

## **Capítulo 9 Estimación de la Vida Útil de la Turbina de Vapor de 110MW Utilizando Redes Neuronales Artificiales**

### **Chapter 9 Steam Turbine Life Estimation at 110 MW Using Artificial Neural Networks**

YOUNESS, El Hamzaoui†\*, ÁLVAREZ-ARELLANO, Juan A y VÁZQUEZ-ÁVILA, José L.

*Universidad Autónoma del Carmen, Campus III, Facultad de Ingeniería, Avenida Central S/N, Esq. Con Fracc. Mundo Maya, C.P. 24115, Ciudad del Carmen, Camp., Mex.*

ID 1<sup>er</sup> Autor: *El Hamzaoui, Younes* / **ORC ID:** 0000-0001-5287-1594, **CVU CONACYT ID:** 292367

ID 1<sup>er</sup> Coautor: *Juan A. Álvarez-Arellano* / **CVU CONACYT ID:** 27636

ID 2<sup>do</sup> Coautor: *José L. Vázquez Ávila* / **ORC ID:** 0000-0002-9654-2431, **CVU CONACYT ID:** 102415

**DOI:** 10.35429/H.2021.16.81.96

E.H. Youness, J. Álvarez y J. Vázquez

\*eyouness@pampano.unacar.mx

S. Figueroa (AA.). Aplicaciones Láser en la Ingeniería. Handbooks-TI-©ECORFAN-Mexico, 2021

## Resumen

Las turbinas de vapor tienen muchas aplicaciones en diversos sectores industriales y, según la experiencia común, los fallos de los álabes son el principal origen de las averías de funcionamiento de estas máquinas, lo que provoca grandes pérdidas económicas en la industria de la turbomaquinaria. Las turbinas están diseñadas para trabajar en condiciones estables de funcionamiento. Sin embargo, se han producido fallos en los álabes tras un corto periodo de tiempo de trabajo.

Estos fallos se atribuyen comúnmente a la tensión de resonancia de los álabes en diferentes etapas a determinadas frecuencias de excitación. Se desarrolló un enfoque de red neuronal artificial (RNA) para predecir la vida útil de los álabes. La configuración 08-40-1 (08 entradas, 40 neuronas ocultas y 1 de salida) presentó una excelente concordancia ( $R^2 = 0,97$  y  $RMSE = 0,000255$ ) entre el valor de la vida útil experimental y el simulado considerando la función de transferencia sigmoideal tangente hiperbólica y lineal en la capa oculta y en la capa de salida, respectivamente.

En el siguiente estudio, se llevó a cabo el análisis de sensibilidad, y mostró, además, que todas las variables de entrada (amortiguación, densidad, presión, temperatura, flujo másico, entalpía, frecuencia y magnitudes de vibración) tienen un fuerte efecto en las turbinas de vapor. Sin embargo, la presión del vapor es el parámetro más influyente, con una importancia relativa del 40%, seguido del flujo másico. Por lo tanto, los resultados mostraron que la modelación y simulación con redes neuronales podía predecir y simular de forma eficaz el comportamiento de la estimación del ciclo de vida en los álabes de las turbinas de vapor tipo 110MW.

## Redes Neuronales Artificiales, Análisis de fatiga, Grietas de turbina de vapor, Evaluación de la vida útil

### Abstract

Steam turbines have many applications in various industrial sectors and, according to common experience, blade failures are the main source of malfunctions of these machines, causing large economic losses in the turbomachinery industry. Turbines are designed to operate under stable operating conditions. However, blade failures have occurred after a short period of operation.

These failures are commonly attributed to blade resonance stress at different stages at certain excitation frequencies. An artificial neural network (ANN) approach was developed to predict blade life. The 08-40-1 configuration (08 inputs, 40 hidden and 1 output neurons) presented an excellent agreement ( $R^2 = 0.97$  and  $RMSE = 0.000255$ ) between the experimental and simulated lifetime value considering the hyperbolic and linear tangent sigmoidal transfer function in the hidden and output layers, respectively. In the following study, the sensitivity analysis was carried out, and further showed that all the input variables (damping, density, pressure, temperature, mass flow, enthalpy, frequency and vibration magnitudes) have a strong effect on steam turbines.

However, steam pressure is the most influential parameter, with a relative importance of 40%, followed by mass flow. Therefore, the results showed that modeling and simulation with neural networks could effectively predict and simulate the life cycle estimation behavior in 110MW type steam turbine blades.

## Artificial Neural Networks, Fatigue Analysis, Steam Turbine Cracking, Life Cycle Assessment

### 1. Introducción

Uno de los equipos más importantes en las centrales de generación de energía, específicamente en las centrales termoeléctricas son las turbinas de vapor (Figura 1.1.). Sus componentes principales son el rotor y el estator. Un rotor está compuesto por varias etapas llamadas L-0, L-1, L-2, etc. Cada etapa contiene un número determinado de álabes que permiten a través de un flujo de vapor generar movimiento.

**Figura 1** Turbina de vapor de 110MW



En una turbina de vapor el fluido entra a alta presión y se expande pasando por varias etapas. Cuando llega a las últimas etapas presenta baja presión y temperatura. En las primeras etapas también llamadas de alta presión, los alabes son cortos creciendo en forma gradual en las etapas intermedias hasta llegar a las últimas etapas también conocidas como de baja presión, estas contienen los alabes más largos, pero igual los cortos que presentan mayor rapidez y por lo tanto sus frecuencias naturales son altas. A medida que el tamaño de los alabes va aumentando, su rigidez va disminuyendo y como consecuencia sus frecuencias naturales también disminuyen.

Esto está en el hecho de que la frecuencia natural depende de la rapidez; a mayor rapidez pues mayor será la frecuencia natural y a menor rigidez, la frecuencia natural será menor. El cálculo de la frecuencia natural es un parámetro de gran importancia ya que permite conocer la frecuencia a la cual un sistema no debe ser excitado, ya que si esto sucede puede aparecer el efecto de la resonancia manifestándose como grandes amplitudes de vibración. Existen diferentes métodos y formas para determinar la frecuencia natural de elementos o sistemas vibratorios, algunos de ellos son analíticos otros son experimentales y en ciertos casos se combinan ambos.

En las turbinas se presenta el fenómeno de resonancia el cual tiene lugar cuando la frecuencia de excitación coincide con la frecuencia natural del sistema. Esto se da porque las turbinas de vapor presentan puntos de resonancia que son causados por la excitación forzada del paso de vapor por las toberas. Las turbinas de vapor están diseñadas para trabajar en condiciones estables de operación, sin embargo, presentan paros y arranques ocasionando que los alabes pasen por sus puntos de resonancia. Estos son causados por las armónicas de velocidad y las NPF (frecuencia de paso de vapor por las toberas, siglas en inglés). Por lo tanto, para conocer las velocidades críticas en cada etapa de la turbina es necesario dibujar el diagrama de Campbell, trazando las frecuencias naturales de los alabes, las NPF y las excitaciones por revolución.

El tamaño de los álabes de la etapa de alta presión está limitado por la disponibilidad de nuevos materiales que puede resistir las condiciones de altas presiones y temperaturas. Mientras que, en los álabes largos de baja presión, las cargas centrífugas dictan la longitud a la cual los álabes pueden ser diseñados. Los alabes de las últimas etapas contribuyen en un mayor rendimiento debido a que producen más energía, pero son componentes que presentan mayores fallas por fractura. Puesto que los álabes de la última etapa son grandes y el flujo se encuentra a bajas presiones se ven afectados considerablemente por la velocidad de rotación y el flujo del vapor. Los álabes son miembros estructurales flexibles dando como resultado que un gran número de sus frecuencias naturales puedan coincidir con frecuencias de excitación. Esto origina que la turbina se diseñe para evitar resonancia en condiciones estables de operación, aunque esto no evita que experimente el fenómeno de resonancia durante paros y arranques. Podemos encontrar en la literatura que la resonancia es un fenómeno que ocurre cuando la frecuencia de excitación coincide con una de las frecuencias naturales, cuando esto sucede los esfuerzos se concentran en una cierta zona del material hasta llegar a la fractura del mismo, estas zonas de concentración de esfuerzos dependen de ciertos factores involucrados con el montaje, las condiciones de inyección del vapor y el diseño de los alabes.

