



Análisis de señales de alta frecuencia para aplicaciones industriales

Reporte de proyecto final de Deep Learning

8 de diciembre de 2023

Autores: Viridiana Itzel Méndez Vásquez, Geraldine González Fernández

Maestría en Cómputo Estadístico

Centro de Investigación en Matemáticas, Unidad Monterrey

Resumen

Con el inicio y adopción del esquema de Industria 4.0, se motivó el análisis de datos de alta frecuencia generados por la sensorización de las líneas de producción y dispositivos relacionados, con aplicaciones principalmente en aseguramiento de calidad y mantenimiento predictivo. El objetivo de éste proyecto es abordar éstas aplicaciones desde una perspectiva de deep learning para una tarea de predicción. Hay dos objetivos principales: explorar y obtener bases de datos relacionadas con ésta aplicación y explorar representaciones adecuadas de los datos, tanto en tiempo como en frecuencia, para aplicar métodos de aprendizaje en una tarea específica, que en este caso será predicción de la vida útil restante (RUL) de rodamientos haciendo uso de un algoritmo de deep learning basado en Transformers, basándose en el artículo [9].

1. Introducción

En la era actual de la transformación digital, la Industria 4.0 emerge como un paradigma revolucionario que redefine la manera en que percibimos y operamos sistemas industriales. En este contexto, el procesamiento de datos de sensores se presenta como un pilar fundamental para el impulso de la eficiencia, la automatización y la toma de decisiones inteligente en entornos industriales avanzados. La capacidad de capturar y analizar datos provenientes de una red extensa de sensores en tiempo real proporciona una nueva oportunidad de conocimiento y control de los sistemas industriales, con esta información se busca construir un camino hacia una producción más inteligente y adaptativa.

Este proyecto se embarca en la exploración y aplicación de técnicas de procesamiento de datos de sensores con el objetivo de potenciar la adopción en la Industria 4.0 en diversos sectores industriales. A través de la implementación de algoritmos de deep learning, se busca optimizar la eficiencia operativa y evitar tiempos de inactividad no planificados de maquinaria a través del indicador de vida útil remanente (*RUL*, por sus siglas en inglés).

Este proyecto aborda los desafíos específicos relacionados con la gestión, procesamiento y análisis de grandes volúmenes de información generada por sensores conectados al equipos de interés.

A medida que el uso de la tecnología incrementa, y se vislumbra un futuro donde la conectividad y la interoperabilidad son imprescindibles, este proyecto busca explorar, desarrollar y aplicar soluciones de procesamiento de datos de sensores con el fin de catalizar la adopción exitosa de la Industria 4.0, brindando beneficios tangibles en términos de eficiencia operativa, calidad de producción y toma de decisiones estratégicas en el panorama industrial actual.

1.1. Industria 4.0

El término de *Industria 4.0* fue introducido por el gobierno alemán en 2010, esto como un plan de alta tecnología para sus industrias. Posteriormente, en 2011 Estados Unidos introdujo los términos “*Smart Manufacturing*” y “*Smart Manufacturing Leadership Coalition*”. Otro país que también se subió a la ola de automatizar y mejorar la operación de la industria a través del uso de sensores y algoritmos computacionales fue China que en 2022 formuló su estrategia de manufactura a la que llamó “*China 2025*”.

El uso de sensores ha cambiado la visión que se tenía sobre el mantenimiento, ya que ahora se visualiza como algo preventivo y no reactivo. Se ha cambiado la visión de verlo como una necesidad cos-

tosa y no deseada, a ver el mantenimiento como una ventaja competitiva, ya que los objetivos del mantenimiento preventivo son tener:

- Alta disponibilidad del equipo.
- Bajos costos de mantenimiento.

1.2. Datos de alta frecuencia

La clasificación de una señal como “alta frecuencia” o “baja frecuencia” depende del contexto específico de los datos y del tipo de aplicación. Como tal no existe una frecuencia de corte única que determine de manera universal si una señal es de alta o baja frecuencia, aunque sí se encuentra en la literatura algunas generalizaciones para ciertos campos de aplicación. Para la realización de este trabajo consideramos que las frecuencias de muestreo en kilohercios (kHz) y superiores serán consideradas como “altas frecuencias”.

1.3. RUL: Vida Útil Remanente

El RUL (Remaining Useful Life), o Vida Útil Remanente en español, es una métrica utilizada para estimar el tiempo o la operación restante antes de que un sistema o componente alcance el final de su vida útil operativa. En el contexto del mantenimiento predictivo y la gestión de activos, el RUL ayuda a prever cuándo es probable que ocurra una falla o deterioro significativo, permitiendo la planificación de actividades de mantenimiento de manera más eficiente. La estimación del RUL se basa en datos históricos, análisis de condición actual y algoritmos de aprendizaje para proporcionar una aproximación del tiempo restante de operación adecuada.

Para estimar el RUL se requiere de mediciones de varios parámetros del equipo, la meta de la predicción del RUL basado en datos es establecer una relación entre el RUL y características del equipo en cuestión. Gráficamente, el problema de estimación del RUL lo observamos en la Figura 1.

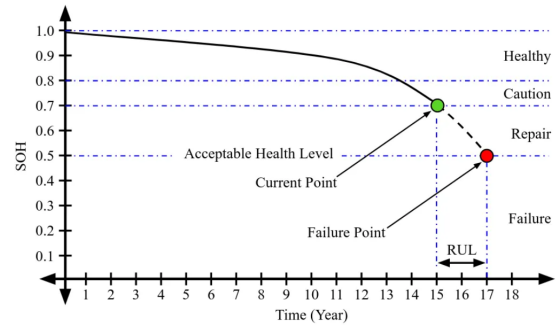


Figura 1: Problema de RUL

El problema de la predicción del RUL formalmente se puede definir como sigue. La entrada es $X_t \in \mathbb{R}^k$, $t = (1, 2, \dots, T)$ donde T es la longitud del paso de tiempo y k es el número de sensores. La salida corresponde a la predicción del RUL Y_t para cada paso de tiempo.

2. Conjuntos de datos

Un desafío importante en los pronósticos basados en datos es la dificultad de obtener una cantidad suficiente de muestras de la progresión del fracaso. Los elevados costos operativos y de inversión, así como la larga vida útil de muchos sistemas técnicos, dificultan la adquisición de conjuntos de datos completos hasta el fallo.

Uno de los objetivos del proyecto es encontrar bases de datos relacionadas con la predicción del tiempo de falla. En esta sección se mencionan algunas de las fuentes de datos más utilizadas en el área de manejo de señales con datos de sensores, sin embargo, en el Apéndice A.1, se hace referencia a un repositorio importante de datos de esta índole.

La NASA cuenta con un repositorio de datos de pronóstico que han sido donados por universidades, agencias o empresas. El repositorio de datos se centra exclusivamente en conjuntos de datos de pronóstico, dicho repositorio se encuentra en Repositorio NASA. . Además, la sociedad de Pronósticos y Gestión de la Salud (PHM society) realiza retos anuales por equipos relacionados con pronósticos por lo que en cada reto liberan el conjunto de datos.

A continuación se enlistan algunas bases de datos.

1. IMS Bearing Data

Los datos fueron generados por el Center for Intelligent Maintenance Systems.

