

UNIVERSIDAD DE COSTA RICA
SISTEMA DE ESTUDIOS DE POSGRADO

RECONOCIMIENTO DE CAVITACIÓN Y ANOMALÍAS EN UNA BOMBA
HIDRÁULICA CON TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Trabajo final de investigación aplicada sometido a la consideración de la Comisión del Programa de Estudios de Posgrado en Computación e Informática para optar al grado y título de Maestría Profesional en Computación e Informática

LUIS CARLOS SÁNCHEZ CÉSPEDES

Ciudad Universitaria Rodrigo Facio, Costa Rica

2022

Dedicatoria

Este trabajo final de investigación aplicada está dedicado a mi esposa Iriabelle Villalta, quien ha estado a mi lado desde el inicio de todo este proceso y quien ha sido mi soporte todo este tiempo. Soy un mejor profesional y un mejor ser humano gracias a ti.

Agradecimientos

En primer lugar, quiero agradecer al Posgrado de Computación e Informática y a su directora la Dra. Gabriela Marín Raventós quienes me abrieron las puertas y me brindaron la oportunidad de cursar este programa de maestría, también a la Escuela de Ingeniería Mecánica y a su director el Dr. Pietro Scaglioni Solano por gentilmente brindarme apoyo para la realización de este proyecto.

Un agradecimiento especial al comité del trabajo final de investigación aplicada conformado por el Dr. Arturo Camacho Lozano, el Mag. Juan Gabriel Monge Gapper y el Dr. José Andrés Guevara Coto, quienes invirtieron su tiempo en participar en las reuniones y las revisiones. Su guía, retroalimentación y apoyo han sido fundamentales para culminar este proyecto y cumplir con las metas propuestas.

Finalmente, un agradecimiento al Ing. Alonso Viquez, quien no dudó en mostrar interés en la idea y brindarme su criterio, indispensable durante la etapa de experimentación de este proyecto.

¡Gracias a todos!

“Este trabajo final de investigación aplicada fue aceptado por la Comisión del Programa de Estudios de Posgrado en Computación e Informática de la Universidad de Costa Rica, como requisito parcial para optar al grado y título de Maestría Profesional en Computación e Informática.”

Dra. Leonora de Lemos Medina
**Representante de la Decana
Sistema de Estudios de Posgrado**

Dr. Arturo Camacho Lozano
Profesor Guía

Mag. Juan Gabriel Monge Gapper
Lector

Dr. José Andrés Guevara Coto
Lector

Dra. Gabriela Marín Raventós
Directora
Programa de Posgrado en Computación e Informática

Luis Carlos Sánchez Céspedes
Sustentante

Índice

1. Dedicatoria	ii
2. Agradecimientos	iii
3. Hoja de aprobación	iv
4. Descripción del problema	1
5. Marco conceptual	2
5.1. Bombas hidráulicas	2
5.1.1. Cavitación	3
5.1.2. Estimación de cavitación en bombas centrífugas	5
5.2. Procesamiento digital de señales	6
5.2.1. Transformada de Fourier discreta	7
5.2.2. Frecuencia fundamental y armónicos	9
5.2.3. Convolución	11
5.3. Inteligencia artificial y aprendizaje automático	11
5.3.1. Modelos de secuencia	13
5.3.2. Reducción de la dimensionalidad y selección de características	14
6. Estado del arte	16
6.1. Inteligencia artificial y diagnóstico de máquinas hidráulicas	16
6.2. Aprendizaje automático y reconocimiento de cavitación	19
7. Justificación	21
8. Objetivo	23
8.1. Objetivo general	23
8.2. Objetivos específicos	23
9. Metodología	23
9.1. Adquisición de datos	23
9.2. Cavitación de la bomba hidráulica	26

9.3. Procesamiento de las señales y selección de las características	29
9.4. Modelos de clasificación	33
9.5. Modelos de detección de anomalías	34
9.6. Modelo con simulaciones	37
9.7. Validación y análisis de error	40
10. Resultados y discusión	42
10.1. Análisis exploratorio de datos	42
10.2. Modelos de clasificación	47
10.3. Modelos de detección de anomalías	52
10.4. Problemas adicionales simulados	56
11. Conclusiones	59
12. Apéndice	67

Índice de figuras

1. Clasificación de bombas	3
2. Diagrama de fase	4
3. Ejemplo de cavitación	5
4. Carga de succión positiva neta	6
5. Señal continua y señal discreta	7
6. Componentes sinusoidales	9
7. Armónicos	10
8. Ejemplo de la operación convolución	11
9. Diagrama de modelo muchos a uno	13
10. Diagrama de modelo muchos a muchos	14
11. Distribución de las investigaciones con aprendizaje automático.	19
12. Módulo de adquisición de datos	24
13. Banco de pruebas	25
14. Curva de la bomba	26
15. Inducir a la cavitación	27

16.	Distintas curvas de operación	28
17.	Puntos operativos del experimento	28
18.	Prueba F ANOVA	30
19.	Metodología para la preparación de datos en el dominio de la frecuencia	31
20.	Metodología para la preparación de datos en el dominio del tiempo . . .	32
21.	Arquitectura del modelo de clasificación usando convolución 1D	35
22.	Arquitectura del modelo de clasificación usando convolución 1D y LSTM anidados	36
23.	Arquitectura del modelo de detección de anomalías con reducción usando convolución 1D	38
24.	Arquitectura del modelo de detección de anomalías con reducción de la dimensionalidad usando LSTM	39
25.	Arquitectura del modelo clasificador para múltiples categorías	39
26.	Ajuste con curva de aprendizaje	41
27.	Comparación en el dominio del tiempo	43
28.	Comparación en el dominio del tiempo	43
29.	Comparación en el dominio de la frecuencia	44
30.	Elementos esperados en frecuencia	44
31.	Factores de ruido	46
32.	Espectro de cavitación	46
33.	Ruido aleatorio en altas frecuencias	47
34.	Reducción de la dimensionalidad por medio de PCA	48
35.	Retención de la información	49
36.	Reducción de la dimensionalidad y clasificación por medio de LDA . . .	49
37.	Relación entre el desempeño del modelo LDA y la cantidad de características de entrada	50
38.	Entrenamiento del modelo de clasificación con convolución 1D	51
39.	Entrenamiento del modelo de clasificación con convolución 1D y LSTM	51
40.	Entrenamiento del modelo de detección de anomalías con convolución 1D	53
41.	Distribución del error esperado con el modelo con convolución 1D . . .	53
42.	Detector de anomalías con convolución 1D detectando cavitación	54