Las turbinas de vapor por estar sujetas a fuerzas provenientes de diferentes fuentes (flujo de vapor, armónicas de giro, desbalance e imperfecciones en la garganta mínima) presentan cargas de tipo alternante y de inestabilidad aerodinámica, originando esfuerzos estáticos y dinámicos. Los esfuerzos estáticos combinados con alguno de los siguientes factores como: la discontinuidad en el material, la velocidad de rotación, el desafinamiento y el diseño del alabe podrían dar lugar al inicio de una grieta, además la resistencia de los alabes se basa únicamente en los esfuerzos estáticos. Los esfuerzos dinámicos son los principales causantes de fatiga ya que son el resultado de cargas cíclicas, estos esfuerzos se presentan por las armónicas de velocidad y por las fuerzas de excitación del vapor. El análisis de los esfuerzos en los alabes es complicado por su geometría y la iteración de varios modos de amortiguamiento en ellos. Las turbinas de vapor de 110 MW pueden ser afectadas ampliamente por la tensión, la fatiga, la corrosión, la fragilidad o el amortiguamiento.

También, el ambiente dentro de las turbinas de vapor y las conexiones de los alabes tiene un efecto negativo en la vida útil de los álabes, principalmente causado por el transporte y concentración de impurezas. De manera similar el material de construcción, sus propiedades como resistencia a la fatiga, esfuerzo de fluencia, esfuerzo último y el factor de amortiguamiento pueden afectar la vida esperada de los alabes. Los principales factores que afectan la vida útil de los álabes son: los esfuerzos, el ambiente, las propiedades del material, cargas estáticas (centrífugas y por presión de vapor) presentes en los álabes móviles, cargas dinámicas (fuerzas de vapor inestables, recirculación) presentes en los alabes móviles, amortiguamiento, frecuencias, vibraciones (se presentan durante el funcionamiento normal de las operaciones propagándose en fatiga), densidades y tensiones.

Nuestra evaluación de vida útil puede maximizar el ciclo de vida de dichas turbinas para el funcionamiento óptimo de una turbina de vapor, se requiere que sus componentes funcionen con eficiencia y con el menor porcentaje de pérdidas. Específicamente, los álabes deben conservar dos características principales: conservar su capacidad de mantener carga sin que sufra una ruptura; y conservar su rigidez dentro de los límites seguros para evitar deformaciones excesivas. Al combinarse diferentes valores de datos de entrada, la vida útil de estos componentes disminuye en gran medida; sin necesidad de presencia de resonancia, esto con la observación de que los valores existentes en la naturaleza no son de carácter determinístico asumiendo ciertos rasgos de aleatoriedad. Sin embargo, el acelerado avance del desarrollo tecnológico en las últimas décadas ha abierto las puertas de diversas áreas del conocimiento humano a la aplicación de la inteligencia artificial donde está el algoritmo Redes Neuronales Artificiales (RNA).

## **2. Materiales and metodología**

La evolución tecnológica, cuyos orígenes se remontan a los comienzos de la historia humana, ha tenido, a partir de la década de los cuarenta, un impulso extraordinario gracias al diseño y la progresiva popularización de máquinas llamadas computadoras u ordenadores. La computadora ha sido definida como una máquina capaz de realizar y controlar a gran velocidad cálculos y procesos complicados que requieren una toma rápida de decisiones. Su función consiste, por tanto, en tratar la información que se le suministra y proveer los resultados requeridos. Sin embargo, este enunciado tan simple está definiendo un hito tradicional. La computadora es, hoy por hoy, incapaz de hacer algo para lo que no ha sido programada. Sin embargo, el progreso que representa en la capacidad para realizar un número muy grande de cálculos a una alta velocidad (milésimas de segundo), permitió el desarrollo de modelos y algoritmos computacionales robustos donde se requiere de muchas operaciones (p.ej., algoritmos de inteligencia artificial: las redes neuronales artificiales, la lógica difusa, los algoritmos genéticos, entre otros, y los métodos estadísticos avanzados como es el caso del método de Monte Carlo).

### **2.1. Redes Neuronales Artificiales (RNA)**

Las neuronas se agrupan en distintas capas y combinados entre sí según una determinada arquitectura. Al igual que en la naturaleza, la función de la red es determinada en gran medida por las conexiones entre dos elementos (neuronas), cada conexión entre dos neuronas tiene un coeficiente de peso que se le atribuye. La estructura de la red estándar para una función de aproximación es la percepción de múltiples capas (feedforward). La red de alimentación hacia adelante tiene a menudo una o más capas ocultas de neuronas sigmoideas seguida de una capa de salida de neuronas lineales. Múltiples capas de neuronas con funciones de transferencia no lineal permiten que la red aprenda relaciones no lineales y lineales entre los vectores de entrada y salida.

La capa de salida lineal de la red permite producir valores fuera del rango de -1 a 1 [Aleboyeh et al. 2008]. Para la red, la anotación apropiada se utiliza en las redes de dos capas (capa oculta y capa de salida). El número de neuronas en las capas de entrada y salida se da, respectivamente, por el número de variables de entrada y de salida del proceso de investigación. El número óptimo de neuronas en la capa oculta (s) ns es difícil de precisar, y depende de la complejidad del proceso. Este número se determina generalmente de manera iterativa. Cada neurona en la capa oculta tiene bias b (umbral), que se añade a las entradas ponderadas para formar una n neurona (Ec. 1). Esta suma, n, es el argumento de la función de transferencia f.

$$n_s = Wi_{(1,1)}ln_1 + Wi_{(1,2)}ln_2 + \dots + Wi_{(1,k)}ln_k + b_1 \quad (1)$$

Los coeficientes asociados a la capa oculta se agrupan en matrices de conexión  $W_i$  (pesos) y  $b_i$  (bias). La capa de salida calcula la suma ponderada de las señales proporcionadas por la capa oculta, y los coeficientes asociados se agrupan en matrices de  $W_o$  y  $b_2$ . Usando la notación de matrices, la salida de la red puede ser dada por (Ec. 2):

$$Out = g(W_o \times f(W_i \times ln + b_1) + b_2) \quad (2)$$

Las neuronas en la capa oculta pueden utilizar cualquier función de transferencia para generar su producción. En este capítulo, se aplica una función de transferencia de tipo tangencial hiperbólica sigmoideal (TANSIG) en la capa oculta y se considera una función de transferencia lineal (PURELIN) en la capa de salida f y g, respectivamente [Demuth y Beale 1998]. El sistema ajusta los pesos de las conexiones internas para minimizar los errores entre la salida de la RNA versus la salida experimental, que pueden resumirse de la siguiente manera: al principio, proponer a un grupo de números aleatorios como los valores iniciales de los pesos y bias, a continuación, calcular la salida de todas las capas por capa de neuronas, comenzando con la capa de entrada, utilizando las Ecs. (3-6):

