes caídas de exactitud de los *dataset* A y D. Ambas Figuras son parecidas a las de exactitud promedio y por etiqueta, con la diferencia que se ven con mayor énfasis las bajas de exactitud.

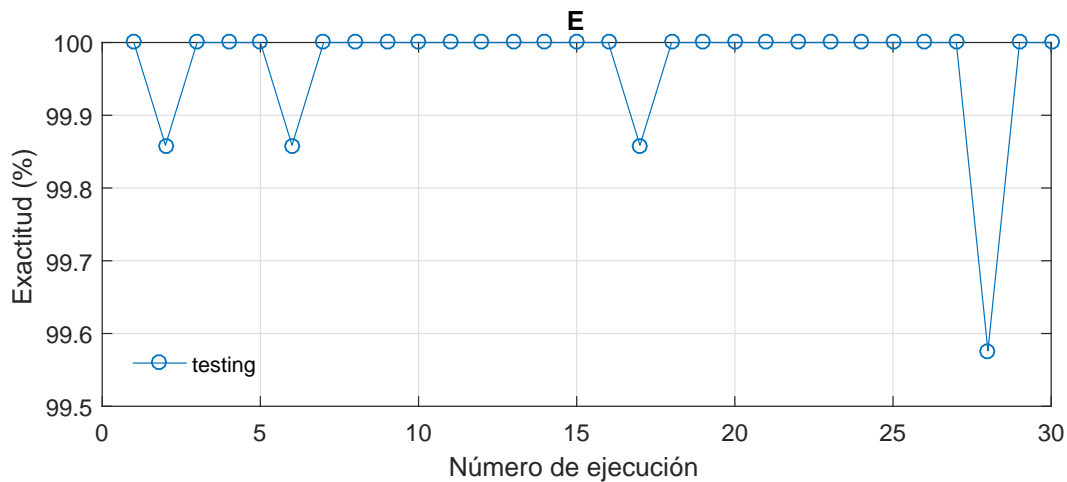


Figura 5: Resultados del diagnóstico en 30 *trials* del *dataset* E.

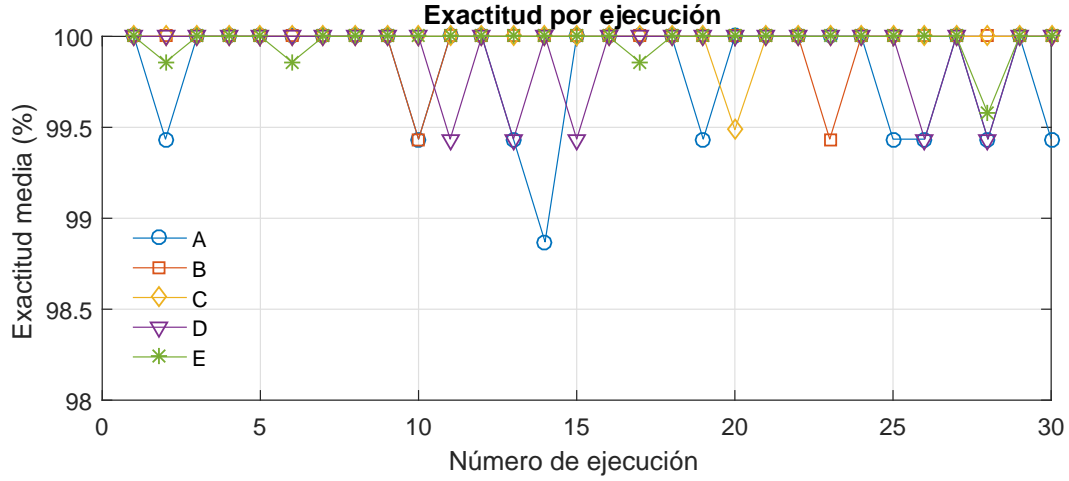


Figura 6: Exactitud promedio del diagnóstico en 30 *trials* de los 5 *datasets*.