43.	Entrenamiento del modelo de detección de anomalías con LSTM	55
44.	Distribución del error esperado con el modelo con LSTM	55
45.	Detector de anomalías con LSTM detectando cavitación	56
46.	Problemas adicionales simulados	57
47.	Comparación con reducción de la dimensionalidad	58
48.	Matriz de confusión en clasificación múltiple	58

Índice de cuadros

1.	Distribución por tipo de máquina o elemento	17
2.	Distribución por tipo de falla	18
3.	Distribución por tipo de aprendizaje	20
4.	Recolección de datos	29
5.	Resultados con modelos de clasificación	52



UNIVERSIDAD DE
COSTA RICA

SEP Sistema de
Estudios de Posgrado

Autorización para digitalización y comunicación pública de Trabajos Finales de Graduación del Sistema de Estudios de Posgrado en el Repositorio Institucional de la Universidad de Costa Rica.

Yo, Luis Carlos Sánchez Céspedes, con cédula de identidad 1-1368-0338, en mi condición de autor del TFG titulado Reconocimiento de cavitación y anomalías en una bomba hidráulica con técnicas de aprendizaje automático.

Autorizo a la Universidad de Costa Rica para digitalizar y hacer divulgación pública de forma gratuita de dicho TFG a través del Repositorio Institucional u otro medio electrónico, para ser puesto a disposición del público según lo que establezca el Sistema de Estudios de Posgrado. SI NO

*En caso de la negativa favor indicar el tiempo de restricción: _____ año (s).

Este Trabajo Final de Graduación será publicado en formato PDF, o en el formato que en el momento se establezca, de tal forma que el acceso al mismo sea libre, con el fin de permitir la consulta e impresión, pero no su modificación.

Manifiesto que mi Trabajo Final de Graduación fue debidamente subido al sistema digital Kerwá y su contenido corresponde al documento original que sirvió para la obtención de mi título, y que su información no infringe ni violenta ningún derecho a terceros. El TFG además cuenta con el visto bueno de mi Director (a) de Tesis o Tutor (a) y cumplió con lo establecido en la revisión del Formato por parte del Sistema de Estudios de Posgrado.

FIRMA ESTUDIANTE

Nota: El presente documento constituye una declaración jurada, cuyos alcances aseguran a la Universidad, que su contenido sea tomado como cierto. Su importancia radica en que permite abreviar procedimientos administrativos, y al mismo tiempo genera una responsabilidad legal para que quien declare contrario a la verdad de lo que manifiesta, puede como consecuencia, enfrentar un proceso penal por delito de perjurio, tipificado en el artículo 318 de nuestro Código Penal. Lo anterior implica que el estudiante se vea forzado a realizar su mayor esfuerzo para que no sólo incluya información veraz en la Licencia de Publicación, sino que también realice diligentemente la gestión de subir el documento correcto en la plataforma digital Kerwá.

4. Descripción del problema

Con el auge del internet de las cosas (IoT) y la revolución industrial 4.0, se procura automatizar en mayor medida la recolección de datos de campo y la forma en la que se dan las operaciones en la industria [1]. Dentro de cuartos de control en las industrias de petróleo, energía y manufactura es común ver a usuarios operadores interactuando con un SCADA (*supervisory control and data acquisition*) o HMI (*human machine interface*) en el que se recopila todo tipo de información para realizar procesos de control, análisis de datos y mejora en la toma de decisiones.

Desde el punto de vista de mantenimiento industrial, se ha tratado de automatizar el monitoreo de máquinas para implementar un mantenimiento preventivo con base en datos. Este tipo de mantenimiento se conoce como CBM (*conditioned based monitoring*). Su objetivo es no depender de tiempos estimados de vida útil o mantenimiento correctivo de elementos que ya fallaron, lo que logra reducir el costo de mantenimiento, al prevenir incluso averías, y aumenta la producción de la industria al mantener la disponibilidad de los equipos para continuar operando [2]. Se estima que usar un enfoque de mantenimiento preventivo puede ahorrar entre un 18 % y un 98 % en costos, donde se calcula que, solamente en 2016, este gasto ascendía a los 50 billones de dólares [3].

Existen al menos tres tipos de metodologías utilizadas para diagnosticar máquinas hidráulicas: el modelado físico, el modelado con base en datos y los sistemas expertos (con base en reglas). Específicamente, en el modelado con base en datos, el uso de señales de vibraciones y sonido para la extracción de características ha proporcionado información valiosa del estado de salud de una máquina rotatoria [4, 5], por lo que distintos autores las han utilizado en múltiples contextos y con muchas variantes con el fin de mejorar su efectividad.

Las bombas hidráulicas son máquinas presentes en todo tipo de industria, por lo que también en los últimos años se ha dado un incremento en este tipo de investigaciones aplicadas a este contexto para determinar qué tipo de metodología es viable para monitorear y diagnosticar problemas. Por lo tanto, se plantea la siguiente pregunta de investigación: **¿cómo registrar señales de vibraciones, procesarlas y generar modelos de aprendizaje automático para reconocer una anomalía, como lo**

es la cavitación, en bombas hidráulicas? El fin es aprovechar las técnicas de ciencias de la computación para resolver problemas en este tipo de industrias.