$$f = TANSIG(W_i * ln_k + b1_s) \quad (3)$$

$$f = \frac{2}{1 + \exp[-2*(W_i * ln_k + b1_s)]} - 1 \quad (4)$$

$$g = PURELIN(W_o * f + b2_l) \quad (5)$$

$$g = Out_l \quad (6)$$

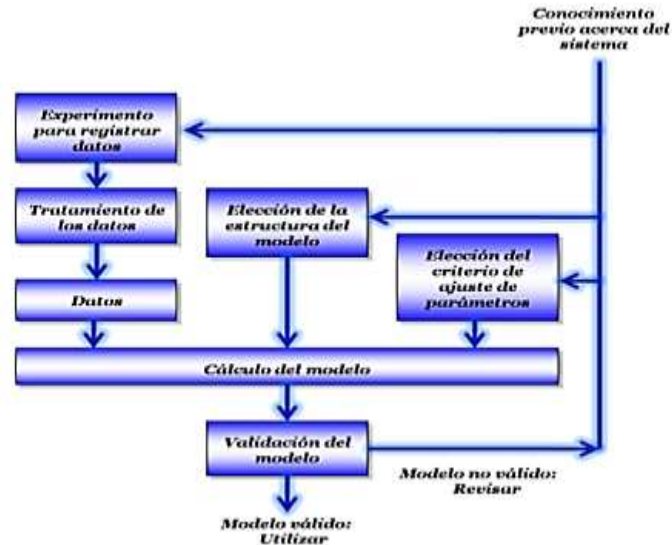
Si tenemos en cuenta las funciones de transferencia, la Ec. 6 puede ser expresada de la siguiente manera (Ec. 7):

$$Out_l = PURELIN\{W_o \times [TANSIG(W_i \times In_k + b1_s)] + b2_l\} \quad (7)$$

Dónde s es el número de neuronas en la capa oculta (S = 3), k es el número de neuronas en la capa de entrada (K=16). l es el número de neuronas en la capa de salida (l = 1).  $W_i$ ,  $W_o$  y  $b1_s$ ,  $b2_l$  son pesos y bias, respectivamente. La Ec. (7) no es compleja porque se compone de una operación aritmética simple. Por lo tanto, puede ser utilizada para la aplicación de estimación en línea de procesos industriales, en tiempo real. La RNA fue entrenada mediante el algoritmo de propagación hacia atrás "Backpropagation".

Cabe mencionar que todos los cálculos se realizaron con el software Matlab considerando la herramienta de neural networks toolbox (RNA). En la figura 2, se muestra el procedimiento de identificación neuronal.

**Figura 2** Procedimiento de identificación neuronal



## 2.2. Aprendizaje de la red neuronal

Un algoritmo de aprendizaje (o entrenamiento) se define como un procedimiento que consiste en ajustar los coeficientes (pesos y bias) de una RNA, para minimizar una función de error entre los valores de salida de la RNA y los valores de salida experimentales [Rumelhart et al. 1986]. Para determinar el algoritmo de aprendizaje más eficiente en el desarrollo de RNA, nosotros hemos evaluado diez algoritmos de propagación hacia atrás (ver Tabla 4.1). Este análisis utiliza cinco neuronas en la capa oculta para todos los algoritmos de propagación. El análisis demostró que el algoritmo de entrenamiento de Levenberg-Marquardt tiene menor valor de la raíz de la sumatoria cuadrada del error promedio (*RMSE*). También, encontramos que este algoritmo realiza menor tiempo de procesamiento computacional, lo cual lo hace muy rápido en comparación con otros algoritmos de entrenamiento. Lo anterior es debido a que el algoritmo de Levenberg-Marquardt fue diseñado para aproximar las segundas derivadas parciales sin tener que calcular la matriz Hessiana. Cuando la función del desempeño tiene la forma de una suma de cuadrados, entonces la matriz Hessiana se puede aproximar como:

$$H = J^T J \quad (8)$$

El gradiente se puede calcular como:

$$g = J^T e \quad (9)$$

Donde,  $J$ , es la matriz Jacobiana que contiene las primeras derivadas de los errores de la red con respecto a los pesos y bias, y  $e$ , es un vector de error de la RNA. La matriz jacobiana puede calcularse por medio de una técnica de retropropagación estándar que es mucho menos compleja que la matriz Hessiana. El algoritmo de Levenberg - Marquardt utiliza esta aproximación (Ec.10):

$$X_{k+1} = X_k - [J^T J + \mu I]^{-1} J^T e \quad (10)$$

Cuando el escalar,  $\mu$ , es cero, esto es sólo el método de Newton utilizando la matriz Hessiana aproximada. Cuando  $\mu$  es grande, esto se convierte en gradiente descendente con un tamaño de pequeño paso [Khatae y Kasiri 2010].

## 2.3. Funciones de transferencias empleadas en redes neuronales

Las funciones de transferencias son funciones matemáticas que juegan un papel significativo en el proceso de aprendizaje de redes neuronales. Las principales funciones de transferencias usadas están mostradas en la figura 3:

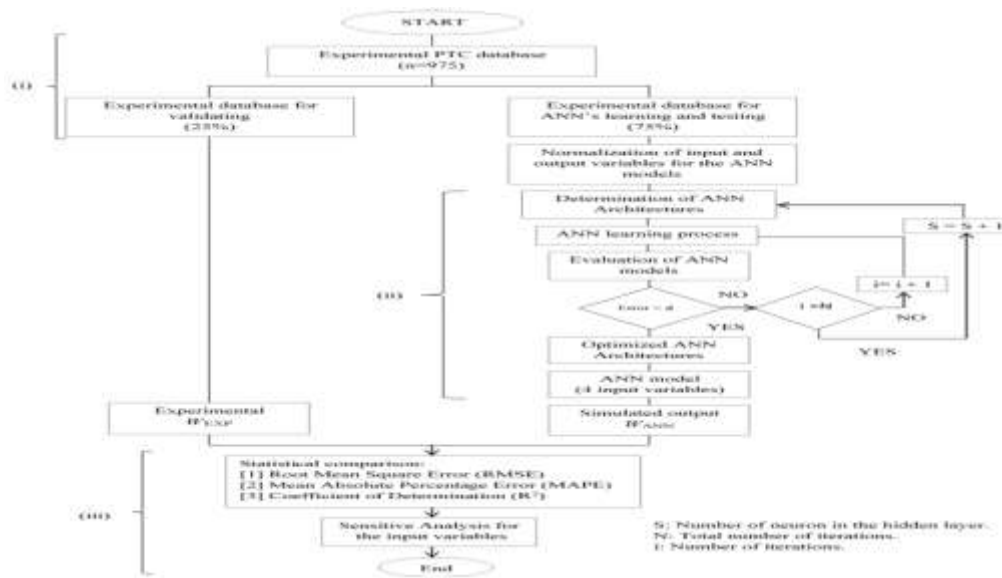
**Figura 3** Funciones de transferencias empleadas en redes neuronales

Función	Ecuación	Símbolo	Descripción
Limitador fuerte	$a = \begin{cases} 1 & \text{si } n \geq 0, \\ 0 & \text{si } n < 0 \end{cases}$		Permite a la neurona clasificar las entradas en dos categorías diferentes, Cierto (1) o Falso (0).
Limitador fuerte simétrico	$a = \begin{cases} 1 & \text{si } n \geq 0, \\ -1 & \text{si } n < 0 \end{cases}$		Al igual que el limitador fuerte, fuerza a la neurona para generar una salida binaria, ahora entre 1 y -1 con lo cual abarca un mayor rango de valores.
Lineal	$a = n$		La salida de esta función de transferencia es idéntica al valor de la entrada neta.
Lineal saturada	$a = \begin{cases} 0 & \text{si } n < 0, \\ n & \text{si } 0 \leq n \leq 1, \\ 1 & \text{si } n > 1 \end{cases}$		La función toma los valores de entrada netas y reduce la salida en un rango de 0 a 1. Se emplea cuando se poseen variaciones suaves de valores positivos.
Lineal saturada simétrica	$a = \begin{cases} -1 & \text{si } n < 0, \\ n & \text{si } 0 \leq n \leq 1, \\ 1 & \text{si } n > 1 \end{cases}$		La función toma los valores de entrada netas y reduce la salida en un rango de -1 a 1. Se emplea cuando se poseen variaciones suaves de valores positivos y negativos.
Logarítmica sigmoidea	$a = \frac{1}{1 + e^{-n}}$		Esta función de transferencia presenta una rápida convergencia. La función toma los valores de entrada netas y reduce la salida en un rango de 0 a 1.
Tangente hiperbólica sigmoidea	$a = \frac{2}{1 + e^{-2n}} - 1$		Similar a la anterior, cubre un mayor rango de valores (-1 a 1). Se emplea cuando la velocidad es más importante que la forma de la función de transferencia.