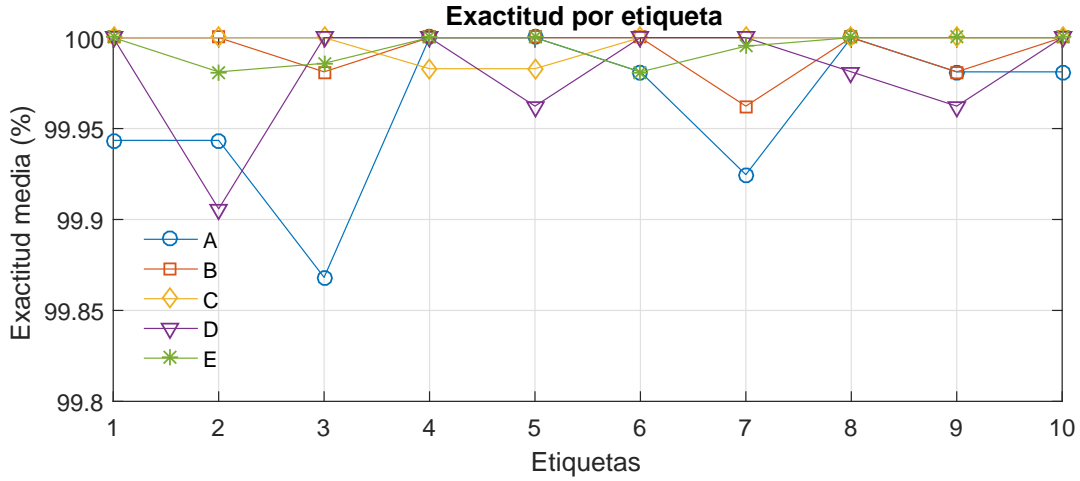


Figura 7: Exactitud por clase en 30 *trials* de los 5 *datasets*.

Con el objetivo de medir la eficiencia del modelo propuesto en este trabajo, se hará la comparación de los resultados obtenidos con dos investigaciones, Rodríguez et al. (2017) [31] y Jia et al. (2016) [8]. El primer autor postula un método de diagnóstico inteligente empleando la transformada wavelet estacionaria, la descomposición en valores singulares de los coeficientes wavelet y una máquina de aprendizaje extremo. El segundo propone un método con la amplitud de los coeficientes de Fourier y una DNN con 3 autoencoders apilados.

Como pudimos observar en los párrafos anteriores, el método propuesto alcanza una exactitud promedio del 99,97% en 30 ejecuciones o *trials*, y una desviación estándar de 0,0862. Mientras que el método postulado por el autor Rodríguez et al. (2017) [31], logra una exactitud del 100% en 50 *trials* con una desviación estándar muy baja equivalente al 0,008. Esta comparación es en base a los resultados del *dataset E* y deja claro que el *deep learning* presenta desventajas frente a un modelo con técnicas de extracción de características más elaboradas y un clasificador con menos capas ocultas.

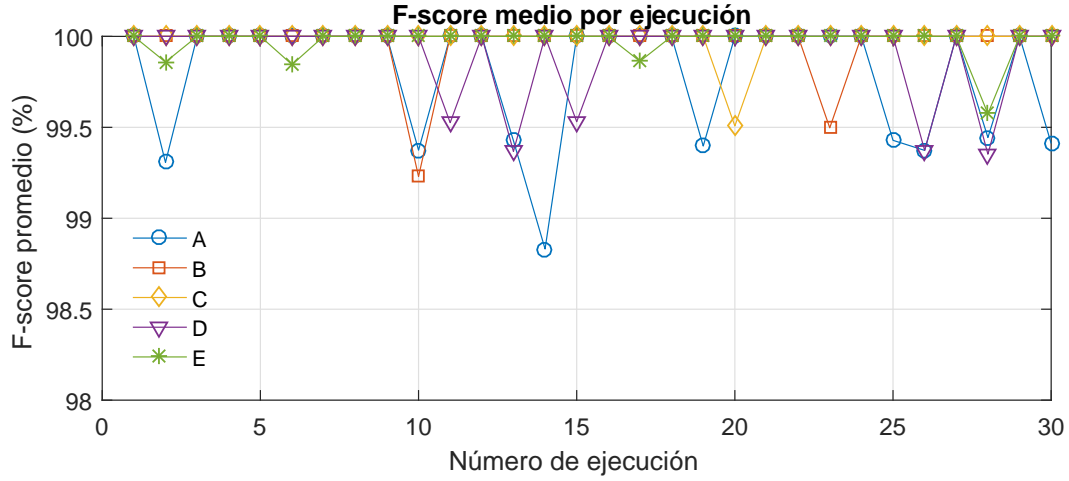


Figura 8: F-score en 30 *trials* de los 5 *datasets*.

