

MONITORIZACIÓN DEL ESTADO DE RODAMIENTOS BASADA EN TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Alberto Villalonga Jaén
alberto.villalongaj@alumnos.upm.es

Fernando Castaño Romero
fernando.castano@car.upm-csic.es

Fernando Matia
fernando.matia@upm.es

Gerardo Beruvides López
gerardo.beruvides@hitachi-eu.com

Rodolfo Haber
Rodolfo.haber@car.upm-csic.es

Resumen

En la actualidad, los procesos de fabricación están adoptando nuevas soluciones basadas en la aplicación de técnicas de aprendizaje automático, que permiten llevar a cabo la monitorización de los procesos en tiempo real, la parametrización y la autoconfiguración óptimas de máquinas, robots y procesos industriales, desde equipos individuales hasta entornos de producción global. Este trabajo presenta la aplicación de dos arquitecturas de redes neuronales, una de aprendizaje supervisado, el perceptrón multicapa (MLP), y otra de aprendizaje no supervisado, los mapas auto-organizados (SOM). Para la parametrización de dichas arquitecturas se utilizan algoritmos evolutivos, con vistas a la monitorización del estado de los rodamientos. Finalmente, se evalúa el potencial de ambas técnicas a través de un caso de estudio, en el que se obtienen cifras de mérito o índices de comportamiento muy positivos.

Palabras clave: Monitorización, Aprendizaje automático, Perceptrón Multicapa, Mapas autoorganizados

1 INTRODUCCIÓN

La ejecución o puesta en marcha de un proceso industrial requiere no sólo administrar y planificar los recursos humanos, las materias primas y la maquinaria, sino también disponer en proceso de toda información necesaria para la toma de decisiones [1]. Por eso es importante disponer de una herramienta que permita visualizar, almacenar y manejar la información del proceso industrial en cuestión.

Los sistemas de monitorización son de vital importancia para el funcionamiento eficiente de

cualquier proceso productivo, ya que ofrecen la posibilidad de comprender de una manera detallada la dinámica de los procesos. Esto posibilita que se pueda realizar una toma de decisiones y posteriormente, una posible corrección de los principales parámetros de ajuste como vía para alcanzar un funcionamiento óptimo y más eficiente [2].

En la actualidad, la industria de fabricación está involucrada de forma activa en la implementación de las nuevas transformaciones digitales para lograr un aumento en la productividad en base a las altas demandas del mercado y los niveles de calidad requeridos por los clientes [3, 4]. Los niveles de conectividad y las capacidades de cómputo disponibles en nuestros días en los sistemas digitales han permitido poder embeber o incorporar a los procesos de nuevos métodos basados en inteligencia computacional para el modelado a través de gemelos digitales, el control y la supervisión, de modo que se puedan realizar tareas de diseño y fabricación cada vez más complejas. Estos métodos permiten o habilitan el aprendizaje automático y por tanto contribuyen a mejorar el funcionamiento de los procesos sin intervención humana [5].

En particular, los sistemas de monitorización del estado (*condition monitoring*, en inglés, CM) identifican las fluctuaciones y variaciones significativas de las variables y señales en relación con los datos proporcionados por los sensores para detectar y prevenir un posible fallo o avería. En los diferentes sectores industriales, las aplicaciones de CM más reportadas se centran en: análisis y diagnóstico de vibraciones, análisis de lubricantes, análisis de señales de emisión acústica, termografía infrarroja, pruebas de ultrasonido, análisis del estado del motor y consumo, entre otros [6].

Debido a las pérdidas que se producen en las paradas de las máquinas herramienta por averías, la monitorización de las condiciones o del estado de los rodamientos en máquinas rotatorias ha ganado gran importancia. De este modo, es posible estimar o predecir los posibles fallos de los elementos y con ello, mejorar la planificación de los mantenimientos preventivos, así como disminuir el tiempo de las paradas por roturas [7]. La monitorización del estado del proceso se basa en la disponibilidad de un modelo del proceso físico, el cual es capaz de predecir con suficiente exactitud el estado de los componentes del sistema, para así poder estimar las posibles roturas y paradas no deseadas y mejorar la planificación de los mantenimientos [8].

En el presente trabajo se trata la monitorización de estado de rodamientos a través de la aplicación de técnicas de aprendizaje automático (*machine learning*, ML). El trabajo está organizado de la siguiente manera: en la sección 2 se explican los algoritmos que forman el método, en la sección 3 se presenta la evaluación a través de un caso de estudio y finalmente se plantean algunas conclusiones.

