

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

**RED NEURONAL SIGMOIDAL CON VALORES
SINGULARES PARA LA SEVERIDAD DE FALLOS
EN EQUIPOS DE ROTACIÓN**

Antonio Alejandro Barría Lazo

Profesor Guía: Nivaldo Rodríguez Agurto.

Profesor Co-referente: Wenceslao Palma.

Carrera: Ingeniería Civil en Informática.

Diciembre de 2016

Índice

Dedicatoria.....	iv
Resumen	v
Lista de Figuras	vi
Lista de Tablas.....	vii
Lista de Siglas.....	viii
1 Introducción.....	1
2 Descripción del problema.....	2
3 Definición de Objetivos.....	5
3.1 Objetivos Generales.....	5
3.2 Objetivos Específicos	5
4 Estado del arte	6
4.1 Redes Neuronales Artificiales	6
4.1.1 Historia de las Redes Neuronales Artificiales.....	8
4.2 Descomposición de Valores Singulares	8
4.3 Trabajos Relacionados.....	9
5 Marco Teórico	11
5.1 Algoritmo de Levenberg-Marquardt	11
5.2 Función de Transferencia	12
5.3 Descomposición de Valores Singulares	13
6 Análisis de los resultados	14
6.1 Muestras Utilizadas	14
6.2 Características.....	14
6.3 Configuración de los datos	15
6.4 Configuración de la red neuronal	17
6.5 Métricas de Rendimiento.....	18
6.6 Modelos	19
6.6.1 Modelo 1	19
6.6.2 Modelo 2	21
6.6.3 Modelo 3	21

6.6.4	Modelo 4	22
6.6.5	Modelo 5	24
6.6.6	Modelo 6	25
8	Conclusión.....	28
9	Referencias	29

Dedicatoria

A mi madre, a mi padre y a mi familia, quienes me convirtieron en quien soy hoy en día, me brindaron apoyo incondicional en todo momento y fueron mi principal fuente de motivación para finalizar mi carrera.

A mis amigos, que estuvieron conmigo en los buenos y malos momentos.

Resumen

Los rodamientos alojados en un motor industrial presentan una gran frecuencia de fallo, por lo que resulta de vital importancia monitorear la condición de la máquina para detectar fallas prematuramente. Una de las ventajas de estas detecciones es que evitarían que el equipo se dañe gravemente y, por otro lado, disminuiría el tiempo dedicado a mantener la máquina, aumentando así su producción. Para lograr esto, se propone un modelo para diagnosticar diez condiciones de salud de un rodamiento en un equipo de rotación. Este modelo fue construido en dos etapas, la primera realiza una extracción de características combinada con la descomposición de valores singulares, que representan el comportamiento del rodamiento. La segunda etapa consiste en un clasificador basado en redes neuronales artificiales, el cual recibe como entrada cinco características (Kurtosis, Raíz valor cuadrático medio, Factor forma, Valor máximo de peak y Entropía) y posee diez salidas que representan las diez condiciones de salud del rodamiento. El modelo propuesto es evaluado desde una base de datos pública de la Case Western Reserve University. Los resultados obtenidos por la mejor topología encontrada (ANN(5, 6, 10)), logran una exactitud promedio de diagnóstico de 99.41%.

Bearings housed in an industrial motor have a high frequency of failure, so it is vital to monitor the condition of the machine to detect faults prematurely. One of the advantages of these detections is that to avoid that the equipment is seriously damaged and, on the other hand, it would reduce the time dedicated to maintain the machine, thus increasing its production. To achieve this, a model is proposed to diagnose ten health conditions of a bearing in rotating equipment. This model was built in two stages, the first one performs a feature extraction combined with the Singular Values Decomposition, which represents the behavior of the bearing. The second stage consists in a classifier based on Artificial Neural Networks, which receives as input five characteristics (Kurtosis, Mean root mean square, Form Factor, Maximum Peak Value and Entropy) and has ten outputs representing the ten health conditions of bearing. The proposed model is evaluated from a public database of Case Western Reserve University. The results obtained by the best topology found (ANN (5, 6, 10)), obtained an average diagnostic accuracy of 99.41%.

Lista de Figuras

Figura 1 Partes de un rodamiento	3
Figura 2 Ciclo de vida de un rodamiento.	4
Figura 3 Estructura de una neurona.....	6
Figura 4 Sinapsis	7
Figura 5 Modelo de Red Neuronal	7
Figura 6 Proceso de carga de datos 1	15
Figura 7 Proceso de carga de datos 2	16
Figura 8 Gráfico Error vs Iteraciones	17
Figura 9 Gráfico Error vs Nodos ocultos.....	18
Figura 10 Red neuronal del modelo 1	20
Figura 11 Estructura del árbol de decisión	20
Figura 12 Red neuronal del Modelo 2	21
Figura 13 Red neuronal del Modelo 3	22
Figura 14 Red neuronal del Modelo 4.....	23
Figura 15 Red neuronal del Modelo	24
Figura 16 Red neuronal del Modelo 6.....	25
Figura 17 Matriz de confusión	26

Lista de Tablas

Tabla 1 Resultados de Trabajos relacionados.....	10
Tabla 2 Resultados Modelo 1	20
Tabla 3 Resultados Modelo 2	21
Tabla 4 Resultados Modelo 3	22
Tabla 5 Resultados Modelo 4	23
Tabla 6 Resultados Modelo 5	24
Tabla 7 Resultados Modelo 6	25
Tabla 8 Pruebas de robustez	27

Lista de Siglas

ADE	: Evolución diferencial auto adaptativa
ANN	: Artificial Neural Network (o en español RNA)
ECG	: Electrocardiograma
EMD	: Descomposición modal empírica
FN	: Falso Negativo
FP	: Falso Positivo
HSSMC	: Hyper-sphere-structured multi-class
IMF	: Función Modal Intrínseca
KJADE	: Kernel joint Approximate diagonalization of eigen-matrices
KNN	: K-Nearest Neighbors
LGPCA	: Local and global principal component analysis
LS	: Laplace Score
MOBDE	: Multi-Objective Binary Differential Evolution
MFE	: Multi-Scale Fuzzy Entropy
NM	: Nearest Mean
NN	: Nearest Neighbors
PNN	: Probabilistic Neuronal Network
SAX	: Symbolic Aggregate ApproXimation
SVD	: Singular Value Decomposition (o en español DVS)
SVM	: Support Vector Machine
VN	: Verdadero Positivo
VP	: Verdadero Negativo
TR-LDA	: Criterio de rastreo LDA
VPMCD	: Variable predictive model based class discrimination

1 Introducción

En la actualidad, la sociedad se encuentra en un estado de comodidad gracias a los avances tecnológicos, los cuales facilitan y automatizan innumerables tareas de la vida cotidiana. Dichos avances se encuentran en la mayoría de las situaciones o momentos del día a día, a tal punto que la ausencia de ellos generaría un cambio drástico en la vida de las personas. Estos cambios tecnológicos comenzaron el año 1832, con la aparición de los convertidores electromagnéticos de energía (máquinas eléctricas que transformaban energía mecánica en eléctrica y viceversa), los que hoy se conocen como máquinas rotacionales [1].

Estas máquinas rotacionales se dividen en dos tipos, motor y generador. La primera máquina y en la cual se centra este documento, se caracteriza por transformar desde cualquier energía a energía mecánica. La segunda, se basa en la transformación de cualquier energía a energía eléctrica [1]. Por ejemplo, se puede mencionar que la primera es usada en electrodomésticos, automóviles, máquinas de trabajo forzado, aviones, etc. Mientras que la segunda es utilizada en, aerogeneradores, centrales hidroeléctricas, reactores nucleares, entre otros.

La principal falla encontrada en los motores es básicamente causada por los rodamientos [2], esto sucede debido a que sobre esta pieza descansa todo el peso del motor y toda la fuerza que este desencadena. Un fallo no detectado en un rodamiento puede generar, con cierta rapidez, una avería aún más grave, poniendo en riesgo la vida útil del motor, la productividad de la máquina y en algunos casos las vidas de las personas. Esto puede llevar a pérdidas monetarias por el no funcionamiento del equipo [3,5], lo que significa por un lado, no producir y por otro lado un gasto en repuestos y mantención. En casos más graves como una falla de una o más turbinas de un avión en pleno vuelo o en el motor de un barco en plena tormenta, puede significar la muerte de cientos de personas [4,5].

Con el fin de extender la vida útil del motor y por tanto la vida útil de la máquina, es de vital importancia realizar un mantenimiento al equipo. Para ello, lo que se pretende abordar en este trabajo es la implementación de un modelo capaz diagnosticar la severidad de una falla en un rodamiento, permitiendo saber el nivel de daño alojado en el rodamiento y dónde este se encuentra. En esta área existen varios métodos para realizar esta tarea, entre ellos están: extracción de características, reducción de características, reducción de ruido y clasificadores. En la fase de extracción de características, estas pueden pertenecer al dominio del tiempo [8, 13, 21], dominio de la frecuencia [12], dominio del tiempo-frecuencia [20, 22] y algunas combinaciones entre estas [6, 24]. Luego opcionalmente puede ir una etapa de reducción de características, por ejemplo LS [27], KJADE [28] y MOBDE [29]. De la misma manera una reducción de ruido, como SVD [8, 12, 22], Wavelet De-Noiseing [23], Matriz de Hankel [6, 9, 13]. Y finalmente un clasificador, que puede ser KNN [28], RNA [20], SVM [12, 22], entre otros. De estos métodos, los que se usarán en esta tesis serán una extracción de características en el dominio del tiempo en conjunto con SVD y como clasificador una RNA.

Este documento está dividido en secciones, comenzando con una descripción del problema. Luego de esto, se presentarán las herramientas y los conocimientos matemáticos necesarios para abordar el problema. Posteriormente, se verán los modelos implementados junto con análisis de sus resultados. Finalmente, como conclusión se mostrarán los principales logros, ventajas, desventajas encontrados en este trabajo.

2 Descripción del problema

La industria minera en Chile forma una parte importante del país, ya que aporta un 11,2 del PIB nacional, en esta industria el activo más importante es la maquinaria pesada. En esta maquinaria, la pieza que presenta mayor frecuencia de fallos es el motor, es por esto que se gastan millones de dólares en paradas programadas y no programadas para lograr realizar una mantención al equipo. Estas paradas corresponden a un cese de funciones por parte del equipo para poder repararlo. En donde la programada es realizada cada cierto tiempo, que puede ser por ejemplo un mes, dos meses, un año, etc., mientras que la no programada surge ante la emergencia de reparar un fallo en el instante que fue encontrado. Las paradas tienen como costos, la mano de obra asociada a la mantención y el tiempo que este equipo está fuera de la producción. Es por esto, que la investigación se centrará en la observación de un motor para minimizar estas detenciones.

Un motor, así como toda máquina rotacional, está constituido por ciertas partes que son de vital importancia para su funcionamiento. Entre ellas se encuentran el rotor, el estator, las escobillas, la culata, el entrehierro y los rodamientos. De estas piezas la que posee mayor frecuencia de fallo en los equipos de rotación, son los rodamientos, llegando a ocupar entre el 40% y 50% de las fallas alojadas en un motor [6]. Es por esto, que esta pequeña pieza pasa a ser una parte crítica para el motor. Ya que, además de soportar la exigencia del motor, debe soportar los esfuerzos dinámicos que se originan como consecuencia de desequilibrios, desalineaciones o excentricidades.