Sin embargo, en la figura 3 muestra el diagrama de flujo para la obtención del modelo RNA para modelar y simular la vida útil de la turbina (VU):

El rendimiento del modelo de RNA es evaluado estadísticamente: por la raíz de la sumatoria cuadrada del error promedio (RMSE) [Hamed et al. 2004; Hernández et al. 2004], por la desviación estándar relativa (RSD) y por el coeficiente de regresión (R<sup>2</sup>). Lo anterior se calcula con los valores experimentales y las predicciones de la RNA. Estas Ecs. 11-13, son consideradas como criterio de evaluación de los modelos de RNA:

**Figura 4** Diagrama de flujo del RNA para modelar y simular VU



$$RMSE = \sqrt{\frac{\sum(y_{q,pred} - y_{q,exp})^2}{Q}} \tag{11}$$

$$RSD = \frac{RMSE}{y_m} \times 100 \tag{12}$$

$$R^2 = 1 - \frac{\sum_{q=1}^Q (y_{q,pred} - y_{q,exp})^2}{\sum_{q=1}^Q (y_{q,exp} - y_m)^2} \quad (13)$$

Dónde Q es el número de datos,  $y_{q,pred}$  es el valor simulado de la RNA,  $y_{q,exp}$  es el valor experimental,  $y_m$  es el promedio de los valores experimentales.

### 3. Resultados

Un algoritmo de aprendizaje (o entrenamiento) se define como un procedimiento que consiste en ajustar los coeficientes (pesos y sesgos) que representen los parámetros de la red neuronal, para minimizar una función de error (generalmente cuadrática) entre las salidas de la red, para un conjunto dado de entradas, y las salidas correctas (ya conocidas). Si se utilizan no linealidades suaves, el gradiente de la función de error se puede calcular mediante el procedimiento clásico de retropropagación. Para determinar el mejor algoritmo de entrenamiento, se estudiaron 10 algoritmos de retropropagación. Además, se usaron cuarenta neuronas en la capa oculta para todos los algoritmos de retropropagación. La Tabla 1, muestra una comparación de diferentes algoritmos de entrenamiento de retropropagación. El algoritmo de entrenamiento de retropropagación de Levenberg-Marquardt podría tener un error cuadrático medio (RMSE) más pequeño; por otro lado, encontramos que el entrenamiento con el algoritmo Levenberg Marquardt puede ejecutarse sin problemas en una computadora con una menor especificación de memoria expandida y el tiempo de entrenamiento es más rápido, al respecto de otros algoritmos de retropropagación. Porque, el algoritmo Levenberg-Marquardt fue diseñado para acercarse a la velocidad de entrenamiento de segundo orden sin tener que calcular la matriz hessiana.

**Tabla 1** Comparación de 10 algoritmos de Retro-propagación con cinco neuronas en la capa oculta

Algoritmo de Retro-propagación	Función	Error cuadrático medio (RMSE)	Época	Coefficiente de correlación ( $R^2$ )	Mejor ecuación lineal
Retropropagación Levenberg-Marquardt	trainlm	0.00235005	1000	0.990	Y=0.990X+0.306
Descenso de gradiente por lotes	traingd	0.01657932	2000	0.998	Y=0.986X+0.927
Descenso de gradiente por lotes con impulso	traingdm	0.01982303	2000	0.987	Y=0.988X+0.837
Retropropagación de gradiente conjugado de Polak-Ribiere	traincgp	0.03267017	2000	0.979	Y=0.957X+2.53
Retropropagación de gradiente conjugado escalado	traincsg	0.44944913	2000	0.974	Y=1.020X-0.783
Retropropagación BFGS cuasi-Newton	trainbfg	0.48619630	2000	0.971	Y=0.982X+1.23
Retropropagación de gradiente conjugado Powell-Beale	traincgb	0.50820237	2000	0.965	Y=0.960X+2.03
Retropropagación secante de un paso	trainoss	0.02753381	2000	0.782	Y=0.617X+45.3
Retropropagación de gradiente conjugado de Fletcher-Reeves	traincgf	0.01756329	2000	0.725	Y=0.425X+34.8
Tasa de aprendizaje variable por retropropagación	traingdx	0.20039637	2000	0.718	Y=0.386X+38

Cuando la función del error tiene la forma de una suma de cuadrados (como es típico en el entrenamiento de redes de retroalimentación), entonces la matriz de Hessiana puede ser aproximada como:

$$H = J^T J \quad (13)$$

Y el gradiente se puede calcular como:

$$g = J^T e \quad (14)$$



Donde  $J$  es la matriz jacobiana que contiene las primeras derivadas de los errores de red con respecto a los pesos y umbrales, y es un vector de errores de red. La matriz jacobiana se puede calcular a través de una técnica de retropropagación estándar que es mucho menos complejo que calcular la matriz de Hessiana. El algoritmo Levenberg-Marquardt utiliza esta aproximación a la matriz Hessiana en la siguiente actualización de Newton:

$$X_{k+1} = X_k - [J^T J + \mu I]^{-1} J^T e \quad (15)$$

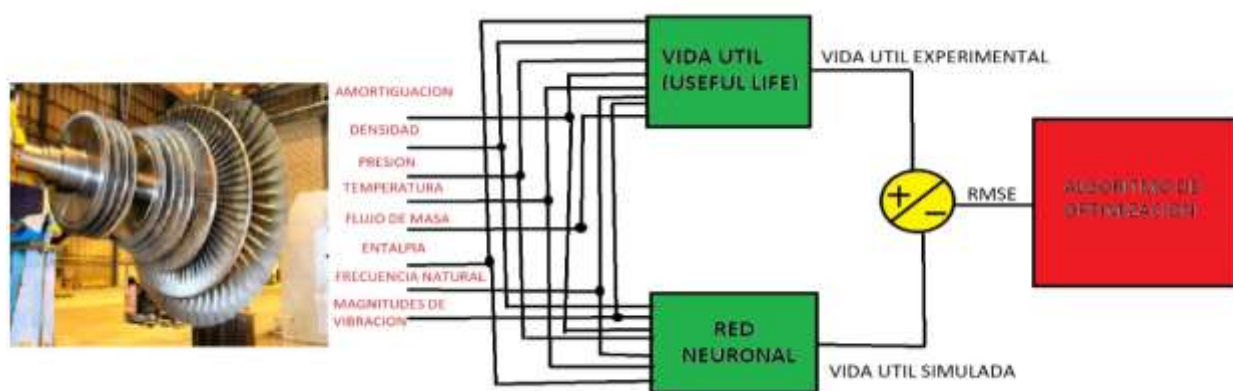
Cuando el sintonizador ( $\mu$ ) es cero, este es solo el método de Newton, utilizando la matriz aproximada Hessiana. Cuando  $\mu = 1$  es grande, éste se convierte en gradiente descendente con un pequeño tamaño de escalón. El método de Newton es más rápido y más preciso cerca de un mínimo de error, por lo que el objetivo es cambiar hacia el método de Newton lo más rápido posible, por lo tanto,  $\mu$  se reduce después de cada paso exitoso (reducción en la función del error) y se incrementa solo cuando un paso tentativo aumentaría la función del desempeño del error. En este contexto, la función del error siempre se reduce en cada iteración del algoritmo [47]. Entonces, para esta motivación, el algoritmo de Levenberg-Marquardt se consideró el algoritmo de entrenamiento en el presente estudio de investigación. Sin embargo, el rendimiento del modelo RNA ha sido medido estadísticamente por el error cuadrático medio (RMSE), como sus siglas en inglés (Root Mean Square Error) y coeficiente de regresión, que se calcula con los valores experimentales y las predicciones de la red. Estos cálculos son utilizados como criterio para encontrar el modelo adecuado (figura 5), obtenido de la siguiente manera:

$$RMSE = \sqrt{\frac{\sum_{q=1}^Q (y_{q,pred} - y_{q,exp})^2}{Q}} \quad (16)$$

$$R^2 = 1 - \frac{\sum_{q=1}^Q (y_{q,pred} - y_m)^2}{\sum_{q=1}^Q (y_{q,exp} - y_m)^2} \quad (17)$$