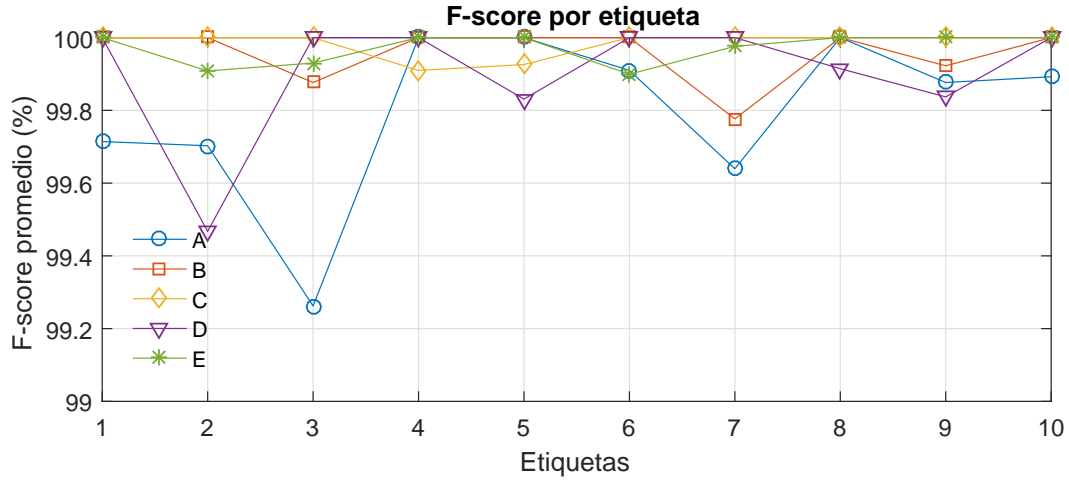


Figura 9: F-score por clase en 30 *trials* de los 5 *datasets*.

Las principales desventajas del modelo propuesto frente a este primer autor, es que no logra igualar la calidad de diagnóstico y la estabilidad de los resultados. La técnica de *deep learning* aplicada en este trabajo requiere grandes tiempos de entrenamiento, mucho mayores que los de la maquina de aprendizaje extremo, y aun así tiene peores resultados. Además, la DNN requiere seleccionar una mayor cantidad de hiperparámetros, los cuales demostraron tener un comportamiento no lineal con respecto a la exactitud del diagnóstico, acrecentando la dificultad de hallar buenos valores para ellos.

El segundo autor Jia et al. (2016) [8], consigue los siguientes resultados para los *datasets* de la A la E: 99,95% 99,61% 99,74% 99,68% y 100% respectivamente en 20 *trials*. Por lo tanto, el método propuesto en este trabajo supera los resultados de este autor en los *dataset* B, C y D en una mayor cantidad de *trials*, además este trabajo logra una mayor estabilidad en los resultados. Sin embargo, la exactitud promedio del *dataset* E en este trabajo no logra igualar a la que alcanza el autor, siendo la diferencia muy pequeña.

A pesar de que los resultados del diagnóstico en los 5 *datasets* no están al nivel del primer autor, para ser un método en el que casi la totalidad del esfuerzo está concentrado en el clasificador, la exactitud lograda en el *dataset* E se aproxima bastante al 100%. La capacidad de diagnóstico del modelo propuesto es satisfactoria, no obstante, la cantidad de parámetros y variables que requieren configuración es una gran desventaja a la que hay que enfrentarse al ocupar técnicas de *deep learning*. El uso del ensayo y error para hallar valores de los hiperparámetros es una tarea agotadora, lo que hace pensar que el *deep learning* requiere mayores avances de tecnología para aplicarse satisfactoriamente en el diagnóstico inteligente de fallas en rodamientos.