5. Marco conceptual

Esta sección presenta los conceptos y la terminología necesaria para entender el contexto de la propuesta. Los conceptos se dividen en tres temas: bombas hidráulicas, procesamiento digital de señales (DSP, por sus siglas en inglés) y aprendizaje automático.

5.1. Bombas hidráulicas

Las bombas hidráulicas son máquinas que se utilizan para impulsar un fluido a través de un sistema de tuberías. Este fluido en estado líquido podría ser agua, aceite, petróleo, lubricante o refrigerante, entre otros. Estas máquinas se dividen en dos grandes grupos: bombas cinéticas y bombas de desplazamiento positivo [6]. En la Figura 1 se resumen los tipos de bombas hidráulicas y su clasificación. Las bombas cinéticas se caracterizan por tener un impulsor radial que rota y descarga el fluido por medio de sus aspas. Por otro lado, las de desplazamiento positivo, conocidas como bombas volumétricas, desplazan un volumen fijo de fluido en cada una de sus pulsaciones. Estas dos clases de bombas hidráulicas tienen casos de uso distintos, por lo que su comportamiento y curvas de desempeño también lo son.

Las bombas cinéticas y las de desplazamiento positivo tienen elementos en común, cabe mencionar por ejemplo cojinetes, eje, acoples y carcasa, entre otros. Por esto, los modos de fallo pueden ser compartidos y afectar a ambos tipos de bomba hidráulica. Algunos ejemplos de problemas con los que se enfrenta la industria son los desalineamientos, la fatiga, las fugas, la corrosión, la sobrepresión y la cavitación. Existen estudios que categorizan hasta 19 tipos de fallas que podrían ocurrir en una bomba [7]. Específicamente la cavitación es un estado en el que la presión de succión es tan baja que se forman burbujas y estas implosionan al pasar a etapas posteriores con cambios abruptos de presión. A continuación se detalla en que consiste este fenómeno.

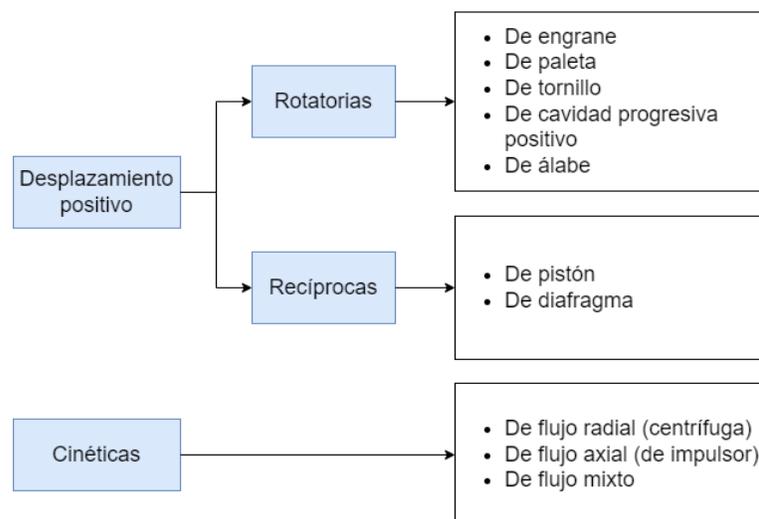


Figura 1: Clasificación de bombas. Las bombas hidráulicas se clasifican con base en su modo de operar: las bombas de desplazamiento positivo donde pueden ser de movimiento rotatorio o recíprocante y las bombas cinéticas. Adaptado de [6].

5.1.1. Cavitación

La cavitación es la aparición de cavidades de vapor en un líquido inicialmente homogéneo [8]. Este fenómeno puede ocurrir en líquidos estáticos o en movimiento, lo que se conoce como cavitación hidrodinámica. Las válvulas, los impulsores de las bombas, las turbinas y los elementos con orificios angostos son algunos de los ejemplos de elementos que tienen contacto con fluido en movimiento y que podrían verse afectados por el fenómeno. Las cavidades o burbujas de vapor se generan cuando la presión del líquido decrece al punto de equilibrio donde ocurre el cambio de estado de líquido a gas. Este punto de equilibrio se conoce como presión de vapor y es dependiente del tipo de fluido y su temperatura. La Figura 2 muestra un ejemplo de las curvas en un diagrama de fases (sólido, líquido y gaseoso). El punto F de la figura se encuentra en estado líquido en su totalidad. Si se asume una temperatura constante, al decrecer la presión se acerca el punto al límite de cambio de estado de líquido a gaseoso. En este momento se pueden empezar a producir burbujas de gas que ante un aumento abrupto de presión implosionan y producen el fenómeno de la cavitación.

La cavitación se puede clasificar en dos regímenes: un inicio de cavitación, que representa el umbral entre un fluido homogéneo y el estado de cavitación, y la cavitación desarrollada, que implica la permanencia prolongada en este estado. Un método para estimar el estado de cavitación en un fluido es por medio del valor

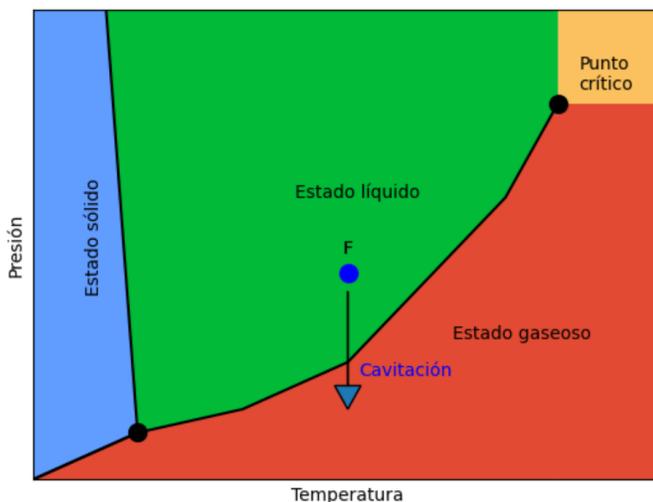


Figura 2: Diagrama de fase. Este diagrama muestra tres estados de la materia y sus cambios en función de la presión y la temperatura. Adaptado de [8].

adimensional conocido como número de cavitación. El número de cavitación tiene variaciones dependiendo del tipo de máquina en la que se esté calculando, pero independiente del contexto, el número de cavitación es una relación entre las presiones involucradas y las propiedades del fluido e indica el potencial de cavitación bajo esas condiciones. El número adimensional se define como

$$Ca = \frac{p_i - p_v}{\rho v^2}, \quad (1)$$

donde p_i es la presión en la entrada de la bomba, p_v es la presión de vapor, ρ es la densidad del fluido y v es la velocidad del fluido en la succión de la bomba.