Debido a los constantes problemas relacionados con los rodamientos o a problemas con otras partes de la máquina, el motor debe someterse a un proceso de mantenimiento, el cual tiene como objetivo extender la vida útil de la máquina o simplemente restaurarla para que pueda volver a ejercer su función. Existen muchos tipos de mantenimiento pero los más comunes son:

- **Mantenimiento correctivo:** Surge para corregir averías o fallas observadas, este mantenimiento puede ser realizado de inmediato generando una parada no programada o puede esperar a que la máquina ejerza una parada programada.
- **Mantenimiento preventivo:** Inicia con el objetivo de prevenir fallos, ya que si un fallo no se detecta a tiempo puede generar un problema mayor en el motor. Este mantenimiento consta de tres etapas:
 - **Detección:** Es la primera etapa a realizar y corresponde al simple hecho de descubrir que existe una condición de falla.
 - **Diagnóstico:** Luego de detectar una condición de falla, esta fase se empeña en averiguar la gravedad de la falla encontrada.
 - **Pronóstico:** El último en realizarse, apunta a estimar cuanto tiempo de vida útil le queda al rodamiento, esta fase sólo sería útil realizarla cuando efectivamente se halló una falla en estado crítico.

De estos mantenimientos el más utilizado es el correctivo, dejando de lado al preventivo. Pero lo que no se sabe, es que este último es el más importante, mientras que el correctivo continúa dañando el equipo luego de que haya ocurrido el fallo, el preventivo evita que el equipo siga dañándose. Esto sucede porque desde que ocurrió el fallo, se detectó y se detuvo la máquina para realizar el mantenimiento correctivo puede transcurrir un tiempo considerablemente largo. Además para poder controlar este fallo, el equipo debe

realizar una parada no programada, lo que provoca que este no pueda trabajar y deje de generar utilidades. Esto se evita con el mantenimiento preventivo, ya que este mantenimiento primero que nada evita paradas en donde no existen fallas, por lo que disminuye la cantidad de paradas programadas y no programadas.

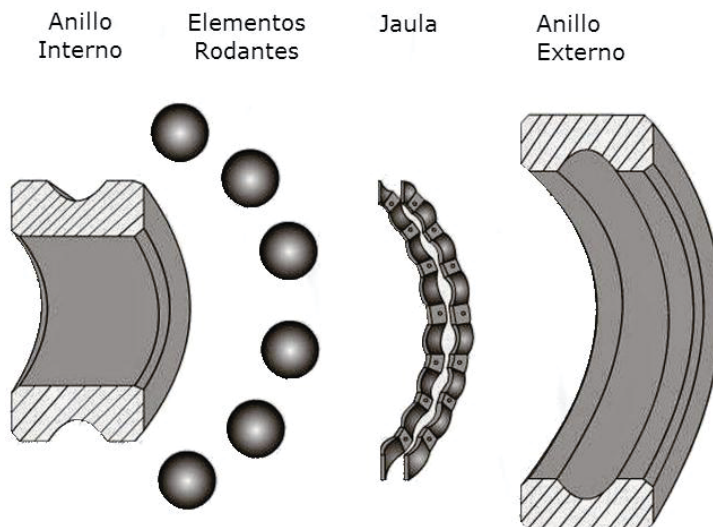


Figura 1 Partes de un rodamiento

Es por esto, que para este proyecto se pretende implementar una herramienta que ayude a realizar un mantenimiento preventivo, enfocado en los rodamientos debido al gran porcentaje de fallos encontrados en una máquina. Este rodamiento está constituido por cuatro partes según se muestra en la Fig. 1: anillo interno, anillo externo, elementos rodantes (bolas) y jaula. Un pequeño defecto en cualquiera de estas partes produce una vibración característica que se puede detectar en una señal de vibración. La señal depende de diversos factores los cuales son: la velocidad de giro y los parámetros geométricos del rodamiento (diámetros primitivo, exterior e interior, diámetro de los elementos rodantes, número de elementos rodantes y ángulo de contacto entre las pistas y los elementos rodantes).

Además de estar constituido por diversas partes, el rodamiento posee un característico ciclo de vida, el cual posee 4 etapas (como se puede apreciar en la Fig. 2). La primera etapa abarca el 90% de su vida útil donde se encuentra en un estado normal. Luego avanza al 5% en donde se detectan fallos leves, luego de que ocurren los fallos, empieza una reacción en cadena detonando más fallos, como se observa en la siguiente etapa en donde los fallos crecen de manera exponencial. En el 5% restante comienzan a ocurrir fallos moderados y fallos graves, de manera casi inmediata y con una frecuencia bastante alta. Por último, dada la cantidad de fallos acumulados es donde el rodamiento termina su vida útil y debe ser reemplazado lo más pronto posible.

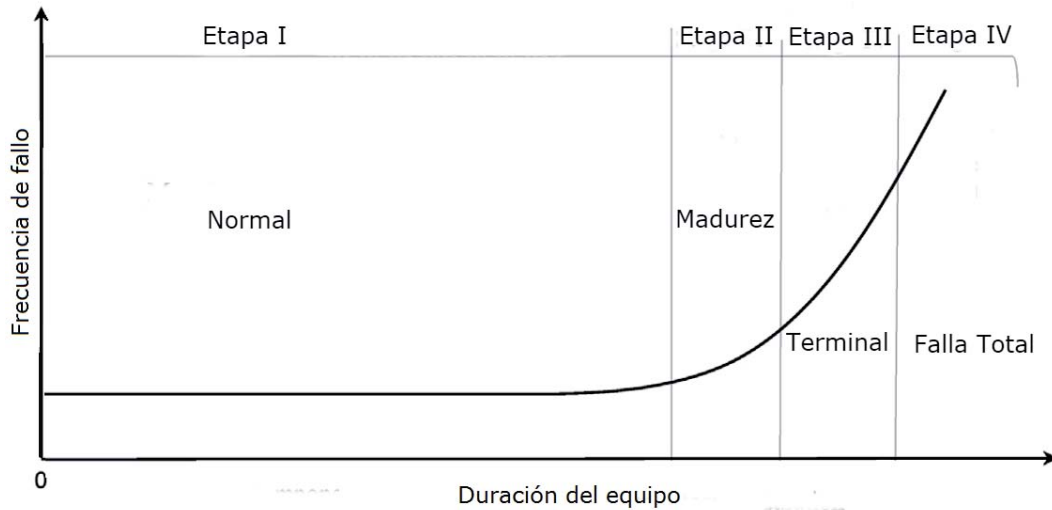


Figura 2 Ciclo de vida de un rodamiento.

En base esto, para lograr un buen mantenimiento preventivo es que la máquina debe ser parada no al momento de encontrar un fallo (Etapa II), ya que esto ocasionaría que la máquina parara una mayor cantidad de veces que el óptimo, pero manteniendo el equipo intacto. Por el contrario, algo similar pero contrario, ocurriría si la máquina se detuviera en una falla grave (Etapa IV), habría una disminución de paradas pero el deterioro del equipo sería mayor. Es por esto, que el mejor momento para realizar una parada es cuando el fallo que se encuentra es de carácter moderado (Etapa III), lo que ofrece un equilibrio entre paradas y deterioro de la máquina.

Para llevar a cabo esta tarea, se utilizará una base de datos de la Case Western Reserve University, la cual consta de dos lecturas de rodamiento, uno es el Drive-End y otro es el Fan-End. Estas lecturas son captadas a través de unos instrumentos llamados acelerómetros, capaces de medir la vibración producida en los rodamientos. Estos rodamientos fueron probados con diferentes especificaciones, entre ellas, cuatro diferentes velocidades, cuatro diferentes diámetros de falla y en cuatro distintas condiciones del rodamiento, estas son detalladas a continuación:

- Velocidades del rodamiento: 1979 RPM – 1972 RPM – 1950 RPM – 1930 RPM
- Diámetros de falla: 0.007' – 0.014' – 0.021' – 0.028'
- Condiciones del rodamiento: Normal – Fallo Interno – Fallo Externo – Fallo bola.

De esta base de datos, se ocuparán las mediciones del rodamiento Drive-End, los tres diámetros de falla más pequeños, las cuatro condiciones del rodamiento y las cuatro velocidades del rodamiento.

3 Definición de Objetivos

3.1 Objetivos Generales

Desarrollar un algoritmo de extracción de características integrado con Redes Neuronales Artificiales para mejorar la exactitud de diagnóstico de fallo de rodamiento bajo condición de operación de velocidad variable.

3.2 Objetivos Específicos

- Extraer características desde la señal de vibración utilizando el método de Descomposición de Valores Singulares.
- Diseñar e implementar la topología de un clasificador neuronal usando algoritmo de aprendizaje de Levenberg-Marquardt.
- Evaluar el desempeño del modelo propuesto y comparar su rendimiento con modelos publicados en revistas científicas.

4 Estado del arte

Para abordar la problemática de este trabajo y comprender como se trabajará, se deben dar a conocer las herramientas indispensables en este estudio. Para ello se comenzará con una explicación básica de estas técnicas y luego se mostrará un pequeño repaso de su evolución a través del tiempo.

4.1 Redes Neuronales Artificiales

Las redes neuronales artificiales (RNA) nacen de un intento de emular parcialmente la forma en que el sistema nervioso procesa la información, es decir, tratan de imitar el funcionamiento del cerebro a través de una computadora. Sin embargo, las tareas realizadas por un cerebro humano no se comparan a las que puede realizar una computadora, aunque hoy en día estas sean extremadamente rápidas. Esto ocurre principalmente porque el cerebro es un sistema altamente complejo y puede realizar tareas simultáneamente, mientras que para una computadora esto resulta imposible, ya que solo puede ejecutar tareas secuencialmente. No obstante, unos de los puntos a favor es que las RNA de igual manera que las redes neuronales biológicas pueden adquirir conocimientos a través de la experiencia (ensayo y error), esta es la característica que pasa a ser la más primordial de las RNA. Otro punto es que ambas son tolerables a fallos, es decir que se puede obtener un buen resultado aunque hayan sufrido un daño considerable.

La red neuronal biológica se compone de varios componentes, pero para objeto de este estudio, solo se analizarán tres partes: las dendritas, el axón y los botones terminales (como se observa en la Fig. 3). En donde la primera corresponde a la unidad encargada de recibir el estímulo que viene hacia la neurona, la segunda viene siendo el canal por el cual se propaga el pulso, y el último como lo dice su nombre vendría siendo la parte final en donde el mensaje se puede comunicar hacia otra neuronas.

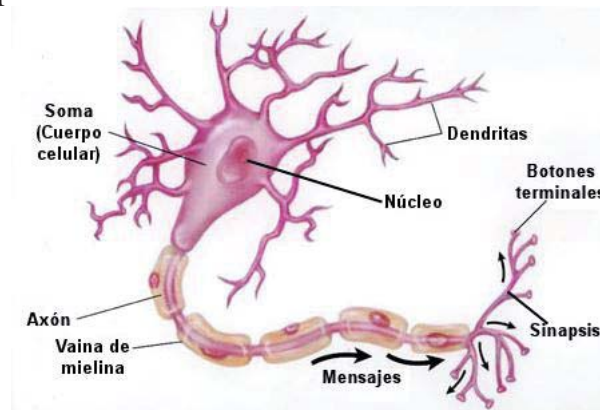


Figura 3 Estructura de una neurona

El fenómeno conocido como sinapsis ocurre cuando se intercomunican dos neuronas, en este proceso el botón sináptico (botón terminal) se comunica con la dendrita de la segunda neurona. A este tipo de sinapsis se le denomina sinapsis química, en donde el botón sináptico hace que las vesículas sinápticas (manchas azules de la Fig. 4) se rompan, liberando una sustancia llamada neurotransmisor. Esta sustancia química se esparce por el

espacio entre las neuronas hasta que es captada por las dendritas generando un nuevo impulso eléctrico que vuelve a hacer el proceso de nuevo.

