



Universidad de Matanzas
Facultad de Ciencias Técnicas



ARQUITECTURA PARA MONITOREO DE ESTADOS BASADA EN COMPUTACIÓN EN LA NUBE PARA SISTEMAS Y PROCESOS MECÁNICOS

Tesis Presentada como Requisito Parcial
para la Obtención del Título de
Máster en Ingeniería Asistida por Computadora

Autor: Ing. Yarens Joaquín Cruz Hernández

Tutores: Dr.C. Gerardo Beruvides López
M.Sc. Alberto Villalonga Jaén

Matanzas, 2019

DECLARACIÓN DE AUTORIDAD Y NOTA LEGAL

Yo, Yarens Joaquín Cruz Hernández, declaro que soy el único autor de la siguiente tesis, titulada *Arquitectura para Monitoreo de Estados basada en Computación en la Nube para Sistemas y Procesos Mecánicos* y, en virtud de tal, cedo el derecho de copia de la misma a la Universidad de Matanzas, bajo la licencia *Creative Commons* de tipo *Reconocimiento No Comercial Sin Obra Derivada*, con lo cual se permite su copia y distribución por cualquier medio siempre que mantenga el reconocimiento de sus autores, no haga uso comercial de la obra y no realice ninguna modificación de ella.

Matanzas, ____ de _____ de 2019.

Yarens Joaquín Cruz Hernández

RESUMEN

En la industria moderna se están incorporando nuevos paradigmas que permiten desarrollar e implementar sistemas más confiables y eficientes. Algunos de los sistemas que más se han beneficiado con estos paradigmas son los de monitoreo y control; sin embargo, no todas las organizaciones cuentan con el capital necesario para adoptar estos cambios debido al alto precio en el mercado de las herramientas requeridas. En este trabajo se propone la arquitectura de un sistema de monitoreo para la detección de fallas en procesos de manufactura, específicamente para Controles Numéricos Computarizados. La arquitectura propuesta se basa en el Monitoreo de Estados mediante el análisis de la señal de vibraciones en el dominio del tiempo y utiliza técnicas de Inteligencia Artificial; además de permitir el almacenamiento de datos en la nube, obteniéndose de esta manera la arquitectura de un sistema de monitoreo acorde a la Industria 4.0.

Palabras claves: Monitoreo de Estados; Detección de Fallas; Inteligencia Artificial

ABSTRACT

New paradigms that allow the development and implementation of more reliable and efficient systems are being incorporated in modern industry. Some of the systems that have benefitted the most of these paradigms are monitoring and control systems; however, not all the enterprises have enough capital to adopt these changes due to the high prices of the necessary tools in the market. The architecture of a monitoring system for fault detection in manufacturing processes, specifically for Computer Numerical Controls, is presented in this work. The proposed architecture is based in Condition Monitoring by means of the analysis of vibration signals in time domain. The system also uses Artificial Intelligence techniques and allows data storage on the cloud, obtaining as a result the architecture of a monitoring system according to Industry 4.0.

Keywords: Condition Monitoring; Fault Detection; Artificial Intelligence

TABLA DE CONTENIDO

Introducción	1
Capítulo 1. Estado del Arte.....	6
1.1 Computación en la nube.....	6
1.2 Sistemas Ciberfísicos Industriales	8
1.3 Sistemas de monitoreo basado en estados	9
1.3.1 Análisis de vibraciones	10
1.3.2 Análisis de lubricantes	16
1.3.3 Análisis de emisiones acústicas, termografía infrarroja y pruebas de ultrasonido.....	17
1.4 Conclusiones parciales del capítulo	19
Capítulo 2. Propuesta de la Arquitectura	20
2.1 Arquitectura general del sistema.....	20
2.2 Almacén de datos	22
2.3 Procesamiento de la señal	24
2.4 Librería de modelos	26
2.4.1 Sistema de Inferencia Difuso Takagi-Sugeno-Kang.....	26
2.4.2 Redes Neuronales Artificiales Perceptrón Multicapa	28
2.5 Entrenamiento	29
2.5.1 Entrenamiento para sistemas difusos Takagi-Sugeno-Kang.....	30
2.5.2 Entrenamiento para Redes Neuronales Artificiales Perceptrón Multicapa..	34
2.6 Optimización.....	37
2.7 Modelo activo	41

2.8 Visualización.....	41
2.9 Conclusiones parciales del capítulo	42
Capítulo 3. Análisis de los Resultados.....	44
3.1 Descripción de la aplicación	44
3.2 Validación del software	48
3.3 Conclusiones parciales del capítulo	55
Conclusiones	56
Recomendaciones	57
Referencias Bibliográficas	58
Anexos	63

INTRODUCCIÓN

Desde el origen del ser humano, este ha tenido la necesidad de transformar su entorno para poder aprovechar de una manera más eficiente y eficaz los recursos a su alcance. Sin embargo, no es hasta finales del siglo XVIII que se produce un conjunto de transformaciones socioeconómicas, tecnológicas y culturales, las cuales condujeron a un desarrollo sin precedente de las técnicas productivas. A esta etapa de cambios radicales se le llamó “Primera Revolución Industrial” y conllevó a que la economía, que antes estaba basada en el trabajo manual, fuera reemplazada por otra, liderada por la industria y la manufactura, donde la máquina de vapor fue el eje central.

En la primera década del siglo XX, los avances en el campo de la electricidad convergen con la utilización del motor de combustión interna, propulsado por combustibles fósiles, lo cual propició otra gran transformación en las técnicas de producción. En este período, llamado “Segunda Revolución Industrial”, se comienzan a electrificar las fábricas; de esta manera, se crean las líneas de producción, con lo que surge la división de tareas y la producción en cadena; y donde el hito en esos años lo marcó la producción de automóviles, especialmente el Ford T.

El final del siglo XX trajo aparejado una nueva transformación basada en el uso de la electrónica y la informática en la industria. La “Tercera Revolución Industrial” hizo posible la automatización de los procesos industriales y las líneas de producción. Como resultado, se sustituyó la presencia del hombre en tareas repetitivas o entornos peligrosos al utilizar máquinas automatizadas.

En los últimos años se han presentado una nueva serie de cambios que han conllevado a varios autores a afirmar que la humanidad se encuentra inmersa en la “Cuarta Revolución Industrial”. En este contexto, a la industria moderna se le ha comenzado a llamar “Industria 4.0” o “Industria Conectada”. Los conceptos de Sistema Ciberfísico, Inteligencia Artificial, Productos Inteligentes, Internet de las cosas, Internet de los servicios, Hiperconectividad, Big Data y Computación en la Nube son cada vez más conocidos y se van incorporando paulatinamente a la industria.

La aplicación de estos últimos avances ha producido notables beneficios en el ámbito industrial, entre los que se pueden destacar el desarrollo tecnológico, la flexibilidad en la producción, la personalización de los productos, la optimización de la toma de decisiones y el aumento de la productividad y la eficiencia en la utilización de los recursos. A su vez, los cambios en la industria han abierto paso a nuevas oportunidades de negocio, lo que ha propiciado la dinamización de la economía.

Sin embargo, todos estos cambios radicales se están dando en el seno de las grandes corporaciones, que son las que cuentan con los fondos suficientes para invertir en investigación y desarrollo. Como consecuencia, el pequeño y mediano sector empresarial de los países desarrollados y el sector industrial de los países subdesarrollados, aunque ya han comenzado a interesarse en la introducción de estos avances, se están quedando rezagados. Constituye entonces una necesidad, para estos sectores, el desarrollo de sistemas que les permitan adoptar estos nuevos paradigmas y den paso al salto cualitativo indispensable para estar acorde a la industria moderna.

En el caso de Cuba, debido a la situación económica, se está optando por la sustitución de importaciones, lo que conlleva a un aumento considerable en el número de procesos de manufactura. Como consecuencia se ha producido un alto nivel de explotación de las maquinarias, provocando malfuncionamientos y accidentes en algunos casos, debido a la falta de sistemas de monitoreo y detección de fallas. Esta situación condujo a plantearse la siguiente hipótesis: ¿Se pudiera diseñar la arquitectura de un sistema basado en el paradigma de la Industria 4.0 que permita el monitoreo del estado de las maquinarias en los sistemas y procesos de manufactura para la detección de fallas?

Tomando como punto de partida esta hipótesis se formulan los objetivos de este trabajo:

Objetivo General.

Diseñar la arquitectura de un sistema de monitoreo de maquinarias en procesos y sistemas mecánicos, que utilice algoritmos de Inteligencia Artificial e incorpore el paradigma de Computación en la nube.

Objetivos Específicos.

- Realizar una revisión bibliográfica sobre algunos paradigmas de la Industria 4.0 y los sistemas de monitoreo del equipamiento de manufactura.
- Realizar el análisis y diseño de la propuesta de arquitectura para el sistema de monitoreo.
- Programar las funcionalidades del sistema.
- Validar el software desarrollado.

Objeto de estudio.

- Sistemas de cómputo inteligentes.
- Programación orientada a objetos.
- Sistemas de monitoreo.

Aportes prácticos esperados.

Se desarrollarán modelos basados en Inteligencia Artificial para un sistema de monitoreo de estados para equipos de manufactura utilizando la librería Qt.

Alcance de la investigación.

Disponer de la arquitectura de un sistema de monitoreo de estados para procesos de manufactura.

Materiales y métodos propuestos.

Computadora personal e IDE Qt Creator. Métodos de programación orientada a objetos utilizando ANSI C++ con librerías de Qt4 y almacenamiento en la nube.

Breve descripción de la estructura del trabajo.

El trabajo consta de 3 capítulos a través de los cuales se expone el proceso de análisis y diseño de la arquitectura de un sistema de monitoreo de estados para equipos de manufactura; además se realiza la validación de las técnicas de Inteligencia Artificial propuestas para el sistema de monitoreo. En el Capítulo 1 se hace un análisis bibliográfico de varios paradigmas de la Industria 4.0 y de las técnicas de monitoreo de equipos de

manufactura. En el Capítulo 2 se aborda el proceso de análisis y diseño de la arquitectura del sistema de monitoreo. Finalmente, en el Capítulo 3 se muestra la validación de las técnicas de monitoreo a incluir en el software.

CAPÍTULO 1. ESTADO DEL ARTE

En este capítulo se realiza una revisión bibliográfica con el objetivo de analizar cómo la Computación en la nube comienza a tener impacto en las industria moderna y cuáles son las principales tendencias en el monitoreo de estados. Estas son dos ramas que convergen hacia una integración ya que en los últimos años el nivel de informatización de las máquinas y procesos en la industria ha hecho posible que la mayoría de los datos que se necesitan para analizar su funcionamiento se puedan enviar, almacenar y procesar utilizando Internet. Además, se explican brevemente algunos de los métodos que se utilizan para la recolección de los datos que permiten realizar el monitoreo.

1.1 Computación en la nube

El término Computación en la nube puede ser definido como: Un paradigma de computación distribuida a gran escala que es impulsado por sus beneficios económicos, en el cual un conjunto de servicios dinámicamente escalables de almacenamiento, plataformas y poder computacional centralmente administrado se ofrece a petición de clientes externos en Internet (Hill, 1990).

Se pueden diferenciar tres tipos de nubes computacionales en dependencia de los servicios que ofrecen a sus usuarios. Las nubes que ofrecen el software como un servicio (Software as a Service, SaaS), las que ofrecen las plataformas como servicio (Platform as a Service, PaaS) y las que ofrecen la infraestructura como un servicio (Infrastructure as a Service, IaaS).

Las nubes SaaS son particularmente útiles para los usuarios finales, ya que muchas de las aplicaciones que estos utilizan se diseñaron y se encuentran instaladas en Internet. De esta forma el usuario no tiene necesidad de ocupar espacio innecesariamente en sus dispositivos y además todo el procesamiento computacional se realiza en los servidores, de manera que se puede contar con muchos más recursos para esta tarea.

En el caso de las nubes PaaS, sus principales usuarios son los desarrolladores de software, ya que los servicios que se ofrecen son plataformas de desarrollo de software e interfaces de programación de aplicaciones (Ahson y Ilyas, 2011). Este tipo de nube permite construir, probar, desplegar y corregir los errores de las aplicaciones con gran facilidad.

El tercer tipo de nubes, las IaaS, ofrecen como servicio recursos de infraestructura tales como almacenamiento, recursos computacionales y de comunicación de una manera flexible. Los recursos computacionales se pueden ofrecer como un servicio gracias a las máquinas virtuales (virtual machines, VM), las cuales permiten personalizar el hardware de acuerdo a las necesidades. El almacenamiento (Data-Storage as a Service, DaaS) es otro de los servicios que se pueden utilizar; permitiendo a los usuarios guardar sus datos en centros o almacenes de datos remotos y luego acceder a ellos desde cualquier lugar en cualquier momento mediante Internet. También las comunicaciones se ofrecen como servicio (Communication as a Service, CaaS) de manera que estas se pueden configurar y programar de acuerdo a horarios, logrando que no se sobrecarguen las líneas y aumentando la seguridad de la red.

Estas ventajas que ofrece la Computación en la nube han sido el principal incentivo para que se haya comenzado a aplicar en las fábricas (Qi, *et al.*, 2018). En la industria moderna,

las demandas de reducción de los costos, el aumento de la confiabilidad, las prestaciones y la seguridad humana se están incrementando cada vez más. Por esta razón, en los procesos de fabricación, se están adoptando soluciones con sistemas que permiten llevar a cabo el monitoreo de los procesos en tiempo real, la parametrización y la autoconfiguración óptima de máquinas-herramienta, robots y procesos industriales, desde equipos individuales hasta entornos de producción global (Villalonga, *et al.*, 2018b).

1.2 Sistemas Ciberfísicos Industriales

Los avances en los sistemas basados en Internet han abierto la posibilidad a las industrias de utilizar el ciberespacio para llevar a cabo colaboraciones eficaces y efectivas, desde cualquier punto, en los sistemas de manufactura distribuida (Qi y Tao, 2018; Takahashi, *et al.*, 2017).

En este contexto surgieron los Sistemas Ciberfísicos (SCF), los cuales son sistemas colaborativos de entidades computacionales que se encuentran en intensa conexión con el entorno físico y sus procesos, proveyendo y utilizando, al mismo tiempo, acceso a datos y a servicios disponibles en Internet (Letichevsky, *et al.*, 2017). Por extensión, se le llama Sistema Ciberfísico Industrial (SCFI) a los SCF aplicados en los ambientes industriales.