Cuatro rodamientos fueron instalados en un eje. La velocidad de rotación se mantuvo constante a 2000 RPM por un motor AC acoplado al eje mediante correas de fricción. Se aplica una carga radial de 6000 lbs sobre el eje y el rodamiento mediante un mecanismo de resorte. Todos los rodamientos están lubricados a presión.

Todas las fallas ocurrieron después de exceder la vida útil diseñada del rodamiento, que es de más de 100 millones de revoluciones.

Estructura de los datos:

Se incluyen tres conjuntos de datos. Cada conjunto de datos describe un experimento de prueba hasta el fallo, cada uno consiste de archivos individuales que son instantáneas de señales de vibración de 1 segundo registradas en intervalos específicos. Cada archivo consiste de 20,480 puntos con una tasa de muestreo de 20kHz. El nombre del archivo indica cuándo se colectaron los datos. Cada registro en el archivo de datos es un punto de datos. Los intervalos más grandes de marcas de tiempo (que se muestran en los nombres de los archivos) indican la reanudación del experimento en el siguiente día hábil.

■ Conjunto 1:

- Duración de la grabación: 22 de Octubre de 2003, 12:06:24 a 25 de Noviembre de 2003, 23:39:56
- No. de archivos: 2,156
- No. de canales: 8
- Orden de los canales: Bearing 1- Ch1&2; Bearing 2- Ch 3&4; Bearing 3-Ch 5&6; Bearing 4-Ch 7 & 8 .
- Intervalo de grabación del archivo: Cada 10 minutos (excepto los primeros 43 archivos los cuales se tomaron cada 5 minutos)
- Formato del archivo: ASCII
- Descripción: Al final del experimento de prueba hasta el fallo, se produjo un defecto en la pista interna en el rodamiento 3 y un defecto en el elemento de rodillo en el rodamiento 4.

■ Conjunto 2:

- Duración de la grabación: 12 de Febrero de 2004, 10:32:39 a 19 de febrero de 2004, 06:22:39
 - No. de archivos: 984
 - No. de canales: 4
 - orden de los canales: Bearing 1- Ch1; Bearing 2- Ch 2; Bearing 3-Ch 3; Bearing 4-Ch 4 .
 - Intervalo de grabación del archivo: Cada 10 minutos
 - Formato del archivo: ASCII
 - Descripción: Al final del experimento de prueba hasta el fallo, se produjo un defecto en la pista externa en el rodamiento 1.
- Conjunto 3:
- Duración de la grabación: 04 de marzo de 2004, 09:27:46 a 4 de abril de 2004, 19:01:57
 - No. de archivos: 4,448
 - No. de canales: 4
 - orden de los canales: Bearing 1- Ch1; Bearing 2- Ch 2; Bearing 3-Ch 3; Bearing 4-Ch 4 .
 - Intervalo de grabación del archivo: Cada 10 minutos
 - Formato del archivo: ASCII
 - Descripción: Al final del experimento de prueba hasta el fallo, se produjo un defecto en la pista externa en el rodamiento 3.

2. C-MAPSS Data

C-MAPSS (Comercial Modular Aero-Propulsion System Simulation) es una herramienta para simular un motor comercial tipo turbofan. El software se codificó en un ambiente de MATLAB y Simulink.

Las fallas con las que se cuentan son:

- Límite de la temperatura de salida la turbina de alta presión (HPT)
- Un límite regulador que previene la presión estática del compresor de alta presión (HPC)
- Degradación del ventilador

El conjunto de datos consiste de múltiples series de tiempo multivariadas. Cada conjunto

de datos está dividido en conjunto de entrenamiento y prueba. Cada serie de tiempo corresponde a un motor diferente, los datos se pueden considerar como provenientes de una flota de motores del mismo tipo.

Cada motor empieza con diferentes grados de desgaste inicial y variación de fabricación la cual es desconocida para el usuario. Este desgaste y variación se consideran normal, es decir, no se consideran como una condición de falla. Hay tres configuraciones operativas que tienen un efecto sustancial en el rendimiento del motor. Dichas características están incluidas en los datos, además los datos están contaminados con sensores de ruido.

El motor opera normalmente al inicio de cada serie de tiempo y desarrollan una falla en algún punto durante la serie. En el conjunto de entrenamiento, la falla crece en magnitud hasta que el sistema falla. En el conjunto de prueba, la serie de tiempo termina algún tiempo antes de que falle el sistema. El objetivo es predecir el número de ciclos operativos restantes antes del fallo en el conjunto de prueba, es decir, el número de ciclos operativos después del último ciclo que el motor seguirá funcionando. También se proporcionó un vector de valores reales de vida útil restante (RUL) para los datos de prueba.

Los datos son proporcionados en un archivo comprimido zip de archivos de texto con 26 columnas de números, separados por espacios. Cada fila es una instantánea de datos tomados durante un único ciclo operativo, cada columna es una variable diferente. Las columnas corresponden a

1. Número de unidad
2. Tiempo, en ciclos
3. Configuración operacional 1
4. Configuración operacional 2
5. Configuración operacional 3
6. Medición del sensor 1
7. Medición del sensor 2
- ⋮
26. Medición del sensor 26

Estructura del conjunto de datos:

- FD001
 - Trayectorias de entrenamiento: 100
 - Trayectorias de prueba: 100
 - Condiciones: Uno (nivel del mar)
 - Modos de falla: Uno (Degradación HPC)
- FD002
 - Trayectorias de entrenamiento: 260
 - Trayectorias de prueba: 259
 - Condiciones: Seis
 - Modos de falla: Uno (Degradación HPC)
- FD003
 - Trayectorias de entrenamiento: 100
 - Trayectorias de prueba: 100
 - Condiciones: Uno (nivel del mar)
 - Modos de falla: Dos (Degradación HPC, degradación ventilador)
- FD004
 - Trayectorias de entrenamiento: 248
 - Trayectorias de prueba: 249
 - Condiciones: Seis
 - Modos de falla: Dos (Degradación HPC, degradación ventilador)

3. FEMTO Bearing (PRONOSTIA)

La IEEE Reliability Society y el FEMTO-ST Institute organizaron el IEEE PHM Data Challenge en 2012, donde este reto se enfocó en estimar el RUL de bearings de sistemas mecánicos. Los datos de este challenge fueron proporcionados por FEMTO-ST Institute, estos datos se generaron por ensayos que se llevaron a cabo en una plataforma experimental de laboratorio llamada *PRONOSTIA*, en la Figura 2 se muestra una imagen de dicha plataforma. Esta plataforma permite la degradación acelerada de los rodamientos bajo condiciones constantes o variables de funcionamiento, un punto importante de PRONOSTIA es que durante la experimentación monitorea y recopila datos de la salud del componente. Las condiciones que se pueden variar durante la experimentación son:

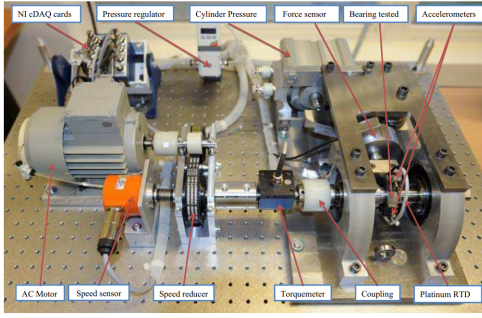


Figura 2: Plataforma PRONOSTIA

- Velocidad de rotación
- Fuerza de carga
- Temperatura de vibración

Para los datos de este challenge se consideraron 3 diferentes cargas de trabajo, modificando los primeros dos parámetros de los antes mencionados.