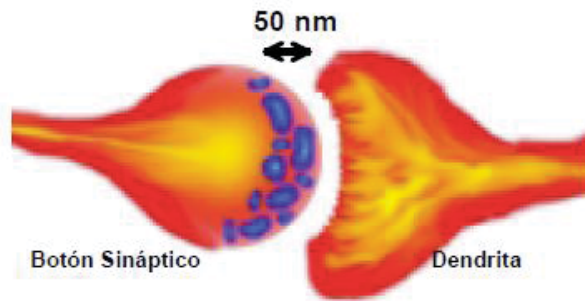


Figura 4 Sinapsis

Adicionalmente se pueden mencionar los siguientes comportamientos de una neurona durante la sinapsis:

- El impulso que llega a una neurona y el que sale no necesariamente tienen que ser iguales, esto depende de la sensibilidad del neurotransmisor que va cambiando constantemente durante el aprendizaje. Dado esto, una sinapsis podría perfectamente debilitar o aumentar el pulso de la neurona.
- En el soma de la neurona se van sumando los distintos impulsos que llegan a esta, de tal manera que si esta sumatoria sobrepasará un umbral, entonces el pulso sería transmitido hacia el axón para que pudiera ser traspasado hacia otra neurona. En caso de que esto no ocurriera simplemente no se transmitirá.

Las características anteriormente mencionadas son esenciales para la construcción del modelo de las RNA. Para explicar este método tomaremos como base la Fig. 5, en la cual podemos notar una similitud con una red neuronal biológica, en donde tenemos las entradas (dendritas), el soma en donde se realiza la sumatoria de los estímulos de entrada y donde posteriormente, dependiendo de su resultado se generan las salidas (botón terminal).

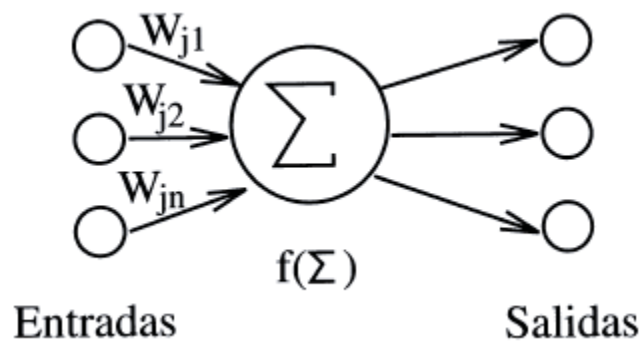


Figura 5 Modelo de Red Neuronal

Sin embargo, para poder generar las salidas se requiere que la sumatoria de las redes supere un cierto umbral, en un cerebro humano esto ya viene construido, pero como implementar esto en un computador. Es por esto, que a las RNA se les debe agregar lo que se llama función de activación (o transferencia), esta le sirve a la red como guía para determinar su umbral y activarlo según sea correspondiente.

Como se mencionó anteriormente las RNA son capaces de adquirir conocimientos a través de la experiencia, pero cómo realiza esta tarea. Las RNA poseen los denominados pesos sinápticos que son los encargados de otorgarle valor a los estímulos que llegan a la red, el valor dependerá de sí el estímulo es relevante para el proceso o si no lo es tanto. Para que la red pueda aprender, primero se le debe enseñar. Este proceso parte con el entrenamiento de la red, en donde se le muestran las entradas y las salidas respectivas, para que vaya asociando las respuestas. Aquí es donde juega un papel importante la técnica del ensayo y error, puesto que si a la red se le envía una entrada y esta no corresponde con la salida, la red puede calibrar los pesos sinápticos de tal manera que llegará a la salida adecuada luego de repetir n-veces este proceso [14].

Con el objetivo de explicar las aplicaciones de las RNA, se debe entender el hecho de que existen dos tipos de red: predicción y clasificación [14]. La primera como lo dice su nombre, se usa para predecir datos, es decir la red se empeña en asociar un patrón, en base a información parcial de este, y luego de adaptarse, la red podría predecir un tiempo futuro de este patrón. Algunos ejemplos de este tipo de red son: la predicción de radiación solar como potencial de energía, predicción del clima global o regional, predicción del valor del euro o dólar, entre otros. La segunda y la que se usa en este trabajo como un diagnosticador, es una red capaz de clasificar datos, esto lo hace adoptando algunas formas del criterio de interpolación o aproximación, que le permiten poder discriminar los datos en distintos tipos. Algunas aplicaciones de esta red son: reconocimiento de voz, reconocimiento de problemas cardiacos (arritmia, bloqueos [15]), etc.

4.1.1 Historia de las Redes Neuronales Artificiales

Los inicios de las RNA datan desde 1943, en donde Warren Mc Coulloc (psiquiatra y neuroanatomista) y Walter Pitts (matemático) desarrollaron el primer modelo de red neuronal biológica, aun cuando no existían los computadores. Este modelo era capaz de resolver problemas de cálculo mediante circuitos lógicos. Sin embargo, el siguiente acontecimiento sucede en el año 1958, donde la primera RNA fue introducida por Frank Rosenblatt, la cual fue conocida como perceptron. Este modelo es el más básico entre las redes, ya que solo posee dos capas (entrada y salida). Luego en 1960, Widrow y Hoff construyeron a ADALINE, una RNA basada en el perceptron pero con una mejora en el balanceo de pesos, lo que le permitía estimar la salida de una manera más exacta. En 1968, Misky y Papert, se dieron cuenta de que ADALINE no era capaz de resolver problemas no lineales, por lo que aquí nace la primera red con perceptron multicapa. En 1974 Werbos, introduce el famoso algoritmo de *backpropagation*. Luego en 1986, nace la primera RNA de perceptron multicapa incluida con el algoritmo de *backpropagation* a manos de Rummelhart y Mclelland.

4.2 Descomposición de Valores Singulares

Para lograr una mayor precisión en este estudio se utilizará la Descomposición de Valores Singulares (DVS), este método pretende eliminar de la señal el ruido de fondo. El ruido de fondo puede ser causado ya sea por impulsos aleatorios, por la captación de la vibración de otro rodamiento, alguna falla en otra parte del sistema o por agentes externos

al sistema. Esto provoca que no se tenga una señal limpia del elemento a analizar, por lo que entorpece el diagnóstico. La DVS dividirá la señal actual en dos tipos de señales, la señal limpia y la señal de ruido, por lo que se espera que los resultados mejoren considerablemente luego de utilizar esta técnica.

Este método numérico se enfoca en descomponer una matriz como el producto de tres matrices, una ortogonal, una diagonal y una ortogonal traspuesta. Esta matriz diagonal será la que contenga las raíces cuadradas de los valores propios (valores singulares). Esta matriz contiene los valores singulares ordenados, de tal forma que los n -primeros resultan ser los correspondientes a la señal limpia y los restantes a la señal de ruido. Para separar esta señal, se toma entre un 50% y 80% de los primeros valores singulares, dividiendo la señal en dos partes. Posteriormente a esto, se procede a reconstruir la señal usando el valor de n , dejando una señal libre de contaminación externa, para que luego puedan ser inyectados a la red neuronal [6].

Si bien esta técnica es usada para un diagnóstico de rodamientos de un motor en base a una señal de vibración, no quiere decir que no se pueda aplicar a otro tipo de problema. De hecho, puede ser aplicada a cualquier tipo de señal, ya sea auditiva, calórica, electromagnética, eléctrica, entre otras. Dejando un sin número de problemas abiertos a la imaginación para poder ser abordados, mejorados o resueltos. Como por ejemplo, la extracción de un electrocardiograma (ECG) de un feto a través de un ECG abdominal de la madre [10], la estimación de neuronas ocultas en una red neuronal, la captación de ruidos y vibraciones en las vigas de un puente [11], entre otros.

4.3 Trabajos Relacionados

Con el objetivo de ambientar el presente trabajo, se presentará un resumen de investigaciones con el mismo propósito y que utilizan la misma base de datos. Entre ellas existen muchas que incluso manejan las mismas herramientas, algunas poseen más o menos características u otras que utilizan un menor número de clases para clasificar. Estos factores llegan a ser claves, dado que, como se ha visto en los primeros modelos de este trabajo, a medida de que el número de clases sea menor, mayor exactitud posee el sistema. También, se ha visto la relevancia de las características correctas, es por esto que algunos de estos trabajos realizan una reducción de características, como por ejemplo de un total de 20 características a 5 características. Cabe destacar, que estos trabajos fueron realizados entre el año 2013 y el 2015, por doctores expertos en el área de vibraciones.

- En [22], se utiliza la Descomposición modal empírica (Empirical Mode Decomposition o EMD), para el análisis de señales no estacionarias o no lineales, el cual consiste en descomponer la señal en una suma de funciones modal intrínsecas (IMF), las cuales en conjunto con la herramienta SVD forman un método de la extracción de características. Posteriormente, estas características son ingresadas a una mejora de un clasificador por reglas, el “Hyper-Sphere-Structured Multi-Class SVM” (HSSMC-SVM).
- En [23], se realiza una extracción de características en el dominio del tiempo, las cuales se introducen a un método de reducción de ruido basado en análisis wavelet, debido a su eficiencia computacional y su gran resolución en los dominios del tiempo y la

frecuencia, sumado a esto se utilizan máquinas de soporte vectorial (SVM) para la clasificación.

- En [25], se hace una fusión de mapeo con la herramienta Local and Global Principal Component Analysis (LGPCA) para la extracción de características, la cual es capaz de revelar la diferencia natural de diferentes fallas a través del análisis y la extracción de la estructura inherente oculta en los datos, luego de esto las características son ingresadas a un clasificador NM.
- En [26], el método de extracción de características es EMD, del cual se rescatan las 5 primeras IMF para capturar 10 características en el dominio del tiempo y 10 en el dominio de la frecuencia, mientras que como método de clasificación utiliza a una Red Neuronal Probabilista (PNN).
- En [28], se realiza un extracción de 16 características, en el dominio del tiempo y de la frecuencia, 8 en cada uno, las cuales son ingresadas al método Kernel Joint Approximate Diagonalization of Eigen-matrices (KJADE). Este algoritmo estima las características más sensibles de la matriz del kernel desde el espacio de características para ofrecer una reducción de ellas. Aquí, se destaca la gran precisión de extracción del modelo, ya que, de 16 características se terminan usando sólo 3. Por último, estas tres características son ingresadas a una SVM.
- Se puede observar que en [29], se logran extraer 29 características, de las cuales 9 pertenecen al dominio del tiempo, mientras que el resto al dominio de la frecuencia, principalmente coeficientes obtenidos de la transformada de wavelet con distintas wavelets madres. De este total sólo se ocupan 5 debido al algoritmo de reducción de características Multi-Objective Binary Differential Evolution (MOBDE), posteriormente estas son ingresadas al algoritmo clasificador K-Nearest Neighbors (KNN).

Todo esto conlleva a lograr resultados similares pero con una gran diferencia, una que va más allá de las características o las salidas del sistema. Cada uno de estos métodos requiere de un conocimiento bastante amplio en el área de análisis de señales, mientras que el modelo propuesto en este trabajo utiliza técnicas excesivamente básicas, tan sencillo como una simple descomposición matricial o un promedio móvil simple. Es aquí donde nace la ganancia del sistema en donde se obtiene el máximo de exactitud con técnicas simples.