En el área de la manufactura se requiere un enfoque sistemático orientado a la introducción de inteligencia para lograr en las fábricas una producción continua y disminuir el tiempo de inactividad. La integración de los SCFI en la producción, la logística y los servicios puede equipar a las fábricas con máquinas autoinformadas y autoadaptables que utilicen

información para optimizar el ajuste de los patrones de producción (Navet, *et al.*, 2017; Srewil y Scherer, 2017).

A pesar de la tendencia a la introducción de las Tecnologías de la Informática y las Comunicaciones (TIC) en las pequeñas y medianas empresas, y del reciente interés de la comunidad científica en los SCF, el grado de despliegue y explotación de estos todavía es muy limitado en la industria de la manufactura. Por tanto, es una prioridad el desarrollo de SCFI, con diferentes arquitecturas, enfocados en aumentar la calidad, flexibilizar la producción y reducir los costos, basándose en la toma inteligente de decisiones y el control adaptativo de los recursos de producción (Ning, *et al.*, 2017).

Una de las aplicaciones más interesantes de los SCFI ha sido su utilización para el monitoreo del estado de las máquinas-herramientas, mediante la observación y análisis de determinados parámetros. Al igual que un doctor recomienda a su paciente chequeos de rutina periódicos de algunos parámetros claves para la salud como son la presión sanguínea, el pulso, el peso y la temperatura para poder detectar a tiempo problemas potenciales de salud, se pueden medir ciertos parámetros de los equipos industriales para evaluar su estado (Wang, 2008).

1.3 Sistemas de monitoreo basado en estados

Los sistemas de monitoreo son de vital importancia para el funcionamiento de cualquier proceso productivo pues nos brindan la posibilidad de comprender, de una manera detallada, su dinámica de trabajo. Una vez que se dispone de toda la información necesaria para la toma de decisiones, se pueden realizar correcciones sobre los principales parámetros

de ajuste del proceso productivo en cuestión, con la meta de lograr un funcionamiento más eficiente.

El monitoreo basado en estados (*condition based monitoring*, CM) es el proceso mediante el cual se determina el estado operacional y la salud de una máquina con el propósito de detectar fallas potenciales antes de que se conviertan en fallas funcionales. El CM consiste en el almacenamiento periódico o continuo de colecciones de datos, su análisis, interpretación y diagnóstico (Kim, *et al.*, 2006). Este tipo de monitoreo ha probado ser efectivo, permitiendo aumentar la seguridad, la rentabilidad y la competitividad, por lo que está asumiendo un papel cada vez más importante en la industria (Liu, *et al.*, 2012).

En sectores industriales, las aplicaciones de CM más reportadas se centran en: análisis y diagnóstico de vibraciones; análisis de lubricantes; análisis de emisiones acústicas; termografía infrarroja; pruebas de ultrasonido, análisis del estado del motor y el consumo, entre otros (Ajith, *et al.*, 2017; Verma, *et al.*, 2017). A continuación se realiza una breve descripción de algunas de estas técnicas de CM.

1.3.1 Análisis de vibraciones

El análisis de vibraciones es una de las técnicas de CM más difundida en la industria. La misma se puede realizar tanto on-line como off-line y se utiliza para detectar funcionamientos defectuosos del sistema basado en la medición de las señales de vibración (Wang, 2008). Una gran variedad de enfoques se han presentado para el análisis de estas señales, los cuales pueden ser clasificados en: análisis en el dominio del tiempo, en el dominio de la frecuencia y en el dominio tiempo-frecuencia. También han sido aplicadas

al análisis de vibraciones diversas técnicas de Inteligencia Artificial (IA) para el modelado como son las Redes Neuronales Artificiales (RNA) y los sistemas difusos (A. Paya, *et al.*, 1997; Li, *et al.*, 2000a; Mechefske, 1998; Subrahmanyam y Sujatha, 1997).

1.3.1.1 Análisis de vibraciones en el dominio del tiempo

Para realizar un análisis de vibraciones en el dominio del tiempo se requiere del conocimiento de una persona altamente calificada, con suficiente experiencia y exposición a este tipo de señales o de modelos de IA entrenados para detectar valores atípicos que indiquen posibles fallos. Los enfoques en el dominio del tiempo presentan como principal ventaja que no se pierde ninguna información antes del análisis. Sin embargo, la desventaja es que se cuenta con muchos datos para poder realizar el diagnóstico de fallos de una manera fácil y clara (Patil, *et al.*, 2008). Generalmente se analizan características de la señal de vibración como son el valor de pico, el promedio, la desviación estándar, la raíz cuadrática media, la distorsión, la curtosis, el factor de cresta, el factor de impulso, el factor de forma y el factor de claridad.

1.3.1.2 Análisis de vibraciones en el dominio de la frecuencia

En este tipo de enfoque, se requiere transformar los datos del dominio del tiempo al dominio de la frecuencia. El método más básico, pero a la vez más difundido, es el análisis espectral basado en la Transformada Discreta de Fourier (TDF). En el análisis espectral se puede ver como la energía se distribuye sobre la frecuencia, lo cual es muy útil para determinar fenómenos periódicos y su intensidad. Las vibraciones de naturaleza repetitiva se muestran de manera muy clara en las frecuencias en las que ocurren. Generalmente, cuando hay defectos en las partes móviles de los mecanismos se producen pulsos de corta

duración, lo cual excita la frecuencia natural del sistema, produciendo un incremento de la energía de las vibraciones (Budynas, *et al.*, 2008).

1.3.1.3 Análisis de vibraciones en el dominio tiempo-frecuencia

En el análisis de vibraciones en el dominio tiempo-frecuencia se utiliza la información de ambos dominios para analizar las características de transición de las señales. Varias técnicas se han aplicado en este enfoque, entre las que se pueden destacar la Transformada de Fourier de Tiempo Reducido (TFTR), la distribución de Wigner-Ville y la Transformada Discreta de Onduleta (TDO), siendo esta última la herramienta más utilizada en la actualidad.

La TDO emplea funciones de onduleta básicas ortonormales y no muestra redundancias. Los coeficientes calculados para la TDO pueden ser utilizados para conformar un conjunto de características que describen sin ambigüedad los diferentes tipos de señal, con lo que se pueden diferenciar las señales anormales que pudieran indicar malfuncionamientos. La función de dilatación de la TDO puede ser representada como un árbol de filtros pasa-bajo y pasa-alto para los componentes de la frecuencia. Los componentes de baja frecuencia de la señal original se filtran sucesivamente para obtener componentes más bajas, mientras que los de alta frecuencia se desechan (Villalonga, *et al.*, 2018a).

1.3.1.4 Análisis de vibraciones mediante RNA

Las RNA pueden ser vistas como una amplia clase de modelos gráficos parametrizados de redes con unidades interconectadas que evolucionan en el tiempo (Baldi y Brunak, 2001).

Estas fueron desarrolladas originalmente con el objetivo de modelar el procesamiento de la información y el aprendizaje en el cerebro.

Las neuronas biológicas son un tipo especial de células que reciben señales eléctricas de las neuronas vecinas a las que están conectadas, procesan esas señales y generan nuevas señales para otras neuronas vecinas conectadas a ellas. La operación de estas células para decidir la señal de salida como una función de las señales de entrada no se conoce con exactitud hasta la fecha. Aunque, la mayoría de los biólogos coinciden en que después de recibir las señales de entrada se realiza la suma ponderada de las mismas y si supera un umbral determinado se produce un intercambio eléctrico entre el interior y el exterior de la neurona y se genera una señal de salida cuya amplitud está limitada mediante una función de activación no lineal (Konar, 2000). Esta idea es la base del modelo propuesto por McCulloch y Pitts (1943) que se describe por la siguiente ecuación:

$$y = \sigma\left(\sum_{i=1}^n W_i U_i + b\right) \quad (1.1)$$

Donde: U_i denota las entradas, b denota un valor de sesgo y W_i denota los pesos de los coeficientes sinápticos, siendo la sinapsis una unión intercelular especializada entre neuronas y σ una función de activación no lineal.

Por su parte, las neuronas artificiales constituyen analogías muy simplificadas de las neuronas biológicas. Hay una gran variedad de modelos de neuronas artificiales, tomando como base el anterior y aplicándole diferentes funciones de activación. Entre ellos se encuentran el perceptrón, el adaline y otros que utilizan funciones sigmoideas, tangentes hiperbólicas, etc. (Patan, 2008).

Las RNA se construyen interconectando varias neuronas artificiales entre sí. En estas redes, las neuronas se suelen agrupar en unidades estructurales denominadas capas, donde las salidas de las neuronas de la capa precedente se conectan a las entradas de la siguiente capa. También las neuronas de una capa pueden agruparse formando grupos neuronales o clústeres. Se distinguen tres tipos de capas: de entrada, de salida y ocultas. La capa de entrada está compuesta por neuronas que reciben datos del entorno. La capa de salida es aquella cuyas neuronas proporcionan la respuesta de la red neuronal. Las capas ocultas no se conectan directamente con el entorno, pero le aportan grados de libertad adicionales a la red, proporcionando una mayor riqueza computacional. La estructura descrita permite obtener sistemas con paralelismo de cálculo, memoria distribuida y adaptabilidad.

Quizás uno de los aspectos más interesantes de las RNA es el proceso de entrenamiento o aprendizaje, en el cual se produce el ajuste de los parámetros libres de la red a partir de un conjunto de datos de entrada-salida. Este proceso es usualmente iterativo y se pueden encontrar muchas formas de realizarlo, entre las que destacan: supervisado, no supervisado, híbrido y reforzado (Martín y Sanz, 2001).

Las RNA pueden ser aplicadas al análisis de vibraciones ya que se pueden entrenar para reconocer cuándo los parámetros de la máquina se encuentran en el régimen de trabajo deseado y cuándo hay síntomas de un funcionamiento defectuoso. Los datos de entrada en este caso serían las principales características de la señal de vibraciones y la salida permitiría determinar si la máquina está funcionando cómo se espera o si hay alguna anomalía.

1.3.1.5 Análisis de vibraciones mediante sistemas basados en lógica difusa

Los sistemas difusos tienen su base en la teoría de conjuntos difusos, que a su vez parten de la teoría clásica de conjuntos, añadiendo una función de pertenencia al conjunto, definida esta como un número real entre 0 y 1. Entre las funciones de pertenencia más utilizadas se encuentran: trapezoidal, singleton, triangular, de tipo S, etc.

Los conjuntos difusos permiten agrupar objetos o sucesos por el valor de una determinada magnitud. La teoría de conjuntos difusos hizo posible el desarrollo de la lógica difusa, la cual, al igual que la lógica clásica, se ocupa del razonamiento formal con proposiciones, pero se diferencia en que los datos pueden tomar valores intermedios entre verdadero y falso. Los sistemas difusos poseen tres componentes fundamentales: el borrosificador, la máquina de inferencias y el desborrosificador (Martín y Sanz, 2001).

El borrosificador establece una relación entre puntos de entrada no borrosos al sistema y sus correspondientes conjuntos borrosos. Esta relación se establece mediante las funciones de pertenencia.

La máquina de inferencias o sistema de inferencia se encarga de generar conclusiones, en forma de conjuntos difusos, a partir de la comparación de los datos de entrada, después de haber sido tratados por el borrosificador, y de una base de reglas, de donde se obtienen las reglas que se aplican a cada situación. Estas reglas son del tipo Si Antecedentes, Entonces Consecuentes.

Por otra parte, el desborrosificador establece una relación entre los conjuntos difusos a la salida de la máquina de inferencias y los datos de salida, dándoles a estos últimos valores

concretos mediante métodos matemáticos. Para obtener estos valores se pueden aplicar distintos métodos como son: método del centro de gravedad, método del promedio de los centros y método del máximo, entre otros.

Los sistemas difusos pueden tener entradas donde la información es imprecisa o ambigua, lo cual es muy útil para tratar con problemas de la vida real (Martín y Sanz, 2001). Precisamente este es uno de los principales incentivos para utilizar este tipo de sistemas para el CM, ya que las características de la señal de vibraciones tienen un rango de valores para los cuales el equipo se encuentra funcionando correctamente, y no un valor único y preciso. El funcionamiento de la máquina o proceso puede ser modelado en la base de reglas para casos donde esté en régimen normal de trabajo y casos donde haya malfuncionamiento y, de esta manera, al pasarle al sistema las características de la señal de vibraciones como datos de entrada, se puede determinar si la máquina está funcionando correctamente o no.

1.3.2 Análisis de lubricantes

El CM basado en el análisis de lubricantes requiere el muestreo y revisión de las propiedades de los lubricantes utilizados en los motores, transmisiones, sistemas hidráulicos, etc. Este tipo de análisis permite la identificación de problemas potenciales antes de que se requiera una reparación de gran envergadura; además se puede reducir la frecuencia de cambios de lubricantes y de esta manera incrementar el valor de reventa de los equipos usados (Wang, 2008).

El muestreo y análisis regular de las características de los lubricantes permite establecer un punto de partida para determinar cuándo se presentan valores anormales o cuando ha ocurrido la contaminación. Los lubricantes permiten conocer el estado de las máquinas ya que se encuentran en contacto directo con los motores o componentes mecánicos, por lo que las partículas metálicas que se desprenden a causa del desgaste pasan a formar parte del lubricante. Estas partículas son tan pequeñas que permanecen en suspensión. Muchos productos de la combustión también quedan atrapados en el lubricante, por lo que se puede afirmar que en el lubricante se refleja la historia de trabajo de la máquina.

En el análisis de lubricantes típico se busca detectar la presencia de diversos materiales, determinando el volumen de las partículas y su naturaleza. Diversas técnicas son usadas para este fin como son el análisis de espectrometría, el escaneo con microscopio de electrones, el análisis de dispersión de rayos X, el escaneo con microscopio óptico de baja potencia y la cuantificación de residuos ferrosos, entre otros (Wang, 2008).

1.3.3 Análisis de emisiones acústicas, termografía infrarroja y pruebas de ultrasonido

El análisis de emisiones acústicas es ampliamente utilizado en el CM de máquinas rotatorias. Comparado con los métodos tradicionales de análisis de vibraciones, el enfoque de las emisiones acústicas tiene la ventaja de mejorar significativamente la relación de la señal con respecto al ruido. También puede ser utilizado para máquinas no rotatorias donde los defectos generen señales no repetitivas, y por tanto no sea posible el análisis en el dominio de la frecuencia. Se debe destacar que los transductores utilizados para el análisis de emisiones acústicas deben tener una banda relativamente estrecha para poder detectar fallos en las altas frecuencias. En esta técnica se analizan generalmente señales que se

encuentran entre 100 kHz y 1 MHz, y se tiene en cuenta el tiempo de subida, la duración y el pico máximo, entre otras características de la señal (Kim, *et al.*, 2006; Wang, 2008).