El conjunto de datos de FEMTO Bearing (PRONOSTIA) contiene tres casos de experimentación que consistieron de las siguientes condiciones de operación respectivamente:

- a) Caso 1: 1800 rpm¹ y 4000 N²
- b) Caso 2: 1650 rpm y 4200 N
- c) Caso 3: 1500 rpm y 5000 N

En total se proporcionaron 6 experimentos ejecutados hasta falla para entrenamiento y 11 para pronóstico, la cantidad de experimentos para cada escenario se puede ver en la Tabla 1. Además, cabe mencionar que no se dio ningún supuesto sobre el tipo de falla.

Condiciones operacionales			
Dataset	Caso 1	Caso 2	Caso 3
Learning set	Bearing1.1	Bearing2.1	Bearing3.1
	Bearing1.2	Bearing2.2	Bearing3.2
Test set	Bearing1.3	Bearing2.3	Bearing3.3
	Bearing1.4	Bearing2.4	
	Bearing1.5	Bearing2.5	
	Bearing1.6	Bearing2.6	
	Bearing1.7	Bearing2.7	

Tabla 1: Rodamientos por condición de operación

¹rpm = revoluciones por minuto es una unidad de frecuencia

²N = Newton es una unidad de fuerza

³g = Fuerza G, es una medida de aceleración, esta métrica se basa en la aceleración que produciría la gravedad de la Tierra sobre un objeto cualquiera.

La caracterización de la degradación de los rodamientos se basa en dos tipos de datos de sensores, uno de vibración/aceleración y otro de temperatura, cabe resaltar que las señales de vibración se miden de forma horizontal y vertical. La frecuencia de muestro de cada uno de estos sensores se muestra en 2.

Sensor	Frecuencia	Registros
Aceleración	25.6 kHz	2560 registro cada 10 seg (i.e. 1/10 s)
Temperatura	10 Hz	600 registros cada minuto

Tabla 2: Frecuencia de muestro de sensores PRONOSTIA

La plataforma PRONOSTIA permite realizar experimentos de ejecución hasta el fallo. Para evitar la propagación de daños en el sistema de pruebas y por razones de seguridad, los experimentos se detuvieron cuando La amplitud de la señal de vibración superó los 20 g³, este comportamiento se puede observar en la Figura 3.

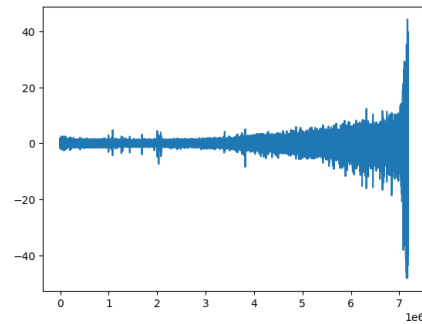


Figura 3: Datos sin procesar: Bearing1.1

3. Representaciones de los datos

Es bien sabido que en la aplicación de métodos de machine learning o deep learning, la representación de los datos es muy importante para obtener buenos resultados. En el caso de análisis de señales de alta frecuencia se han explorado diferentes repre-

sentaciones, a continuación se mencionan algunas de estas representaciones encontradas en la literatura.

- En [6] preprocesan los datos usando la **transformada wavelet continua (CTW)**, en particular utilizan la wavelet de Morlet basándose en el criterio de energía máxima. Se calculan 5 estadísticos que son:
 - Error cuadrático medio
 - Valor pico
 - Factor de cresta
 - Kurtosis
 - Sesgo rectificado

Estas métricas se calculan para la señal antes y después del preprocesamiento, con el objetivo de determinar la coherencia en las tendencias observadas. Las estadísticas de valor pico la usan posteriormente como una variable para predicción, y esta misma acompañada del error cuadrático medio se usaron para algoritmos de aprendizaje.

- En el artículo [7] se propone representar los datos a partir de una imagen donde se muestren las características de tiempo-frecuencia, con esto la señal de vibración se transforman en una imagen de espectro multi-escala que mostrará visualmente los componentes de frecuencia que han cambiado en el tiempo. Para transformar la serie de tiempo unidimensional a 2-D, se utiliza la transformada wavelet continua (CWT) para extraer las características de tiempo-frecuencia, y así aprovechar la ventaja visual de descomponer la señal y localizar fallos. Una vez generada la imagen, esta se modela con una Red neuronal convolucional en 2-D en la cual entra directamente, con esto se construye un indicador de salud (HI) con el cual posteriormente se usa para el pronóstico del RUL.
- En [1] utilizan análisis de componentes principales kernel (KPCA, por sus siglas en inglés) para la extracción de características no lineales y reducción de dimensionalidad buscando evitar problemas de sobreajuste causado por demasiados parámetros del modelo, posteriormente aplican una red GRU para la estima-

ción del RUL. KPCA busca retener información importante y remover la redundante. KPCA transforma los datos originales de baja dimensión a un espacio de alta dimensión mediante la función de mapeo, para en ese nuevo espacio lograr la separabilidad lineal, al mismo tiempo, a través de la función kernel KPCA transforma la operación de producto punto de vectores del espacio de alta dimensión en la operación entre los vectores originales lo que ayuda a reducir la complejidad del proceso de cálculo. Al aplicar KPCA se realiza sobre los datos normalizados y se quedan con las componentes asociadas a los primeros M dimensiones, y estas se usan como datos de entrenamiento.

- En [2] usan la transformada ortogonal de Hilbert-Huang (OHHT) la cual es una mejora a la de la transformada de Hilbert-Huang (HHT) que es computacionalmente más eficiente. La transformada OHHT usa el método de Gram-Schmidt para generar IMF's funciones de modo intrínseco ortogonales. El Espectro de Hilbert otorga una representación 2-D energía-frecuencia que se conoce como espectro marginal de Hilbert (MHS). El espectro de Hilbert y el espectro marginal de Hilbert son análogos al espectrograma, la representación de densidad espectral de potencia, con la ventaja de devolver resultados instantáneos de frecuencia y energía. Las características extraídas con OHHT se introdujeron en la red neuronal. La arquitectura de la red neuronal utilizada fue la siguiente: Capa de entrada con 12 nodos, dos capas ocultas con 118 y 122 nodos respectivamente, capa de salida con siete nodos. Se usó función de activación ReLu para todas las capas, excepto para la capa de salida en la que utilizaron softmax. Usaron Adam como optimizador y cross-entropy categórica como función de pérdida. Accuracy como métrica adicional. El modelo se entrenó con un tamaño de batch de 16 y 150 épocas.
- En el artículo [9] normalizan los datos considerando el mínimo y el máximo, posteriormente se utiliza una ventana deslizante para el preprocesamiento de segmentación de datos para que el modelo obtenga la mayor cantidad de información valiosa posible de series de tiempo

multivariadas. En este caso utilizan un modelo de Auto atención de doble aspecto basada en Transformer (DAST) el cual es una estructura encoder-decoder puramente basado en la autoatención sin ningún módulo RNN/CNN. La arquitectura de DAST utilizada en esta referencia se muestra en la Tabla 3.