Ref.	Método	% Exactitud	N° Características	N° Clases
[21] (2015)	SAX-NN	98.94 - 99.43	5	10
[22]	EMD-SVD-Improved HSSMC-SVM	94.35 - 96.85	6	8
[23] (2014)	TR-LDA-Kmeans	93.35 - 97.55	15	10
[24]	WaveletDenoising-SVM	100	2	7
[25]	LGPCA-Fusion-NM	99.4 - 99.8	4	10
[26]	EMD-PNN	99.17	20	6
[27]	MFE-LS-VPMCD	93.13 - 100	5 (20)	8
[28] (2016)	KJADE-SVM	100	3 (16)	10
[29] (2016)	MOBDE-KNN	99.9	29 (5)	10
Modelo Propuesto	SVD-S-ANN	100	5	10

Tabla 1 Resultados de Trabajos relacionados

5 Marco Teórico

En base a lo mencionado anteriormente, se procederá a definir los tipos de métodos que se utilizarán en este proyecto. Uno de ellos será la implementación de una red neuronal de perceptron multicapa, que poseerá como entradas cuatro características y cuatro salidas (cuatro tipos de clasificación), una función de transferencia sigmoideal/softmax y como algoritmo de distribución de pesos Levenberg-Marquardt. Esta red deberá ser capaz de identificar la cantidad de fallos y el tipo de fallo correspondiente.

5.1 Algoritmo de Levenberg-Marquardt

El algoritmo de Levenberg-Marquardt es un algoritmo de back-propagation que si se utiliza como algoritmo de aprendizaje en las redes neuronales se convierte una herramienta bastante poderosa. Este algoritmo es generalmente el más usado debido a su estabilidad y rápida capacidad de convergencia. Además es conocido por su gran capacidad de minimizar el error cuadrático medio, generando así modelos con altos índices de correlación.

Este algoritmo se basa en calibración de los pesos sinápticos a través de numerosas iteraciones. Para ello parte con una aproximación de una matriz Hessiana (matriz que posee la segunda derivada respecto a sus pesos) expresada en la siguiente ecuación:

$$H \approx J^T J + \mu I \quad (1)$$

Donde J es una matriz Jacobiana que contiene la primera derivada de los errores de salida respecto a los correspondientes pesos; μ es una constante llamada coeficiente de combinación, la cual es siempre positiva y aumenta o decrece directamente según sea el cambio en el error; I es una matriz identidad la cual posee una diagonal con unos y el resto con ceros.

$$J = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \frac{\partial e_{1,1}}{\partial w_2} & \dots & \frac{\partial e_{1,1}}{\partial w_N} \\ \frac{\partial e_{1,2}}{\partial w_1} & \frac{\partial e_{1,2}}{\partial w_2} & \dots & \frac{\partial e_{1,2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{1,M}}{\partial w_1} & \frac{\partial e_{1,M}}{\partial w_2} & \dots & \frac{\partial e_{1,M}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{P,1}}{\partial w_1} & \frac{\partial e_{P,1}}{\partial w_2} & \dots & \frac{\partial e_{P,1}}{\partial w_N} \\ \frac{\partial e_{P,2}}{\partial w_1} & \frac{\partial e_{P,2}}{\partial w_2} & \dots & \frac{\partial e_{P,2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{P,M}}{\partial w_1} & \frac{\partial e_{P,M}}{\partial w_2} & \dots & \frac{\partial e_{P,M}}{\partial w_N} \end{bmatrix} \quad (2)$$

En base a esta ecuación complementada con los algoritmos de *Steepest Descent*, Newton y Newton-Gauss, se llega a una ecuación de calibración de pesos:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I})^{-1} \mathbf{J}_k \mathbf{e}_k \quad (3)$$

Donde \mathbf{w}_k es el peso sináptico actual usado para calibrar el peso siguiente, y \mathbf{e}_k es el k-ésimo error de salida de la red.

Gracias a la formula recién mencionada es que este algoritmo logra ser uno de los mejores, en especial por la matriz Jacobiana ya que esta la responsable de que el algoritmo converja rápidamente, aunque esto también demanda un poco más de recursos debido a la gran cantidad de cálculos. Sin embargo, en base a los resultados que otorga y a su rapidez de convergencia, definitivamente conviene usar este algoritmo de aprendizaje [16,17].

5.2 Función de Transferencia

Como se mencionaba anteriormente las redes neuronales necesitan de una función capaz de poder simular el umbral y así decidir si se transmite o no el impulso hacia otra neurona. A continuación se nombraran las más comunes [14]:

- **HardLim (escalón):** Esta función acerca la salida de la red a cero, si el argumento de la función es menor que cero y la lleva a uno si este argumento es mayor que uno.

$$Y(n) = \begin{cases} 1 & \text{si } n \geq 0 \\ -1 & \text{si } n < 0 \end{cases} \quad (4)$$

- **Purelin (lineal):** Como lo dice su nombre se expresa de forma lineal. Esta función de transferencia es utilizada en la red tipo Adaline.

$$Y(n) = n \quad (5)$$

- **Sigmoidal:** Esta fue seleccionada para formar parte de este trabajo, particularmente para ser usada en la capa oculta de la red. Esta función toma los valores de entrada, los cuales pueden oscilar entre más y menos infinito, y restringe la salida a valores entre cero y uno, de acuerdo a la siguiente expresión:

$$Y(x_i) = \frac{1}{1 + e^{x_i}} \quad (6)$$

- **Tangente Hiperbólica:** Los valores de salida de la función tangente hiperbólica están comprendidos dentro de un rango que va de -1 a 1.

$$Y(x_i) = \frac{e^{x_i} - e^{-x_i}}{e^{x_i} + e^{-x_i}} \quad (7)$$

- Softmax: Esta es otra de las funciones escogidas para este estudio, debido a que se caracteriza por ser usada en redes neuronales en su capa de salida destinadas al ámbito de la clasificación de problemas no lineales. A continuación se mostrará su ecuación:

$$Y(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (8)$$

En donde x_i corresponde a una entrada de la red y n es el número de salidas de la red. Con esto la función estandariza los valores de salida para que todos los valores estén entre cero y uno.

5.3 Descomposición de Valores Singulares

Como se mencionó anteriormente la descomposición se basa en una división matricial [6], la cual será mostrada a continuación:

$$[A]_{m \times n} = [U]_{m \times m} [S]_{m \times n} [V]_{n \times n}^T \quad (9)$$

Siendo $[A]$ la matriz a descomponer, por lo tanto $A^T A = S$, donde S es una matriz diagonal contiene las raíces de los valores propios ($S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_L)$, donde L es $\min(m, n)$). Mientras que U y V son matrices ortonormales (tal que, $U^T U = I$ y $V^T V = I$). Luego sea $\overline{[A]}$, la matriz reconstruida solo con los valores singulares más grandes, dejando a esta matriz con un rango de $l < L$, como se muestra a continuación:

$$\overline{[A]} = [U_l] [S_l] [V_l^T] \quad (10)$$

De esta la forma la separación deja, en primer lugar a la señal sin ruido ($\overline{[A]}$), y luego la señal contaminada ($[N]$), quedando $A = \overline{[A]} + N$. Esta separación se hace a través de un umbral ε , como muestra en las siguientes condiciones:

$$\sigma_i > \varepsilon, i = 1 \dots l \quad (11)$$

$$\sigma_i \leq \varepsilon, i = l + 1 \dots L \quad (12)$$

Este umbral se definió entre el 50% y 80% de los primeros valores singulares, debido a que este pertenece a la baja frecuencia de la señal (señal de rotación), mientras que el restante se encuentra en la alta frecuencia de la señal (señal de daño y de ruido). Como se muestra l viene siendo el número óptimo de valores singulares, por lo que la expresión para reconstruir la matriz queda expresada de la siguiente forma:

$$\overline{[A]} = [U]_{m \times l} [S]_{l \times n} [V]_{n \times l}^T \quad (13)$$

6 Análisis de los resultados

Para analizar los resultados de los distintos modelos primero se explicará todo el proceso necesario para construir una red, de principio a fin. Posteriormente se ahondará en la particularidad de cada modelo.

6.1 Muestras Utilizadas

Como se mencionaba al comienzo de este trabajo, se utilizarán los datos obtenidos de la base de datos de la Case Western Reserve University, la cual, posee datos para un rodamiento con estado normal, tres distintos tipos de fallos.

Los datos corresponden a un rodamiento del eje de salida, con los cuales se armará una configuración de red a una velocidad variable, incluyendo sus tres tipos de fallos con sus tres magnitudes de fallo.

6.2 Características

A lo largo de la investigación se utilizan diferentes métricas estadísticas para recoger información de los datos. Estas son:

- Desviación estándar

$$s = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (14)$$

- Valor medio absoluto

$$\bar{x}_{abs} = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (15)$$

- Skewness

$$\sum_{i=1}^N \frac{(x_i - \bar{x})^3}{N\sigma^3} \quad (16)$$

- Kurtosis

$$\sum_{i=1}^N \frac{(x_i - \bar{x})^4}{N\sigma^4} \quad (17)$$

- Raíz valor cuadrático medio

$$\bar{x}_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (18)$$

- Valor máximo de peak

$$\hat{x} = \max(x) \quad (19)$$

- Factor Forma

$$\frac{\bar{x}_{rms}}{\bar{x}_{abs}} \quad (20)$$

- Entropía

$$H(x) = - \sum_{i=1}^N p(x_i) * \log_2 p(x_i) \quad (21)$$

6.3 Configuración de los datos

Para poder empezar a trabajar con las redes primero se deben preparar los datos. A continuación se explicará detalladamente este proceso.

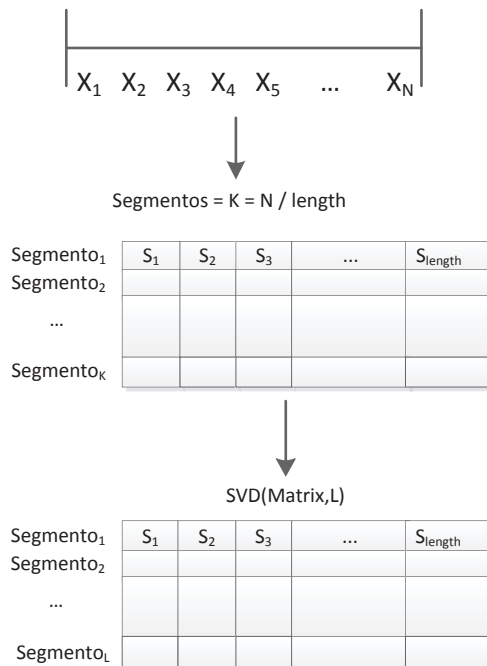


Figura 6 Proceso de carga de datos 1

Como se puede observar en la Fig. 6, el primer paso es tomar la señal (120.000 datos) y dividirla en una longitud (length), en este caso 1000 datos, formando una matriz de 120x1000, donde 120 equivale al número de segmentos. El siguiente paso, es aplicar la herramienta de reducción de ruido SVD con un l igual 75 como número de segmentos, por lo que la matriz se reduce a 75x1000. Luego de esto, a cada segmento se le deben extraer