Otro de los enfoques para el CM es el de la termografía infrarroja, en el cual se busca medir la energía electromagnética infrarroja emitida por las máquinas. Todos los objetos emiten energía en forma de calor o de ondas electromagnéticas, pero solo una pequeña porción es visible para los humanos. Por esta razón a bajas temperaturas se requiere de una cámara infrarroja para poder observar estas emisiones de energía. La imagen mostrada por la cámara permite hacer visible la energía térmica, de manera que se puede analizar y detectar cualquier anomalía (Wang, 2008).

Por otra parte, el CM también puede ser realizado mediante la técnica de ultrasonido. Se le llama ultrasonido a las ondas sonoras con una frecuencia superior a 20 kHz, las cuales son inaudibles para los humanos. La mayoría de las máquinas emiten patrones de sonido consistentes bajo condiciones normales de operación, muchos de ellos en el rango de los ultrasonidos. Estos patrones de sonido pueden ser estudiados y reconocidos; y cualquier alteración que se presente en ellos puede indicar deterioro en los componentes de las máquinas. Mediante esta técnica se pueden detectar el deterioro de partes rotatorias, fugas de fluidos hidráulicos o aire comprimido, fugas de vacío, pérdida de la hermeticidad en válvulas, etc. El análisis de ultrasonidos se centra en las señales desde 20 kHz hasta 100 kHz; en este sentido se puede decir que es complementario con el análisis de emisiones acústicas, ya que en este último se tienen en cuenta señales con una frecuencia superior. En el análisis de ultrasonido las características que generalmente se miden de la señal son el valor de la raíz cuadrática media y los decibeles (Kim, *et al.*, 2006).

Las técnicas antes descritas pueden ser de ayuda para determinar el estado o condición de las máquinas o plantas industriales. Basándose en un análisis técnico de los datos adquiridos mediante el CM se pueden tomar decisiones para el mantenimiento; de forma que el funcionamiento de la máquina o planta industrial sea seguro y eficiente.

1.4 Conclusiones parciales del capítulo

Una vez finalizado el presente capítulo, se ha podido arribar a las siguientes conclusiones parciales:

1. El paradigma de Computación en la nube permite utilizar servicios a través de Internet como pueden ser aplicaciones, plataformas o infraestructura.
2. Los SCFI permiten recolectar información del entorno de producción y procesarla en el ciberespacio.
3. Es necesario desarrollar nuevos SCFI que puedan ser utilizados en las empresas de manufactura.
4. Las principales técnicas de CM se basan en determinar cuándo algún parámetro se encuentra fuera de sus valores típicos para indicar una posible falla en la máquina o proceso.

CAPÍTULO 2. PROPUESTA DE LA ARQUITECTURA

El objetivo de este capítulo es presentar la propuesta de la arquitectura del SCFI para realizar el CM en procesos de manufactura. Se profundiza en la composición de los módulos que conforman la arquitectura, se brinda más información sobre el servicio de almacenamiento en la nube utilizando almacenes de datos y se detallan los algoritmos de IA propuestos para el sistema y sus métodos de entrenamiento. También se explica el método de optimización para estos algoritmos de IA y el flujo de información para poder visualizar los datos del proceso.

2.1 Arquitectura general del sistema

La arquitectura del SCFI que se propone en este capítulo está orientada a la detección temprana de eventos que pudieran conducir a una falla en los procesos de manufactura. La misma se compone de un conjunto de módulos que permiten realizar el CM mediante el análisis de vibraciones en el dominio del tiempo utilizando técnicas de IA. Se optó por realizar el CM basado en la señal de vibraciones ya que el 90% de los fallos en las maquinarias industriales vienen precedidos por cambios en dicha señal (Devendiran y Manivannan, 2016).

El CM se realiza de manera local en la industria con datos obtenidos de equipos de Control Numérico Computarizado (CNC). Los datos que generan estos equipos, además de ser analizados localmente, son almacenados en la nube, ya que cada CNC cuenta con conexión a Internet mediante una red cableada o inalámbrica como se muestra en la Figura 2.1.

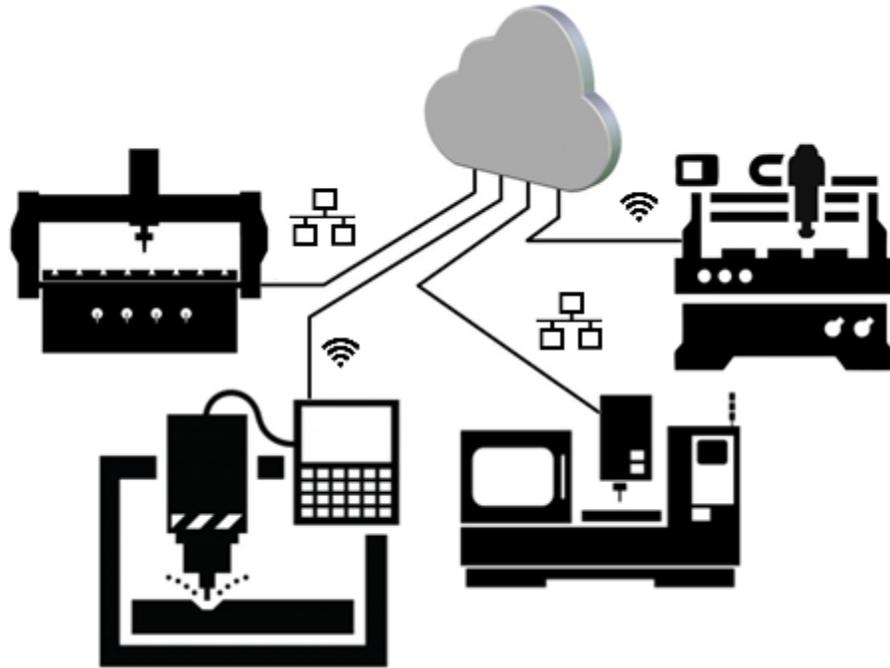


Figura 2.1 Conexión de los CNC a Internet

Una descripción general de la distribución de los módulos que componen el sistema de CM y el servicio de almacenamiento en la nube se muestra en la Figura 2.2. El análisis de vibraciones se realiza a través de un modelo que se selecciona de una librería disponible localmente. Sin embargo, para el entrenamiento y la optimización del modelo se requiere acceder a los datos almacenados en la nube. El módulo de visualización permite observar el estado reciente y actual del CNC; pero para poder ver los datos históricos es necesario acceder a estos en el almacén de datos.

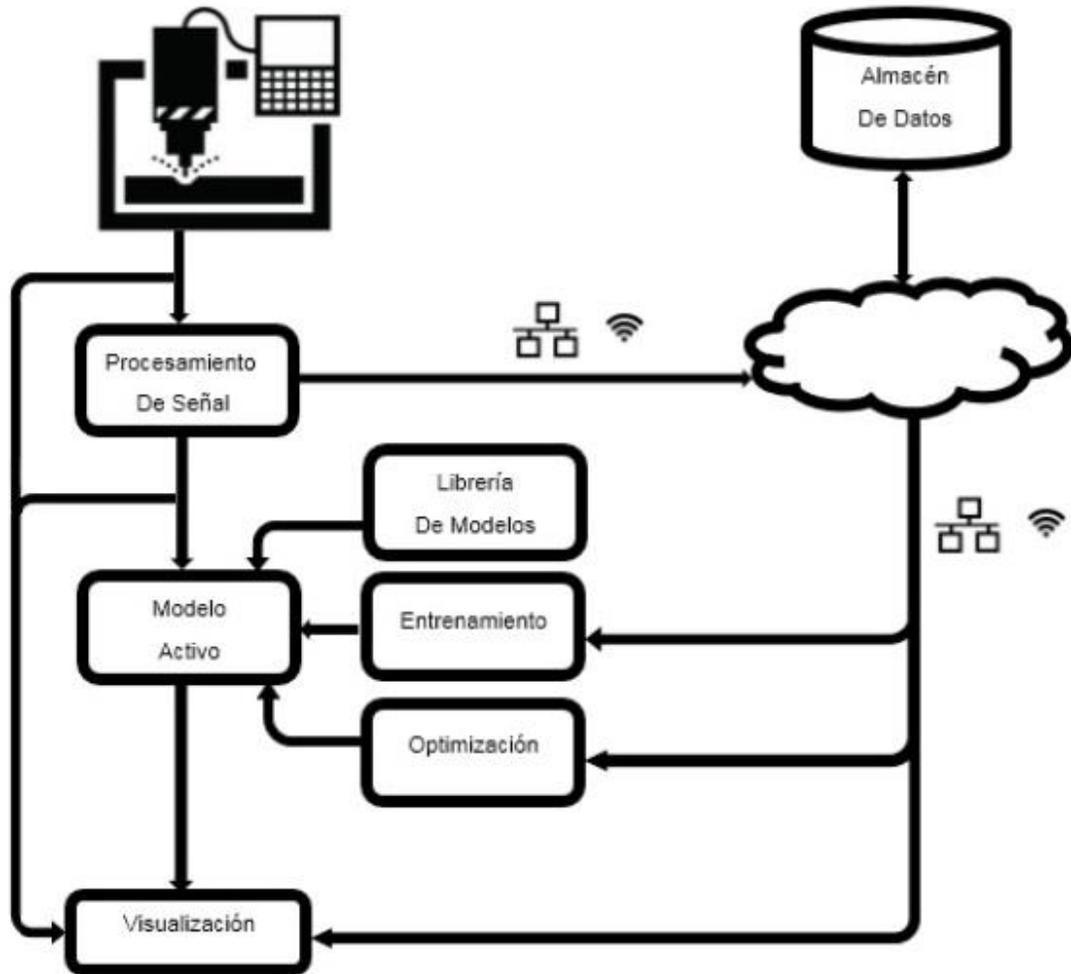


Figura 2.2 Distribución general de los módulos del sistema de monitoreo

2.2 Almacén de datos

Uno de los servicios de Computación en la nube más utilizados en el ámbito industrial es el DaaS, ya que debido a la gran cantidad de información que se genera en la industria es más rentable almacenarla y gestionarla en Internet que en servidores propios. El aumento en el costo de los centros de almacenamiento de datos no está dado por un incremento en el precio del hardware, sino de la demanda energética para su correcto funcionamiento (Hwang, 2017). Al utilizar el DaaS la responsabilidad de la gestión energética pasa de las

industrias al proveedor del servicio de almacenamiento. Además, al utilizar un almacén de datos se incrementa la capacidad de procesamiento de la información.

Debido al gran volumen de información a almacenar en las industrias, una base de datos convencional resulta inapropiada y por esta razón se utilizan almacenes de datos. Los almacenes de datos son repositorios integrados de datos que fueron concebidos para facilitar la compilación de reportes regulares sobre el estado de algún proceso mediante la recolección continua, limpieza y resumen de los datos principales del sistema analizado (Chakrabarti, *et al.*, 2009).

Los almacenes de datos están optimizados para llevar a cabo operaciones de consulta de alto rendimiento, ya que en ocasiones la información que se necesita conocer requiere que miles o cientos de miles de datos sean analizados para poder generar una simple respuesta (Kimball y Ross, 2013). Resulta más conveniente que estas operaciones que requieren de un potente hardware sean llevadas a cabo en un almacén de datos de un proveedor de DaaS, que en la industria.

Los almacenes de datos típicamente contienen una gran cantidad de datos históricos pudiendo proceder estos de múltiples bases de datos. Los almacenes de datos constituyen también una fuente de información limpia y organizada para aplicar la Minería de Datos (MD). Por tanto, utilizar un almacén de datos en el CSFI aporta la ventaja de que no haya que realizar modificaciones en la arquitectura si se desea hacer MD posteriormente. Además los datos en el almacén pueden ser explorados utilizando el Procesamiento Analítico en línea, generadores de reportes y herramientas de consulta.

2.3 Procesamiento de la señal

La arquitectura propuesta está diseñada para realizar el CM mediante el análisis de la señal de vibraciones en el dominio del tiempo. La señal de vibraciones se obtiene de los CNC mediante el uso de acelerómetros que generalmente vienen integrados en estos equipos; de no ser así, estos sensores se deben instalar cerca de los rodamientos. La medición de la señal se realiza de manera continua para asegurar que cualquier desgaste que se comience a producir en los rodamientos sea detectado rápidamente (Hemmati, *et al.*, 2016). Cada vez que se mide la señal se debe registrar su valor durante 10 segundos. Se debe tener presente que la frecuencia de muestreo debe ser, como mínimo, superior al doble de la frecuencia de la señal de vibraciones para de esta forma asegurar que se pueda reconstruir la señal fielmente. En Chebil, *et al.* (2011) se propone una frecuencia de muestreo de 12 kHz para equipos operando a frecuencias cercanas a 30 Hz. En el Anexo 1 se muestran gráficos de señales de vibración de rodamientos operando a valores de frecuencia inferiores y cercanos a 30 Hz.

La señal de vibraciones al ser medida contiene mucho ruido asociado, por lo que, antes de extraer sus principales características o realizar cualquier análisis con ella, este ruido debe ser eliminado. Para lograr la eliminación del ruido se utiliza un filtro pasa alto. Luego de haber suprimido el ruido de la señal se pasa a determinar los valores de las características que serán utilizadas para el CM. En (Chebil, *et al.*, 2011) y (Villalonga, *et al.*, 2018a) se proponen utilizar para el análisis de la señal de vibraciones en el dominio del tiempo los estadígrafos valor de pico (V_p), promedio (μ), desviación estándar (σ), raíz cuadrática media (RCM), factor de cresta (F_{cr}), factor de claridad (F_{cl}), factor de impulso (F_i),

factor de forma (F_f), *distorsión* y *curtosis*, los cuales se calculan de acuerdo a las siguientes ecuaciones:

$$V_p = \frac{1}{2} [\max(x_i) - \min(x_i)] \quad (2.1)$$

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.2)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (2.3)$$

$$RCM = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i)^2} \quad (2.4)$$

$$F_{cr} = \frac{V_p}{RCM} \quad (2.5)$$

$$F_{cl} = \frac{V_p}{\left(\frac{1}{N} \sum_{i=1}^N \sqrt{|x_i|}\right)^2} \quad (2.6)$$

$$F_i = \frac{V_p}{\frac{1}{N} \sum_{i=1}^N |x_i|} \quad (2.7)$$

$$F_f = \frac{RCM}{\frac{1}{N} \sum_{i=1}^N |x_i|} \quad (2.8)$$

$$distorsión = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^3}{\sigma^3} \quad (2.9)$$

$$curtosis = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^4}{\sigma^4} \quad (2.10)$$

Una vez que se tienen los valores de las principales características de la señal de vibraciones, estos se almacenan en la nube y, al mismo tiempo, se seleccionan los que se pasarán al módulo en el cual se realizará el monitoreo de los mismos.