La cavitación tiene un efecto indeseado en la operación de una máquina hidráulica, la cual está diseñada para trabajar con un fluido en estado líquido y completamente homogéneo. Debido al movimiento que ocurre en estas máquinas, existen aumentos de presión abruptos que implosionan estas burbujas de vapor. Dentro de las consecuencias más comunes de este fenómeno se pueden listar las siguientes: una baja eficiencia mecánica, un aumento de las vibraciones, un aumento del ruido, la erosión de las paredes en las máquinas y la falla mecánica de los elementos. En la Figura 3 se muestra un ejemplo de los efectos de la cavitación.



Figura 3: Daño ocasionado por la cavitación en una turbina tipo Francis horizontal. Como se evidencia en la figura, la cavitación ha ocasionado un desgaste significativo en casi todos los álabes de esta turbina y como consecuencia, esta máquina requiere mantenimiento correctivo. Fotografía proporcionada por AV Ingeniería. Sitio web <https://www.avingeneria.net/>

5.1.2. Estimación de cavitación en bombas centrífugas

Los fabricantes de bombas hidráulicas miden la presión absoluta requerida en la succión de una bomba con el fin de ayudar a los ingenieros en el diseño de las tuberías y así minimizar el riesgo de que ocurra cavitación una vez instalada. Este valor se denomina *carga de succión positiva neta requerida* (NPSHr, por sus siglas en inglés), algunos fabricantes usualmente la proveen usando como unidades el metro de columna de líquido equivalente de presión (o pies en el sistema Imperial de unidades). Durante la selección de una bomba, el diseñador debe tomar el punto de operación y calcular que su *carga de succión positiva neta disponible* (NPSHa, por sus siglas en inglés) sea mayor que la NPSHr indicada por la curva de la bomba. La Figura 4 muestra un ejemplo del NPSHr en una curva real proporcionada por un fabricante.

El valor de NPSHa se calcula como

$$\text{NPSHa} = h_{sp} \pm h_s - h_f - h_{pv}, \quad (2)$$

donde h_{sp} es la carga de presión estática sobre el fluido en almacenamiento, h_s es la carga por diferencia de elevación entre el depósito de almacenamiento y la entrada de succión de la bomba, h_f es la carga por pérdida de presión debido a fricción y h_{pv} es la carga por presión de vapor. Es posible forzar la cavitación en la succión de una bomba alterando cualquiera de estas cargas de presión, pero resulta usualmente más práctico cerrar parcialmente las válvulas ubicadas antes de la succión de una bomba. De esta forma aumenta el valor por carga en pérdidas por fricción h_f y así el NPSHa disminuye.

En la sección siguiente se introducen algunos conceptos sobre señales que van a

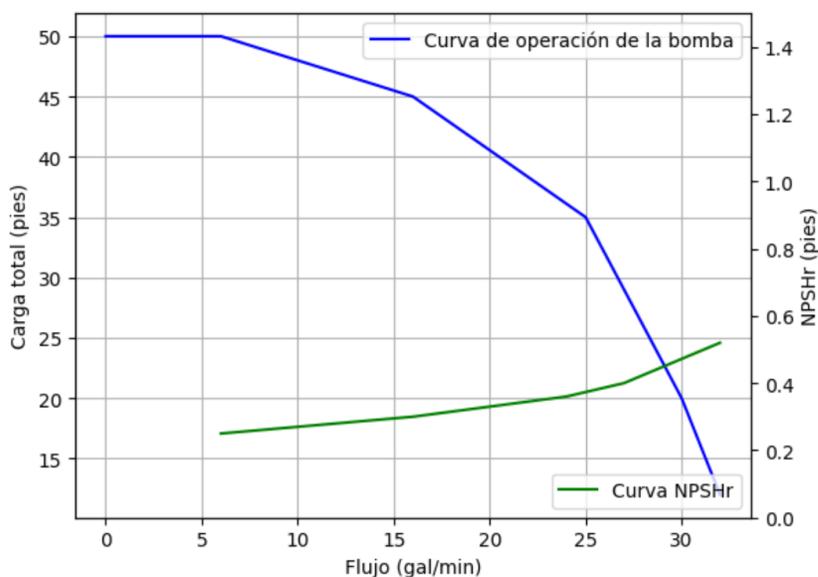


Figura 4: NPSHr de una bomba centrífuga. Este es un ejemplo de una curva de funcionamiento de una bomba centrífuga. Es importante notar que conforme aumenta el flujo disminuye la presión y aumenta el NPSHr. Adaptado de [6].

permitir entender los tipos de datos que se esperan analizar en este trabajo.

5.2. Procesamiento digital de señales

Una señal es una descripción de cómo un parámetro depende de otro. El tipo de señal más común es el voltaje que varía en el tiempo [9]. Se dice que una señal es *continua* cuando representa una función continua (con un ámbito de valores continuos). Mediante un proceso conocido como conversión análogo-digital, se discretizan las señales continuas en elementos individuales, lo que da como resultado una señal discreta que puede ser analizada numéricamente. En la naturaleza todas las señales son continuas, mientras que en el procesamiento computacional se trabaja solo con señales discretas. La Figura 5 muestra un ejemplo de una señal continua y una discreta.