6 Conclusión

Al iniciar este trabajo de investigación, el objetivo fue desarrollar un método de diagnóstico inteligente capaz de distinguir severidades de falla en rodamientos bajo velocidades variables. Empleando la transformada de Fourier, implementada con la función FFT, y una técnica de *deep learning* denominada DNN. También fue un objetivo alcanzar resultados parecidos a los de publicaciones que aplicaran técnicas similares y comprender la complejidad del problema subyacente del diagnóstico de fallas en rodamientos.

Investigando sobre el tema del diagnóstico inteligente de fallas en rodamientos y otros elementos mecánicos, me tope con información muy valiosa respecto al fenómeno mecánico detrás de las señales de vibración de un rodamiento. La cantidad de herramientas y técnicas de extracción de características derivadas de la estadística y procesamiento de señales es enorme, por lo que este campo de estudio resulta muy enriquecedor en términos de formación académica. Estudiar un tema tan interesante y complejo como el análisis de vibraciones y populares técnicas de *deep learning*, efectúa un cambio de paradigma en un alumno de pregrado.

Para cumplir con mi objetivo en esta investigación, fue necesario comenzar a estudiar acerca de las necesidades del mantenimiento moderno y de cómo la ingeniería ha propuesto técnicas de *machine learning* para automatizar procesos que hasta hoy en día se hacen a mano. También tuve que entender cómo surgió la aplicación de técnicas de *representation learning*, ampliamente usadas en el reconocimiento de imágenes, en nuevos enfoques de *deep learning* empleados en el diagnóstico de fallas de elementos mecánicos. Además, tuve que aprender a usar y calibrar las funciones del *toolkit* de MATLAB que permiten crear la red con *sparse* autoencoders apilados.

En este trabajo de investigación corroboré la aplicación de técnicas de *deep learning*, como las DNN con SAE's apilados, en problemas de diagnóstico de fallas de rodamientos. La exactitud promedio de 30 *trials* del *dataset* E fue de un 99,97% y la desviación estándar un 0,0862, lo que indica una buena capacidad de generalización de clasificador y resultados relativamente estables. Sin embargo, los resultados del método propuesto están lejos de acercarse a los de algunas investigaciones como la de Rodríguez et al. (2017) [31], en términos de tiempo de ejecución y homogeneidad de los resultados.

Respecto a las limitaciones de este trabajo, las principales dificultades estuvieron relacionadas con la configuración del clasificador. Esto se refiere a hallar valores de hiperparámetros y

topologías de la red con las que fuera mejorando paulatinamente el desempeño de la red profunda. Otra dificultad no menor, fue comprender la naturaleza subyacente de las vibraciones mecánicas de los rodamientos, y de cómo esta se relaciona con las señales de vibración que son medidas por los sensores.

En una próxima investigación me gustaría aprender más acerca del fenómeno físico detrás de las vibraciones mecánicas. Lo que me permitiría probar algún método de diagnóstico inteligente una técnica de procesamiento de señales mas avanzada y alguna herramienta de *machine learning* que parezca interesante. Por ahora, el *deep learning* queda descartado de nuevas investigaciones personales de problemas de diagnostico de fallas, debido a sus extenuante proceso de configuración.