2.4 Librería de modelos

En este módulo se encuentran los modelos de IA que pueden ser utilizados para realizar el diagnóstico de los CNC. Aquí los modelos se encuentran en una forma genérica y sin parametrizar, ya que estos deben poder adaptarse a los requerimientos específicos de cada CNC, por ejemplo, es posible que las RNA que se utilicen para el CM en dos CNC distintos tengan un número diferente de neuronas en cada capa.

Inicialmente se prevé la inclusión de dos tipos de modelos en la librería; uno basado en un Sistema de Inferencia Difuso Takagi-Sugeno-Kang (TSK) y el otro basado en RNA Perceptrón Multicapa, pero se deja abierta la posibilidad de agregar más modelos de IA, ya sean basados en RNA, lógica difusa u otras técnicas que sean efectivas para el CM.

2.4.1 Sistema de Inferencia Difuso Takagi-Sugeno-Kang

Los Sistemas de Inferencia Difusos TSK fueron desarrollados por Takagi y Sugeno (1985) y Sugeno y Kang (1988). La principal particularidad de estos sistemas es que no requieren un desborrosificador, ya que el valor de la variable de salida calculado en cada regla no es un dato difuso, sino que es preciso. Esto es posible ya que el consecuente de cada regla se plantea en forma de una función lineal que depende de las variables de entrada. La k -ésima regla en un Sistema de Inferencia Difuso TSK tiene la siguiente forma:

Si x_1 es Q_{1k} y x_2 es Q_{2k} y ... y x_n es Q_{nk} ,

$$\text{Entonces } z = p_0^k + p_1^k x_1 + p_2^k x_2 + \dots + p_n^k x_n$$

Para calcular el valor de la variable de salida del sistema difuso TSK se utiliza la siguiente ecuación:

$$y = \frac{\sum_{i=1}^M w_i z_i}{\sum_{i=1}^M w_i} \quad (2.11)$$

Donde: M es el número de reglas difusas; w_i es el coeficiente de ponderación calculado para la i -ésima regla; y z es la consecuencia de la i -ésima regla. Los coeficientes de ponderación se determinan por la ecuación que se muestra a continuación:

$$w_i = \prod_{j=1}^N u_{ij}, i = 1, \dots, M \quad (2.12)$$

Donde: u_{ij} son las salidas de las funciones de pertenencia de la j -ésima variable en la i -ésima regla. Para funciones de pertenencia gaussianas se utiliza la siguiente ecuación:

$$u_{ij} = e^{-\frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2}}, i = 1, \dots, M; j = 1, \dots, N \quad (2.13)$$

Donde: x_j es el valor de la j -ésima variable; mientras que μ_{ij} y σ_{ij} son los centros y las desviaciones estándar de las funciones de pertenencia respectivamente. La salida de cada regla se calcula utilizando la ecuación lineal del consecuente donde p_0 es el coeficiente

del término independiente y p_1, \dots, p_n son los coeficientes asociados a cada variable de entrada. Estos coeficientes pueden tener valores diferentes en cada regla.

Los Sistemas de Inferencia Difusos TSK son más difíciles de implementar que otros sistemas basados en lógica difusa ya que el conocimiento no se puede traducir a una base de reglas intuitivamente como es el caso de los Sistemas de Inferencia Difusos Mamdani. Sin embargo, los sistemas TSK presentan un mejor rendimiento computacional y son más fáciles de integrar con técnicas adaptativas y de optimización, por esta razón son ampliamente utilizados para tratar con sistemas que tengan una dinámica no lineal (Chou, *et al.*, 2014). Además, para resolver el problema de la creación de la base de reglas se han presentado varias estrategias que permiten realizar esta tarea de manera automática.

2.4.2 Redes Neuronales Artificiales Perceptrón Multicapa

Las RNA Perceptrón Multicapa son un tipo de modelo de red neuronal de avance hacia adelante, es decir, donde las conexiones entre las capas se realizan en una sola dirección. Estas redes están compuestas generalmente de tres o más capas de neuronas, donde todas las neuronas artificiales son de tipo Perceptrón y la salida de cada neurona se calcula utilizando la ecuación 1.1.

Las RNA Perceptrón Simple pueden ser consideradas como los modelos más simples de neuronas artificiales debido a su función de activación, la cual es una función de paso discontinuo unipolar como la que se muestra en la siguiente ecuación:

$$\sigma(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases} \quad (2.14)$$

Sin embargo, las RNA Perceptrón Multicapa generalmente utilizan funciones de activación sigmoidales en las neuronas de las capas ocultas y funciones lineales en las neuronas de la capa de salida (Bishop, 2006). El hecho de utilizar este tipo de funciones en una arquitectura de varias capas hace que sea posible usar este tipo de RNA para problemas de clasificación donde los conjuntos sean linealmente separables y para la aproximación de funciones. De hecho, se ha demostrado que una RNA Perceptrón Multicapa se puede adaptar para estimar aproximadamente cualquier función con la precisión deseada (Deo y Biazar, 2018; Hagan, *et al.*, 2014).

La convolución de este tipo de red se puede cambiar variando el número de capas y el número de unidades en cada capa, pero lo más común es que se utilice una arquitectura compuesta por 3 capas, ya que con este tipo de redes se pueden resolver una gran cantidad de problemas a los que son aplicables las RNA.

2.5 Entrenamiento

El entrenamiento o aprendizaje es el proceso mediante el cual se ajustan los parámetros de un modelo de IA para que este sea capaz de reproducir, con una precisión aceptable, una relación entre los datos de entrada y los de salida. Entre las diversas técnicas de entrenamiento que existen, una de las más utilizadas es el supervisado. En este tipo de entrenamiento se utiliza un conjunto de datos de entrada y otro conjunto de datos que representa la salida que se desea produzca el sistema. Los valores de los parámetros del modelo se varían de acuerdo a alguna estrategia que permita que las salidas reales que produzca el sistema sean razonablemente similares a las salidas deseadas. La estrategia utilizada depende del tipo de modelo de IA con que se esté tratando; de esta manera, el

entrenamiento de los sistemas difusos se diferencia al de las RNA aunque se esté realizando un entrenamiento supervisado en ambos casos.

2.5.1 Entrenamiento para sistemas difusos Takagi-Sugeno-Kang

El entrenamiento para los sistemas TSK consiste en la construcción de la base de reglas. Esta tarea puede resultar bastante difícil incluso para expertos en los procesos que se estén analizando ya que no se puede realizar de forma intuitiva, debido a que el consecuente de cada regla es una ecuación (Díaz-Contreras, *et al.*, 2014; Schnitman, *et al.*, 2001). Es por esta razón que comúnmente se utiliza alguna técnica de agrupamiento que permita el diseño automático de la base de reglas. Las técnicas de agrupamiento permiten asociar datos similares en una misma categoría o clúster. Una vez que se tienen los clústeres, cada uno de ellos puede ser traducido en el antecedente una regla. Por otra parte, para el consecuente de cada regla se pueden calcular los coeficientes de la ecuación lineal como parámetros de regresión mediante Mínimos Cuadrados (Ren, *et al.*, 2006).

2.5.1.1 Agrupamiento sustractivo

Una de las técnicas más difundidas para la generación automática de reglas en Sistemas de Inferencia Difusos es el Agrupamiento Sustractivo, el cual fue presentado por Li, *et al.* (2000b). Esta técnica presenta la ventaja de ser un algoritmo no iterativo frente a otras técnicas con la misma finalidad, que sí lo son, como es el caso de K-medios o C-medios; además el Agrupamiento Sustractivo ha demostrado ser un algoritmo más consistente (Raoa, *et al.*, 2015).

El método del Agrupamiento Sustractivo fue diseñado para lidiar con el inconveniente del crecimiento exponencial de los cálculos debido al aumento de la dimensión del problema en los algoritmos de agrupamiento. En este método se utilizan valores del conjunto de datos como centros de clúster potenciales. Aunque el centro de un clúster no tiene por qué ser necesariamente un dato, en muchos casos resulta ser una muy buena aproximación (Widodo, 2017).

Para poder aplicar el Agrupamiento Sustractivo a un conjunto de datos m -dimensionales de entrada-salida, es necesario proyectar los datos en el espacio de entrada. Además se requiere que todos los datos sean normalizados, de manera que estén contenidos en un hipercubo unitario (Ren, *et al.*, 2006). Una vez que se hayan cumplido estos 2 requisitos se procede al cálculo del potencial de cada punto utilizando la ecuación:

$$p_i = \sum_{j=1}^q e^{-a\|x_i - x_j\|^2} \quad (2.15)$$

Donde: q representa el número de puntos y $\| \cdot \|$ representa la distancia euclidiana. El valor de a se puede calcular mediante la siguiente ecuación:

$$a = \frac{4}{r_a^2} \quad (2.16)$$

Donde: r_a representa el radio del clúster.

Después de haber calculado el potencial de cada punto, el que tenga mayor valor es seleccionado como centro del primer clúster. Si se asume que x_1^* es la ubicación del centro del primer clúster y p_1^* su potencial, se puede recalculer el potencial de cada punto x_i mediante la siguiente expresión:

$$p_i \leftarrow p_i - p_1^* e^{-\beta \|x_i - x_1^*\|^2} \quad (2.17)$$

El cálculo del valor de β requiere que se determine previamente el valor de r_b . Las ecuaciones que permiten calcular los valores de estos dos parámetros se muestran a continuación:

$$\beta = \frac{4}{r_b^2} \quad (2.18)$$

$$r_b = \eta r_a \quad (2.19)$$

Donde: η es el factor de reducción.

Al ser reducido el potencial de todos los puntos mediante la ecuación 2.17, el punto que quede con el mayor potencial es escogido como centro del segundo clúster. Seguidamente se vuelve a reducir el potencial de cada punto y el que quede con mayor potencial se elige como próximo centro de clúster y así sucesivamente. La expresión que se utiliza para reducir el potencial de los puntos después de encontrado el k -ésimo centro de clúster es la siguiente:

$$p_i \leftarrow p_i - p_k^* e^{-\beta \|x_i - x_k^*\|^2} \quad (2.20)$$

Donde: x_k^* es la ubicación del k -ésimo centro de clúster y p_k^* es el valor de su potencial.

El proceso de adquirir un nuevo centro de clúster y recalcular los potenciales se debe repetir hasta que se cumpla con el criterio de parada que se muestra en el siguiente pseudocódigo:

Código 2.1 Pseudocódigo del criterio de parada del Agrupamiento Sustractivo

```
Si ( $p_k^* > \delta p_1^*$ )
    Aceptar a  $x_k^*$  como centro de clúster, recalculando los
    potenciales y continuar.
Si no Si ( $p_k^* > \varepsilon p_1^*$ )
    Rechazar a  $x_k^*$  y finalizar el proceso de agrupamiento.
Si no
     $d_{\min}$  = distancia más corta entre  $x_k^*$  y todos los centros
    de clúster previos
    Si ( $d_{\min} / r_a + p_k^* / p_1^* \geq 1$ )
        Aceptar a  $x_k^*$  como centro de clúster, recalculando
        los potenciales y continuar.
    Si no
        Rechazar a  $x_k^*$  y hacer el potencial de ese punto
        igual a 0. Asignar el nuevo punto con mayor
        potencial a  $x_k^*$  y resetear
Fin
```

Donde: δ es el radio de aceptación y ε es el radio de rechazo. Cada clúster puede ser traducido en el antecedente de una regla difusa como las presentadas en la sección anterior, donde los Q_{nk} son funciones gaussianas y representan a un conjunto difuso. Los valores de μ_{ij} serían los centros de clúster y los valores de σ_{ij} pueden calcularse como se muestra en la siguiente ecuación:

$$\sigma_{ij} = \sqrt{\frac{1}{2a}} \quad (2.21)$$

Donde: el valor de a se obtiene de la ecuación 2.16.

Este algoritmo de agrupamiento ha sido utilizado con resultados satisfactorios en problemas de modelación y control de sistemas no lineales (Dhandapani y Chandralekha, 2017; Mohan y Rani, 2017). Además, gracias a su rendimiento también es posible su aplicación a la Minería de Datos (Castañon-Puga, *et al.*, 2015).

2.5.1.2 Mínimos Cuadrados

Al aplicar el algoritmo de Agrupamiento Sustractivo se obtienen todos los elementos suficientes para definir los antecedentes de las reglas, siendo necesario a continuación, crear los consecuentes. Los parámetros p_n de los consecuentes pueden ser obtenidos con el algoritmo de estimación de mínimos cuadrados. Este método se ha utilizado para muchas aplicaciones, siendo una de las más importantes el ajuste a datos (Boyd y Vandenberghe, 2018). El algoritmo de mínimos cuadrados permite calcular un hiperplano de regresión para los valores de salida de los puntos que pertenecen a cada clúster de manera que se minimice la varianza residual. En notación matricial, el hiperplano se calcula por la ecuación:

$$\mathbf{P}_i = (\mathbf{X}_i' * \mathbf{X}_i)^{-1} * \mathbf{X}_i' * \mathbf{Y}_i \quad (2.22)$$

Donde: \mathbf{X}_i es la matriz de datos de entrada de los puntos del i -ésimo clúster y \mathbf{Y}_i es la matriz de datos de salida de los puntos del i -ésimo clúster.

Cada regla del Sistema de Inferencia Difuso TSK requiere que se calcule un hiperplano diferente para la formación del consecuente. Estos hiperplanos calculados mediante el método de mínimos cuadrados se utilizan posteriormente para predecir el comportamiento del sistema en la vecindad de los clústeres a los cuales están asociados.

2.5.2 Entrenamiento para Redes Neuronales Artificiales Perceptrón Multicapa

En el caso de las RNA el proceso de entrenamiento consiste en la asignación adecuada de los pesos sinápticos. El método que se utiliza generalmente para entrenar RNA Perceptrón

Multicapa es la retropropagación de errores. En la retropropagación de errores se calcula la pendiente de una función de pérdida con referencia a todos los pesos en el modelo de red y se equilibran los pesos mediante el gradiente descendiente, en un intento de minimizar la función de error (Villalonga, *et al.*, 2018b).

La retropropagación de errores permite calcular simultáneamente todos los pesos de una capa en cada paso, siendo su costo computacional mucho menor que el de otros enfoques y su tiempo de ejecución mucho más rápido. Este algoritmo se comenzó a utilizar ampliamente a partir de 1986 y es el método de entrenamiento que se usa incluso para las redes neuronales más modernas con un gran número de capas ocultas debido a su eficacia (Nielsen, 2015).