Otro concepto importante es el dominio de una señal. El dominio indica el tipo de parámetro utilizado por la variable independiente (eje X o abscisa). Por lo que para señales medidas en tiempo, se dice que pertenecen al dominio del tiempo. Otro tipo de dominio utilizado en esta área de estudio es el dominio en frecuencia, que es una representación de las mediciones (eje Y u ordenadas) en términos de su repetición.

Dentro del área de procesamiento digital de una señal existen diversos algoritmos

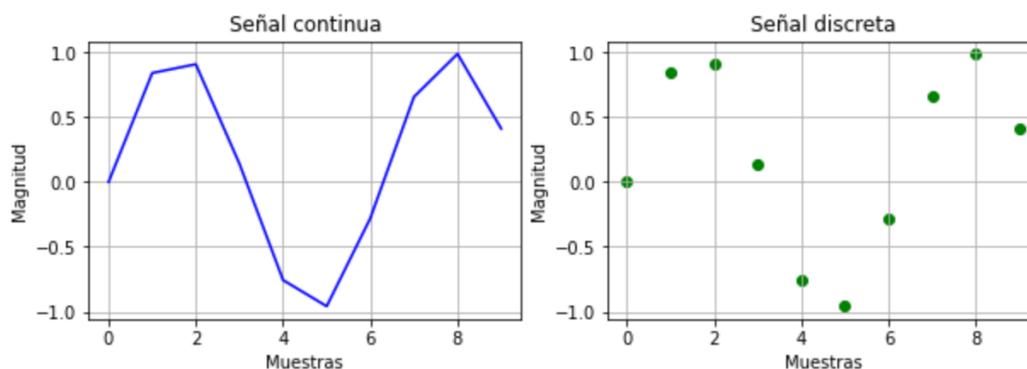


Figura 5: Señal continua y señal discreta.

utilizados con distintos propósitos. Por ejemplo, algunas herramientas de uso extendido son la convolución, la ventana de Hamming y la transformada de Fourier discreta, por mencionar algunas. De igual forma, el uso de estas herramientas está muy extendido a diversas áreas y es común encontrarlos en aplicaciones de múltiples industrias: telecomunicaciones, militares y científicas.

En la sección siguiente se explica detalles en la transformada de Fourier discreta. Se elige este método para convertir una señal al dominio de la frecuencia, ya que existe un algoritmo computacionalmente eficiente, preciso y frecuentemente utilizado en aplicaciones de DSP para calcularlo.

5.2.1. Transformada de Fourier discreta

La transformada de Fourier discreta (DFT, por sus siglas en inglés) pertenece una familia de técnicas matemáticas para descomponer una señal en sinusoides. En la sección se mencionó que una señal puede ser considerada continua o discreta. Estas señales también pueden ser clasificadas como periódicas o aperiódicas, así como una combinación de ambas características que resultan en cuatro categorías para señales: aperiódica-continua, periódica-continua, aperiódica-discreta y periódica-discreta. La DFT es la técnica que permite descomponer una señal periódica-discreta.

En la teoría, ninguna de las transformadas de Fourier permite descomponer una señal finita, ya que por definición, se extiende hacia el infinito en ambos sentidos negativo y positivo. La forma práctica de aplicar este concepto en una señal con una aplicación real es definir una *ventana* con una cantidad finita de muestras, usualmente una potencia de dos y asumir que esta ventana se repite con en el mismo patrón

infinitamente en ambos sentidos. Con esta aclaración, es posible calcular la DFT en una señal como

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i k n}{N}}, \quad (3)$$

donde x_n es la muestra en el índice n y N es el total de muestras. El resultado de la DFT es una secuencia de números complejos X_k con índice k . Una forma alternativa de ver esta ecuación y separarla en su componente real e imaginario, es recordando la *fórmula de Euler*, definida como

$$e^{i\phi} = \cos(\phi) + i\text{sen}(\phi). \quad (4)$$

Por lo tanto, sustituyendo obtenemos el método de calcular DFT por medio de correlación [9]:

$$\text{Re } X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{2\pi k n}{N}\right) \quad (5)$$

e

$$\text{Im } X_k = - \sum_{n=0}^{N-1} x_n \text{sen}\left(\frac{2\pi k n}{N}\right). \quad (6)$$

A pesar de que es más usual representar números complejos con coordenadas rectangulares, en la práctica resulta también útil analizar los resultados en coordenadas polares con una magnitud y un ángulo de desfase entre su componente real e imaginario. Usando notación polar, las ecuaciones 5 y 6 toman la forma siguiente:

$$\text{Mag}X_k = \left(\text{Re}X_k^2 + \text{Im}X_k^2\right)^{\frac{1}{2}} \quad (7)$$

y

$$\text{Fase}X_k = \arctan\left(\frac{\text{Im}X_k}{\text{Re}X_k}\right). \quad (8)$$

La Figura 6 muestra una señal con un único componente senusoidal a 30 Hz en el dominio de la frecuencia, resultado de la transformación DFT. Asimismo, se muestra