Donde  $Q$  es el número de datos experimentales,  $y_{q,pred}$  es la predicción de la red,  $y_{q,exp}$  es la respuesta experimental,  $y_m$  es el promedio de valores reales y  $q$  es un índice de datos. En consecuencia, RMSE se utilizó como la función de error que mide el rendimiento de la red. Por lo tanto, la red que tiene un RMSE mínimo y un  $R^2$  máximo se seleccionó como el mejor modelo neuronal de RNA.

**Figura 5** Arquitectura del algoritmo neuronal al respecto de la vida útil y el procedimiento utilizado para el algoritmo aprendizaje



Como mencionamos anteriormente, las variables de entrada a RNA fueron la presión, flujo de masa, amortiguamiento, frecuencia, vibración, entalpia, temperatura y densidad del vapor. Sin embargo, la vida útil fue salida. Las características de las variables de entrada y salida se muestran en la tabla 2, respectivamente.

**Tabla 2** Características de las variables de entrada y salida del modelo de red neuronal artificial

Variables de capa de entrada	Rango
Amortiguación ( $N \frac{s}{m}$ )	[133,171]
Frecuencia natural (Hz)	[8139,11255]
Magnitudes de vibración ( $\frac{m}{s^2}$ )	[4472,8236]
Densidad ( $\frac{Kg}{m^3}$ )	[370,398]
Presión (MPa)	[1.46,3.34]
Temperatura (°C)	[385,420]
Flujo de masa ( $\frac{Kg}{s}$ )	[551,592]
Entalpia (KJ)	[421062,816958]
Variable de capa de salida	Rango
UL (Vida útil, por siglas en inglés) (Min)	[4850000,5270000]

La topología de una red neuronal artificial está determinada por el número de capas, el número de nodos en cada capa y La naturaleza de las funciones de transferencia. La optimización de la topología RNA es probablemente el paso más importante en el desarrollo de un modelo [47].

Para determinar el número óptimo de neuronas en la capa oculta, se utilizó una serie de topologías, en las que el número de neuronas varió de 1 a 40. Todos los RNA fueron entrenados utilizando diez algoritmos de retropropagación, como ha sido mencionado en la tabla 5.1. La capacitación en red es un proceso mediante el cual el peso y el umbral de conexión en el RNA se adaptan a través de un proceso continuo de simulación por el entorno en el que está incrustada la red. El objetivo principal del entrenamiento es minimizar la función de error (RMSE) buscando un conjunto de pesos (weight) y bias (umbrales) de conexión que provoque que el RNA genera salidas que sean iguales o cercanas al valor objetivo. En otras palabras, el algoritmo de retropropagación minimiza el RMSE entre la salida observada y la predicha en la capa de salida, a través de dos fases. En la fase de avance, la entrada externa señales de información en las neuronas de entrada que se propagan hacia adelante para calcular la señal de información de salida en la neurona de salida.

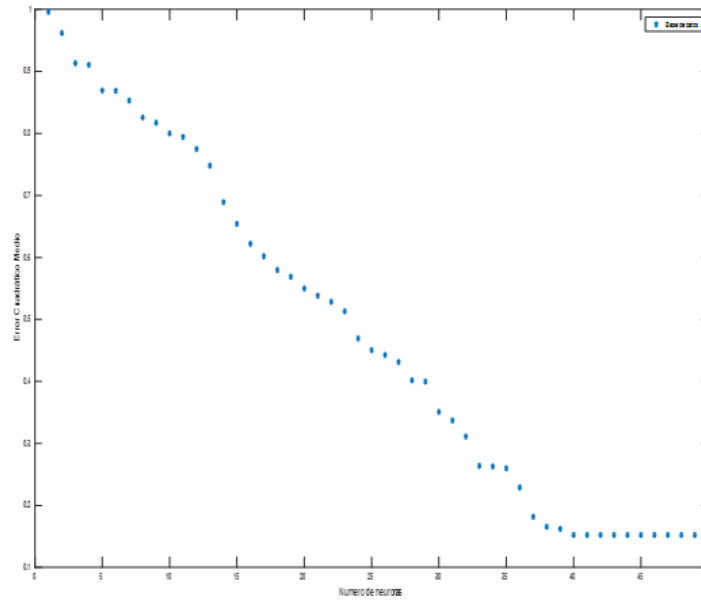
En la fase hacia atrás, se realizan modificaciones a las fuerzas de conexión, en función de la diferencia en las señales de información predichas y observadas en la neurona de salida [47].

Los conjuntos de datos se dividieron en subconjuntos de entrenamiento, validación y prueba, cada uno de los cuales contiene 5000, 2500 y 2500 muestras, respectivamente. Los conjuntos de validación y prueba, para la evaluación y el modelado de las redes, se seleccionaron aleatoriamente de los datos experimentales. Como la función de transferencia utilizada en la capa oculta era sigmoideal (tansig), todas las muestras deben ser normalizado en el rango de 0.1-0.9. Entonces, todos los conjuntos de datos de entrada  $x_i$  (de los conjuntos de entrenamiento, validación y prueba) se escalaron a un nuevo valor  $x_i$  de la siguiente manera:

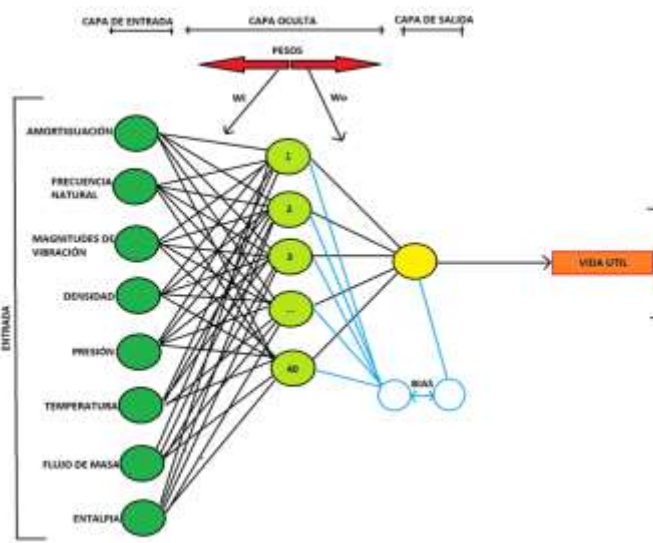
$$x_i = 0.8 \left( \frac{x_i - x_{min}}{x_{max} - x_{min}} \right) + 0.1 \quad (18)$$

La topología final se obtuvo después de 20000 ejecuciones de 1000 iteraciones (epoch) a partir de pesos iniciales aleatorios. Para cada corrida, se calculó el error de red contra el número de neuronas en la capa oculta. En la figura 6 se ilustra el error de red frente al número de neuronas en la capa oculta. Se descubrió que el rendimiento de la red se estabilizó con cuarenta neuronas en la capa oculta (8-40-1). Entonces, en base a la aproximación de la función del erro (RMSE), se utilizaron una cantidad de neuronas en la capa oculta igual a cuarenta, y una red neuronal de propagación de retroalimentación de tres capas para modelar el proceso como se muestra en la figura 7.