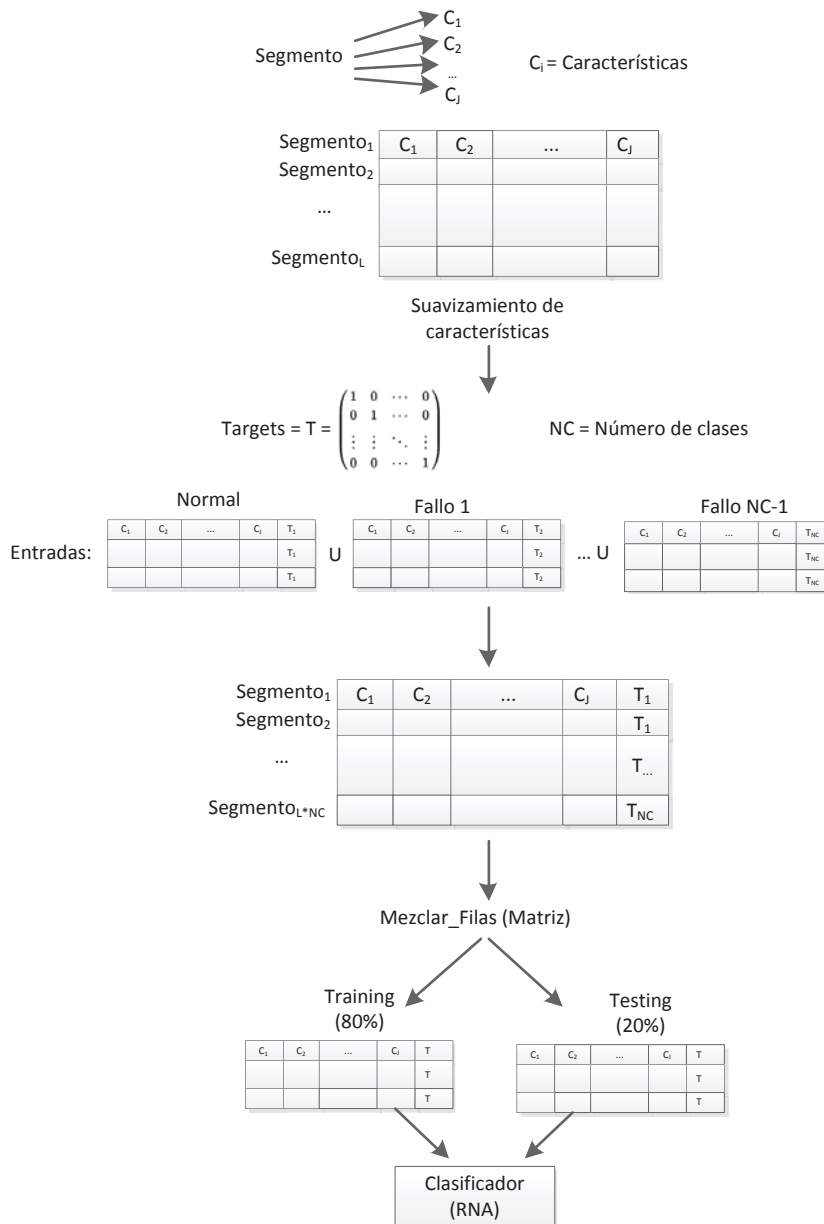


Figura 7 Proceso de carga de datos 2

las características propuestas (Kurtosis, Raíz valor cuadrático medio, Factor forma, Valor máximo de peak y Entropía), por lo que, por cada fila (1000 datos) se obtienen estas cinco características, donde se obtiene una matriz de 75x5. Además a cada bloque de características se le debe adherir el estado correspondiente (targets), representado con una asociación binaria. Este paso resulta ser clave, ya que los denominados targets, son los

encargados ayudar a la red indicándole a qué tipo de clase pertenecen estos datos (normal, anillo interno fallo, anillo externo o bola, entre otros). Al final de este paso se consigue obtener un matriz de 75x15.

Los pasos anteriormente descritos deben ser ejecutados para cada los 10 tipos de clases (normal, anillo interno, anillo externo y bola (tres por cada fallo)). Luego de esto, procede a unificar estas matrices en una sola, quedando en 750x15. Después, esta matriz debe ser mezclada, para evitar que la red aprenda diagnosticar solo datos que estén ordenados. Y por último, se deben dividir la cantidad de datos en dos categorías, una para entrenar la red (training) y otra para probar la red (testing). Se escogió la división de un 80% de los datos para training y 20% para testing. De donde se obtienen dos matrices, una de 600x15 y otra de 150x15.

6.4 Configuración de la red neuronal

Una vez que los datos estén preparados se procede a calibrar la red, para esto se debe realizar una serie de pasos, los cuales serán mencionados a continuación:

- 1) Lo primero, es ver la configuración del algoritmo de Levenberg-Marquardt, es decir con cuántas iteraciones el algoritmo tiende a converger. Como se puede observar en el gráfico en una prueba con 500 iteraciones, el algoritmo comienza a converger desde aproximadamente en las 60 iteraciones y termina de converger alrededor de las 300.

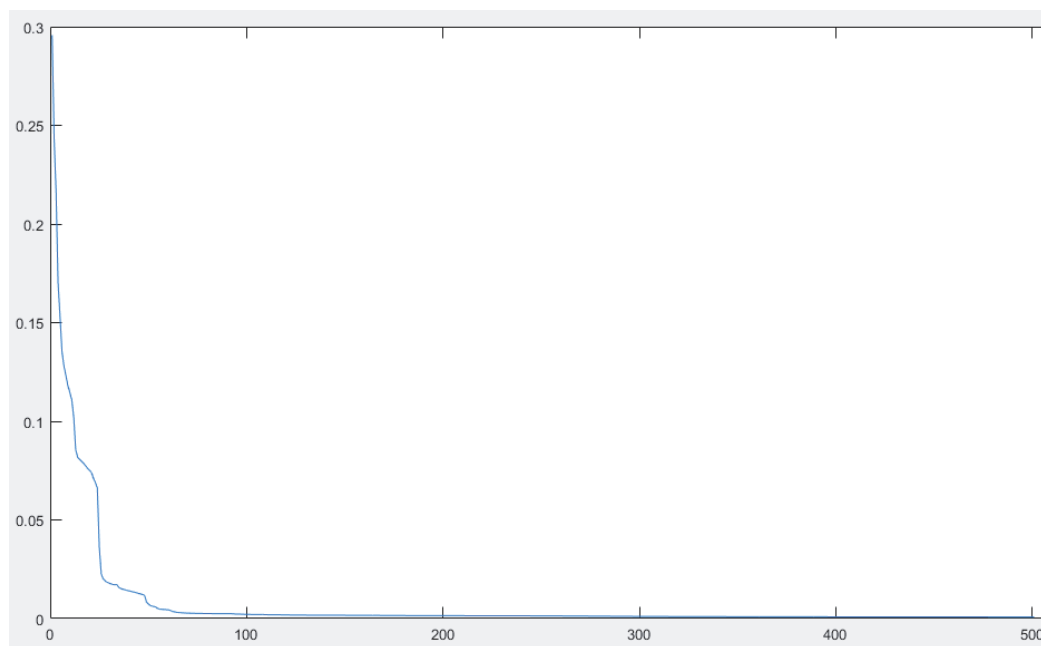


Figura 8 Gráfico Error vs Iteraciones

- 2) Luego se debe hacer variar la cantidad de nodos ocultos, ya que no se sabe qué número de nodos ocultos es el óptimo. A su vez, por cada nodo oculto la red se ejecuta a lo menos cinco veces, ya que la red funciona de manera casi aleatoria, por lo que puede funcionar muy bien como muy mal. Todo esto se hace a través de una medición del

error, comparando las salidas de la red con los targets de las entradas, quedándonos obviamente con el que posea menor error. Una demostración de esto se puede observar en la Fig. 9.

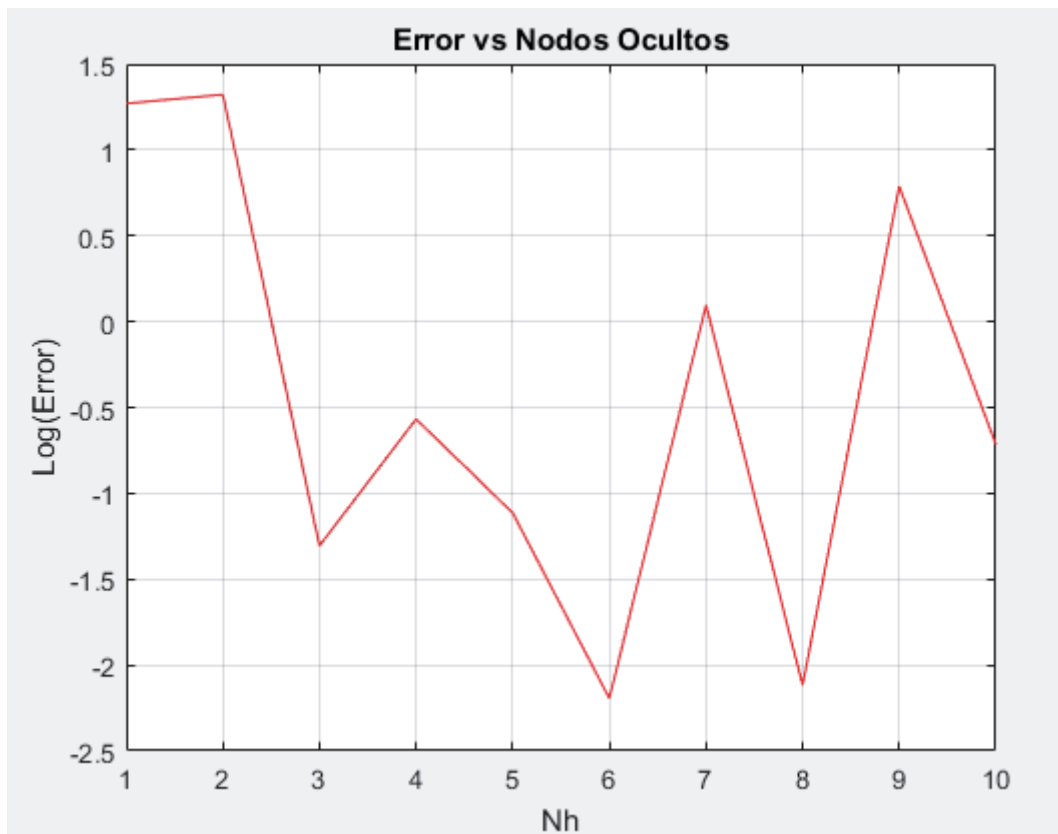


Figura 9 Gráfico Error vs Nodos ocultos

- 3) Teniendo el número óptimo de nodos ocultos, se procede a hacer correr la red con la configuración óptima.

6.5 Métricas de Rendimiento

Para poder captar la calidad con la que la red neuronal está trabajando, es que nacen métricas, las cuales son la precisión, la sensibilidad y la exactitud. Estas métricas se calculan en base a los siguientes variables:

- Verdaderos Positivos (VP): Es el número de aciertos que corresponden a una clase, cuando efectivamente pertenecían a esta.
- Verdaderos Negativos (VN): Es el número de aciertos que no corresponden a una clase, cuando efectivamente no pertenecían a esta.

- Falsos Positivos (FP): Es el número de aciertos que no corresponden a una clase, cuando efectivamente pertenecían a esta. Estos son representados por las columnas de una matriz de confusión.
- Falsos Negativos (FN): Es el número de aciertos que corresponden a una clase, cuando efectivamente no pertenecían a esta. Estos son representados por las filas de una matriz de confusión.

En base a esto, las métricas se definen de la siguiente forma:

- Precisión: Corresponde a la probabilidad de acertar la clasificación en una clase con datos de la misma clase.

$$P_i = \frac{VP_i}{VP_i + FP_i} = \frac{N_{i,i}}{\sum_{j=1}^K N_{i,j}} \quad (20)$$

- Sensibilidad: Corresponde a la probabilidad de acertar la clasificación en una clase con datos de la otra clase.

$$R_j = \frac{VP_j}{VP_j + FN_j} = \frac{N_{j,j}}{\sum_{i=1}^K N_{i,j}} \quad (21)$$

- Exactitud: Corresponde al porcentaje de datos que fueron correctamente clasificados por la red.

$$A = \frac{VP + FP}{VP + FP + FN + FP} = \frac{N_{j,j}}{\sum_{i=1}^K \sum_{j=1}^K N_{i,j}} \quad (22)$$

- F-Score: Puede ser interpretado como una media armónica entre la precisión y la sensibilidad, en donde puede tomar valores entre 1 y 0, siendo 1 el mejor y 0 el peor.

$$Fscore_i = 2 * \frac{R_i * P_i}{R_i + P_i} \quad (23)$$

6.6 Modelos

Para explicar la forma de trabajo, se presentará una descripción de todos los modelos utilizados en este proyecto. Comenzando con modelos de detección, para luego avanzar a modelos de diagnóstico.