La retropropagación de errores exige una salida conocida y deseada para cada valor de entrada para el cálculo del gradiente de la función de pérdida. Por tanto, es considerado un método de aprendizaje supervisado. Requiere que la función de activación utilizada por las neuronas artificiales sea de naturaleza diferenciable. El aprendizaje se produce actualizando los pesos de conexión, de cada neurona, después de procesar cada información, según el error del resultado en comparación con el valor esperado (Sethy, *et al.*, 2016).

Se representa el error para el n -ésimo dato de entrenamiento en el nodo de salida j mediante la ecuación siguiente:

$$e_j(n) = d_j(n) - y_j(n) \quad (2.23)$$

Donde: d es el valor esperado, mientras que y es el valor producido por el Perceptrón. Luego se actualizan los pesos de las neuronas en función de las correcciones que reducen el valor del error, dada por:

$$\varepsilon(n) = \frac{1}{2} \sum [e_j(n)]^2 \quad (2.24)$$

Utilizando el gradiente descendiente, el cambio en cada peso se expresa de la siguiente forma:

$$\Delta W_{ji}(n) = -\eta \frac{d\varepsilon(n)}{dv_j(n)} y_i(n) \quad (2.25)$$

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 tomen un valor adecuado lo suficientemente rápido, 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. Se puede probar que para una neurona de salida esta derivada se puede simplificar a:

$$-\frac{d\varepsilon(n)}{dv_j(n)} = e_j(n) f'(v_j(n)) \quad (2.26)$$

Donde: f' es la derivada de la función de activación. El análisis es más difícil para el cambio del peso a un nodo oculto, pero se puede demostrar que la derivada es:

$$-\frac{d\varepsilon(n)}{dv_j(n)} = f'(v_j(n)) \sum_k -\frac{d\varepsilon(n)}{dv_j(n)} W_{kj}(n) \quad (2.27)$$

Dependiendo de los cambios en los pesos de las k -ésimas neuronas, 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.

2.6 Optimización

Una vez que los modelos han sido entrenados, estos ya se pueden utilizar para realizar el CM. Sin embargo, no se puede asegurar que los valores asignados a los parámetros requeridos para definir el modelo o entrenarlo hayan sido los más idóneos; es por esta razón que se propone antes de usar los modelos, aplicarles alguna técnica de optimización para la parametrización.

Muchos de los algoritmos de optimización aplicados a la simulación no requieren de una descripción algebraica del sistema, sino que dependen solamente de los datos de entrada-salida que recibe y devuelve el modelo. Cabe destacar que estos algoritmos son no determinísticos, por lo que se hace difícil evaluar su desempeño; sin embargo, se han alcanzado resultados muy alentadores en este campo (Amaran, *et al.*, 2016).

Para realizar la optimización se decidió utilizar el método de la Entropía Cruzada, el cual fue propuesto por Rubinstein (1997) como un algoritmo de aprendizaje estocástico para la estimación de la probabilidad de eventos aislados. Su uso ha sido expandido como una herramienta genérica y eficiente para resolver problemas numéricos de gran dificultad y problemas de optimización combinatoria NP-Hard (Boer, *et al.*, 2005; Rubinstein y Kroese, 2004).

Como plantean Ünveren y Acan (2007), el método se ejecuta iterativamente en dos fases, el muestreo aleatorio de los parámetros del objeto y el algoritmo de adaptación de los parámetros basado en las mejores aproximaciones durante la primera fase. La idea principal es tomar más muestras alrededor de las mejores aproximaciones, donde la medida de que tan buena es la aproximación se realiza utilizando una función objetivo predefinida.

Se considera un espacio de soluciones X y una medida de idoneidad $S(X)$ definida sobre X , de manera que $S(X) \in R$, para toda $x \in X$. Se desean encontrar aquellos elementos de X para los cuales $S(.)$ alcanza su valor mínimo o máximo global. Si se define el valor extremo de interés para $S(.)$ como γ^* , entonces hay un conjunto de muestras $X^* \subset X$ tales que:

$$S(x^*) = \gamma^*, x^* \in X^* \quad (2.28)$$

Usando las ideas de la simulación de eventos aislados, el método de la Entropía Cruzada considera el problema descrito de búsqueda u optimización como un problema de estimación en el cual los parámetros de una familia de funciones de densidad probabilística se estiman basados en las características extraídas de muestras aleatorias de X .

Para la estimación iterativa de muestras de X^* , se define un conjunto de funciones indicadoras $I\{S(x) \geq \gamma\}$, las cuales indican si $S(x)$ se encuentra por encima del nivel γ para la muestra x . Si $\{f(., \nu), \nu \in R^m\}$ es una familia de funciones de densidad probabilística para X , entonces para un vector $u, |m|$, de parámetros de funciones de densidad probabilística, el problema de optimización se transforma en el problema de

estimar la probabilidad $P_u(S(X) \geq \gamma)$. Usando las funciones indicadoras esta probabilidad es igual a:

$$L(\gamma) = P_u(S(X) \geq \gamma) = \sum I_{\{S(x) \geq \gamma\}} f(x, u) = E_u I_{\{S(x) \geq \gamma\}} \quad (2.29)$$

Donde: P_u es la distribución de probabilidad asociada a la función de densidad probabilística $f(., u)$ y E_u denota al operador de expectativa correspondiente. Cuando $\gamma = \gamma^*$, $L(\gamma)$ puede ser estimado usando el estimador de radio de semejanza con referencia a $v^* \in R^m$ de la siguiente manera:

$$v^* = \arg \max \left(\frac{1}{N} \sum I_{\{S(x) \geq \gamma\}} \ln f(X_i, v) \right) \quad (2.30)$$

Donde las X_i se generan utilizando la función de densidad probabilística $f(., v)$. Entonces, si γ se acerca a γ^* , $f(., v^*)$ toma la mayoría de las muestras cercanas a x^* y puede ser utilizado para generar una solución aproximada al problema de búsqueda u optimización con respecto a la medida de rendimiento $S(x)$. Es importante destacar que en lugar de encontrar soluciones óptimas x^* directamente para un problema particular, el método de la Entropía Cruzada apunta a encontrar la muestra de densidad más favorable $f(., v^*)$, tal que la solución óptima o cercana a la óptima pueda ser tomada de dicha muestra.

El método de la Entropía Cruzada se puede definir mediante el siguiente procedimiento, en el cual se asume que las funciones de densidad probabilística usadas son de tipo gaussianas:

Paso 1. Inicializar los parámetros del algoritmo, es decir, la media (μ) y la desviación estándar (σ) de las funciones de densidad probabilística, el tamaño de la muestra (N) y la variable de tolerancia (eps) usada para evaluar el criterio de parada.

Paso 2. Generar N vectores de muestras X_i de la forma:

$$x_{ij} = \mu + \sigma randn(), i = 1, \dots, N; j = 1, \dots, M \quad (2.31)$$

Donde $randn()$ produce números aleatorios y M define el número de parámetros del objeto.

Paso 3. Calcular el rendimiento de las muestras $S(X_i), i = 1, \dots, N$.

Paso 4. Determinar un subconjunto de mejores muestras y actualizar los parámetros de la función de densidad probabilística, utilizando la media y la desviación estándar.

Paso 5. Evaluar el criterio de parada, por ejemplo: si el máximo número de iteraciones fue excedido o los parámetros de la función de densidad probabilística son inferiores a la tolerancia eps . Si se cumple el criterio de parada, escoger la mejor de todas las muestras y terminar, si no, regresar al Paso 2.

Los parámetros a optimizar en el caso de la RNA Perceptrón Multicapa serían el número de capas ocultas y la cantidad de neuronas en cada una de ellas; mientras que en el caso del modelo TSK serían los parámetros necesarios para el agrupamiento sustractivo, es decir, el radio del clúster, el factor de reducción, el radio de rechazo y el radio de aceptación. Por

otra parte, la función objetivo que se propone en ambos casos es la reducción del error cuadrático medio.

2.7 Modelo activo

Anteriormente se describieron los modelos contenidos en la librería, los métodos de entrenamiento para ellos y además cómo se pueden optimizar; sin embargo, todos los modelos no se pueden utilizar simultáneamente para realizar el CM, sino que se debe escoger uno. Una vez que se haya seleccionado el modelo y se haya parametrizado correctamente para adaptarlo al CNC sobre el que va a actuar, ya se puede activar el modelo, es decir, ya se pueden recibir los datos del módulo de procesamiento de señal para realizar el diagnóstico.

El modelo activo solo se debe utilizar cuando el CNC se encuentre en régimen de trabajo estable y no en períodos transitorios como arranque o parada, ya que los datos que se adquieran en estos períodos pudieran indicar un funcionamiento anormal cuando en realidad no lo hay. Una vez que se procesan los datos y se determina si el equipo presenta síntomas de una posible falla, esta información se envía al módulo de visualización.

2.8 Visualización

El módulo de visualización recibe datos de varias fuentes, lo que permite conocer la información que se está procesando y transmitiendo en cada módulo del sistema, en cada momento. Mediante la arquitectura propuesta se pueden observar los valores de la señal de vibraciones al ser medida o después de haber sido procesada y que se hayan determinado los valores de sus principales características. Por otra parte, también se puede conocer el

diagnóstico emitido por el módulo donde se realiza el CM. En caso de detectarse una posible falla, esta información sería mostrada mediante una advertencia o alarma y las particularidades se podrían conocer a través del módulo de visualización, de manera tal que el operador pueda valerse de esta herramienta para tomar medidas al respecto y llevar a cabo un mantenimiento.

El módulo de visualización también permite mostrar los datos almacenados en la nube, de manera que el operador puede tener acceso a todos los registros históricos que se hayan hecho del equipo. Por otra parte, el módulo de visualización también puede vincularse a algún SaaS, ya que existe, por ejemplo, el servicio de aplicaciones para el monitoreo de información y la construcción de gráficos en Internet, como es el caso de Grafana, Power BI, Pentaho, Past3, entre otros.

2.9 Conclusiones parciales del capítulo

Una vez finalizado el presente capítulo, se ha podido arribar a las siguientes conclusiones parciales:

1. La arquitectura propuesta se compone de una serie de módulos que permiten crear modelos y entrenarlos para detectar las posibles fallas de los CNC en procesos de manufactura.
2. Basándose en la arquitectura propuesta, el CM se realizará de manera local con los datos de la señal de vibraciones recibidos de cada CNC.
3. El almacenamiento se realizará utilizando el DaaS, por lo que cada CNC debe contar con acceso a Internet.

4. El módulo de visualización permitirá observar tanto la información actual del sistema, como los datos históricos.

CAPÍTULO 3. ANÁLISIS DE LOS RESULTADOS

El objetivo de este capítulo es presentar brevemente las funcionalidades de la aplicación que se desarrolló para el CM de CNC en procesos de manufactura. Se explica cómo utilizar las diferentes ventanas de la interfaz visual para entrenar, optimizar y poner en funcionamiento los modelos de IA programados. Además se describen las pruebas realizadas al software para validar su correcto funcionamiento.

3.1 Descripción de la aplicación

Tomando como base la arquitectura propuesta se desarrolló una aplicación para realizar el diagnóstico de fallas en CNC mediante el CM. Esta aplicación permite escoger el tipo de modelo que se utilizará para supervisar el proceso, escoger la fuente de datos de entrenamiento, optimizar el modelo, utilizarlo para analizar una señal de vibraciones y visualizar los datos. Además se almacenan los datos en la nube de manera tal que después puedan ser accedidos mediante algún SaaS para graficarlos.

En el caso de utilizarse el modelo basado en Sistema de Inferencia Difuso TSK, el primer paso para su entrenamiento consiste determinar cuáles son las características de la señal de vibraciones que se van a tener en cuenta, las cuales deben coincidir con las que se tenga planeado monitorear más adelante durante la ejecución. Esta selección se puede llevar a cabo con ayuda de una interfaz visual como se muestra en la Figura 3.1.

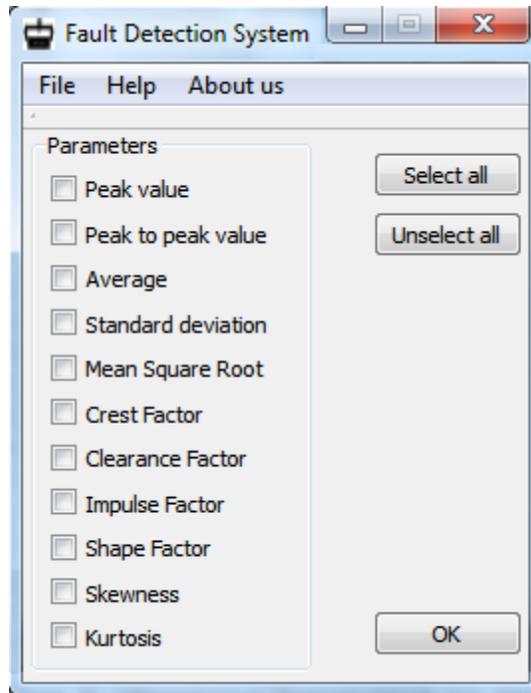


Figura 3.1 Selección de parámetros

Seguidamente se deben cargar los datos, los cuales se obtienen del almacén de datos. Una vez que se hayan cargado los datos, se debe determinar qué cantidad se utilizará para el entrenamiento y qué cantidad se utilizará para la validación. A continuación, se debe parametrizar el algoritmo de Agrupamiento Sustractivo y finalmente poner en marcha el entrenamiento. Todos estos pasos se llevan a cabo con ayuda de la interfaz visual que se muestra en la Figura 3.2. Utilizando esta misma interfaz se puede realizar la optimización mediante la Entropía Cruzada, la cual se programó para tener como función objetivo la minimización del ECM.