**Figura 6** Numero óptimo de neuronas en la capa oculta



**Figura 7** Modelo de predicción de los valores de la vida útil



**3.1. Modelo propuesto de red neuronal**

Se encontró que una red neuronal con cuarenta neuronas en la capa oculta (que involucra 361 coeficientes: 320 pesos y 41 bias) es eficiente en la predicción de la vida útil. La figura 8, presenta datos experimentales y simulados disponibles para la base de datos de aprendizaje y también ilustra datos experimentales y simulados para probar la base de datos.

Parámetros ajustables obtenidos (pesos y bias) en el modelo propuesto S=40, K=8.														
Pesos de la capa oculta									Pesos de la capa de salida		Bias de la capa oculta		Bias de la capa de salida	
$IW_{(s)}$	$W_{i(1,1)}$	$W_{i(1,2)}$	$W_{i(1,3)}$	$W_{i(1,4)}$	$W_{i(1,5)}$	$W_{i(1,6)}$	$W_{i(1,7)}$	$W_{i(1,8)}$	$W_{0(s)}$	$W_{0(1)}$	$b1_{(s)}$	$b1_{(1)}$	$b2$	$b2$
	1.51074	1.71357	1.28095	1.69061	1.54533	1.73064	1.34225	1.35083		1.25169		3.86021		1.25175
	55e+01	66e+01	00e+01	98e+01	18e+01	13e+01	98e+01	76e+01		77e+05		71e+01		92e+05
	$W_{i(2,1)}$	$W_{i(2,2)}$	$W_{i(2,3)}$	$W_{i(2,4)}$	$W_{i(2,5)}$	$W_{i(2,6)}$	$W_{i(2,7)}$	$W_{i(2,8)}$		$W_{0(2)}$		$b1_{(2)}$		
	5.12820	7.54318	5.63457	5.29120	5.00392	5.86888	5.57222	5.60623		1.25140		1.25411		
	82e+01	83e+01	77e+01	83e+01	40e+01	87e+01	55e+01	62e+01		59e+05		74e+02		
	$W_{i(3,1)}$	$W_{i(3,2)}$	$W_{i(3,3)}$	$W_{i(3,4)}$	$W_{i(3,5)}$	$W_{i(3,6)}$	$W_{i(3,7)}$	$W_{i(3,8)}$		$W_{0(3)}$		$b1_{(3)}$		
	-	-	-	-	-	-	-	-		-		-		
	3.79821	5.71293	4.02184	4.66912	4.53767	4.73570	4.25122	4.51770		1.25110		9.54908		
	51e+01	88e+01	35e+01	32e+01	28e+01	93e+01	82e+01	53e+01		83e+05		46e+01		
	$W_{i(4,1)}$	$W_{i(4,2)}$	$W_{i(4,3)}$	$W_{i(4,4)}$	$W_{i(4,5)}$	$W_{i(4,6)}$	$W_{i(4,7)}$	$W_{i(4,8)}$		$W_{0(4)}$		$b1_{(4)}$		
	-	-	-	-	-	-	-	-		-		-		
	4.89639	7.41272	5.82318	5.50429	5.02979	6.12251	5.42797	6.02551		1.25142		1.23950		
	45e+01	32e+01	28e+01	15e+01	56e+01	26e+01	68e+01	65e+01		65e+05		78e+02		
	$W_{i(5,1)}$	$W_{i(5,2)}$	$W_{i(5,3)}$	$W_{i(5,4)}$	$W_{i(5,5)}$	$W_{i(5,6)}$	$W_{i(5,7)}$	$W_{i(5,8)}$		$W_{0(5)}$		$b1_{(5)}$		



-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4.38806 29e+01	5.72364 21e+01	3.92301 87e+01	3.72452 01e+01	4.90864 16e+01	4.88434 60e+01	3.80825 86e+01	2.86852 40e+01	1.25167 34e+05	-	9.49940 78e+01	-	-	-	-
$W_{i(30,1)}$	$W_{i(30,2)}$	$W_{i(30,3)}$	$W_{i(30,4)}$	$W_{i(30,5)}$	$W_{i(30,6)}$	$W_{i(30,7)}$	$W_{i(30,8)}$	$W_{o(30)}$	-	$b_{1(30)}$	-	-	-	-
5.56560 41e+00	4.03255 91e+00	2.80019 79e+00	3.03525 50e+00	1.17332 51e+00	1.23531 85e+00	6.82084 62e-01	3.61902 29e+00	1.25174 46e+05	-	9.94676 50e+00	-	-	-	-
$W_{i(31,1)}$	$W_{i(31,2)}$	$W_{i(31,3)}$	$W_{i(31,4)}$	$W_{i(31,5)}$	$W_{i(31,6)}$	$W_{i(31,7)}$	$W_{i(31,8)}$	$W_{o(31)}$	-	$b_{1(31)}$	-	-	-	-
2.04825 61e+01	2.35039 07e+01	1.82245 97e+01	1.88478 93e+01	1.86643 86e+01	1.98395 01e+01	1.64030 02e+01	1.59650 35e+01	1.25165 91e+05	-	4.13629 02e+01	-	-	-	-
$W_{i(32,1)}$	$W_{i(32,2)}$	$W_{i(32,3)}$	$W_{i(32,4)}$	$W_{i(32,5)}$	$W_{i(32,6)}$	$W_{i(32,7)}$	$W_{i(32,8)}$	$W_{o(32)}$	-	$b_{1(32)}$	-	-	-	-
8.88676 99e-01	5.18111 29e+00	2.39821 15e+00	6.36144 46e+00	1.92355 35e+00	6.27824 62e+00	1.66083 35e-01	6.46595 74e+00	1.24278 11e+05	-	8.74544 99e+00	-	-	-	-
$W_{i(33,1)}$	$W_{i(33,2)}$	$W_{i(33,3)}$	$W_{i(33,4)}$	$W_{i(33,5)}$	$W_{i(33,6)}$	$W_{i(33,7)}$	$W_{i(33,8)}$	$W_{o(33)}$	-	$b_{1(33)}$	-	-	-	-
2.39881 26e+01	2.77675 92e+01	1.26300 66e+01	2.62105 01e+01	2.39902 70e+01	2.99946 54e+01	1.93047 96e+01	1.60488 11e+01	1.25086 25e+05	-	4.59130 29e+01	-	-	-	-
$W_{i(34,1)}$	$W_{i(34,2)}$	$W_{i(34,3)}$	$W_{i(34,4)}$	$W_{i(34,5)}$	$W_{i(34,6)}$	$W_{i(34,7)}$	$W_{i(34,8)}$	$W_{o(34)}$	-	$b_{1(34)}$	-	-	-	-
1.80723 64e+01	2.28558 12e+01	2.56585 85e+01	1.09565 55e+01	1.11596 42e+01	1.73082 52e+01	1.57228 50e+01	1.71819 67e+01	1.25087 31e+05	-	4.83774 67e+01	-	-	-	-
$W_{i(35,1)}$	$W_{i(35,2)}$	$W_{i(35,3)}$	$W_{i(35,4)}$	$W_{i(35,5)}$	$W_{i(35,6)}$	$W_{i(35,7)}$	$W_{i(35,8)}$	$W_{o(35)}$	-	$b_{1(35)}$	-	-	-	-
5.60621 69e+01	8.09629 48e+01	6.40119 01e+01	5.64879 39e+01	5.32657 79e+01	5.96255 23e+01	5.96085 51e+01	6.71414 40e+01	1.25100 92e+05	-	1.30451 78e+02	-	-	-	-
$W_{i(36,1)}$	$W_{i(36,2)}$	$W_{i(36,3)}$	$W_{i(36,4)}$	$W_{i(36,5)}$	$W_{i(36,6)}$	$W_{i(36,7)}$	$W_{i(36,8)}$	$W_{o(36)}$	-	$b_{1(36)}$	-	-	-	-
5.03013 45e+00	3.24844 06e+00	1.42112 58e+00	9.91871 86e-01	4.69167 14e+00	1.09066 12e+00	3.46455 25e+00	5.57922 10e+00	1.25175 59e+05	-	9.57877 87e+00	-	-	-	-
$W_{i(37,1)}$	$W_{i(37,2)}$	$W_{i(37,3)}$	$W_{i(37,4)}$	$W_{i(37,5)}$	$W_{i(37,6)}$	$W_{i(37,7)}$	$W_{i(37,8)}$	$W_{o(37)}$	-	$b_{1(37)}$	-	-	-	-
6.51286 71e+00	1.01817 02e+01	1.10794 33e+01	9.02267 37e+00	7.07419 18e+00	8.62556 19e+00	4.75489 80e+00	6.27608 70e+00	1.25173 50e+05	-	1.69752 64e+01	-	-	-	-
$W_{i(38,1)}$	$W_{i(38,2)}$	$W_{i(38,3)}$	$W_{i(38,4)}$	$W_{i(38,5)}$	$W_{i(38,6)}$	$W_{i(38,7)}$	$W_{i(38,8)}$	$W_{o(38)}$	-	$b_{1(38)}$	-	-	-	-
6.47420 37e+01	9.36881 15e+01	7.18838 81e+01	6.92895 99e+01	6.85421 77e+01	7.94263 16e+01	6.77379 83e+01	7.36591 70e+01	1.25117 34e+05	-	1.48468 67e+02	-	-	-	-
$W_{i(39,1)}$	$W_{i(39,2)}$	$W_{i(39,3)}$	$W_{i(39,4)}$	$W_{i(39,5)}$	$W_{i(39,6)}$	$W_{i(39,7)}$	$W_{i(39,8)}$	$W_{o(39)}$	-	$b_{1(39)}$	-	-	-	-
4.30747 41e+00	5.28675 74e+00	1.78862 94e+00	2.56407 56e+00	2.03445 63e+00	5.38071 35e+00	5.27702 07e+00	3.68993 02e+00	1.25174 74e+05	-	1.40001 31e+01	-	-	-	-
$W_{i(40,1)}$	$W_{i(40,2)}$	$W_{i(40,3)}$	$W_{i(40,4)}$	$W_{i(40,5)}$	$W_{i(40,6)}$	$W_{i(40,7)}$	$W_{i(40,8)}$	$W_{o(40)}$	-	$b_{1(40)}$	-	-	-	-
2.11706 49e+01	3.08652 14e+01	2.69385 68e+01	3.40781 73e+01	2.70030 56e+01	2.75369 36e+01	2.26441 81e+01	3.54436 53e+01	1.25085 40e+05	-	5.63768 68e+01	-	-	-	-