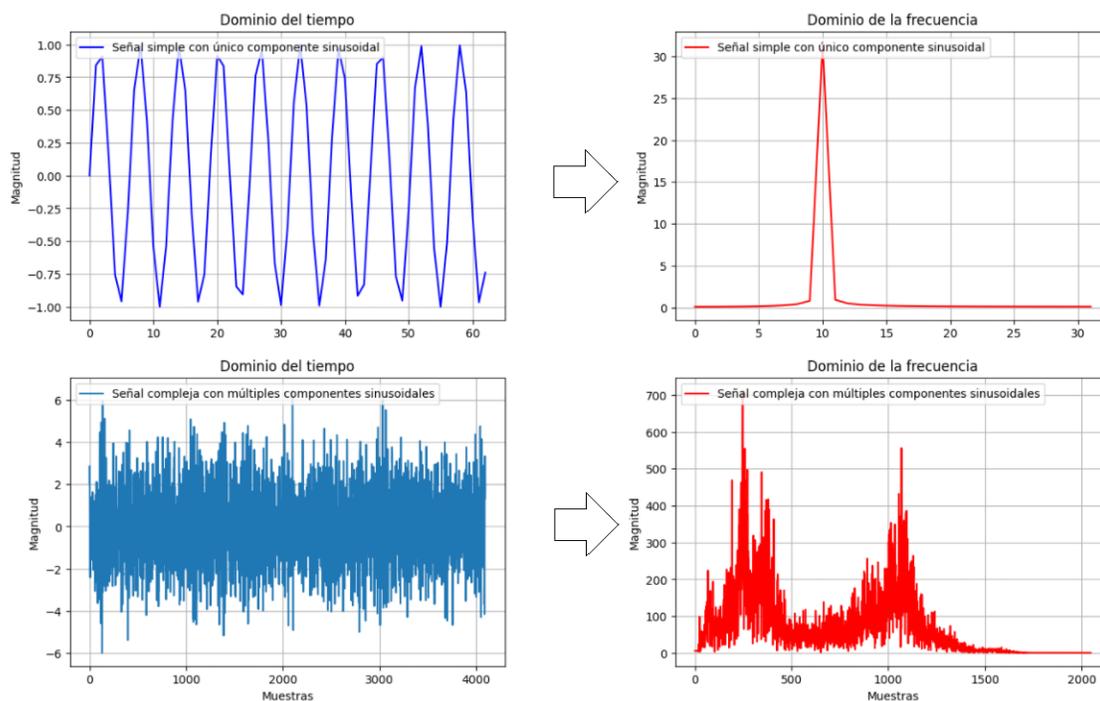


Figura 6: Se muestra una señal simple con un único componente sinusoidal (arriba) y otra con múltiples componentes sinusoidales (abajo). En la columna de la izquierda se muestran estas señales en el dominio del tiempo y en la columna de la derecha en el dominio de la frecuencia. El dominio de la frecuencia proporciona información útil para el análisis de datos de secuencia, por ejemplo en este caso, muestra de forma visual los componentes sinusoidales que conforman la señal y todo su espectro de frecuencias.

un ejemplo de este mismo cálculo, pero con una señal con muchos más componentes sinusoidales y, por lo tanto, similar a una aplicación real como las que se ven en ciencia de datos. En términos de desempeño y notación *Big-O*, las ecuaciones 5 y 6 tienen una complejidad cuadrática N^2 . Por esta razón, para aplicaciones reales es más usual usar un algoritmo mucho más eficiente. La versión más eficiente del algoritmo se conoce como transformada rápida de Fourier [10] (FFT, por sus siglas en inglés) y trabaja descomponiendo una señal del dominio del tiempo con N muestras en N señales con una sola muestra. Este abordaje, en conjunto con operaciones en los arreglos permite reducir la complejidad a $N \log_2 N$ y resulta en una alternativa muy conveniente conforme aumenta la cantidad de muestras en una señal.

5.2.2. Frecuencia fundamental y armónicos

En este trabajo de investigación se usarán los armónicos para el análisis exploratorio de datos y así entender la naturaleza del fenómeno. Por esta razón, es

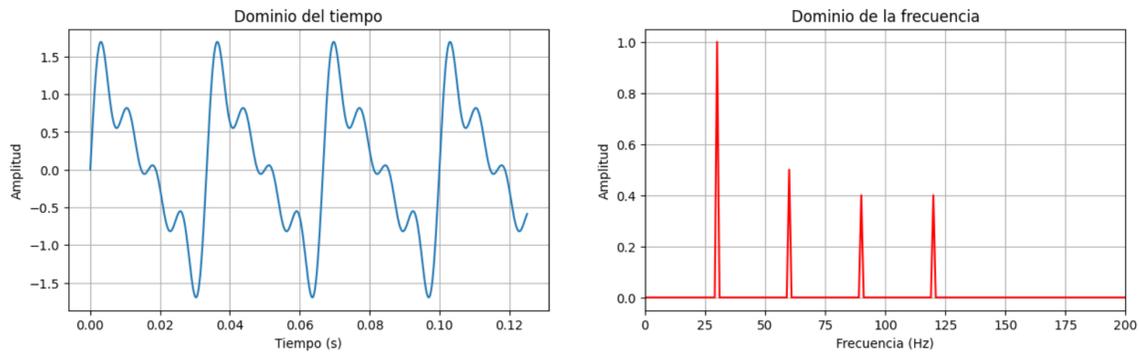


Figura 7: En la figura de la izquierda se muestra una señal en el dominio del tiempo. En la figura de la derecha, se muestra su espectro en el dominio de la frecuencia. La frecuencia fundamental se encuentra en 30 Hz, mientras que sus armónicos en 60 Hz, 90 Hz y 120 Hz.

importante entender el concepto de un armónico para interpretar adecuadamente la información que provee el dominio de la frecuencia. Todas las ondas se componen por una combinación de una frecuencia fundamental y armónicos [11]. Para ilustrar el concepto de la frecuencia fundamental, si se toma una señal periódica con un único componente sinusoidal de frecuencia f y se convierte al dominio de la frecuencia, el espectro resultante mostrará el único componente principal que compone a esta onda y que se conoce como frecuencia fundamental. Por otro lado, si adicionalmente se distorsiona esta onda con nuevos componentes sinusoidales, aquellos que sean múltiplos enteros de la frecuencia fundamental se conocen como armónicos. En la Figura 7 se muestra un ejemplo de un espectro que muestra la frecuencia fundamental y otros armónicos. En la Sección 10.1 se le dará una explicación a los armónicos que se encuentran en los datos. Por último, es importante mencionar que adicionalmente se hará uso de la relación señal a ruido (SNR, por sus siglas en inglés) para entender cuando los armónicos y, en general, cualquier otro elemento con información se ubican en relación con respecto al ruido que es esperado en las señales. El SNR se define como la relación de la media con la desviación estándar de la secuencia que se está analizando. Eso significa que entre más pequeño sea el valor SNR, los elementos con información se pierden entre el ruido y son difíciles de distinguir, incluso para un algoritmo como el que se desea implementar. Idealmente se procura mantener una relación SNR por encima de la unidad que facilite el reconocimiento de la información de interés.