2 RODAMIENTOS. CARACTERÍSTICAS PRINCIPALES.

Los rodamientos son elementos mecánicos y forman parte de máquinas rotatorias, los cuales cumplen la función de reducir la fricción entre el eje y las piezas conectadas a éste. Dada su función son elementos de vital importancia, ya que si se averían producen la parada de las máquinas, y con ello del proceso productivo en el cual están involucradas.

Se componen de 2 partes fundamentales: elementos de rodadura y las jaulas o pistas de rodadura (interna y externa). La mayoría de los fallos que se producen en los rodamientos se pueden clasificar de acuerdo con el elemento dañado como: (1) defecto en la pista exterior de rodadura; (2) defecto en la pista interna de rodadura; (3) defecto del elemento de rodadura. El fallo o la avería se suele modelar como un pequeño agujero creado a partir de un faltante de material en el elemento correspondiente. La frecuencia del fallo tiene una relación directa, con la geometría del rodamiento y la velocidad relativa de cada pista individual [9].

Las frecuencias de resonancia asociadas a cada elemento que componen el rodamiento se pueden calcular teóricamente. Cada elemento de rodamiento tiene una frecuencia característica de rotación. Con la aparición de un defecto o fallo en un elemento de rodamiento particular, se suele producir un aumento de las vibraciones y de la amplitud en esta frecuencia de rotación. Esta frecuencia se puede calcular a partir

de la geometría, de la velocidad de rotación del elemento y del rodamiento.

Las frecuencias asociadas con los rodamientos defectuosos son las siguientes:

$$ORF = \frac{n}{2} \left(\frac{Ns}{60} \right) \left(1 - \frac{d}{D} \cos \alpha \right) \quad (1)$$

$$IRF = \frac{n}{2} \left(\frac{Ns}{60} \right) \left(1 + \frac{d}{D} \cos \alpha \right) \quad (2)$$

$$BF = \frac{D}{d} \left(\frac{Ns}{60} \right) \left(1 + \frac{d^2}{D^2} \cos \alpha \right) \quad (3)$$

Donde Ns son las rpm del eje, d el diámetro medio de los elementos rodantes, D el diámetro de paso del rodamiento, n el número de elementos rodantes, α el ángulo de contacto, siendo ORF la frecuencia asociada a los fallos en la pista externa, IRF asociada a la pista interna y BF al elemento de rodadura [10].

Para la detección de fallos de rodamiento han sido propuestas varias metodologías en la bibliografía. Una de las más utilizadas es el análisis de la señal de vibración de los rodamientos. Para la realización de este análisis se utilizan métodos basados en el análisis de la señal obtenida en tres dominios fundamentales: el tiempo, la frecuencia y el tiempo-frecuencia. Para el análisis en el dominio del tiempo se utilizan estadígrafos temporales como el valor RMS de la señal, el factor de cresta, el pico máximo, la función de densidad de probabilidad, entre otros [11]. En el análisis en el dominio de la frecuencia, la transformada rápida de Fourier FFT es la herramienta más ampliamente utilizada [12]. Por último, en el dominio tiempo-frecuencia la transformada wavelet es la elegida [13].

3 ALGORITMOS.

Entre los modelos más usados en los sistemas de monitorización están las técnicas de aprendizaje automático. Entre estos destacan las redes neuronales artificiales (ANN), las regresiones lineales y no lineales, la máquina de soporte vectorial (SVM) y las técnicas de aprendizaje por refuerzo. En este trabajo se implementan dos modelos de redes neuronales artificiales, a saber: el *perceptrón* multicapa y los mapas autoorganizados, que han sido utilizados en el modelado.

3.1 PERCEPTRÓN MULTICAPA

El *perceptrón multicapa* (MLP) pertenece a la categoría de redes neuronales artificiales anticipativas, que utiliza generalmente tres o más capas de neuronas con una función de activación no lineal. Las redes MLP están diseñadas para uso

general, modelos flexibles no lineales que consisten en varias unidades organizadas en múltiples capas. El resultado de la red MLP se puede cambiar variando el número de capas y el número de unidades en cada capa. Se ha demostrado que los MLP pueden estimar aproximadamente cualquier función con la precisión deseada [14, 15]. Cada neurona en una capa está conectada con un cierto peso w_{ij} a cada nodo en la siguiente capa [16]. Matemáticamente se puede escribir como:

$$y = f\left(\sum_{i=1}^n \omega_i x_i + \theta\right) \quad (4)$$

donde, f es una función de paso discontinuo unipolar (0 o 1) o bipolar (valores -1 o 1); n es el número de entradas en una neurona; x las señales de entrada; ω los pesos sinápticos; θ el valor umbral, también llamado sesgo, e y el valor de salida de la neurona.