4. Arquitectura del modelo

La principal arquitectura de red neuronal que se utilizó en este trabajo se describirá en esta sección, no obstante, en el Apéndice A.2 se mencionan algunas otras arquitecturas ampliamente utilizadas en el campo de procesamiento de señales de sensores.

En [9] proponen un método de predicción de RUL con un método de Auto-atención de Aspecto Dual basado en Transformer (DAST). Esta arquitectura profunda basada en auto-atención captura los pesos de las características para el sensor y el paso de tiempo.

DAST sigue la estructura encoder-decoder de un Transformer original y consiste de tres subestructuras principales en el framework: capa de encoder (incluyendo capa de encoder del sensor y capa de encoder del paso de tiempo), capa de fusión de características y capa de decoder. A diferencia del método de predicción de RUL basado en arquitecturas RNN y CNN, DAST captura la información de dependencia a largo plazo entre entradas y salidas de la secuencia a través de mecanismos de auto-atención sin considerar la distancia.

El flujo de trabajo del modelo DAST es el siguiente:

- Extracción de características de los datos CBM recopilados por múltiples sensores. Para esto diseñaron un mecanismo de encoding de aspecto dual que aplica el encoder de sensor y el encoder de paso del tiempo que trabajan en paralelo para capturar los pesos de características de diferentes sensores y paso de tiempo. Ambos encoders están diseñados basados en mecanismo de auto-atención multi-cabezal.
- Las características extraídas de los dos aspectos son integradas por la capa de fusión de características para obtener un nuevo mapa de

características con información importante de diferentes sensores y pasos de tiempo.

- El mapa de características fusionadas es enviada al decoder, el cual adopta mecanismos de auto-atención para captar la atención de la información del ciclo de trabajo actual y los diferentes pasos de tiempo anteriores y la información de los sensores y genera el RUL previsto a través de una red feed-forward (FFN) densamente conectada.

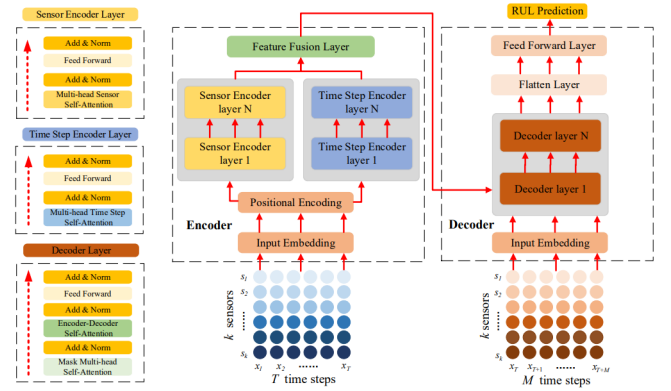


Figura 4: Arquitectura de DAST. Imagen tomada de [9].

4.1. Encoder de DAST

El encoder está principalmente conformado por una **capa de entrada de embedding**, una **capa de encoding posicional**, **múltiples capas de encoder de sensor** y **capas de encoder del paso del tiempo**. La capa de embedding asigna los datos de monitoreo del estado de entrada a un vector de dimensión D_{model} a través de una FFN(feed-forward network), con el fin de prepararse para el siguiente proceso de extracción de características.

1. Capa de encoding posicional. DAST no contiene estructuras basadas en RNN o CNN. En consecuencia, es necesario colocar tokens de posiciones relativas dentro de la secuencia, así el modelo puede hacer uso de la información de la posición de la secuencia. En este caso como método de encoding posicional usan las funciones seno y coseno de diferentes frecuencias, como sigue

$$P_t(2k) = \sin(t/10000^{2k/D_{model}}) \quad (1)$$

$$P_t(2k+1) = \cos(t/10000^{2k/D_{model}}) \quad (2)$$

donde t es el paso de tiempo y k es la dimensión del sensor. P_t tiene una relación lineal con P_{t+l} donde l cualquier paso de tiempo fijo. Esto hace que el modelo aprenda más fácilmente la atención según las posiciones relativas.

2. Capa de encoder de sensor

La capa de encoder de sensor incluye principalmente dos sub-capas: capa de auto-atención de sensor de múltiples cabezales y una capa FFN.

En la Figura 4 se puede ver que hay una conexión residual y una capa de normalización (Add y Norm, respectivamente) después de cada sub-capa. El propósito de la conexión residual es aliviar la dificultad de entrenamiento de la red neuronal profunda. La capa de normalización puede acelerar el proceso de entrenamiento y hacer que el modelo converja más rápido al normalizar el valor de activación de la capa.

La capa de encoder del sensor usa el mecanismo de auto-atención con múltiples cabezales para extraer la importancia de los diferentes sensores a lo largo de la dimensión del sensor, por lo tanto, puede aprender automáticamente a enfocarse en aquellas características del sensor con pesos más altos sin intervención de la experiencia humana en el proceso de entrenamiento.

El proceso de trabajo de la auto-atención de sensor con múltiples cabezales se describe a continuación.

Se definen los datos recolectados por los k sensores en una ventana de tiempo de longitud T como $X = \{X_1, X_2, \dots, X_T\} \in \mathbb{R}^{d_k \times T}$. Se define además X'_s como los datos obtenidos después de ser procesados por la capa de encoding posicional. El proceso de trabajo de la auto-atención se muestra en la Figura 5. Primero se generan tres matrices (Queries, Keys, Values) procesando los datos de entrada usando los siguientes cálculos

$$Q_s = X'_s W_s^q, \quad K_s = X'_s W_s^k, \quad V_s = X'_s W_s^v, \quad (3)$$

donde W_s^q, W_s^k, W_s^v son parámetros entrenables, $Q_s, K_s, V_s \in \mathbb{R}^{d_k \times D_{model}}$, D_{model} es la

dimensión de la entrada. Entonces se calcula el producto punto de Q y K (escalado por $\sqrt{D_{model}}$, y se aplica la función softmax a lo largo de la dimensión del sensor para obtener los pesos de diferentes sensores en X'_s . Además, el vector de pesos de diferentes sensores en el paso de tiempo t es:

$$\alpha_t = \text{softmax}_{\text{sensors}} \left(\frac{Q_s K_s^T}{\sqrt{D_{model}}} \right) \quad (4)$$

donde $\alpha_t = (\alpha_{t,1}, \alpha_{t,2}, \dots, \alpha_{t,k})$, $t = (1, 2, \dots, T)$. Finalmente, las características de los diferentes sensores son pesadas por el mecanismo de atención son calculadas como una suma ponderada de V_s

$$\text{Atención}_{\text{sensors}}(Q_s, K_s, V_s) = \alpha_t V_s \quad (5)$$

Adoptan un mecanismo de auto-atención multi-cabezal para permitir que el modelo atienda información de diferentes subespacios de representación en diferentes posiciones, tal que mejore el rendimiento de la predicción. En la Figura 5 se muestra la auto-atención multi-cabezal, el cual se puede expresar como

$$\text{MultiHead}(Q_s, K_s, V_s) = \text{Concat}(\{head_i\}_{i=1}^h) W^s, \quad (6)$$

donde las matrices de parámetros $W^s \in \mathbb{R}^{h D_{model} \times D_{model}}$, h es el número de cabezales y $head_i = \text{Atención}(Q_s, K_s, V_s)_i$.