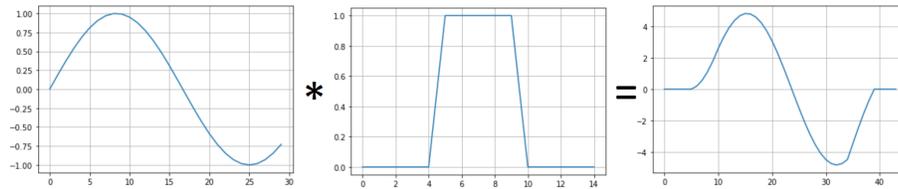


Figura 8: Ejemplo de la operación convolución.

5.2.3. Convolución

La convolución es una operación matemática que toma dos señales y produce una tercera señal. Estos tres elementos son la señal de entrada, la señal de salida y la señal impulso. Dependiendo del tipo de aplicación o de literatura, la señal impulso puede tomar distintos nombres. En este trabajo se usa el término *kernel* para referirse a este elemento. Para describir el concepto simplificado, se descompone la señal de entrada, pasa por el kernel y se produce una señal de salida. Formalmente

$$y[n] = x[n] * h[n], \quad (9)$$

donde $x[n]$ es la señal de entrada, $h[n]$ es el kernel, $y[n]$ es la señal de salida y n el índice. Formalmente se define como

$$y[i] = \sum_{j=0}^{M-1} h[j]x[i-j], \quad (10)$$

donde M es el tamaño del kernel, el índice j representa el punto del kernel y el índice i representa el punto de la señal de entrada. Esta ecuación da como resultado una señal de salida cuyo tamaño equivale a la sumatoria del tamaño de la señal de entrada y el tamaño del kernel menos la unidad. En la Figura 8 se muestra un ejemplo de una operación convolución.

5.3. Inteligencia artificial y aprendizaje automático

El reconocimiento de patrones en el contexto de detección de fallas en máquinas sigue un proceso de al menos cinco etapas definidas [12]: la adquisición de los datos, el procesamiento de la señal, la extracción de características, la fusión o reducción de características y el reconocimiento de patrones.

El aprendizaje automático es una rama de la inteligencia artificial cuyo objetivo es lograr que un modelo aprenda de los ejemplos. Existen tres instancias en la que el aprendizaje se puede dar a partir de datos: de forma no supervisada, de forma supervisada y por refuerzo. En el aprendizaje no supervisado, un agente aprende de las señales existentes sin que un agente externo proporcione información sobre estos. El método más común en este tipo de aprendizaje es por agrupamiento en clústeres, siendo la técnica de *K-medias* muy utilizada en estos modelos. En el aprendizaje supervisado un agente analiza el par entrada-salida, su relación y aprende una función que vincule ambas variables. Algunas de las técnicas utilizadas en el aprendizaje supervisado son la regresión lineal, la regresión logística, las redes neuronales y las máquinas de soporte vectorial (SVM, por sus siglas en inglés). Por último, el aprendizaje por refuerzo es aquel donde un agente aprende por medio de una serie de recompensas y penalizaciones que le indican si una acción fue correcta o no. Un ejemplo de este tipo de aprendizaje es el proceso de decisión de Markov (MDP, por sus siglas en inglés). De los tres métodos de aprendizaje, el no supervisado y el supervisado representan una mejor alternativa para resolver un problema como el planteado en la pregunta de investigación, por lo que el enfoque de este trabajo es experimentar con estos dos aprendizajes.

La meta del aprendizaje supervisado es que de un conjunto de ejemplos N de prueba, que relacionan entradas y salidas de la forma $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ y donde existe una función $y = f(x)$ desconocida que relaciona cada salida y , lograr estimar una hipótesis h que aproxime la función f descrita [13]. Para poner a prueba la exactitud de esta hipótesis, es usual preparar dos conjuntos de ejemplos y que no han sido usados para el entrenamiento: un conjunto de validación cruzada y un conjunto de pruebas. Idealmente, estos conjuntos deben mantener una misma distribución de los ejemplos disponibles. El objetivo de estos conjuntos de ejemplos es que la hipótesis sea tan generalizada como sea posible, de tal forma que también tenga un buen desempeño con ejemplos nuevos. Existen dos tipos de problemas comunes que el aprendizaje supervisado resuelve. Cuando la salida y representa un número y el problema es predecir un valor con base en atributos, entonces se dice que es un problema de *regresión*. Por otro lado, cuando la salida y es un conjunto finito de posibles valores (por ejemplo, estado normal o de falla), se dice que es un problema de *clasificación*.