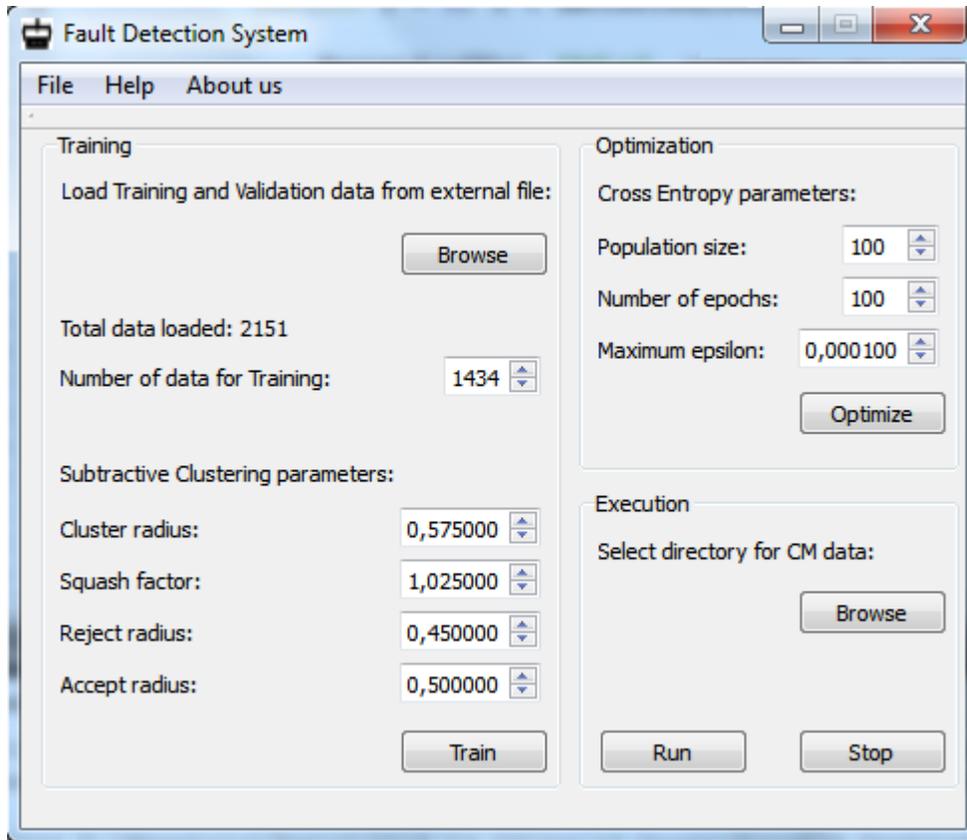


Figura 3.2 Interfaz visual para el entrenamiento, optimización y uso del Modelo Difuso TSK

En el caso del modelo basado en RNA Perceptrón Multicapa, los pasos son los mismos, cambiando solamente la parametrización de este algoritmo, la cual se realiza con ayuda de la interfaz visual de la Figura 3.3. La ejecución de cada paso que se lleva a cabo en el sistema puede ser seguida mediante el módulo de visualización como se muestra en la Figura 3.4.

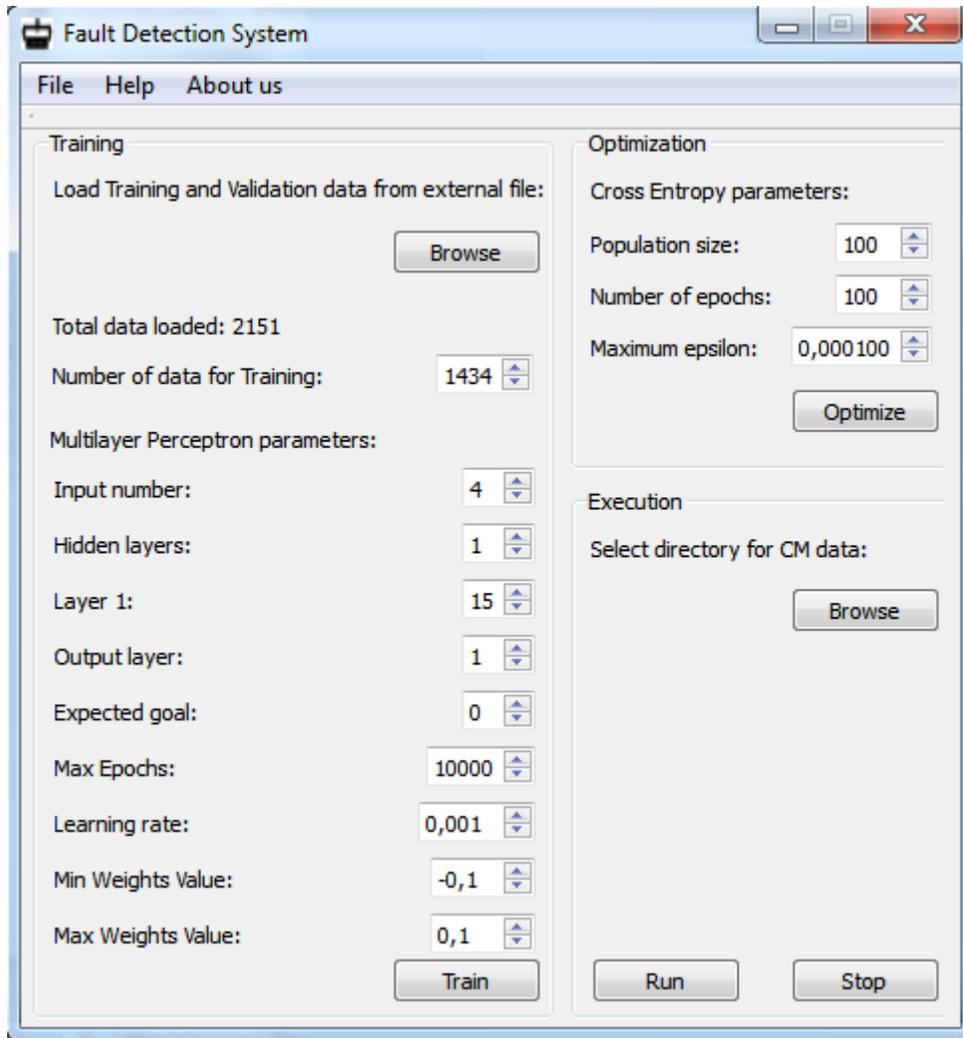


Figura 3.3 Interfaz visual para el entrenamiento, optimización y uso de la RNA Perceptrón Multicapa

Una vez que el sistema ha sido entrenado y optimizado, ya se puede comenzar a realizar el CM. El sistema puede ser configurado para que se generen alarmas cuando se prediga un valor que indique fallo; aunque es recomendable que las alarmas se produzcan cuando se predigan al menos 3 valores consecutivos que indiquen fallo. De esta forma, se reduce la posibilidad de generar una alarma ante falsos positivos. Las alarmas serán presentadas en forma visual como se muestra en la Figura 3.5.

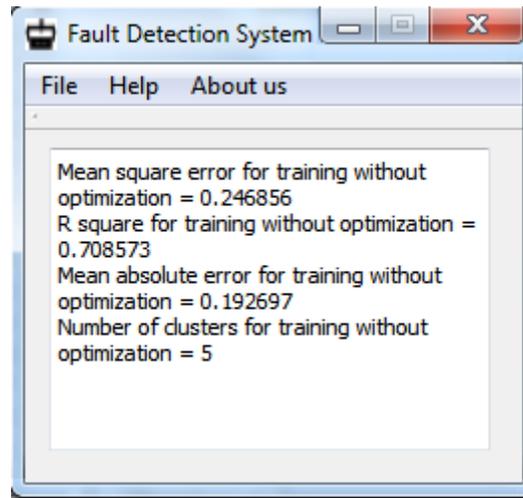


Figura 3.4 Visualización de la etapa de entrenamiento

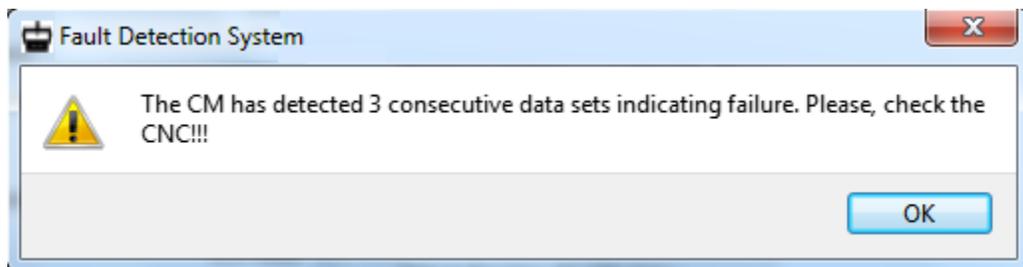


Figura 3.5 Visualización de alarmas

3.2 Validación del software

Las pruebas funcionales son pruebas basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para un software. Estas pruebas se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático. Dicho de otro modo, son pruebas específicas, concretas y exhaustivas para probar y validar que el software hace lo que debe y sobre todo, lo que se ha especificado. Existen diferentes tipos de pruebas funcionales, entre las que se

encuentran las pruebas de compatibilidad, de aceptación, pruebas de caja blanca y pruebas de caja negra.

Las pruebas de caja negra se centran principalmente en estudiar los módulos de software desde el punto de vista de las entradas que reciben y las salidas o respuestas que producen, sin tener en cuenta su funcionamiento interno. Las pruebas de caja negra son, ni más ni menos que, pruebas funcionales dedicadas a observar desde el exterior lo que se prueba. Estas pruebas pueden tener otras denominaciones como son: pruebas de entrada/salida, pruebas inducidas por datos, etc.

Por lo conveniente que resulta a partir de un conjunto de datos de entrada, determinar si la salida es correcta o no, se decidió realizar pruebas de caja negra para validar el desempeño de los algoritmos en la librería de modelos, el módulo de entrenamiento, el módulo de optimización y el acceso a datos históricos. Para determinar si se podían aplicar estas técnicas a la detección de fallas mediante el CM se decidió utilizar datos correspondientes a una señal de vibraciones en una máquina rotatoria.

Las señales de vibración utilizadas fueron seleccionadas de un estudio realizado por el Centro de Inteligencia Sistemas de mantenimiento (IMS), Universidad de Cincinnati (Lee, *et al.*, 2007). El conjunto de datos experimentales se generó a partir de ensayos de fallas en condiciones de carga constante en una plataforma de prueba con la configuración que se muestra en la Figura 3.6. El eje fue accionado por un motor de corriente alterna acoplado por correas. La velocidad de rotación se mantuvo constante a 2000 rpm. Se agregó una carga radial de 6000 libras al eje y. Las vibraciones medidas fueron las generadas por cuatro rodamientos Rexnord ZA-2115 y se captaron con acelerómetros de alta sensibilidad.

La frecuencia de muestreo fue de 20 kHz y la longitud de la señal de 20480 valores por muestra. Se realizó el análisis de la señal de vibración del tercer rodamiento, el cual, luego de 30 días de trabajo presentó una falla en la pista interna.

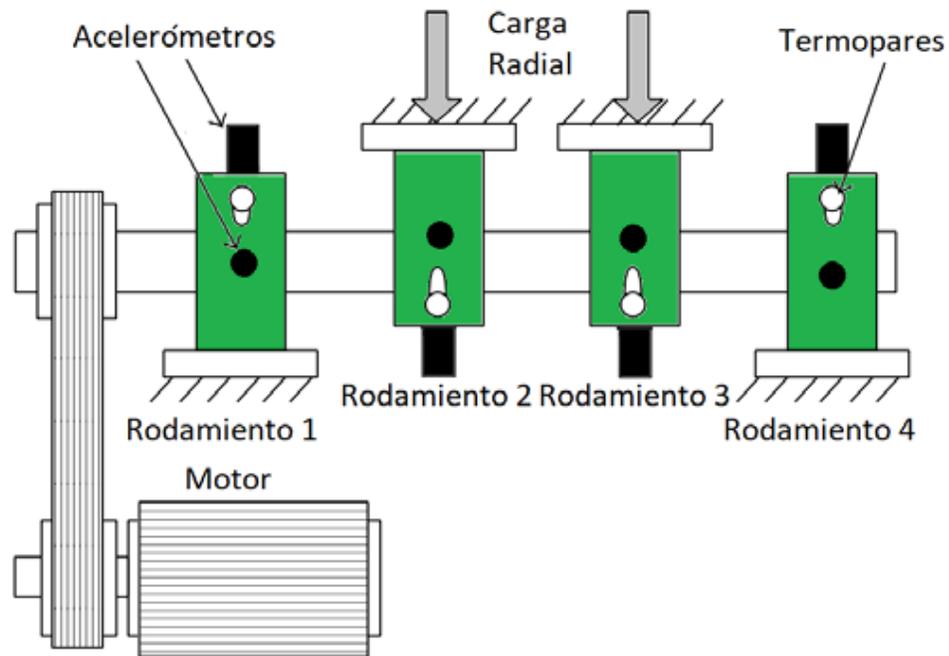


Figura 3.6 Plataforma de pruebas de rodamientos

Se tuvieron en cuenta los estadígrafos curtosis, raíz cuadrática media, desviación estándar y pico máximo; convirtiéndose estos cuatro valores en las entradas de los modelos. Dichos valores pueden ser seleccionados en el sistema al procesar la señal como ya se explicó anteriormente.

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 finalización del entrenamiento en el sistema difuso TSK, se tomó el arribo a la cantidad de épocas de entrenamiento, mientras que, en la RNA Perceptrón Multicapa,

se utilizaron como parámetros de parada el error de ajuste y las épocas de entrenamiento. Como salidas del modelo se prefijaron los valores 0 (indicando buen estado del rodamiento) y 1 (indicando fallas en el rodamiento).

Toda la información que se obtiene del CNC se va almacenando en la nube después de ser procesada, de manera tal que se pueda acceder a ella mediante el sistema para utilizarla en el entrenamiento y la optimización o mediante algún SaaS para analizarla. En la Figuras 3.7 y 3.8 se muestran los datos históricos a partir de los cuales se realizó el entrenamiento y la validación del sistema. La construcción de los gráficos en este caso se realizó con el software Past3, el cual, a pesar de ser una herramienta básica de construcción de gráficos y análisis de datos, presenta las ventajas de ser software libre y soportar varios tipos de archivo como origen de datos.