Además, los modelos de redes neuronales artificiales necesitan ser entrenados y para ello, un método comúnmente utilizado es la propagación hacia atrás de errores o retropropagación. Este método suele utilizarse combinado con un método de optimización como es el gradiente descendiente. Este proceso calcula la pendiente de una función de pérdida con referencia a todos los pesos en el modelo de red. Posteriormente, ésta pendiente calculada es suministrada al método de optimización que a su vez lo usa para equilibrar los pesos, en un intento de minimizar la función de error.

La retropropagación requiere que exista un conjunto de entrada / salida para el aprendizaje supervisado, es decir, para el cálculo del gradiente o la Hessiana. Además, incorpora una función de activación utilizada por las neuronas artificiales que debe ser diferenciable. El aprendizaje se produce actualizando los pesos de conexión, de cada neurona, después de procesar cada información, según el error resultante de la comparación con el valor esperado. Este es un paradigma de aprendizaje supervisado, y se lleva a cabo a través de una generalización del algoritmo de mínimos cuadrados en el perceptrón lineal [17]. Los diferentes pasos se muestran a continuación:

En primer lugar, se representa el error en el nodo de salida j en el n -ésimo conjunto de datos por:

$$e_j(n) = d_j(n) - y_j(n) \quad (5)$$

Seguidamente, se actualizan los pesos de las neuronas en función de las correcciones que reducen el valor del error, dada por:

$$\varepsilon(n) = \frac{1}{2} \sum e_j^2(n) \quad (6)$$

utilizando el gradiente descendiente, el cambio en cada paso se expresa:

$$\Delta \omega_{ji}(n) = -\eta \frac{d\varepsilon(n)}{dv_j(n)} y_i(n) \quad (7)$$

donde y_i es la salida de la neurona anterior y η es la tasa de aprendizaje, que se selecciona cuidadosamente para asegurar que los pesos se encuentran con una respuesta lo suficientemente rápida, sin producir oscilaciones. En aplicaciones de programación, este marco generalmente varía de 0.2 a 0.8.

El cálculo de la derivada depende del campo local inducido, que a su vez varía. Es fácil probar que para un nodo de salida, esta derivada se puede simplificar a:

$$-\frac{d\varepsilon(n)}{dv_j(n)} = e_j(n) \vartheta'(v_j(n)) \quad (8)$$

donde ϑ' es la derivada de la función de activación. El análisis es más difícil para el cambio en pesos a un nodo oculto, pero se puede demostrar que la derivada es:

$$-\frac{d\varepsilon(n)}{dv_j(n)} = \vartheta'(v_j(n)) \sum_k -\frac{d\varepsilon(n)}{dv_j(n)} \omega_{kj}(n) \quad (9)$$

dependiendo de los cambios en los pesos de los k -ésimos nodos, que representan la capa de salida.

Entonces, para cambiar los pesos de la capa oculta, primero se deben cambiar los pesos de la capa de salida de acuerdo con la derivada de la función de activación. Este proceso se aproxima a la retropropagación de la función de activación. En el MLP se utilizan funciones de activación no lineales lo cual permite simular la frecuencia de activación o disparo de las neuronas en el cerebro humano.

3.2 MAPAS AUTORGANIZADOS

Los mapas autoorganizados son un tipo de red neuronal artificial que pertenecen a la categoría de los métodos de aprendizaje no supervisados, es decir, no hay salidas “objetivo” explícitas asociadas con cada entrada y el objetivo es construir representaciones de la entrada que puedan usarse para la toma de decisiones. Está conformado por unidades llamados nodos o neuronas. Asociado con cada neurona hay un vector de pesos, de la misma dimensión de los vectores de entrada, y una posición en el mapa. En el SOM, un patrón se proyecta desde un espacio de entrada a una posición en el mapa donde la información se codifica como la ubicación