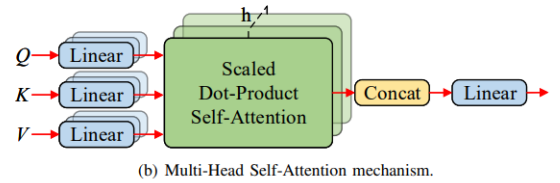
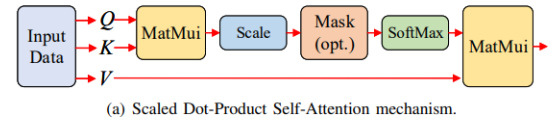


Figura 5: Proceso de auto-atención multi-cabezal. Imagen tomada de [9]

3. Capa de encoder del paso de tiempo

En la Figura 4 se puede ver que la capa de encoder del paso de tiempo tiene la misma estructura que la capa de encoder de sensor. Incluye principalmente dos sub-capas: capa de auto-atención multi-cabezal del paso de

tiempo y una capa de una red neuronal feed-forward. La diferencia es que la capa de encoder de paso de tiempo extrae características a lo largo de la dimensión del paso de tiempo, y permite que el modelo DAST tome atención en el paso de tiempo que es más importante en la predicción de RUL.

Los datos de entrada del encoder del paso de tiempo es la transpuesta de X y es procesada por la capa de encoding posicional como X'_t . La capa de encoder del paso de tiempo primero obtiene las matrices Queries, Keys y Values

$$Q_t = X'_t W_t^q, \quad K_t = X'_t W_t^k, \quad V_t = X'_t W_t^v, \quad (7)$$

con $Q_t, K_t, V_t \in \mathbb{R}^{T \times D_{model}}$ donde W_t^q, W_t^k, W_t^v son parámetros entrenables. Entonces, el vector de pesos de los diferentes pasos de tiempo correspondientes al sensor s se pueden obtener aplicando softmax a lo largo de la dimensión de paso de tiempo

$$\beta_s = \text{softmax}_{time_steps} \left(\frac{Q_t K_t^T}{\sqrt{D_{model}}} \right) \quad (8)$$

donde $\beta_s = (\beta_{s,1}, \beta_{s,2}, \dots, \beta_{s,T})$, $s = (1, 2, \dots, k)$. Las características de los diferentes pasos de tiempo son ponderadas por el mecanismo de atención que se obtiene como

$$\text{Atención}_{\text{sensores}}(Q_t, K_t, V_t) = \beta_s V_t. \quad (9)$$

Se aplica auto-atención multi-cabezal a la capa de encoder de paso de tiempo

$$\text{MultiHead}(Q_t, K_t, V_t) = \text{Concat}(\{head_j\}_{j=1}^h) W^t, \quad (10)$$

donde las matrices de parámetros $W^t \in \mathbb{R}^{h D_{model} \times D_{model}}$, h es el número de cabezales y $head_j = \text{Atención}(Q_t, K_t, V_t)_j$.

4. Capa de fusión de características

Después de extraer características de las dimensiones de sensor y de paso de tiempo de los datos CBM, DAST realiza fusión de características para integrar información de dos aspectos.

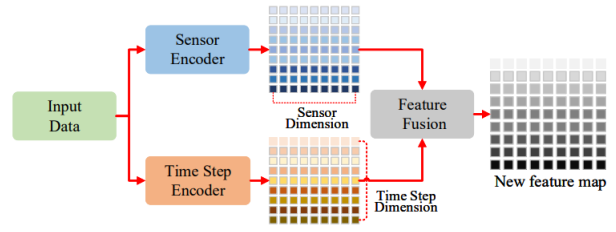


Figura 6: Fusión de características

En la Figura 6, la capa de fusión de características combina las características del paso de tiempo y el sensor en un nuevo mapa de características. Denotamos la extracción de características del encoder de sensor y del encoder del paso de tiempo como $F_s \in \mathbb{R}^{d_k \times D_{model}}$ y $F_t \in \mathbb{R}^{T \times D_{model}}$, respectivamente. DAST realiza la fusión de características realizando el siguiente cálculo

$$F_r = \text{Concat}(F_s, F_t) W^f, \quad (11)$$

donde las matrices de parámetros entrenables $W^f \in \mathbb{R}^{(d_k+T) \times D_{model}}$, lo que puede hacer que el modelo capture información de características tanto de F_s como de F_t .

4.2. Decoder de DAST

El decoder de DAST fue diseñado de manera similar al del transformer vainilla. El decoder consiste de lo siguiente:

- Capa de entrada de embeddings
- Múltiples capas idénticas de decoder
- Una Flatten layer
- Capa FFN

Las capas del decoder incluyen dos subcapas de autoatención multi-cabezal, cada subcapa trabaja con diferentes fragmentos de la información que descompuso el **Encoder**, estas subcapas son de la siguiente manera:

- Una máscara multi-cabezal auto-atención, que trabaja con los *Queries*.
- Una subcapa encoder-decoder de autoatención de múltiples cabezales, esta recibe los *Keys* y *Values*

Asimismo, los pesos de los sensores y los puntos en el tiempo generados por el encoder, son analizados por el decoder. El decoder realiza la atención de la información del ciclo actual, y de los diferentes ciclos previos de los sensores y los pasos de tiempo, con toda esta información arroja la predicción del *RUL* a través de la FFN layer.

Además, para garantizar que la predicción de un punto dependerá únicamente de los puntos previos en los datos, la mascarará multicabezal que es aplicada para la autoatención, establece los productos escalares correspondientes a $-\infty$. En otras palabras, con esto se busca que el mecanismo de auto-atención solo preste atención a los datos X_{T_1} y previos cuando predice el RUL de X_T .

5. Metodología

5.1. Descripción del problema

Este trabajo se enfocó en la predicción de la vida útil remanente (RUL, por sus siglas en inglés) de rodamientos, el desgaste de los rodamientos se puede observar en las componentes como se muestra en la figura 7.



Figura 7: Bearing desgastado

Los “bearings” o rodamientos son componentes esenciales en muchos sistemas mecánicos utilizados en múltiples industrias como: automotriz, aeroespacial, de construcción, marítima, entre otras. El desgaste de los rodamientos puede verse reflejado de diferentes formas, entre las que encontramos:

- Incremento de fricción
- Aumento de temperatura
- Juego y holgura

- Incremento en el ruido
- Vibraciones no deseadas

Todos estos síntomas repercuten en la reducción de la vida útil de los rodamientos. Un rodamiento desgastado tiene una mayor probabilidad de fallar prematuramente, lo que puede resultar en tiempos de inactividad no planificados y costosas reparaciones, si el desgaste no se atiende a tiempo, una falla total de los rodamientos puede ocasionar daños catastróficos en el sistema de maquinaria.

Para prevenir estos problemas es importante realizar un mantenimiento regular de inspeccionar y lubricar adecuadamente los rodamientos, y reemplazarlos cuando sea necesario. Esto ayuda a mantener el rendimiento óptimo del sistema, prolongar la vida útil de los componentes, y aumentar la disponibilidad de los equipos.