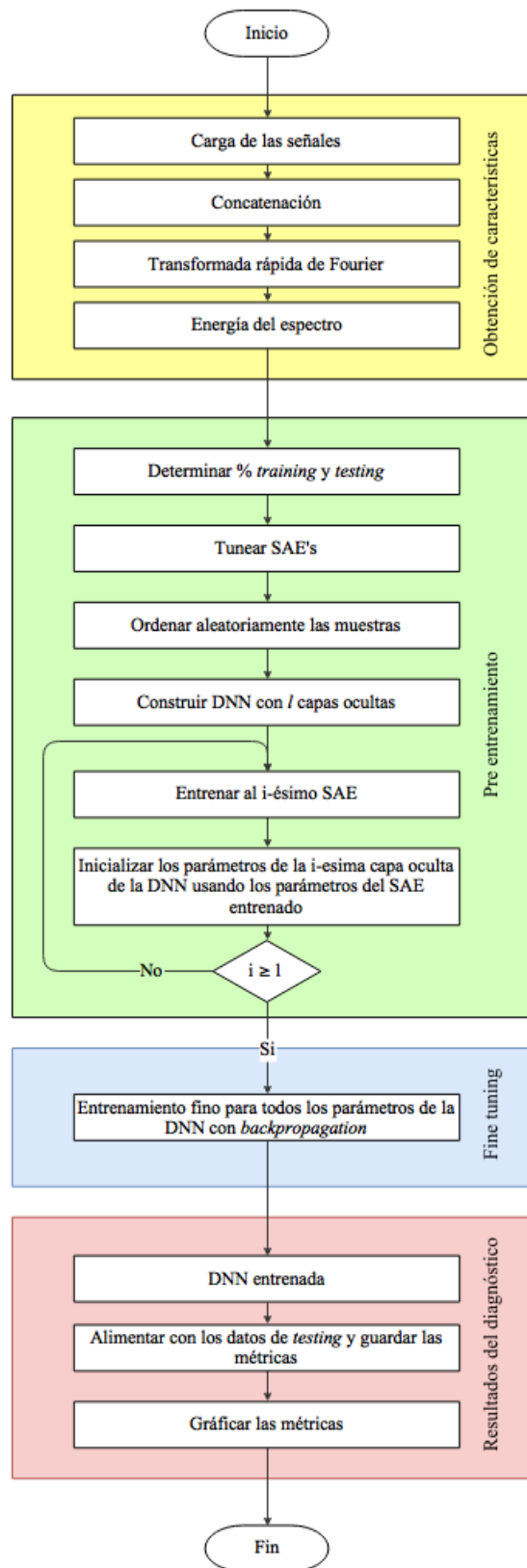


Figura 10: Diagrama de flujo de los procesos que se llevan a cabo para entrenar y probar el modelo propuesto.

Referencias

- [1] Manjeevan Seeraa, M.L. Dennis Wonga & Asoke K. Nandic (2017). Classification of ball bearing faults using a hybrid intelligent model. *Applied Soft Computing*, 57, 427–435.
- [2] Yimin Zhan & Chris K. Mechefske (2007). Robust detection of gearbox deterioration using compromised autoregressive modeling and Kolmogorov–Smirnov test statistic—Part I: Compromised autoregressive modeling with the aid of hypothesis tests and simulation analysis. *Mechanical Systems and Signal Processing*, 21, 1953–1982.
- [3] Issam Attoui, Nadir Fergani, Nadir Boutasseta, Brahim Oudjani & Adel Deliou (2017). A new time–frequency method for identification and classification of ball bearing faults. *Journal of Sound and Vibration*, 397, 241–265.
- [4] Yunxiao Fu, Limin Jia, Yong Qin & Jie Yang (2016). Product Function Correntropy and Its Application in Rolling bearing Fault Identification. *Measurement*, 88–99.
- [5] Wei Li, Zhencai Zhu, Fan Jiang, Gongbo Zhou & Guoan Chen (2015). Fault diagnosis of rotating machinery with a novel statistical feature extraction and evaluation method. *Mechanical Systems and Signal Processing*, 50–51, 414–426.
- [6] Hong Guo, Lindsay B. Jack & Asoke K. Nandi (2005). Feature Generation Using Genetic Programming With Application to Fault Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(1), 89–99.
- [7] Xiaoran Zhu, Youyun Zhang & Yongsheng Zhu (2012). Intelligent fault diagnosis of rolling bearing based on kernel neighborhood rough sets and statistical features. *Journal of Mechanical Science and Technology*, 26(9), 2649–2657.
- [8] Feng Jia, YaguoLei, Jing Lin, Xin Zhou & Na Lu (2015). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, 72–73, 303–315.
- [9] Yang Yu, YuDejie & Cheng Junsheng (2006). A roller bearing fault diagnosis method based on EMD energy entropy and ANN. *Journal of Sound and Vibration*, 294, 269–277.
- [10] Changting Wang & Robert X. Gao (2002). Wavelet Transform with Spectral Post-Processing for Enhanced Feature Extraction. *IEEE Instrumentation and Measurement Technology Conference*, 52(4), 1296–1301.
- [11] Ran Zhang, Zhen Peng, Lifeng Wu, Beibei Yao & Yong Guan (2017). Fault Diagnosis from Raw Sensor Data Using Deep Neural Networks Considering Temporal Coherence. *Sensors*, 17(3), 549–566.
- [12] Jaouher Ben Ali, Nader Fnaiech, Lotfi Saidi, Brigitte Chebel-Morello & Farhat Fnaiech (2015). Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Applied Acoustics*, 89, 16–27.
- [13] Lifeng Wu, Beibei Yao, Zhen Peng & Yong Guan (2017). Fault Diagnosis of Roller Bearings Based on a Wavelet Neural Network and Manifold Learning. *Applied Science*, 7, 158–168.