de la neurona activada. La conservación topológica del SOM lo hace especialmente útil en la clasificación de datos, que incluye variaciones suaves en los mismos [18]. Un SOM funciona como una matriz de neuronas dispuestas en una cuadrícula, y asigna un espacio de entrada de alta dimensión a un espacio de salida de baja dimensión. El mapeo del SOM se realiza mediante vectores de características asociados con cada unidad. Durante la fase de aprendizaje, los puntos operativos (Ops) se inicializan aleatoriamente [19]. La selección de la unidad de salida ganadora se basa en determinar la neurona que posea los pesos con la mayor similitud al vector de entrada x . La unidad ganadora satisface la siguiente ecuación:

$$|x - \omega_c| = \min |x - \omega_i| \quad (10)$$

Definida la neurona ganadora, se procede a determinar la vecindad del ganador, utilizando una función de vecindad alrededor de una unidad ganadora. Por ejemplo, la función Gaussiana se puede usar como la función de vecindario de la siguiente manera:

$$\Omega_c(i) = \exp\left(-\frac{|p_i - p_c|}{2\sigma^2}\right) \quad (11)$$

donde p_i y p_c son las posiciones de las unidades de salida i y c respectivamente, y σ es un coeficiente que representa el alcance del vecindario. Después de la definición de la función de vecindad, el vector de peso ω_c de la neurona seleccionada y los vectores ω_i de sus vecinos se actualizan de acuerdo con la siguiente fórmula:

$$\Delta\omega_i = \alpha(n) \cdot \Omega_c(i) \cdot (x - \omega_i) \quad (12)$$

donde $\alpha(n)$ se define como el índice de aprendizaje el cual decrece exponencialmente con el aumento de las muestras de entrenamiento. Este se calcula según la expresión:

$$\alpha(n) = \alpha_0 \exp\left(-\frac{n}{\lambda}\right) \quad (13)$$

siendo λ una constante.

El proceso de entrenamiento es iterativo, los pesos de las neuronas que se encuentran en la vecindad de la ganadora en cada iteración son actualizados modificándose así a los patrones utilizados para el entrenamiento.

3.3 OPTIMIZACIÓN

Las técnicas de modeladas utilizadas en la biblioteca digital (*digital toolset*) desarrollada contienen un

conjunto de parámetros a ajustar por cada uno de los métodos predefinidos para los que no existe una regla clara de ajuste o de selección, lo que dificulta una parametrización óptima que permita mejorar la precisión del modelo que se desea obtener. Una vía para solucionar este problema, es la hibridación entre los métodos de modelado y técnicas de optimización [20]. De este modo las estrategias de optimización se convierten en una poderosa herramienta en la obtención de los parámetros óptimos o cuasi-óptimos de los modelos [21].

3.3.1 Algoritmos genéticos

Los algoritmos genéticos (GA) son técnicas de búsqueda generales y una de sus aplicaciones es la optimización, basada en las operaciones de selección natural y genética. Operan sobre una población compuesta por individuos. Los individuos se codifican como cadenas (cromosomas) construidas sobre un código particular, por ejemplo, el código binario (0.1), de modo que los valores de los cromosomas se mapean de forma única en el dominio de la variable de decisión. Una vez que se calcula la representación del dominio de la variable de decisión de la población actual el rendimiento se asume de acuerdo con la función objetivo que caracteriza el problema a resolver. También es posible usar los parámetros variables directamente para representar los cromosomas en la solución [22].

En la etapa de reproducción, un valor de aptitud se deriva de la medida de rendimiento individual en bruto dado por la función objetivo, y se usa para sesgar el proceso de selección. Los individuos altamente capacitados tendrán mayores oportunidades de transmitir material genéticamente importante a generaciones sucesivas.

Los individuos seleccionados se modifican mediante la aplicación de operadores genéticos. Para obtener la próxima generación. Los operadores genéticos manipulan los caracteres (genes) que constituyen los cromosomas directamente, siguiendo la suposición de que ciertos genes codifican, en promedio, para individuos más idóneos que otros genes [23]. Los operadores genéticos se pueden dividir en tres categorías principales. Reproducción, cruzamiento y mutación.

Los algoritmos genéticos tienen más probabilidades de converger a cuasi-óptimos globales que las técnicas de optimización convencionales ya que realizan búsquedas desde una población de puntos y se basan en reglas de transición probabilísticas. Las técnicas de optimización convencionales se basan normalmente en métodos deterministas que, por definición, tienen mayor probabilidad de encontrar óptimos locales. Los algoritmos genéticos también pueden tolerar discontinuidades y evaluaciones de funciones ruidosas.