6.6.1 Modelo 1

Este modelo fue construido en un proyecto [18,19] que fue realizado el año pasado por unos compañeros, el cual consistía en hacer diagnóstico de fallas de rodamientos pero utilizando tres redes y un árbol binario de decisión. Estas redes como se observa en la Fig.

10, eran capaces de distinguir solo dos estados del rodamiento, un sí o un no, estas tres redes eran puestas una después de la otra para poder analizar si correspondían o no a un determinado fallo. Primero el árbol analizaba si existía algún fallo o no, los datos que no correspondían fallos quedaban registrados como normales mientras que los demás

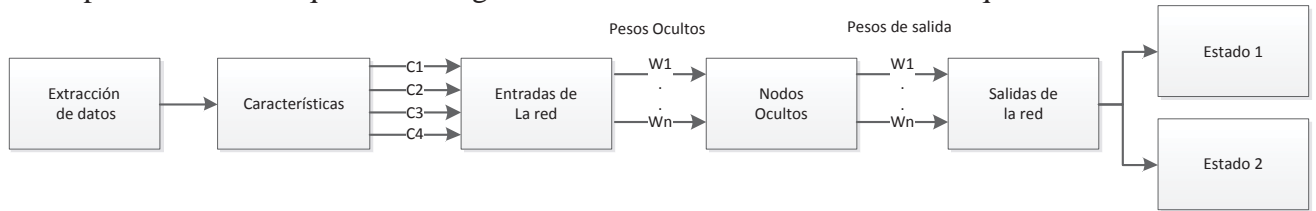


Figura 10 Red neuronal del modelo 1

por tener fallos eran reingresados a la siguiente red. Para determinar qué tipo de fallo tenía el rodamiento, se comenzaba de nuevo el proceso, pero esta vez verificando si era un fallo de anillo interno o no, los que no eran anillo interno se reingresaban a la red clasificando un fallo externo o si no lo era, y en caso de no ser fallo externo correspondían a fallo de bola. Este esquema se puede visualizar en la Fig. 11. De esta manera terminaban recorriendo todas las tipologías de errores.

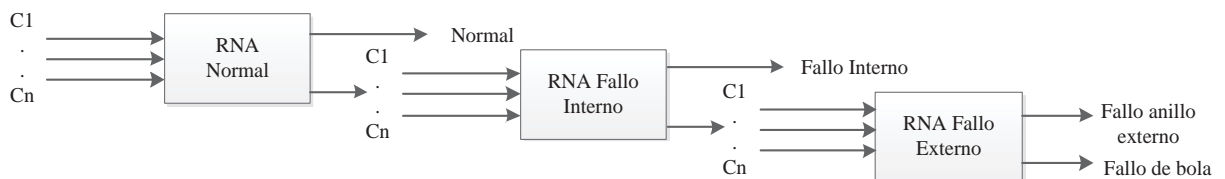


Figura 11 Estructura del árbol de decisión

Como se puede ver en los esquemas anteriores tenemos una variable C, lo que representa una de las 5 características que se usan en esta red. Las cuales son: desviación estándar, skewness, valor máximo de peak, factor de forma y kurtosis.

Los resultados de la red en distintos tipos de falla y distintas velocidades fueron los siguientes:

Velocidad del motor / Diámetro de la falla	1797 RPM	1772 RPM	1750 RPM	1730 RPM
0.007	100%	100%	100%	100%
0.014	100%	100%	100%	100%
0.021	100%	100%	100%	100%

Tabla 2 Resultados Modelo 1

En este trabajo cómo se puede ver en la tabla 2, tuvo muy buenos resultados, de hecho se le hicieron más pruebas que variando el diámetro de la falla y las velocidades, obteniendo el mismo desempeño.

6.6.2 Modelo 2

El modelo actual consta de una red neuronal con algoritmo de aprendizaje Levenberg-Marquardt y funciones de transferencia sigmoidal y softmax, en la capa oculta y la otra en la capa de salida, respectivamente. A su vez, esta cuenta con cuatro entradas que son las características obtenidas desde la señal, las cuales son, el valor medio absoluto, la varianza, la skewness y la kurtosis. En el contexto de salidas se analizarán cuatro estados de un rodamiento, normal, fallo anillo interno, fallo anillo externo y fallo de bola. Esto último es la gran diferencia respecto del otro modelo ya que une estas tres redes en una sola, como se muestra en la Fig. 12.

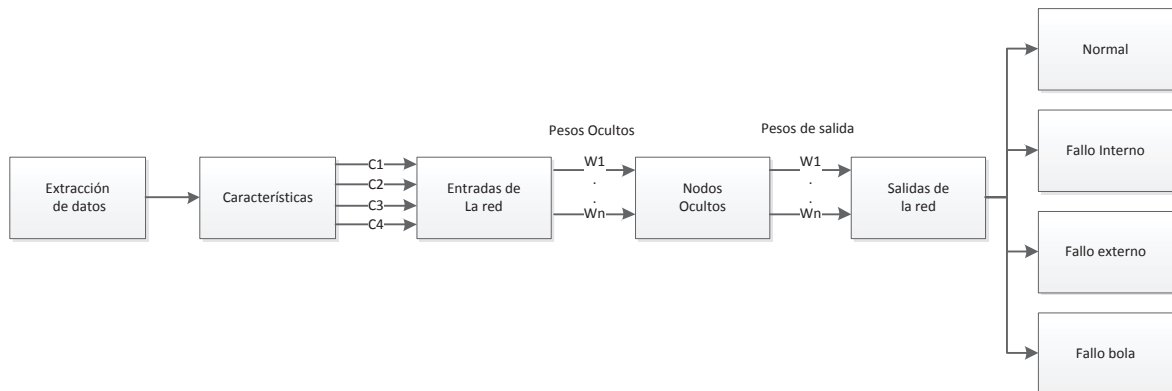


Figura 12 Red neuronal del Modelo 2

Como se puede observar la tabla 2 versus la tabla 3, los resultados son bastantes distintos, esto se debe principalmente a la exigencia de la red, ya que para esta obviamente resulta bastante más fácil clasificar entre dos que entre cuatro.

Velocidad del motor / Diámetro de la falla	1797 RPM	1772 RPM	1750 RPM	1730 RPM
0.007	86.0%	81.3%	82.0%	81.8%
0.014	72.2%	63.4%	63.0%	64.8%
0.021	78.1%	75.9%	77.2%	76.5%

Tabla 3 Resultados Modelo 2

6.6.3 Modelo 3

Este modelo resulta ser una mejora del modelo anterior, el cual se diferencia solo en la implementación de la descomposición de valores singulares antes del cálculo de las características, como se muestra a continuación:

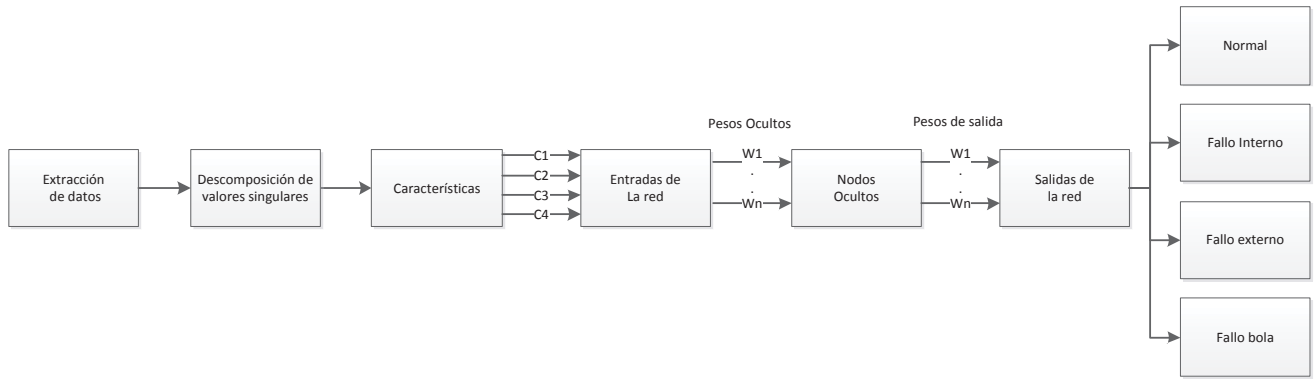


Figura 13 Red neuronal del Modelo 3

Si bien, el cambio es bastante pequeño, los cambios en los resultados son mejores. Como se puede observar en la siguiente tabla, están las 7 muestras y justo con ellas el desempeño que tuvo en el nuevo modelo. Las primeras cuatro presentan la detección frente a una velocidad constante con diámetros de fallo variable, mientras que las tres restantes presentan diámetro de falla constante y velocidad variable.

Muestra de datos	Exactitud
Velocidad 1 Diámetro variable	100%
Velocidad 2 Diámetro variable	100%
Velocidad 3 Diámetro variable	100%
Velocidad 4 Diámetro variable	100%
Diámetro 0.007 Velocidad variable	95%
Diámetro 0.014 Velocidad variable	85%
Diámetro 0.021 Velocidad variable	100%

Tabla 4 Resultados Modelo 3

Como se pudo observar en los resultados, estos fueron bastante buenos. Al compararlos con la tabla 3, se nota que los resultados incrementaron considerablemente gracias a la descomposición de valores singulares, por lo que se puede decir que, la incorporación de esta técnica fue un éxito. A su vez, se puede notar que estos resultados comparados con los de la tabla 2 son bastantes parecidos, a pesar de que las nuevas configuraciones tienen mayor dificultad dada a la mezcla que fueron sometidas. Teniendo esto en cuenta, se puede contrastar sus arquitecturas, en donde, la de este modelo resulta ser tremendamente compacta, es decir se obtiene prácticamente el mismo desempeño con una red que con tres redes.

6.6.4 Modelo 4

El modelo que se presentará a continuación, es una extensión del anterior dado los grandes resultados obtenidos en este. Este nuevo modelo tiene como objetivo lograr un mayor control de la gravedad de los fallos, por lo que la idea es ir contrastando entre gravedades para cada uno de los fallos y su estado normal. Aquí se probarán tres configuraciones con el modelo de la figura 14. La primera tiene como objetivo captar el

momento en que una falla leve pasa a ser una falla moderada, lo que arrojaría un diagnóstico en las primeras fases de fallo. La segunda evalúa la transformación de un fallo moderado hacia un fallo grave, esta etapa es de vital importancia, ya que aquí es donde se debe efectuar la detención de la máquina. La última configuración si bien es poco probable trata de simular la detección en un evento donde una falla leve se transforme inmediatamente en una falla grave, pero esto no deja de ser un caso interesante de analizar.