5.2. Datos de prueba y validación

Para la realización de este trabajo se usaron únicamente los datos de PRONOSTIA descritos en la Sección 3, conviene subrayar que no se utilizaron la totalidad de los datos, únicamente se utilizaron los asociados al Caso 1 de condiciones operativas (3a), asimismo, solamente se usaron las medidas de vibración ya que estas son las únicas que tienen una frecuencia de muestreo de alta frecuencia.

5.3. Preprocesamiento de los datos

A las mediciones correspondientes a cada bearing se le realizó el siguiente pre-procesamiento, vale la pena comentar que al tener mediciones de alta frecuencia la ejecución de este procesamiento llevó un tiempo considerable. Para los datos de entrenamiento del Bearing1_1 se requirió de un tiempo aproximado de 3 horas.

5.3.1. Normalización

Los datos provenientes de los diferentes sensores tienen diferentes unidades y escalas, lo cual puede afectar el accuracy de la predicción del RUL. Se usa el método min-max para normalizar los datos del sensor. De manera específica, para los datos CBM $X_i = \{X_1, X_2, \dots, X_T\}$, se normaliza como sigue

$$\tilde{X}_i = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)},$$

donde \tilde{X}_i son los datos normalizados, $\max(X_i)$ y $\min(X_i)$ denota el máximo y el mínimo de X_i .

5.3.2. Procesamiento de ventana deslizante

Una ventana deslizante a menudo es utilizada para el preprocesamiento de segmentación de datos para que el modelo obtenga información valiosa de series de tiempo multivariadas tanto como sea posible. T_w es el tamaño de la ventana de tiempo y el paso de deslizamiento se establece en uno. El RUL del último punto de datos en la ventana de tiempo funciona como el RUL de esa ventana.

5.4. Características estadísticas

El valor promedio y los coeficientes de regresión estimados entre el paso de tiempo y la secuencia de datos de la secuencia son dos características explícitas que son usualmente utilizadas en datos de secuencias de tiempo, las cuales proveen de secuencias estadísticas útiles.

5.4.1. Creación de etiquetas en conjuntos de datos

En [10] se detalla que en la forma tradicional de asignar valores objetivo para la estimación de RUL, este disminuye linealmente con el tiempo. Esta definición implica que la salud de un sistema se degrada linealmente con el tiempo. En aplicaciones prácticas, la degradación de un componente es insignificante al comienzo de su uso y aumenta cuando el componente se acerca al final de su vida útil. Para modelar mejor los cambios de la vida útil restante junto con el tiempo, en [3], se propuso una función objetivo de RUL lineal por partes, como en la Figura 8, que limita el RUL máximo a un valor constante y luego inicia la degradación lineal después de un cierto grado de uso. Cuando no se tiene un valor de RUL máximo se sigue una idea similar, en [8] utilizan la idea donde la etiqueta de datos del ciclo de vida del rodamiento generalmente se divide en dos etapas, la etiqueta RUL de la etapa de funcionamiento normal es 1 y la etiqueta RUL desde el FPT (Primer tiempo de predicción) hasta la falla completa del rodamiento se reduce gradualmente de 1 a 0. Según técnica la basada en el criterio 3σ mostrada en [5]. En [8] proporcionan los estados de funcionamiento normal y estado de degradación incluyendo el valor del FPT

para los rodamientos de las condición 1, en la Tabla 4 se resumen dichos valores.

Bearings	Estado normal (s)	FPT	Estado de degradación
Bearing1_1	1-11420	11420	11420-28030
Bearing1_2	1-8220	8220	8220-8710
Bearing1_3	1-9600	9600	9600-23750
Bearing1_4	1-10180	10180	10180-14280
Bearing1_5	1-24070	24070	24070-24630
Bearing1_6	1-6270	16270	16270-24480
Bearing1_7	1-22040	22040	22040-22590

Tabla 4: Estados de funcionamiento de los rodamientos del Caso 1. Valores tomados de [8].

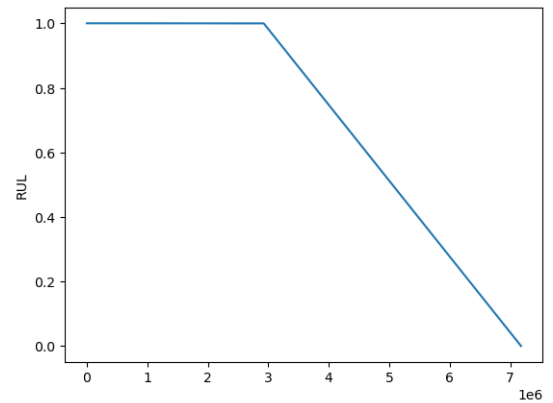


Figura 8: RUL

6. Resultados y discusión

Conviene subrayar que antes de la implementación de los modelos utilizados en este trabajo con los datos de PRONOSTIA, se realizaron múltiples ejercicios con un conjunto de datos menos complejo replicando el artículo base [9], estos procesos previos se describen en el Apéndice B.

6.1. Recursos de Implementación

La implementación se realizó en dos ambientes distintos con diferente capacidad de recursos descritos en la Tabla 5, el pre-procesamiento de los datos se ejecutó en el ambiente de Google Colab y el entrenamiento de la arquitectura en los núcleos de los servidores de CIMAT. Dada la arquitectura del modelo utilizada, el entrenamiento se realizó en CPU por lo que este proceso se llevó a cabo en aproximadamente 20 horas.

Recurso	Colab	CIMAT
RAM	51 GB	128 GB
Disco	225.8 GB	
Procesador		Intel Xeon Silver 4314 3.4 Ghz

Tabla 5: Recursos de ambientes

Rodamiento	Diferencia	Relativa
Bearing1_3	60	1 %
Bearing1_4	69	20 %
Bearing1_5	180	11 %
Bearing1_6	510	35 %
Bearing1_7	2210	29 %

Tabla 7: Error en Pronóstico RUL

6.2. Resultados estimación de RUL

Como se muestra en la Tabla 1, existe información de 5 rodamientos en el conjunto de datos de prueba de PRONOSTIA para las condiciones de operación del caso 1. Para estos 5 rodamientos se estimó el tiempo de vida remanente, los resultados se muestran en la Tabla 6. La columna de tiempo actual es la cantidad de historia observada que tenemos para cada rodamiento, cabe aclarar que las cifras se reportan en segundos, en las siguientes dos columnas tenemos el valor observado y el pronosticado de RUL respectivamente.

Rodamiento	Tiempo Actual	RUL Real	RUL Pronosticado
Bearing1_3	18,010	5,730	5,670
Bearing1_4	11,380	339	270
Bearing1_5	23,010	1,610	1,430
Bearing1_6	23,010	1,460	950
Bearing1_7	15,010	7,570	5,360

Tabla 6: Pronostico RUL

Analizando las diferencias entre los valores reales y pronosticados de RUL (Tabla 7), se puede observar que el rodamiento con el mayor error es el *Bearing1_6* y el que tiene el pronóstico más preciso es el *Bearing1_3*. Otro aspecto que resalta en los resultados, es que todas las estimaciones obtenidas son menores que el valor real del RUL, lo cual es benéfico ya que estas estimaciones podrían ayudar a que los mantenimientos preventivos se realicen antes de la falla, cumpliendo con el objetivo de anticiparse a la ocurrencia de deficiencias en la operación que podrían llegar a ser catastróficas.