4 ANÁLISIS DE RESULTADOS

La evaluación de los dos métodos considerados para la monitorización se realizó mediante el análisis de la señal de vibraciones de rodamientos en máquinas rotacionales. Las señales de vibración utilizadas fueron proporcionadas por el Centro de Inteligencia Sistemas de mantenimiento (IMS) de la Universidad de Cincinnati [24, 25]. El conjunto de datos experimentales se generó a partir de ensayos de fallos en condiciones de carga constante en una plataforma de prueba diseñada como se muestra en la Figura 1. La velocidad de rotación se mantuvo constante a 2000 rpm. Se agregó una carga radial de 2722 kg al eje y se consideró el sistema con un comportamiento de mecanismo de resorte. La frecuencia de muestreo fue de 20 kHz y el tamaño de la señal fue de 20.480 muestras. Se realizó el análisis de las señales del rodamiento tres, el cual después de 30 días de trabajo, produjo un fallo en la pista interna.

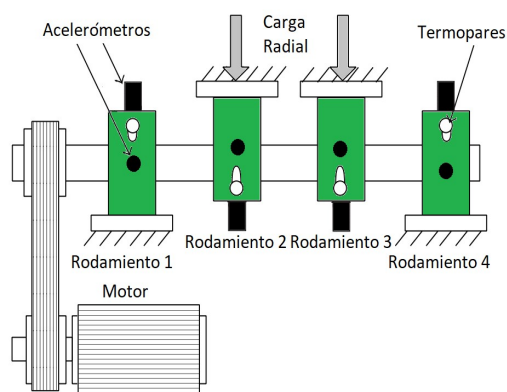


Figura 1: Diagrama del sistema físico real.

Para obtener un mejor ajuste en los modelos, a la señal de vibraciones se le realizó extracción de rasgos característicos, dividida en dos etapas. En la primera etapa se aplicó la transformada de *wavelet* a la señal de entrada. En la etapa posterior a los coeficientes obtenidos de la transformada de *wavelet* se le aplicaron los estadígrafos: Kourtosis, valor cuadrático medio (RMS), desviación estándar (STD) y pico máximo. Estas fueron las cuatro variables consideradas como entradas en ambos modelos.

Para la selección de los conjuntos de entrenamiento y validación se utilizó el método *hold-out*, tomándose para el entrenamiento el 67% de las muestras y el 33% para la validación. Como criterio de detección del entrenamiento en el SOM, se utilizó el alcanzar el total de muestras de entrenamiento, mientras que en el MLP se utilizaron como parámetros de finalización el error de ajuste y el tamaño del entrenamiento. Una vez procesados los datos se realizó la parametrización de los modelos (ver tablas 2 y 3) por medio de la interfaz de configuración del sistema y se procedió al entrenamiento de ambos. En la

parametrización de los modelos, en el MLP se tuvieron que definir: el número de entradas, número de capas ocultas, neuronas por cada capa oculta, cantidad de neuronas en la capa de salida. Como salidas del modelo se prefijaron los valores 0 (indicando buen estado del rodamiento) y 1 (indicando fallo en el rodamiento).

Tabla 1: Configuración SOM

Filas	12
Columnas	2
Pesos	4
Muestras de entrenamiento	1000
Distancia	2

Tabla 2: Configuración MLP

Capas ocultas	1
Neuronas en la capa oculta	15
Neuronas en la capa de salida	1
Tamaño del entrenamiento	5000
Tasa de aprendizaje	0.01
Valor mínimo normalización	-1
Valor máximo normalización	1
Valor mínimo pesos	-0.1
Valor máximo pesos	0.1

En la tabla 3 se muestran los índices de rendimiento: verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN), falsos negativos (FN), tasa de acierto (hR), especificidad (Sp) y sensibilidad (Sn). Si bien ambos presentan buenos comportamientos con tasas de acierto por encima del 80%, la optimización de los parámetros de estos modelos podría mejorar estos índices.