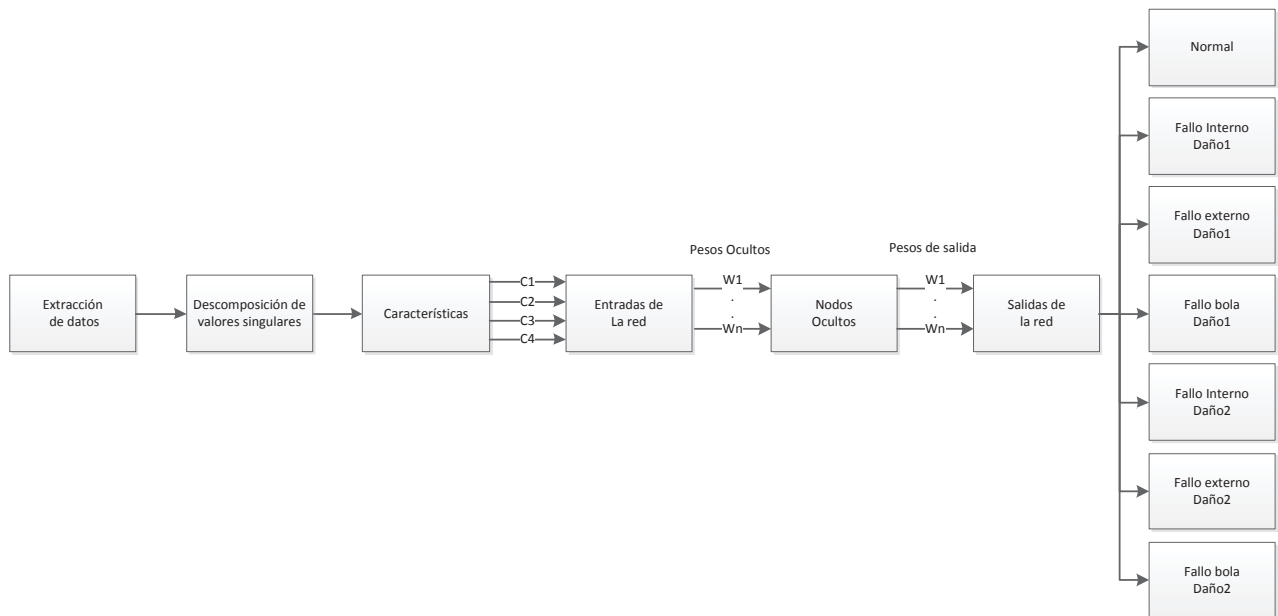


Figura 14 Red neuronal del Modelo 4.

- Leve: Falla de tamaño 0.007 pulgadas o menor.
- Moderado: Corresponde a una falla de un tamaño de 0.014 pulgadas.
- Grave: Falla que posee un tamaño mayor igual a 0.021 pulgadas.

Los resultados obtenidos de estas combinaciones se muestran en la siguiente tabla, en los cuales la exactitud bajo un poco respecto al modelo anterior. Esta baja se debe en gran parte al aumento de las salidas de la red, lo que produce que la red neuronal le sea más difícil clasificar. En los resultados se puede apreciar el hecho de que los porcentajes más altos de exactitud se encuentran entre Leve-Moderado y Leve-Grave, de aquí se puede inferir de que los datos de las fallas poseen una diferencia notoria entre ellos. Mientras que para el caso de Moderado-Grave se puede especular que los datos podrían tender a parecerse y entorpecer el funcionamiento de la red.

Muestra de datos	Exactitud
Leve-Moderado	81.0%
Moderado-Grave	77.1%
Leve-Grave	87.6%

Tabla 5 Resultados Modelo 4

6.6.5 Modelo 5

Este modelo surge como una mejora del modelo 4. Un punto importante encontrado en el modelo anterior fue que, dado los bajos resultados obtenidos, se concluyó que había un problema con las características, por lo que estas fueron cambiadas por: Kurtosis, Raíz valor cuadrático medio, Factor forma, Valor máximo de peak y Entropía. Además de este cambio, se implementó una suavización de las características antes de ingresarlas a la red con un promedio móvil simple de cinco, obteniendo como resultado una reducción de máximos y mínimos junto a características más acotadas en ciertos intervalos, evitando que estas se solapen y entorpezcan el desempeño de la red.

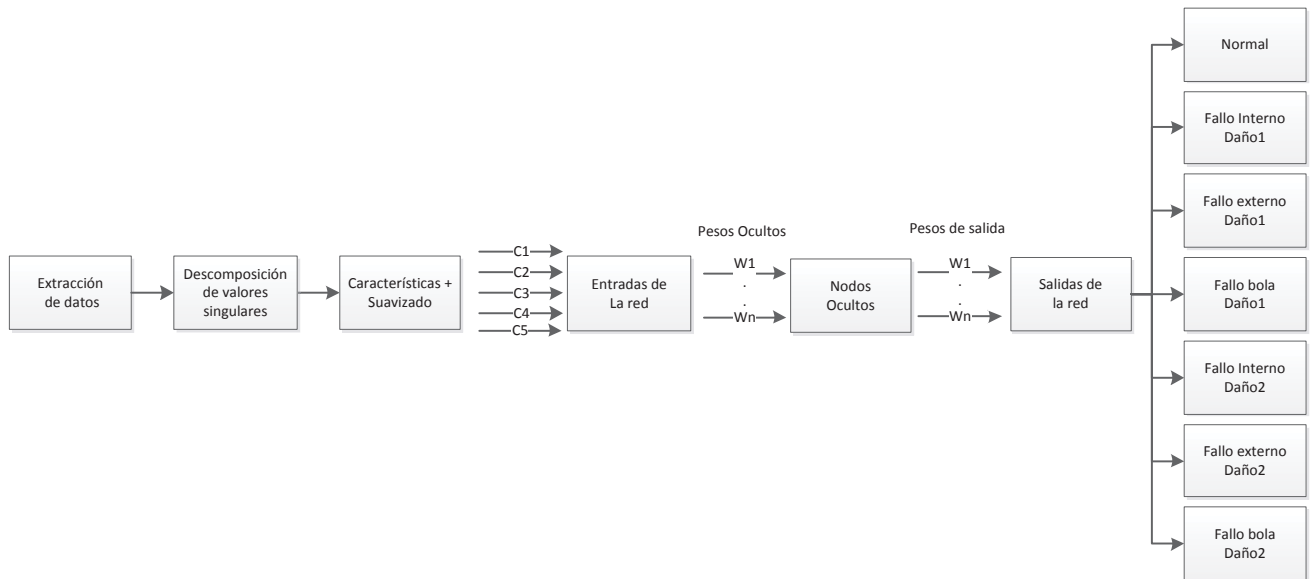


Figura 15 Red neuronal del Modelo

Dada la mejora obtenida en los resultados se puede decir, lo poderoso que son las nuevas características junto con el suavizado realizado en ellas. Esta mejora alcanza en promedio una diferencia 15 puntos porcentuales, una brecha bastante alta, lo que mejora el modelo considerablemente.

Muestra de datos	Exactitud
Leve-Moderado	97.3%
Moderado-Grave	95.2%
Leve-Grave	98.8%

Tabla 6 Resultados Modelo 5

6.6.6 Modelo 6

Dado que la mejora obtenida en el modelo 5 tuvo tan buenos resultados, se utilizará la misma configuración de datos. Además se agrandó el modelo llegando a obtener una red neuronal de diez salidas, la cual es capaz de reconocer su estado normal, los tres tipos de fallos y en cada uno de ellos sus tres dimensiones de fallo.

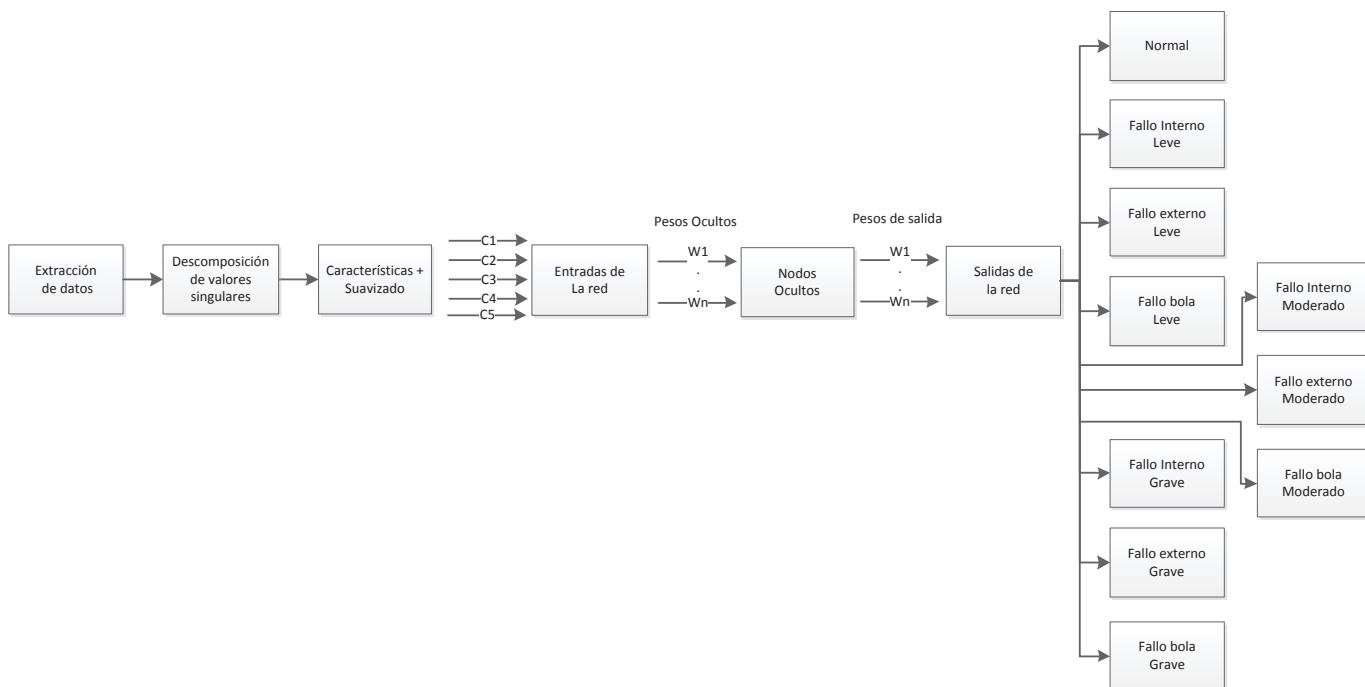


Figura 16 Red neuronal del Modelo 6

En base a los resultados obtenidos con esta última configuración de red se da por terminada la fase de diagnóstico, siendo capaz de clasificar en un 100%, tanto la localización como la magnitud de un fallo.

Muestra de datos	Exactitud
Leve-Moderado-Grave	100%

Tabla 7 Resultados Modelo 6

A continuación se muestra en la Fig. 18, las matrices de confusión correspondiente al resultado obtenido por el modelo, esta matriz representa el resumen de la ejecución de una red de clasificación. En esta matriz podemos observar las distintas clases del modelo, descritas a continuación:

- 1-Normal
- 2-Fallo Anillo interno 0.007
- 3- Fallo Anillo externo 0.007
- 4-Fallo bola 0.007
- 5- Fallo Anillo interno 0.014

- 6- Fallo Anillo externo 0.014
- 7-Fallo bola 0.014
- 8- Fallo Anillo interno 0.021
- 9- Fallo Anillo externo 0.021
- 10-Fallo bola 0.021

En la diagonal verde se pueden ver las clasificaciones correctamente hechas, mientras que en la zona roja de las matrices se observan las que no fueron correctamente clasificadas. En las columnas grises se localizan las relaciones entre clasificaciones correctas y falsos positivos (Precisión). Por el contrario en las filas grises se encuentran las relaciones entre clasificaciones correctas y falsos negativos de cada clase (Sensibilidad). Por último, las casillas azules poseen la exactitud de los modelos.

Output Class	1	2	3	4	5	6	7	8	9	10	
1	17 14.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	13 10.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	14 11.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	11 9.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 11.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	13 10.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 8.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 6.7%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 7.5%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11 9.2%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	4	5	6	7	8	9	10	
	Target Class										

Figura 17 Matriz de confusión

Para poder visualizar la robustez del modelo, luego de encontrar la mejor topología se procede a ejecutar 20 veces la red, con la diferencia que por cada ejecución se vuelven a mezclar los datos. Como se puede observar en la tabla 8, la exactitud en algunas

ejecuciones varia, pero esto sigue siendo un porcentaje menor, en donde se obtiene un promedio de 99.4, el cual es un resultado bastante satisfactorio.