$$UsefulLife (UL) = \sum_{S=1}^S \left[ W_{o(1,S)} \left( \frac{2}{1 + \exp(-2(\sum_{k=1}^K (W_{i(s,k)} \ln(k)) + b_{1(s)}))} - 1 \right) \right] + b_{2(l)} \quad (19)$$

Donde  $S$  es el número de neuronas en la capa oculta ( $S = 40$ ),  $k$  es el número de la entrada ( $K = 8$ ), y  $W_i$ ,  $W_o$  y  $b$  son peso y bias, respectivamente.

$$\begin{aligned}
 UL = & 2 \left[ \frac{W_{o(1,1)}}{1+e^{X_1}} + \frac{W_{o(1,2)}}{1+e^{X_2}} + \frac{W_{o(1,3)}}{1+e^{X_3}} + \frac{W_{o(1,4)}}{1+e^{X_4}} + \frac{W_{o(1,5)}}{1+e^{X_5}} + \frac{W_{o(1,6)}}{1+e^{X_6}} + \frac{W_{o(1,7)}}{1+e^{X_7}} + \frac{W_{o(1,8)}}{1+e^{X_8}} + \frac{W_{o(1,9)}}{1+e^{X_9}} + \frac{W_{o(1,10)}}{1+e^{X_{10}}} + \frac{W_{o(1,11)}}{1+e^{X_{11}}} + \frac{W_{o(1,12)}}{1+e^{X_{12}}} + \right. \\
 & \frac{W_{o(1,13)}}{1+e^{X_{13}}} + \frac{W_{o(1,14)}}{1+e^{X_{14}}} + \frac{W_{o(1,15)}}{1+e^{X_{15}}} + \frac{W_{o(1,16)}}{1+e^{X_{16}}} + \frac{W_{o(1,17)}}{1+e^{X_{17}}} + \frac{W_{o(1,18)}}{1+e^{X_{18}}} + \frac{W_{o(1,19)}}{1+e^{X_{19}}} + \frac{W_{o(1,20)}}{1+e^{X_{20}}} + \frac{W_{o(1,21)}}{1+e^{X_{21}}} + \frac{W_{o(1,22)}}{1+e^{X_{22}}} + \frac{W_{o(1,23)}}{1+e^{X_{23}}} + \frac{W_{o(1,24)}}{1+e^{X_{24}}} + \\
 & \frac{W_{o(1,25)}}{1+e^{X_{25}}} + \frac{W_{o(1,26)}}{1+e^{X_{26}}} + \frac{W_{o(1,27)}}{1+e^{X_{27}}} + \frac{W_{o(1,28)}}{1+e^{X_{28}}} + \frac{W_{o(1,29)}}{1+e^{X_{29}}} + \frac{W_{o(1,30)}}{1+e^{X_{30}}} + \frac{W_{o(1,31)}}{1+e^{X_{31}}} + \frac{W_{o(1,32)}}{1+e^{X_{32}}} + \frac{W_{o(1,33)}}{1+e^{X_{33}}} + \frac{W_{o(1,34)}}{1+e^{X_{34}}} + \frac{W_{o(1,35)}}{1+e^{X_{35}}} + \frac{W_{o(1,36)}}{1+e^{X_{36}}} + \\
 & \left. \frac{W_{o(1,37)}}{1+e^{X_{37}}} + \frac{W_{o(1,38)}}{1+e^{X_{38}}} + \frac{W_{o(1,39)}}{1+e^{X_{39}}} + \frac{W_{o(1,40)}}{1+e^{X_{40}}} \right] \dots \\
 & - (W_{o(1,1)} + W_{o(1,2)} + W_{o(1,3)} + W_{o(1,4)} + W_{o(1,5)} + W_{o(1,6)} + W_{o(1,7)} + W_{o(1,8)} + W_{o(1,9)} + W_{o(1,10)} + W_{o(1,11)} + \\
 & W_{o(1,12)} + W_{o(1,13)} + W_{o(1,14)} + W_{o(1,15)} + W_{o(1,16)} + W_{o(1,17)} + W_{o(1,18)} + W_{o(1,19)} + W_{o(1,20)} + W_{o(1,21)} + W_{o(1,22)} + \\
 & W_{o(1,23)} + W_{o(1,24)} + W_{o(1,25)} + W_{o(1,26)} + W_{o(1,27)} + W_{o(1,28)} + W_{o(1,29)} + W_{o(1,30)} + W_{o(1,31)} + W_{o(1,32)} + W_{o(1,33)} + \\
 & W_{o(1,34)} + W_{o(1,35)} + W_{o(1,36)} + W_{o(1,37)} + W_{o(1,38)} + W_{o(1,39)} + W_{o(1,40)}) + b_{2(l)} \quad (20)
 \end{aligned}$$

En la ecuación 19, describe el comportamiento de la vida útil de la turbina de vapor de 110MW, que ha sido inspirado del algoritmo neuronal. Sin embargo, antes de desarrollar la ecuación de la vida útil, tenemos que denotar las siguientes variables para simplificar el cálculo:

- V1 = Amortiguación (N s/m)
- V2 = Frecuencia natural (Hz)
- V3 = Magnitudes de vibración ( $m/s^2$ )
- V4 = Densidad ( $Kg/m^3$ )
- V5 = Presión (MPa)
- V6 = Temperatura ( $^{\circ}C$ )
- V7 = Flujo de masa ( $Kg/s$ )
- V8 = Entalpía (KJ)

Sin embargo, la salida está representada la vida útil, por sus siglas en inglés (useful life) (UL) en minutos.