Con el objetivo de minimizar el error de ajuste de los modelos se optimizaron los parámetros de ambas redes neuronales mediante el uso de algoritmos genéticos. En el caso del MLP se utilizaron como variables de decisión el tamaño del entrenamiento y el número de neuronas en la capa oculta, mientras en SOM se utilizaron el número de muestras del entrenamiento y el coeficiente de vecindad. Se utilizó un algoritmo genético de un objetivo con selección por torneo con los siguientes parámetros: tamaño de la población 100, número de generaciones 200 y probabilidad de mutación de 0.00001. El resto de parámetros fueron los que normalmente se utilizan en estos métodos. En las figuras 2 y 3 se muestran el comportamiento de los modelos obtenidos. Se observa que en ambos casos se produce una mejora cercana al 10% tras aplicar la optimización de los parámetros de ambos métodos. Esto demuestra que los algoritmos genéticos son muy buenos buscadores de óptimos globales debido a su mecanismo de búsqueda basado en poblaciones, frente a las técnicas clásicas que fácilmente quedan atrapadas en óptimos locales.

Tabla 3: Índices de rendimiento de los modelos.

	Total	TP	FP	TN	FN	hR	Sp	Sn
SOM	700	168	58	443	31	0.875	0.899	0.798
MLP	700	183	71	430	16	0.881	0.877	0.896

Tabla 4: Análisis comparativo de los dos métodos considerados.

	Total	TP	FP	TN	FN	hR	Sp	Sn
SOM	700	199	16	485	0	0.977	0.968	1.0
MLP	700	199	11	490	0	0.984	0.978	1.0

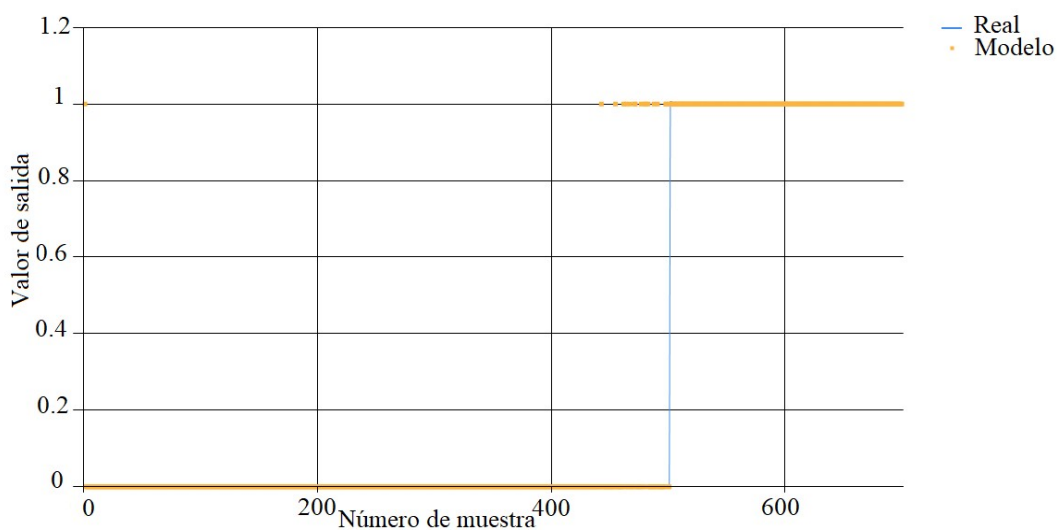


Figura 2: Comportamiento del SOM frente al conjunto de validación tras la optimización

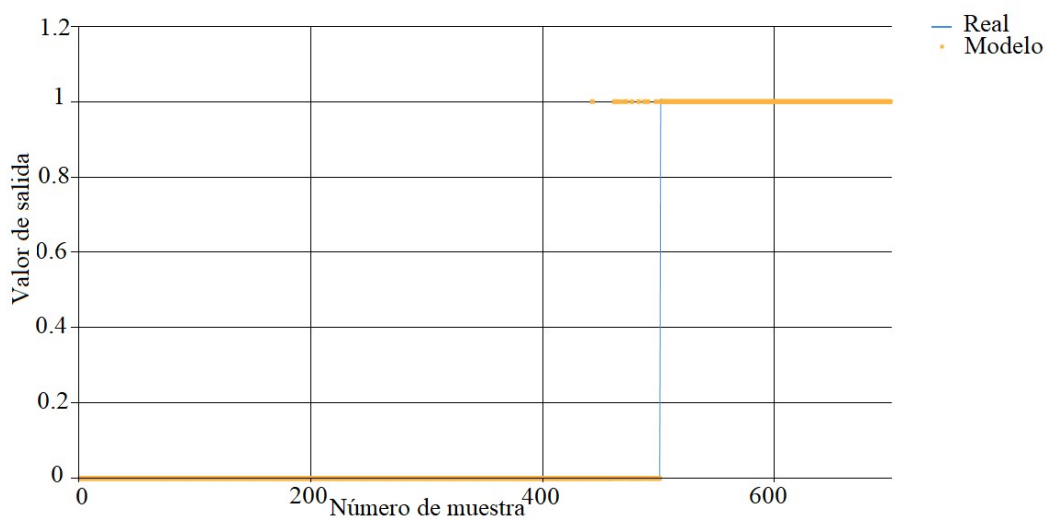


Figura 3: Comportamiento del MLP frente al conjunto de validación tras la optimización