	Exactitud	Precisión	Sensibilidad	F-Score
1	100	100	100	100
2	99.2	99.2	99.4	99.1
3	100	100	100	100
4	100	100	100	100
5	97.5	98.0	98.6	99.8
6	100	100	100	100
7	98.3	98.5	98.7	99.6
8	98.3	98.6	98.6	98.6
9	100	100	100	100
10	100	100	100	100
11	100	100	100	100
12	100	100	100	100
13	98.3	98.2	98.7	98.4
14	100	100	100	100
15	99.2	99.4	99.2	99.3
16	100	100	100	100
17	100	100	100	100
18	97.5	97.6	97.7	97.4
19	100	100	100	100
20	100	100	100	100
Promedio	99.41	99.47	99.54	99.63
Desv. Estándar	0.9	0.8	0.7	0.72

Tabla 8 Pruebas de robustez

8 Conclusión

Primero, como se pudo observar, el modelo en que se basa este estudio es contrastado con el que fue implementado el año 2015, junto con las entregas anteriores de este proyecto. De la comparación de estas situaciones se puede inferir la gran capacidad de las herramientas utilizadas, las cuales ni siquiera recurre a grandes cálculos numéricos, pero terminan siendo bastante poderosas. Estas herramientas mejoran los resultados considerablemente de las entregas pasadas y gracias a ella se puede hacer uso de una arquitectura mucho más compacta, lo que el modelo termina superando con creces a los modelos que lo anteceden.

Un punto destacable, es el impacto que generan las características en la red neuronal, como se pudo observar en el modelo cuatro versus el número cinco, la brecha entre las exactitudes fue bastante notoria (alrededor de 15%). Aunque esta diferencia, fue lograda con la ayuda del suavizamiento de las características, no deja de ser un punto menor, la elección de las características más relevantes para el sistema. Esto puede ser observado en algunos trabajos relacionados [27, 28, 29], en donde más allá de hacer un proceso de extracción de características, lo que hace es reducir el número de características, eliminando las menos significativas para el modelo.

Como se ha visto en otras oportunidades, el número de características y el número de clases impactan tremendamente en los resultados obtenidos por el modelo. El presente modelo utiliza cinco características, lo cual resulta ser bastante bajo, mientras que para el número de clases diez, lo que provoca que el sistema sea muy complejo, ya que si comparamos esta arquitectura con otros trabajos, la mayoría de los trabajos no generan el mismo resultado aun cuando sus sistemas poseen una arquitectura mucho más flexible, poseen más características y menos clases, lo que debería facilitar su trabajo y lograr una exactitud perfecta. Pero, no siempre es así, mientras que este modelo logra un 100% de exactitud, el resto es un su gran mayoría no.

Otro punto importante, es la capacidad de lograr buenos resultados con herramientas que no requieren grandes cálculos computacionales o una gran cantidad de estudio. En los trabajos relacionados hablados anteriormente, se tienen un número bastante alto de técnicas para la extracción de características, reducción de características, eliminación del ruido y clasificación de clases. Pero, la mayorías de estas, requiere quizás no una gran eficiencia computacional, sino más bien, un gran consumo de tiempo para comprender estas técnicas. Es aquí donde este modelo, toma una gran ventaja, ya que las técnicas utilizadas en él, no requieren de grandes conocimientos matemáticos, sino una simple descomposición matricial y un cálculo de promedio móvil simple.

Por último, vale la pena destacar el hecho de que los resultados logrados por este trabajo, superan o igualan a trabajos realizados por doctores expertos en el área de vibración. Además, esto lo hace con técnicas sumamente básicas en comparación con los otros trabajos. Por lo tanto, en base a esto y dado a los resultados obtenidos por el modelo y la robustez del modelo, se puede afirmar que, el presente modelo está altamente calificado para desarrollar, tanto una detección como un diagnóstico de fallos alojados en un rodamiento de un motor.

9 Referencias

- [1] Tomomichi Nakamura, Shigehiko Kaneko, Fumio Inada, Minoru Kato, Kunihiko Ishihara, Takashi Nishihara, Njuki Mureithi, Mikael Langthjem. “*Vibrations in Rotary Machines*”. Flow-induced Vibrations (2014) Capítulo 7, Páginas 311-358.
- [2] P. V. J. Rodriguez and A. Arkkio. "Detection of Stator Winding Fault in Induction Motor Using Fuzzy Logic". Applied Soft Computing Volumen (2008), Páginas 1112-1120.
- [3] Yongxiang Zhang, R.B. Randall. “*Rolling element bearing fault diagnosis based on the combination of genetic algorithms and fast kurtogram*”. Mechanical Systems and Signal Processing Volumen 23 (2009), Páginas 1509–1517.
- [4] Douglas D. Boyd, Alan Stolzer. “*Accident-precipitating factors for crashes in turbine-powered general aviation aircraft*”. Accident Analysis and Prevention Volumen 86 (2016), Páginas 209–216.
- [5] M. A. G. Calle, R. E. Oshiro, M. Alves. “*Ship collision and grounding: Scaled experiments and numerical analysis*”. International Journal of Impact Engineering Volumen 103 (2017), Páginas 195-210.
- [6] Reza Golafshan, Kenan Yuce Sanliturk. “*SVD and Hankel matrix based de-noising approach for ball bearing fault detection and its assessment using artificial faults*”. Mechanical Systems and Signal Processing Volumen 64-65 (2015), Páginas 36–50.
- [7] Wade A. Smith, Robert B. Randall. “*Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study*”. Mechanical Systems and Signal Processing Volumen 64-65 (2015), Páginas 100–131.
- [8] Feiyun Cong, Jin Chen, Guangming Dong, Fagang Zhao. “*Short-time matrix series based singular value decomposition for rolling bearing fault diagnosis*”. Mechanical Systems and Signal Processing Volumen 34 (2013), Páginas 218–230.
- [9] Xuezhi Zhao, Bangyan Ye. “*Similarity of signal processing effect between Hankel matrix based SVD and wavelet transform and its mechanism analysis*”. Mechanical Systems and Signal Processing Volumen 23 (2009), Páginas 1062–1075.
- [10] A. Al-Zaben, A. Al Smadi. “*Extraction of foetal ECG by combination of singular value decomposition and neuro-fuzzy inference system*”. Phys. Med. Biol. Volumen 51 (2006), Páginas 137-43.
- [11] Nicolás Ponso, Leonardo Molisani, Adriana Zapico, Juan C. del Real y Yolanda Ballesteros. “*Eliminación de ruido de señales acústicas usando descomposición de valores singulares y análisis de componentes principales*”. Mecánica Computacional Vol. XXXI, Páginas 4013-4027.

- [12] M. Kang, J. M. Kim. “*Singular value decomposition based feature extraction approaches for classifying faults of induction motors*”. Mechanical Systems and Signal Processing Volumen 41 (2013), Páginas 348–356
- [13] H. Jiang, J. Chen, G. Dong, T. Liu, G. Chen. “*Study on Hankel matrix-based SVD and its application in rolling element bearing fault diagnosis*”. Mechanical Systems and Signal Processing Volumen 52–53 (2015), Páginas 338–359.
- [14] Winfried Just, Sungwoo Ahn, David Terman. “*Neuronal Networks: A Discrete Model*”. Mathematical Concepts and Methods in Modern Biology (2013), Capítulo 6, Páginas 179–211.
- [15] Fatin A. Elhaj, Naomie Salim, Arief R. Harris, Tan Tian Swee, Taqwa Ahmed. “*Arrhythmia Recognition and Classification using Combined Linear and Nonlinear Features of ECG Signals*”. Computer Methods and Programs in Biomedicine Volumen 127, (2016), Páginas 52–63.
- [16] A. Reynaldi, S. Lukas and H. Margaretha. “*Backpropagation and Levenberg-Marquardt algorithm for training finite element neural network*”. 2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation, Valetta, 2012, Páginas 89-94.
- [17] Xiancheng Shi, Yucheng Feng, Jinsong Zeng, Kefu Chen. “*Chaos time-series prediction based on an improved recursive Levenberg–Marquardt algorithm*”. Chaos, Solitons and Fractals Volumen 100 (2017) Páginas 57–61.
- [18] Diego López Cataldo. “*Diagnóstico de fallas de rodamientos y engranajes de equipos de rotación utilizando redes neuronales artificiales*”. Pontificia Universidad Católica de Valparaíso, 2015.
- [19] Daniel Valenzuela Sepúlveda. “*Diagnóstico de fallas de rodamientos y engranajes de equipos de rotación utilizando redes neuronales artificiales*”. Pontificia Universidad Católica de Valparaíso, 2015.
- [20] Kumar H. S., Dr. Srinivasa Pai P., Dr. Sriram N. S., Vijay G. S. “*ANN based evaluation of performance of wavelet transform for condition monitoring of rolling element bearing*”. Procedia Engineering Volumen 64 (2013), Páginas 805 – 814.
- [21] George Georgoulas, Petros Karvelis, Theodoros Loutas, Chrysostomos D. Stylios. “*Rolling element bearings diagnostics using the Symbolic Aggregate approXimation*”. Mechanical Systems and Signal Processing Volumen 60–61 (2015), Páginas 229–242.
- [22] Yujing Wang, Shouqiang Kang, Yicheng Jiang, Guangxue Yang, Lixin Song, V.I.Mikulovich. “*Classification of fault location and the degree of performance degradation of a rolling bearing based on an improved hyper-sphere-structured multi-class support vector machine*”. Mechanical Systems and Signal Processing Volumen 29 (2012), Páginas 404–414.

- [23] Mingbo Zhao, Xiaohang Jin, Zhao Zhang, Bing Li. “*Fault Diagnosis of Rolling Element Bearings via Discriminative Subspace Learning: Visualization and Classification*”. *Expert Systems with Applications* 41 (2014), Páginas 3391–3401.
- [24] S. Abbasiona, A. Rafsanjania, A. Farshidianfarb, N. Irani. “*Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine*”. *Mechanical Systems and Signal Processing* Volumen 21 (2007), Páginas 2933–2945.
- [25] Xiaoxi Ding, Qingbo He, Nianwu Luo. “*A fusion feature and its improvement based on locality preserving projections for rolling element bearing fault classification*”. *Journal of Sound and Vibration* Volumen 335 (2015) Páginas 367–383.
- [26] Xiaoyue Chen, Jianzhong Zhou, Jian Xiao, Xinxin Zhang, Han Xiao, Wenlong Zhu, Wenlong Fu. “*Fault diagnosis based on dependent feature vector and probability neural network for rolling element bearings*”. *Applied Mathematics and Computation* Volumen 247 (2014), Páginas 835–847.
- [27] Jinde Zheng, Junsheng Cheng, Yu Yang, Songrong Luo. “*A rolling bearing fault diagnosis method based on multi-scale fuzzy entropy and variable predictive model-based class discrimination*”. *Mechanism and Machine Theory* Volumen 78 (2014), Páginas 187–200.
- [28] Yongbin Liu, Bing He, Fang Liu, Siliang Lu, YileiZhao. “*Feature fusion using kernel joint approximate diagonalization of eigen-matrices for rolling bearing fault identification*”. *Journal of Sound and Vibration* Volumen 385 (2016), Páginas 389–401.
- [29] Piero Baraldi, Francesco Cannarile, Francesco Di Maio & Enrico Zio. “*Hierarchical k-nearest neighbours classification and binary differential evolution for fault diagnostics of automotive bearings operating under variable conditions*”. *Engineering Applications of Artificial Intelligence* Volumen 56 (2016), Páginas 1–13.