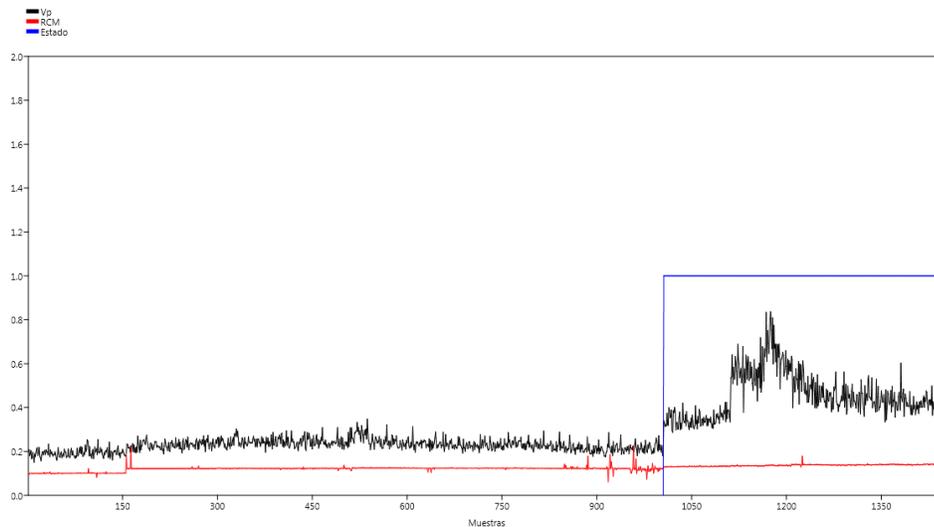


Figura 3.7 Datos de entrenamiento de los modelos de IA

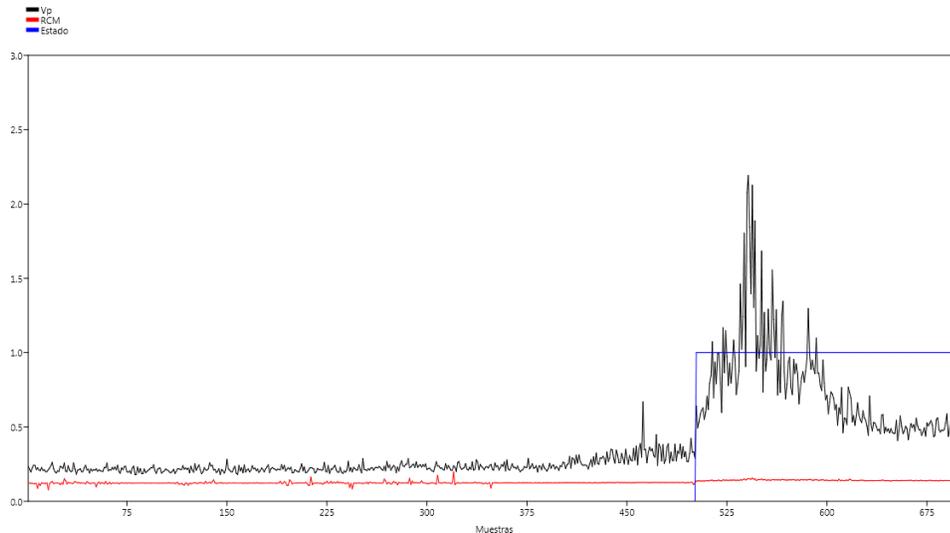


Figura 3.8 Datos de validación de los modelos de IA

El uso de este tipo de herramientas es muy útil, ya que los modelos de IA no brindan mucha información acerca del funcionamiento interno del sistema. De esta manera se tiene un enfoque complementario que permite conocer cuáles son los parámetros más influyentes para determinar el estado de un CNC.

Por otra parte, el sistema también accede a los datos históricos almacenados en la nube para el entrenamiento y la optimización, como se mencionó anteriormente. En las Figuras 3.9 y 3.10 se muestran los gráficos obtenidos para la validación de ambos modelos después del entrenamiento, mientras que en la Tabla 3.1 se muestran los índices de desempeño: verdaderos positivos (VP), falsos positivos (FP), verdaderos negativos (VN), falsos negativos (FN), tasa de acierto (TA), especificidad (E) y sensibilidad (S). Si bien ambos modelos presentaron buenos comportamientos con tasas de acierto superior al 80%, la optimización de los parámetros debería mejorar el desempeño.

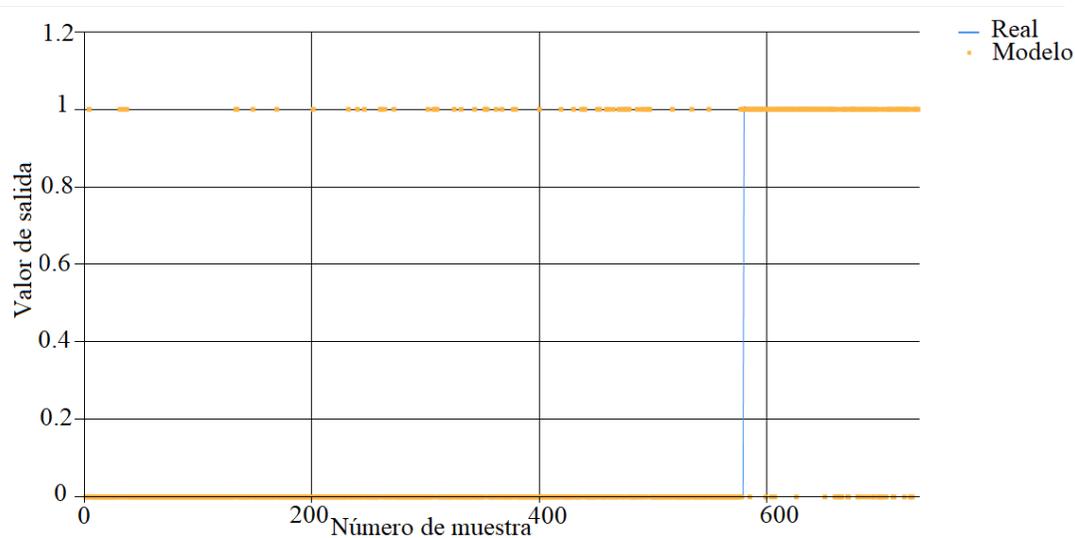


Figura 3.9 Comportamiento del sistema difuso TSK frente a la respuesta real

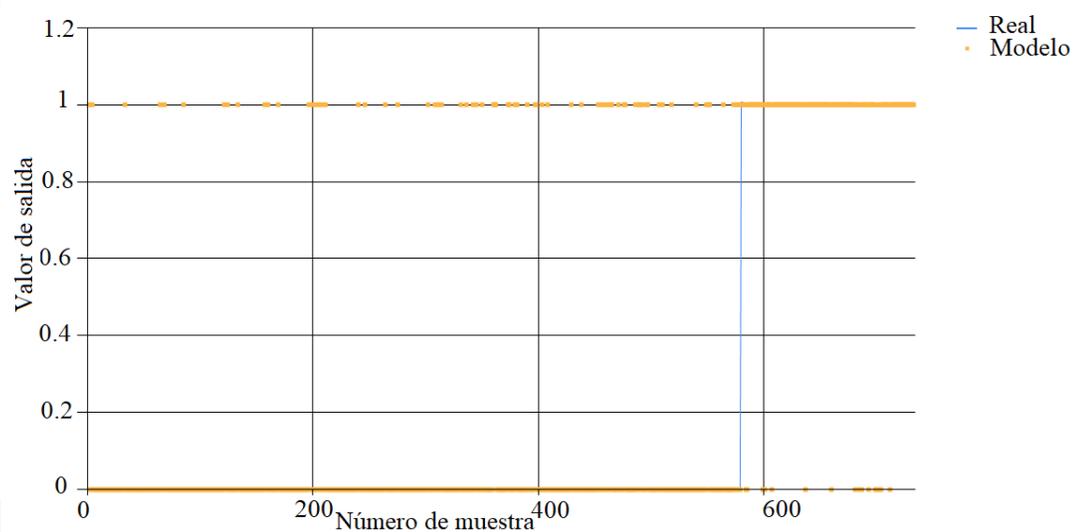


Figura 3.10 Comportamiento de la RNA Perceptrón Multicapa frente a la respuesta real

Tabla 3.1 Índices de desempeño de los modelos

Modelo	VP	FP	VN	FN	TA	E	S
Sistema difuso TSK	168	58	443	31	87.5%	0.899	0.798
RNA Perceptrón Multicapa	183	71	430	16	88.1%	0.877	0.896

Para aplicar la optimización mediante la Entropía Cruzada se asignó un valor de 100 al tamaño de la población y al número de épocas; y de 0,0001 a ϵ . En las Figuras 3.11 y

3.12 se puede observar el comportamiento de los modelos frente a la respuesta real después de la optimización. En la Tabla 3.2 se muestran los índices de desempeño de ambos modelos, donde se puede ver cómo mejora el sistema después de la optimización, especialmente en la TA, cuyo valor se incrementa casi en un 10% .

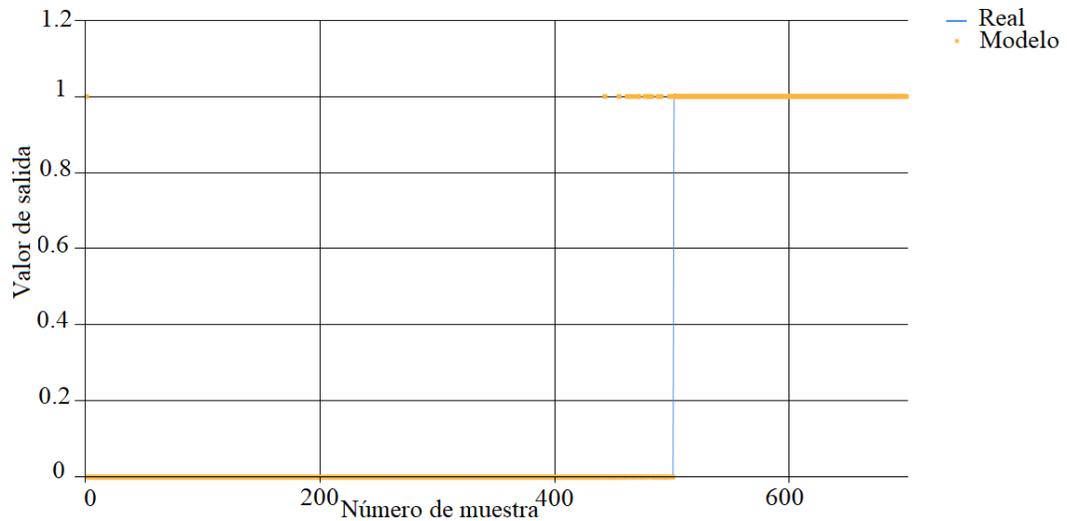


Figura 3.11 Comportamiento del sistema difuso TSK frente a la respuesta real después de la optimización

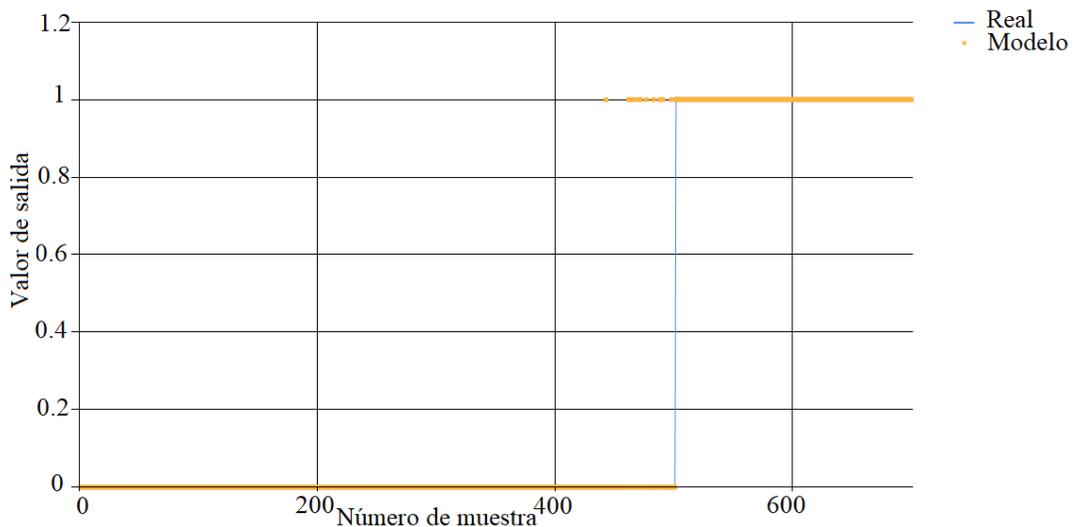


Figura 3.12 Comportamiento de la RNA Perceptrón Multicapa frente a la respuesta real después de la optimización

Tabla 3.2 Índices de desempeño de los modelos después de la optimización

Modelo	VP	FP	VN	FN	TA	E	S
Sistema difuso TSK	199	16	485	0	97.7%	0.968	1.0
RNA Perceptrón Multicapa	199	11	490	0	98.4%	0.978	1.0

3.3 Conclusiones parciales del capítulo

Una vez finalizado el presente capítulo, se ha podido arribar a las siguientes conclusiones parciales:

1. Las principales funcionalidades del sistema de monitoreo pueden llevarse a cabo mediante el software desarrollado.
2. El acceso a datos históricos almacenados en la nube se puede realizar mediante el sistema de monitoreo desarrollado o mediante algún software de análisis de datos.
3. Las pruebas realizadas para la validación del sistema demostraron la robustez y confiabilidad del mismo.

CONCLUSIONES

Como resultado final del trabajo desarrollado, se ha podido arribar a las siguientes conclusiones:

1. Es posible realizar el CM de CNC en procesos de manufactura mediante el análisis de la señal de vibraciones en el dominio del tiempo con técnicas de IA.
2. La arquitectura propuesta para el sistema de monitoreo permite realizar el CM de manera local, mientras que el almacenamiento de información se lleva a cabo en la nube.
3. Las pruebas realizadas al software desarrollado demostraron el correcto desempeño del mismo.

RECOMENDACIONES

Basadas en las conclusiones obtenidas y para la extensión futura del presente trabajo, se realizan las siguientes recomendaciones:

1. Incorporar otras técnicas de IA que prueben ser efectivas para el CM de procesos de manufactura.
2. Desarrollar módulos que incrementen las funcionalidades del sistema para contar con una herramienta de monitoreo y control integrados.
3. Desarrollar el software de monitoreo para que las alarmas puedan ser enviadas mediante Internet a todo el personal involucrado con el funcionamiento del equipo.