#### 4. Conclusiones

Las redes neuronales artificiales se han utilizado ampliamente en el modelado y la simulación de turbinas de vapor dentro de la industria de la energía eléctrica que proporcionan energía eléctrica a los clientes. Las redes neuronales también se han aplicado a diversos problemas en el diseño y desarrollo del sistema de energía desde la década de 1990. El análisis de fallas de ingeniería en alabes de turbinas de vapor es, en general, bastante complejo, debido a la complejidad de resolver la ecuación que involucra las variables operaciones como: presión, flujo de masa, amortiguamiento, frecuencia, vibración, entalpía, temperatura y densidad de vapor. Este trabajo de tesis de Ingeniería Mecatrónica confirma que el modelado de RNA es un enfoque efectivo y simple para describir con éxito el comportamiento de estos procesos complejos, en los que las variables operacionales muestran un efecto combinado, dentro del rango experimental de condiciones de operaciones investigadas. Las RNA son capaces de simular las complejas relaciones existentes entre las variables de proceso de entrada y salida en el análisis de fallas de ingeniería en alabes de turbinas de vapor.

Por lo tanto, el parámetro de vida útil ha sido predicho exitosamente, mediante redes neuronales de tres capas, una capa de entrada de 08 variables, capa oculta de 40 neuronas y capa de salida de una sola neurona, y usando diferentes algoritmos de aprendizaje de tipo retropropagación (backpropagation), lo cual el mejor fue el algoritmo de Levenberg Marquardt. Se realizaron simulaciones basadas en el modelo RNA para estimar el comportamiento del sistema en diferentes condiciones. Los resultados obtenidos por el modelo neuronal (RNA) muestran un alto acuerdo con los resultados experimentales: muy buena correlación ( $R^2 = 0.97$ ) y pequeño error ( $\{RMSE = 0.000255\}$ ).

En general, a medida que aumenta el número de componentes con fallas simultáneas, disminuye la confiabilidad de la red para evaluar con precisión la falla. Una forma de mejorar esta fiabilidad sería el aumento de la información de sensores al considerar los datos en diferentes puntos de operación en las turbinas de vapor, también conocido como análisis de múltiples puntos de operación. Sin embargo, la metodología descrita se ha probado con datos no utilizados para la generalización. Además, se confirmó un nivel muy alto de confianza para el modelo neuronal con la prueba estadística de intercepción y pendiente (99%). Según el análisis de sensibilidad, encontramos que la presión del vapor es el parámetro más influyente. Brevemente, en esta tesis de licenciatura del programa educativo de ingeniero mecatrónica, de la facultad de ingeniería de la Universidad Autónoma del Carmen, hemos logrado que las redes neuronales artificiales representen una metodología relativamente nueva para el modelado predictivo en turbinas de vapor de 110MW. El algoritmo de RNA es una forma de inteligencia artificial basada libremente en el cerebro, que ha demostrado la capacidad de aprender relaciones complejas y sutiles entre las variables de la turbina de vapor. A diferencia de las técnicas estadísticas tradicionales, los RNA son capaces de resolver automáticamente estas relaciones sin la necesidad de suposiciones a priori sobre la naturaleza de las interacciones entre las variables.

En este momento, la mayoría de los científicos e ingenieros de todo el mundo estaban dispuestos a aceptar la red neuronal artificial, como una herramienta poderosa de modelar fenómenos complejos y caóticos, porque (RNA) ha sido perfectamente de acuerdo con el experimento.

A pesar del importante papel que desempeñan las redes neuronales. Sin embargo, los RNA tienen debilidades como:

- La estructura de red óptima para un problema dado generalmente no se conoce.
- El criterio para la selección de los mejores algoritmos de entrenamiento para la convergencia rápida de los nuevos patrones dados no es entendido tampoco trivial.
- Las reglas para seleccionar la cantidad y el tipo de datos para la generalización a fin de mejorar la calidad de la red son mínimas.
- La convergencia de los algoritmos de entrenamiento no está garantizada
- Largos tiempos de entrenamiento / adaptación, incluido simulaciones

## Referencias

- [1] Stultz SC, Kitto JB. Steam, its generation and use. 40th ed. Barberton: The Babcock & Wilcox; 1992.
- [2] Mazur Zdzislaw, Garcia-Illescas Rafael, Aguirre-Romano Jorge, Perez-Rodriguez Norberto. Steam turbine blade failure analysis. Eng Fail Anal 2008;15:129–41.
- [3] Dorfman LS, Trubelja M. Torsional monitoring of turbine-generators for incipient failure detection. In: Proceedings of the 6th EPRI steam turbine/generator workshop, St. Luis, Missouri; 1999.
- [4] Rosario DA. L-0 failure investigation at south Texas project. In: Proceedings of the 8th EPRI turbine-generator conference, Nashville, Tennessee; 2003.
- [5] Trumpler WE, Owens HM. Turbine-blade vibration and strength. Trans ASME 1955;77:337–41.
- [6] Garcia JC. Determination of induced forces by steam flow in turbines. Doctoral thesis, CIICAp, Cuernavaca, Morelos; 2008 [in Spanish].
- [7] Vyas NS, Rao JS. Fatigue life estimation procedure for a turbine blade under transient loads. J Eng Gas Turb Power 1994;116:198–206.
- [8] Smith SM. Vibration of turbine blades in packets. In: Proceedings of the 7th international congress for applied mechanics, London; 1948.
- [9] Prohl MA. A method for calculating vibration frequency and stress of banded group of turbine buckets. Trans ASME 1958;80:169–80.
- [10] Rao JS. Turbomachine blade vibration. 1st ed. New Delhi: V.K. Printers; 1991.
- [11] Kubiak J. Diagnostic of fracture of the blade of the steam turbine. Internal report of EM-CIICAp, No 6P/DM CIICAp. University of the State of Morelos; 2002 [in Spanish].
- [12] Logan E, Roy R. Handbook of turbomachinery. 2nd ed. Tempe, Arizona: Taylor & Francis Group; 2003.
- [13] Talib A, Hasan YA, Rahman NNA. Predicting biochemical oxygen demand as indicator of river pollution using artificial neural networks. In: Proceedings of the 18th world IMACS/MODSIM congress, Cairns, Australia; 2010.
- [14] McCulloch WW, Pitts WA. A logical calculus of ideas imminent in nervous activity. Bull Math Biophys 1943;5:115–33.
- [15] Abad A, Hahn M, Es-Said OS. Corrosion of 15-5PH H1025 stainless steel due to environmental conditions. Eng Fail Anal 2010;17:208–12.
- [16] Azevedo CRF, Sinátora A. Erosion-fatigue of steam turbine blades. Eng Fail Anal 2009;16:2293–303.

- [17] Kubiak J, Segura JA, Gonzalez G, García JC, Sierra F, Nebradt JG, et al. Failure analysis of the 350 MW steam turbine blade root. *Eng Fail Anal* 2009;16:1270–81.
- [18] Kubiak J, Urquiza G, Rodriguez JA, González G, Rosales I, Castillo G, et al. Failure analysis of the 150 MW gas turbine blades. *Eng Fail Anal* 2009;16:1794–804.
- [19] Otegui JL, Kunert H, Booman J, Barcia P, Guidi G. Two competing crack growth mechanisms in very high pressures tubes. *Eng Fail Anal* 2011;18:564–71.