- [14] V.K. Rai & A.R. Mohanty (2007). Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert–Huang transform. *Mechanical Systems and Signal Processing*, 21, 2607–2615.
- [15] Wheeler, Anthony J. & Ganji, Ahmad R. (2009). *Introduction to Engineering Experimentation* (3rd ed., pp. 107-117). New Jersey: Prentice Hall.
- [16] Gray, Robert M. & Goodman, Joseph W. (1995). *Fourier Transforms: An Introduction for Engineers* (1st ed., pp. 53-57). New York: Springer.
- [17] Shao Haidong, Jiang Hongkai, Zhao Huiwei & Wang Fuan (2017). A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. *Mechanical Systems and Signal Processing*, 95, 187-204.
- [18] Song-Zhi Su, Zhi-Hui Liu, Su-Ping Xu, Shao-Zi Li & Rongrong Ji (2014). Sparse auto-encoder based feature learning for humanbody detection in depth image. *Signal Processing*, 112, 43-52.
- [19] Yufeng Zhang, Zhenyu Guo, Weilian Wang, Side He, Ting Lee & Murray Loew (2003). A comparison of the wavelet and short-time fourier transforms for Doppler spectral analysis. *Medical Engineering & Physics*, 25, 547-557.
- [20] Aurelien Prudhom, Jose Antonino-Daviu, Hubert Razik & Vicente Climente-Alarcon (2015). Time-frequency vibration analysis for the detection of motor damages caused by bearing currents. *Mechanical Systems and Signal Processing*, 84, 747-762.
- [21] Zhiqiang Chen, Shengcai Deng, Xudong Chen, Chuan Li, René-Vinicio Sanchez & Huafeng Qin (2017). Deep neural networks-based rolling bearing fault diagnosis. *Microelectronics Reliability*, 75, 327-333.
- [22] P. Konar & P. Chattopadhyay (2011). Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs). *Applied Soft Computing*, 11, 4203-4211.
- [23] Wentao Mao, Jianliang He, Yuan Li & Yunju Yan (2017). Bearing fault diagnosis with auto-encoder extreme learning machine: A comparative study. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(8), 1560-1578.
- [24] Kanpaj Gupta & M.K. Pradhan (2017). Fault detection analysis in rolling element bearing: A review, *Materials Today: Proceedings*, 4, 2085-2094.
- [25] Witten, Ian H. & Frank, Eibe & Hall, Mark A. & Pal, Christopher J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques-Morgan Kaufmann* (4th ed., pp.), Morgan Kaufmann.
- [26] Haykin, Simon. (2009). *Neural Networks and Learning Machines* (3rd ed., pp.), Pearson.
- [27] Marc’Aurelio Ranzato, Y-Lan Boureau & Yann LeCun (2008). Sparse feature learning for deep belief networks. *Advances in Neural Information Processing Systems*, 20, 1185–1192.

- [28] Erzhu Li, Peijun Du, Alim Samat, Member, Yaping Meng & Meiqin Che (2017). Mid-Level Feature Representation via Sparse Autoencoder for Remotely Sensed Scene Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 10, 1068–1081.
- [29] Case Western Reserve University Bearing Data Center, *Bearing vibration data set*, <http://csegroups.case.edu/bearingdatacenter/pages/12k-drive-end-bearing-fault-data>, (visitado el 30-08-2017).
- [30] Jont B. Allen & Lawrence R. Rabiner (1977). A Unified Approach to Short-Time Fourier Analysis and Synthesis. *Proceedings Of The IEEE*, 65(11), 1548-1564.
- [31] Nibaldo Rodriguez, Carolina Lagos, Enrique Cabrera & Lucio Cañete (2017). Extreme Learning Machine Based on Stationary Wavelet Singular Values for Bearing Failure Diagnosis. *Studies in Informatics and Control*, 26(3), 287-294.