REFERENCIAS BIBLIOGRÁFICAS

- A. Paya, B.; Esat, I. ; Badi, M., 1997. “Artificial Neural Network Based Fault Diagnostics of Rotating Machinery Using Wavelet Transforms as a Preprocessor”. *Mechanical Systems and Signal Processing*, 11 (5), pp. 751-765, DOI: 10.1006/mssp.1997.0090.
- Ahson, S.A. ; Ilyas, M., 2011. *Cloud Computing and Software Services: Theory and Techniques*. Boca Raton: CRC Press, ISBN 978-1-4398-0316-5.
- Ajith, R.; Tewari, A. ; Gupta, D., 2017. “Low-Cost Vibration Sensor for Condition-Based Monitoring Manufactured From Polyurethane Foam”. *IEEE Sensors Letters*, 1 (6), pp. 1-4.
- Amaran, S.; Sahinidis, N.; Sharda, B. ; Bury, S., 2016. “Simulation optimization: a review of algorithms and applications”. *Annals of Operations Research*, 240 (1), pp. 351-380.
- Baldi, P. ; Brunak, S., 2001. *Bioinformatics: The Machine Learning Approach*. Massachusetts: The MIT Press.
- Bishop, C.M., 2006. *Pattern Recognition and Machine Learning*. New York: Springer, ISBN 978-0387-31073-2.
- Boer, P.T.d.; Kroese, D.P.; Mannor, S. ; Rubinstein, R.Y., 2005. “A tutorial on the cross-entropy method”. *Annals of Operations Research*, 134, pp. 19-67.
- Boyd, S. ; Vandenberghe, L., 2018. *Introduction to Applied Linear Algebra*. United Kingdom: Cambridge University Press, ISBN 978-1-316-51896-0.
- Budynas, R.G.; Nisbett, J.K. ; Shingley, J.E., 2008. *Shingley's mechanical engineering design*. 8th Ed. Singapore: McGraw-Hill.
- Castañon-Puga, M.; -Parra, J.F.; Castro, J.; Gaxiola-Pacheco, C. ; Palafox-Maestre, L.E., 2015. “Building java intelligent applications data mining for java type-2 fuzzy inference systems”. *Procedia Computer Science*, 51, pp. 2719-2728.
- Chakrabarti, S.; Cox, E.; Frank, E.; Güting, R.H.; Han, J.; Jiang, X.; Kamber, M.; Lighstone, S.S.; Nadeau, T.P.; Neapolitan, R.E.; Pyle, D.; Refaat, M.; Schneider, M.; Teorey, T.J. ; Witten, I.H., 2009. *Data Mining: Know It All*. Amsterdam: Morgan Kaufmann.
- Chebil, J.; Hrairi, M. ; Abushikhah, N., 2011. “Signal Analysis of Vibration Measurements for Condition Monitoring of Bearings”. *Australian Journal of Basic and Applied Sciences*, 5 (1), pp. 70-78.
- Chou, K.P.; Prasad, M.; Lin, Y.Y.; Joshi, S.; Lin, C.T. ; Chang, J.Y., 2014 of Conference. “Takagi-Sugeno-Kang type collaborative fuzzy rule based system”. *IEEE Symposium on Computational Intelligence and Data Mining*. Florida.

Deo, R.C. ; Biazar, M., 2018. "Multi-layer perceptron hybrid model integrated with the firefly optimizer algorithm for windspeed prediction of target site using a limited set of neighboring reference station data". *Renewable Energy*, 116, pp. 309-323.

Devendiran, S. ; Manivannan, K., 2016. "Vibration Based Condition Monitoring and Fault Diagnosis Technologies For Bearing and Gear Components - A Review". *International Journal of Applied Engineering Research*, 11 (6), pp. 3966-3975.

Dhandapani, J. ; Chandralekha, R., 2017. "Analysis of faults in three phase induction motor using neuro fuzzy logic and wavelet transformation". *International Journal of Control Theory and Applications*, 10 (34), pp. 1-14.

Díaz-Contreras, C.A.; Aguilera-Rojas, A. ; Guillén-Barrientos, N., 2014. "Lógica difusa vs. modelo de regresión múltiple para la selección de personal". *Ingeniare*, 22 (4).

Hagan, M.T.; Demuth, H.B.; Beale, M.H. ; Jesús, O.D., 2014. *Neural Network Design*. 2nd Ed., ISBN 978-0971732117.

Hemmati, F.; Orfali, W. ; Gadala, M.S., 2016. "Roller bearing acoustic signature extraction by wavelet packet transform, applications in fault detection and size estimation". *Applied acoustics*, 104, pp. 101-118.

Hill, M.D., 1990. "What is scalability?". *ACM SIGARCH Computer Architecture News*, 18 (4), pp. 18-21, DOI: 10.1145/121973.121975.

Hwang, K., 2017. *Cloud Computing for Machine Learning and Cognitive Applications*. Massachusetts: The MIT Press.

Kim, Y.-H.; Tan, A.C.C.; Mathew, J. ; Yang, B.-S., 2006. "Condition Monitoring of Low Speed Bearings: A Comparative Study of the Ultrasound Technique Versus Vibration Measurements". En: Mathew, J.; Kennedy, J.; Ma, L.; Tan, A. ; Anderson, D. (ed.)^(eds.), *Engineering Asset Management*. London: Springer London, 2006, vol. 5, pp. 182-191.

Kimball, R. ; Ross, M., 2013. *The Data Warehouse Toolkit*. 3rd Ed. Indianapolis: John Wiley & Sons, ISBN 978-1-118-53080-1.

Konar, A., 2000. *Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain*. Boca Raton: CRC Press LLC, ISBN 0-8493-1385.

Lee, J.; Qiu, H.; Yu, G. ; Lin, J., 2007. *Bearing Data Set*.

Letichevsky, A.A.; Letychevskiy, O.O.; Skobelev, V.G. ; Volkov, V.A., 2017. "Cyber-Physical Systems". *Cybernetics and Systems Analysis*, 53 (6), pp. 821-834.

Li, B.; Chow, M.-Y.; Tipsuwan, Y. ; Hung, J.C., 2000a. "Neural-Network_Based Motor Rolling Bearing Fault Diagnosis". *IEEE Transactions on Industrial Electronics*, 47 (5), pp. 1060-1069.

- Li, H.; Chen, C.L.P. ; Huang, H.-P., 2000b. *Fuzzy Neural Intelligent Systems: Mathematical Foundation and the Applications in Engineering*. 1st Ed.: CRC Press.
- Liu, T.; Lemeire, J.; Cartella, F. ; Meganck, S., 2012. “An improved segmentation-based HMM learning method for Condition-based Maintenance”. *Journal of Physics: Conference Series*, 364 (1).
- Martín, B. ; Sanz, A., 2001. *Redes Neuronales y Sistemas Difusos*. 2nd Ed.: RA-MA.
- McCulloch, W.S. ; Pitts, W.H., 1943. “A Logical Calculus of the Ideas Immanent in Nervous Activity”. *Bulletin of Mathematical Biophysics*, 5, pp. 115-133.
- Mechefske, C.K., 1998. “Objective machinery fault diagnosis using fuzzy logic”. *Mechanical Systems and Signal Processing*, 12 (6), pp. 855–862, DOI: 10.1006/mssp.1998.0173.
- Mohan, A.M. ; Rani, M.J., 2017. “Application of Soft Computing Techniques for Non-linear Process Modeling”. *Journal of Innovative Research In Technology*, 2 (10).
- Navet, N.; Bertolotti, I.C. ; Hu, T., 2017 of Conference. “Software patterns for fault injection in CPS engineering”. *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation*. pp. 1-6.
- Nielsen, M.A., 2015. *Neural Networks and Deep Learning*. Determination Press.
- Ning, Z.; Hou, W.; Hu, X. ; Xiaoxue, G., 2017 of Conference. “A cloud-supported cps approach to control decision of process manufacturing: 3D ONoC”. *2017 13th IEEE Conference on Automation Science and Engineering*. pp. 458-463.
- Patan, K., 2008. *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes*. Springer, ISBN 978-3-540-79871-2.
- Patil, M.; Mathew, J. ; Rajendrakumar, P., 2008. “Bearing Signature Analysis as a Medium for Fault Detection: A Review”. *Journal of Tribology*, 130 (1), DOI: 10.1115/1.2805445.
- Qi, Q. ; Tao, F., 2018. “Digital Twin and Big Data Towards Smart MAnufacturing and Industry 4.0: 360 Degree Comparison”. *IEEE Acces*, 6 (99), pp. 3585-3593, DOI: 10.1109/ACCES.2018.2793265.
- Qi, Q.; Zhao, D.; Liao, T.W. ; Tao, F., 2018 of Conference. “Modeling of Cyber-Physical systems and Digital Twin based on Edge Computing, Fog Computing and Cloud Computing towards smart manufacturing”. *13th International Manufacturing Science and Engineering Conference*. Texas: ASME.
- Raoa, U.M.; Soodb, Y.R. ; Jarialb, K., 2015. “Subtractive clustering expert system for engineering applications”. *Procedia Computer Science*, 48, pp. 77-83.

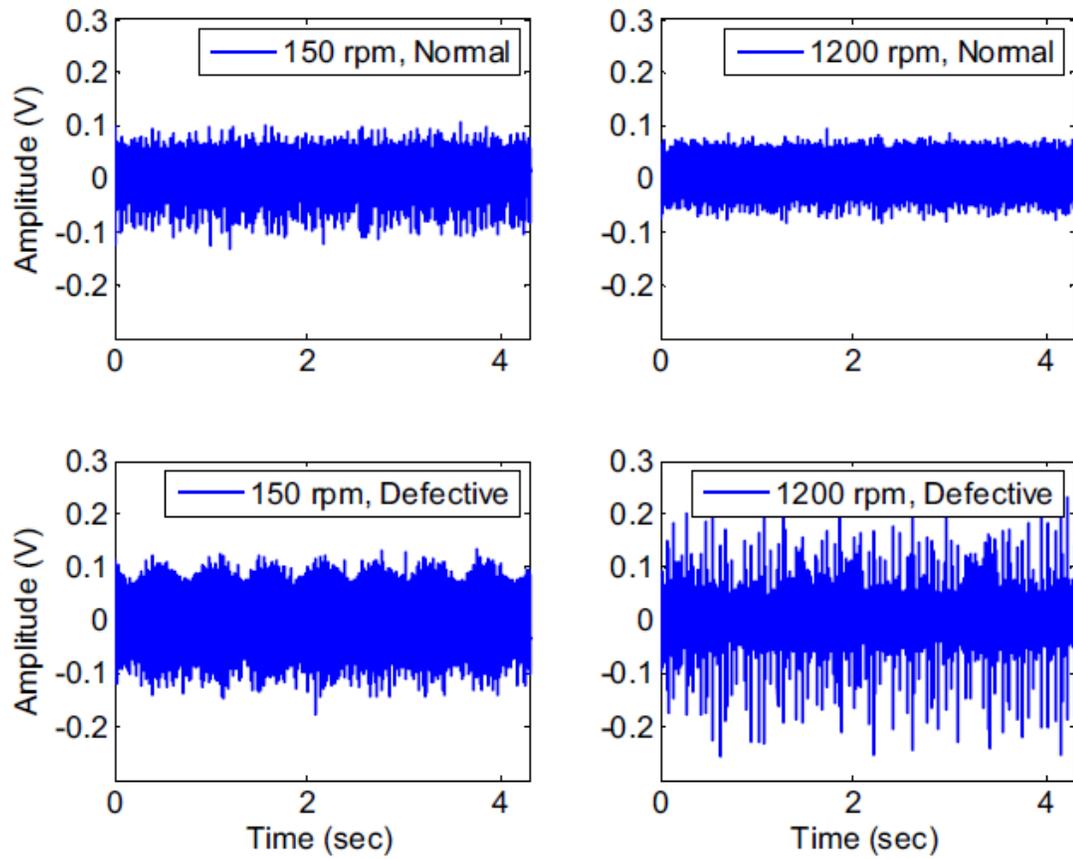
- Ren, Q.; Baron, L. ; Balazinski, M., 2006 of Conference. “Type-2 Takagi-Sugeno-Kang Fuzzy Logic Modeling using Subtractive Clustering”. En: Xplore, I. (ed.), *2006 Annual Conference of the North American Fuzzy Information Processing Society*. Montreal.
- Rubinstein, R.Y., 1997. “Optimization of computer simulation models with rare events”. *European Journal of Operations Research*, 99, pp. 89-112.
- Rubinstein, R.Y. ; Kroese, D.P., 2004. *The Cross-Entropy Method: A unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. New York: Springer-Verlag.
- Schnitman, L.; Souza, J.A.M.F.d. ; Yoneyama, T., 2001 of Conference. “Takagi-Sugeno-Kang Fuzzy Structures in Dynamic Systems Modeling”. *International Conference on Control and Application*. Canada, pp. 160-165.
- Sethy, P.; Panda, L. ; Behera, S., 2016 of Conference. “Ann based image restoration in approach of multilayer perceptron”. *2016 International Conference on Inventive Computation Technologies*. pp. 1-4.
- Srewil, Y. ; Scherer, R.J., 2017. “Construction objects recognition in framework of CPS”. En: (ed.)^(eds.), *2017 Winter Simulation Conference*. vol., pp. 2472-2483.
- Subrahmanyam, M. ; Sujatha, C., 1997. “Using neural networks for the diagnosis of localized defects in ball bearings”. *Tribology International*, 30 (10), pp. 739-752.
- Sugeno, M. ; Kang, G.T., 1988. “Structure identification of fuzzy model”. *Fuzzy Sets and Systems*, 28 (1), pp. 15-33.
- Takagi, T. ; Sugeno, M., 1985. “Fuzzy Identification of Systems and its Applications to Modeling and Control”. *IEEE Transactions on Systems, Man, and Cybernetics*, 15 (1), pp. 116-132.
- Takahashi, K.; Ogata, Y. ; Nonaka, Y., 2017 of Conference. “A proposal of unified reference model for smart manufacturing”. *2017 13th IEEE Conference on Automation Science and Engineering*. Singapore, pp. 964-969.
- Ünveren, A. ; Acan, A., 2007 of Conference. “Multi-Objective Optimization with Cross Entropy Method: Stochastic Learning with Clustered Pareto Fronts”. *2007 IEEE Congress on Evolutionary Computation*. IEEE Xplore, pp. 3065-3071.
- Verma, N.K.; Sevakula, R.K. ; Thirukovalluru, R., 2017. “Pattern Analysis Framework With Graphical Indices for Condition-Based Monitoring”. *IEEE Transactions on Reliability*, 66 (4), pp. 1085-1100.
- Villalonga, A.; Beruvides, G.; Castaño, F. ; Haber, R., 2018a. “Industrial Cyber-Physical System for Condition-based Monitoring in Manufacturing Processes”. En: (ed.)^(eds.), *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*. 15-18 May 2018, vol., pp. 637-642.

Villalonga, A.; Quiza, R. ; Cruz, Y.J., 2018b of Conference. “Módulo de monitoreo basado en estados para sistema inteligente de supervisión de procesos mecánicos”. *19 Convención Científica de Ingeniería y Arquitectura*.

Wang, W., 2008. “Condition-based Maintenance Modelling”. En: Kobbacy, K.A.H. ; Murthy, D.N.P. (eds.), *Complex System Maintenance Handbook*. London: Springer London, ISBN 978-1-84800-011-7, pp. 111-131.

Widodo, I.D., 2017 of Conference. “Fuzzy subtractive clustering based prediction model for brand association analysis”. En: Ma'mun, S.; Tamura, H. ; Purnomo, M.R.A. (eds.), *2nd International Conference on Engineering and Technology for Sustainable Development*. Indonesia: EDP Sciences, vol. 154.

ANEXOS



Anexo 1. Señal de vibración para rodamientos normales y defectuosos a 150 rpm y 1200 rpm