Gráficamente la estimación del RUL a lo largo del tiempo de operación se observa que tiene un comportamiento inicialmente creciente alcanzando su máximo alrededor del tiempo promedio de vida de la componente y luego decrece. Este comportamiento se atribuye a que los componentes suelen fallar al inicio de su operación por defectos de fabrica, y una vez sobrepasando su tiempo de vida promedio, por lo que en estos intervalos el tiempo de vida remanente es menor. Este comportamiento para el *Bearing1_3* se observa en la imagen 9, que fue tomada de [7], en este artículo almacenaron la estimación de RUL para cada momento del ciclo de vida por lo que se puede apreciar este comportamiento graficamente.

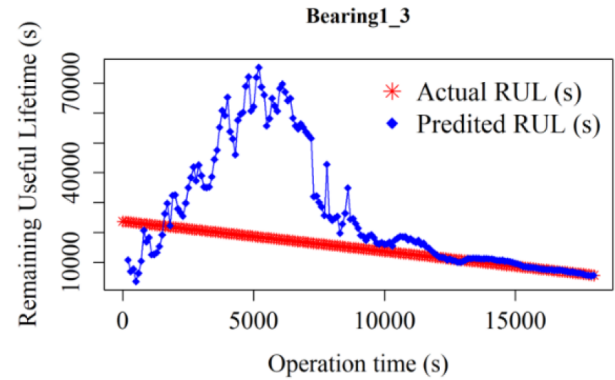


Figura 9: Predicción RUL Bearing1_3

7. Conclusiones

La utilización de datos de alta frecuencia permite mejorar las estimaciones del RUL, ya que se pueden capturar patrones de comportamiento temporal más finos, lo que se traduce a la identificación de cambios sutiles en el rendimiento de los equipos que ayudan a una predicción más precisa del tiempo restante de vida.

En este trabajo, un aspecto de gran importancia fue tener en cuenta las consideraciones asociadas a

la disponibilidad de recursos computacionales para la implementación de modelos con datos de sensores en alta frecuencia, esto debido a que la volumetría de los conjuntos de datos es muy alta y fácilmente se agotan los servidores en este entorno de datos.

El uso de Deep Learning en la Industria 4.0 ha brindado beneficios sustanciales, pero la gestión eficiente de los recursos computacionales sigue siendo un aspecto crítico, es por esto que la hay un área de investigación creciente que se centra en encontrar soluciones que equilibren la complejidad de los modelos con la capacidad de los recursos disponibles.

A. Recursos Adicionales

A.1. Repositorio de Datos

El artículo [3] contiene un repositorio de conjuntos de datos de señales, con su respectivo URL para descargar. Si se desea trabajar con datos de esta naturaleza esta referencia es valiosa, para la obtención de datos. En la imagen 10, se muestran un fragmento del repositorio encontrado.

A.2. Repositorio Algoritmos

En el artículo [4] se enlistan varios algoritmos muy utilizados en distintas aplicaciones de DL en datos de sensores. Las arquitecturas CNN son el algoritmo que se menciona más frecuentemente, aunque también encontramos las arquitecturas RNN, LSTM y Virtual A-E. En la imagen ??, se muestra esta tabla con las características de que tipo de red y en que contexto se han utilizado esos algoritmos.

B. Análisis C-MAPSS

Se llevó a cabo un ensayo de la construcción de redes neuronales profundas utilizando un conjunto de datos preliminar antes de la implementación oficial. Recordemos que la realización de este trabajo se basó en los datos de PRONOSTIA [3], pero estos contemplan dificultades por su alta volumetría, por lo que previamente a la utilización de esta base se realizaron ejercicios con la base de C-MAPSS descrita en [2].

En este proceso de entendimiento, se exploraron las siguientes arquitecturas de redes neuronales profundas con el objetivo de evaluar y comprender

mejor el comportamiento de estos modelos en un conjunto de datos más pequeño:

- DAST
- LSTM
- Bi-LSTM

Las implementaciones con los datos de C-MAPSS se realizaron con los datos asociados a la primer condición de funcionamiento, que son datos a nivel del mar que consideran únicamente la falla por degradación. Este conjunto considera 100 trayectorias para entrenamiento y 100 trayectorias para prueba.

Se realizaron ajustes sistemáticos en los hiperparámetros, tales como la tasa de aprendizaje, el tamaño del lote y la complejidad de la red, con el fin de analizar su impacto en el rendimiento del modelo, en este proceso no se consiguió encontrar una combinación de hiperparámetros que arrojara buenos resultados con la arquitectura bi-LSTM, por lo que esta se omitió en la implementación con los datos de PRONOSTIA. Durante este proceso de experimentación, se recopilaban métricas de desempeño clave, como el tiempo de ejecución, la cantidad de épocas necesarias, y se visualizaron los resultados de los pronósticos del RUL.

Este enfoque preliminar permitió adquirir conocimientos valiosos sobre diferentes arquitecturas y parámetros de los modelos propuestos. Los resultados obtenidos durante este experimento sirvieron como base para la toma de decisiones, para la posterior implementación en un conjunto de datos más complejo.

C. Contribuciones

C.1. VIMV

Revisión de artículos, Implementación DAST con datos C-MAPSS y PRONOSTIA, Realización de sección de representación de los datos, ejecuciones en servidor de Colab y CIMAT. Generación de sección de preprocesamiento: normalización, ventana deslizante, generación de etiqueta. Generación de GitHub para entrega. Generación de sección de Encoder DAST.

C.2. GGF

Revisión de artículos, Implementación LSTM con datos C-MAPSS, revisión de arquitectura BiLSTM, Realización de sección, redacción de sección de conclusiones. Redacción sección de datos de PRONOSTIA. Generación de sección de Decoder DAST.

Referencias

- [1] J. CHEN, H. JING, Y. CHANG, AND Q. LIU, *Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process*, Reliability Engineering & System Safety, 185 (2019), pp. 372–382.
- [2] T. FURLONG AND K. REICHARD, *Machining tool and bearing failure analysis using the orthogonal hilbert-huang transform on vibration and motor current datasets*, in Annual Conference of the PHM Society, vol. 14, 2022.
- [3] S. HAGMEYER, F. MAUTHE, AND P. ZEILER, *Creation of publicly available data sets for prognostics and diagnostics addressing data scenarios relevant to industrial applications*, International Journal of Prognostics and Health Management, 12 (2021).
- [4] A. JAMWAL, R. AGRAWAL, AND M. SHARMA, *Deep learning for manufacturing sustainability: Models, applications in industry 4.0 and implications*, International Journal of Information Management Data Insights, 2 (2022), p. 100107.
- [5] N. LI, Y. LEI, J. LIN, AND S. X. DING, *An improved exponential model for predicting remaining useful life of rolling element bearings*, IEEE Transactions on Industrial Electronics, 62 (2015), pp. 7762–7773.
- [6] A. MALHI, R. YAN, AND R. X. GAO, *Prognosis of defect propagation based on recurrent neural networks*, IEEE Transactions on Instrumentation and Measurement, 60 (2011), pp. 703–711.
- [7] Y. YOO AND J.-G. BAEK, *A novel image feature for the remaining useful lifetime prediction of bearings based on continuous wavelet transform and convolutional neural network*, Applied Sciences, 8 (2018), p. 1102.
- [8] Y. ZHANG, K. FENG, J. C. JI, K. YU, Z. REN, AND Z. LIU, *Dynamic model-assisted bearing remaining useful life prediction using the cross-domain transformer network*, IEEE/ASME Transactions on Mechatronics, 28 (2023), pp. 1070–1080.
- [9] Z. ZHANG, W. SONG, AND Q. LI, *Dual-aspect self-attention based on transformer for remaining useful life prediction*, IEEE Transactions on Instrumentation and Measurement, 71 (2022), pp. 1–1.
- [10] S. ZHENG, K. RISTOVSKI, A. FARAHAT, AND C. GUPTA, *Long short-term memory network for remaining useful life estimation*, in 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), 2017, pp. 88–95.