5 Conclusiones

En el presente trabajo se ha presentado el diseño e implementación de dos arquitecturas de redes neuronales para la monitorización del estado de los rodamientos en máquinas de producción industrial con el objetivo de disminuir el tiempo de parada en la industria. Estas son el perceptrón multicapa (MLP) y el mapa autoorganizado (SOM).

Además, se ha propuesto la aplicación de algoritmos evolutivos para la optimización de los parámetros de entrenamiento. Los resultados obtenidos en el proceso de ajuste de los parámetros de las redes, demuestran unos buenos resultados en los experimentos realizados, con tasas de acierto superiores al 95%.

Debido a la alta potencialidad de estos métodos, se han sentado las bases de cara a comparar sus posibilidades con redes neuronales entrenadas con algoritmos más actuales, como es el aprendizaje profundo.

Agradecimientos

Este trabajo ha sido realizado en el marco del Proyecto ECSEL IA 826417 “Power2Power: The next-generation silicon-based power solutions in mobility, industry and grid for sustainable decarbonisation in the next decade”. Además, también ha sido cofinanciado por el proyecto DPI2017-86915-C3-1-R “COGDRIVE: Cognitive inspired navigation for autonomous driving” financiado por el Ministerio de Economía, Industria y Competitividad.

English summary

CONDITION MONITORING OF BEARINGS BASED ON MACHINE LEARNING METHODS.

Abstract

Nowadays, the manufacturing processes are adopting solutions based on the application of machine learning techniques, which allow to carry out the monitoring of the processes in real time, the parameterization and the optimal auto-configuration of machine tools, robots and industrial processes from individual equipment to global production environments. The present work presents the application of a multilayer perceptron (MLP), and self-organized maps (SOM). Evolutionary algorithms are used to parametrize said architectures, with a view to monitoring the bearing conditions. Finally, the performance of both techniques is assessed on the basis of a case study for condition monitoring in

bearings, obtaining high performance indices in both cases.

Keywords: Condition monitoring, Machine learning, Multilayer Perceptron, Self-organized maps.

Referencias

- [1] I. la Fé-Perdomo, G. Beruvides, R. Quiza, R. Haber, and M. Rivas, "Automatic Selection of Optimal Parameters Based on Simple Soft-Computing Methods: A Case Study of Micromilling Processes," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 800-811, 2019.
- [2] G. Beruvides, R. Quiza, R. del Toro, and R. E. Haber, "Sensing systems and signal analysis to monitor tool wear in microdrilling operations on a sintered tungsten-copper composite material," *Sensors and Actuators A: Physical*, vol. 199, pp. 165-175, 2013.
- [3] A. Villalonga, G. Beruvides, F. Castaño, and R. Haber, "Industrial Cyber-Physical System for Condition-based Monitoring in Manufacturing Processes," in *ICPS 2018 - 1st. International Conference on Industrial Cyber Physical Systems*, 2018, pp. -.
- [4] G. Adamson, L. Wang, and P. Moore, "Feature-based control and information framework for adaptive and distributed manufacturing in cyber physical systems," *Journal of Manufacturing Systems*, vol. 43, pp. 305-315, 2017/04/01/ 2017.
- [5] A. Villalonga, G. Beruvides, F. Castaño, and R. Haber, "Condition-Based Monitoring Architecture for CNC Machine Tools Based on Global Knowledge," in *INCOM 2018 - 16th IFAC Symposium on Information Control Problems in Manufacturing*, 2018, pp. -.
- [6] D. Mourtzis, E. Vlachou, N. Milas, and N. Xanthopoulos, "A Cloud-based Approach for Maintenance of Machine Tools and Equipment Based on Shop-floor Monitoring," *Procedia CIRP*, vol. 41, pp. 655-660, 2016/01/01/ 2016.
- [7] Y. Pan, R. Hong, J. Chen, Z. Qin, and Y. Feng, "Incipient fault detection of wind turbine large-size slewing bearing based on circular domain," *Measurement*, vol. 137, pp. 130-142, 2019/04/01/ 2019.
- [8] S. Dong, S. Yin, B. Tang, L. Chen, and T. Luo, "Bearing Degradation Process Prediction Based on the Support Vector Machine and Markov Model," *Shock and Vibration*, vol. 2014, p. 15, 2014, Art. no. 717465.