Componentes	Capas	Parámetros
Embedding de entrada	Capa densamente conectada	Unidades ocultas: 64 Activación: Lineal
Encoder	Encoder del sensor	Bloques de encoder N=2 Cabezales de autoatención H=4
	Encoder del paso del tiempo	Bloques de encoder N=2 Cabezales de autoatención H=4
Decoder	Decoder	Bloques de decoder N=1 Cabezales de autoatención H=4
Salida	Capa densamente conectada	Unidades ocultas: 64 Activación: ReLU
	Capa de salida	Unidades ocultas: 1 Activación: Lineal

Tabla 3: Arquitectura modelo DAST

No.	Data Set Designation and Reference to data origin	URL for Data Download
1	Data Challenge PHM Soc. 2008 - Turbofan NASA Prognostics Center of Excellence	https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#phm08_challenge
2	Data Challenge PHM Soc. 2009 - Gearbox Fault Detection PHM Society, Gearbox fault detection data set, 2010	https://c3.nasa.gov/dashlink/resources/997/
3	Data Challenge PHM Soc. 2010 - CNC milling machine cutters X. Li ¹ , B.S. Lim ¹ , J.H. Zhou ¹ , S. Huang ¹ , S.J. Phua ¹ , K.C. Shaw ¹ , and M.J. Er ² ¹ Singapore Institute of Manufacturing Technology, 71 Nanyang Drive, Singapore 638075 ² School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798	https://www.phmsociety.org/competition/phm/10
4	Data Challenge PHM Soc. 2011 - Anemometer Fault Detection Creators Unknown	https://www.phmsociety.org/competition/phm/11/problem
5	Data Challenge PHM Soc. 2013 Creators Unknown	https://www.phmsociety.org/events/conference/phm/13/challenge
6	Data Challenge PHM Soc. 2014 Creators Unknown	https://www.phmsociety.org/events/conference/phm/14/data-challenge
7	Data Challenge PHM Soc. 2015 - Plant Fault Detection Creators Unknown	https://www.phmsociety.org/events/conference/phm/15/data-challenge
8	Data Challenge PHM Soc. 2016 - Semiconductor CMP Crystec Technology Trading GmbH	https://www.phmsociety.org/events/conference/phm/16/data-challenge
9	Data Challenge PHM Soc. 2017 - Bogie Vehicle Creators Unknown	https://www.phmsociety.org/events/conference/phm/17/data-challenge
10	Data Challenge PHM Soc. 2018 - Ion Mill in Wafer Manufacturing A dataset shared by Kai Goebel (NASA)	https://www.phmsociety.org/events/conference/phm/18/data-challenge

Figura 10: Fragmento de repositorio de datos de señales

S.No.	Model Name	Training model/Equation	Application in manufacturing	Sustainability Impact	Reference
1.	D-CNN	$v^j = \beta_1 v_{j-1} + (1 - \beta_1) g_t^2$ $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$ $\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\eta_t + \epsilon}} \hat{m}_t$	Cost estimation in manufacturing	Economic (✓) Environment (✓) Social (✓)	(Imoto et al., 2018; Ning et al., 2020)
2.	R-CNN	$L_{hsc}(u^s, v) = \sum_{\tilde{u} \in \{x, y, w, h\}} smooth \ h_{L_1}(u^s - V_j)$ $L_{mask} = \frac{1}{m^2} \sum_{1 \leq i, j \leq m} [y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k)]$	Spectroscopic analysis in manufacturing	Economic (✓) Environment (✓) Social (✓)	(Bhuvaneswari et al., 2021)
3.	CNN	$S_j(X) = \frac{e^{f_j}}{\sum_{k=1}^K e^{f_k}}$ $L(y_i, \hat{y}_i) = -\frac{1}{N} \sum_i y_i \log(\hat{y}_i)$	Crack detection in manufacturing	Economic (✓) Environment (✓) Social (✓)	(Alipour et al., 2019)
4.	CNN	$L_{indl} = \sum_{k=1}^{n_{ex}^{indl}} \frac{1}{N_k} V_{\omega}((g(P_k), S^s, S^k)_1)$ $P_k \in \left\{ \left\{ x_{i,j}^{an} \right\}_{j=1}^{n_k} \right\}_{i=1}^{n_{ex}}$ $S^s = \left\{ g(P_{k,m}^s) \right\}_{m=1}^{N_{indl}}, S^k = \left\{ g(P_{k,m}^k) \right\}_{m=1}^{N_{indl}}$	Additive manufacturing quality assessment	Economic (✓) Environment (✓) Social (✓)	(Li et al., 2020)
5.	RBM, DBM and AE	Bayesian belief network, tanh functions, Deionizing, contractive and sparse autoencoder stochastic gradient descent	Fault diagnosis	Economic (✓) Environment (✓) Social (✓)	(J. Wang et al., 2018)
6.	RNN and CNN	Point-wise thermal concatenation and thermal imaging	Distortion prediction	Economic (✓) Environment (✓) Social (✓)	(Alipour et al., 2019)
7.	D-CNN and Virtual-AE	$OS = 2 \times 2 \left[1 - \frac{\sum_{k=1}^Q \Phi_k - \frac{1}{2} Q^2 + Q}{\frac{1}{2} (2N - Q + 1) Q - \frac{1}{2} Q^2 + Q} \right] = \frac{NQ + Q - 2 \sum_{k=1}^Q \Phi_k}{NQ - Q^2}$	Crystal structure	Economic (✓) Environment (✓) Social (✓)	(Ryan et al., 2018)
8.	Virtual-AE	$S_i^p = \frac{1}{N} \sum_{j=1}^N \cos(\phi_i^{exc}, \phi_j^{non-exc}), N = 5$	Virtual material synthesis	Economic (✓) Environment (✓) Social (✓)	(Bhuvaneswari et al., 2021)
9.	MFNN	$MAPE = \frac{1}{N} \sum_{i=1}^N \left \frac{A_i - \hat{F}_i}{A_i} \right $	Mechanical properties	Economic (✓) Environment (✓) Social (✓)	(Sun et al., 2019)
10.	CNN, RNN and LSTM	Stochastic gradient descent, transfer learning and large-scale visual representation	Material degradation	Economic (✓) Environment (✓) Social (✓)	(Alipour et al., 2019)

Figura 11: Algoritmos para trabajar datos de señales