- [9] M. Cerrada *et al.*, "A review on data-driven fault severity assessment in rolling bearings," *Mechanical Systems and Signal Processing*, vol. 99, pp. 169-196, 2018/01/15/ 2018.
- [10] E. Palomino-Marín, *Elementos de medición y análisis de vibraciones en máquinas rotatorias*. Cuba, 2012.
- [11] S. S. Stalin, "Fault Diagnosis and Automatic Classification of Roller Bearings Using Time-Domain Features and Artificial Neural Network," *International Journal of Science and Research (IJSR)*, vol. 3, no. 11, 2014.
- [12] J. P. Noël and G. Kerschen, "Nonlinear system identification in structural dynamics: 10 more years of progress," *Mechanical Systems and Signal Processing*, vol. 83, pp. 2-35, 2017/01/15/ 2017.
- [13] J. Chebil, M. Hrairi, and N. Abushikhah, "Signal Analysis of Vibration Measurements for Condition Monitoring of Bearings," *Australian Journal of Basic and Applied Sciences*, vol. 5, p. 70, 2011.
- [14] R. E. Haber, R. Haber, A. Alique, S. Ros, and J. R. Alique, "Dynamic model of the machining process on the basis of neural networks: from simulation to real time application," in *International Conference on Computational Science*, 2002, pp. 574-583: Springer, Berlin, Heidelberg.
- [15] R. E. Haber, R. Haber, C. Peres, A. Alique, and S. Ros, "Model of the milling process on the basis of cutting force: A neural network-based approach," *Advances in Neural Networks and Applications*, pp. 378-383, 2001.
- [16] Y.-C. Hu, "Pattern classification by multi-layer perceptron using fuzzy integral-based activation function," *Applied Soft Computing*, vol. 10, no. 3, pp. 813-819, 2010/06/01/ 2010.
- [17] P. K. Sathy, L. Panda, and S. K. Behera, "Ann based image restoration in approach of multilayer perceptron," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, 2016, vol. 2, pp. 1-4.
- [18] U. Fidan, N. Ozkan, and I. Calikusu, "Clustering and classification of dermatologic data with Self Organization Map (SOM) method," in *2016 Medical Technologies National Congress (TIPTEKNO)*, 2016, pp. 1-4.
- [19] E. M. Voumvoulakis and N. D. Hatziaegyriou, "Decision Trees-Aided Self-Organized Maps for Corrective Dynamic Security," *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 622-630, 2008.
- [20] A. Gajate, R. E. Haber, J. R. Alique, and P. I. Vega, "Transductive-weighted neuro-fuzzy inference system for tool wear prediction in a turning process," in *International Conference on Hybrid Artificial Intelligence Systems*, 2009, pp. 113-120: Springer, Berlin, Heidelberg.
- [21] G. Beruvides, R. Quiza, and R. E. Haber, "Multi-objective optimization based on an improved cross-entropy method. A case study of a micro-scale manufacturing process," *Information Sciences*, vol. 334, pp. 161-173, 2016.
- [22] Y. L. Abdel-Magid and M. M. Dawoud, "Genetic algorithms applications in load frequency control," in *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, 1995, pp. 207-213.
- [23] F. Millo, P. Arya, and F. Mallamo, "Optimization of automotive diesel engine calibration using genetic algorithm techniques," *Energy*, vol. 158, pp. 807-819, 2018/09/01/ 2018.
- [24] W. Mao, W. Feng, and X. Liang, "A novel deep output kernel learning method for bearing fault structural diagnosis," *Mechanical Systems and Signal Processing*, Article vol. 117, pp. 293-318, 2019.
- [25] H. Qiu, J. Lee, J. Lin, and G. Yu, "Robust performance degradation assessment methods for enhanced rolling element bearing prognostics," *Advanced Engineering Informatics*, Article vol. 17, no. 3-4, pp. 127-140, 2003.



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).