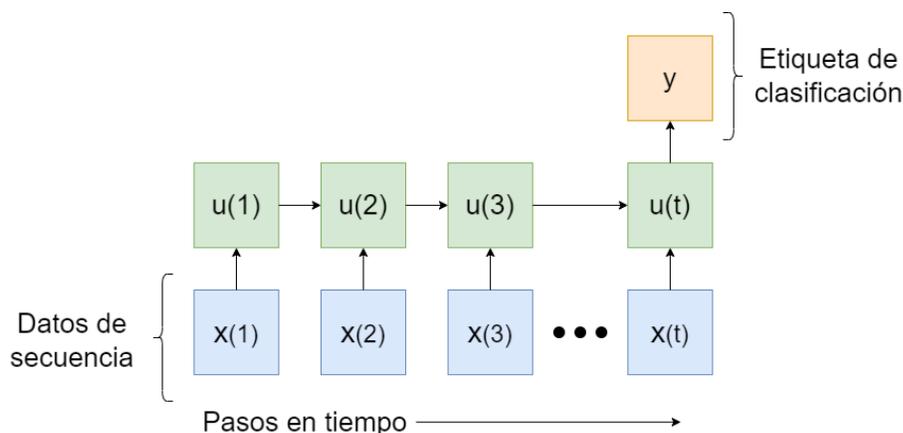


Figura 9: Para resolver el problema de clasificación se puede recurrir a un modelo *muchos a uno*. Esto significa que se tienen múltiples valores de x en secuencia cambiando con el tiempo, luego estas entradas introducen datos nuevos con cada paso de tiempo a un nodo que acumula y procesa esta información (representado con u). Finalmente, la salida consiste en una sola etiqueta y (no necesariamente binaria). Adaptado de [14].

5.3.1. Modelos de secuencia

Los datos que se analizarán en esta investigación son de naturaleza secuencial. A diferencia de otros modelos donde existe una relación de un par (x, y) , los datos de entrenamiento consisten en secuencias de pares (x, y) . Estas secuencias podrían ser de varias formas, como por ejemplo, uno a uno, uno a muchos, muchos a uno o muchos a muchos. Para el contexto de este trabajo es esperado trabajar con modelos de *muchos a uno* donde se espera una cadena de datos de entrada (x_1, x_2, \dots, x_T) , en donde cada dato de entrada x comparte una relación con su vecino y el objetivo es una única etiqueta de clasificación y . En la Figura 9 se muestra un ejemplo de un diagrama conceptual de un modelo *muchos a uno*, similar al que se propone construir en este ejercicio. Es también esperado recurrir a modelos *muchos a muchos* para reducción de la dimensionalidad de los datos de entrada, generación de datos sintéticos y aprendizaje no supervisado, para detección de anomalías. En la Figura 10 se muestra un ejemplo con el diagrama conceptual de otro modelo, *muchos a muchos*, que también se pretende en este trabajo.

Las series de tiempo son una secuencia temporal. En la literatura se encuentran distintos métodos de aprendizaje automático para abordar el análisis con series de tiempo [15]. Los primeros métodos implicaban crear una *ventana deslizante* que ajusta la secuencia en forma de un par (x, y) , tal que podría ser analizado por medio técnicas como una SVM. Otra técnica es usar modelos ocultos de Markov, en donde se calculan

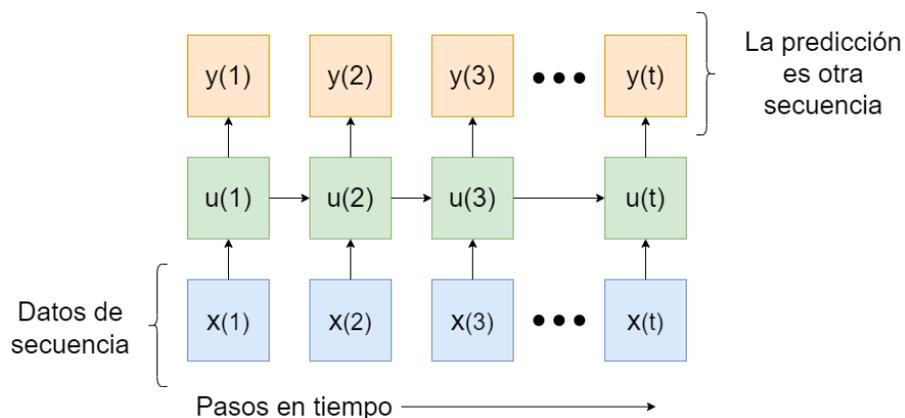


Figura 10: Para construir un detector de anomalías se puede recurrir a un modelo *muchos a muchos*. En este caso, ingresan datos en forma de secuencia y a la salida del modelo, la predicción es otra secuencia de datos. La nueva secuencia puede ser del mismo tamaño que la original o podría ser distinta y esto depende del problema que se desee resolver. Adaptado de [14].

dos modelos de probabilidad: la probabilidad $P(y_i, t | y_i, t - 1)$, que relaciona los valores adyacentes de y con la probabilidad de $P(x|y)$, que relaciona todos los valores de x con los valores ocultos de y . Unos métodos de aprendizaje profundo usualmente usados para abordar el problema de secuencias son las redes neuronales recurrentes (RNN, por sus siglas en inglés) en donde se hace uso del algoritmo conocido como retropropagación a través del tiempo [16]. Dentro de las RNN existen variantes más modernas que logran solventar algunos problemas muy estudiados de la retropropagación a través del tiempo, como por ejemplo, el *desvanecimiento de gradiente*. Una de estas variaciones es *long short-term memory* [17]. Una LSTM es una red neuronal recurrente cuyos nodos siguen un proceso interno más complejo con la transformación de los datos de entrada. Dentro de estos nodos ocurren distintas etapas, conocidas como *compuertas*, en donde con un juego de pesos el nodo es capaz de olvidar cierta información, decidir qué datos actualizar en la secuencia y filtrar qué datos deben salir del nodo.

5.3.2. Reducción de la dimensionalidad y selección de características

Una de las técnicas para reducir la dimensionalidad se conoce como análisis de componentes principales (PCA, por sus siglas en inglés). El PCA es un análisis no paramétrico que permite reducir la dimensionalidad de un conjunto de datos para encontrar información difícil de descifrar con los datos crudos solamente. Es además uno de los métodos más usados de álgebra lineal aplicada [18]. PCA es un método que asume linealidad y trabaja reduciendo las dimensiones en componentes principales,

los cuales son ortogonales entre sí. Una alternativa a lineal a PCA es el análisis discriminante lineal (LDA, por sus siglas en inglés). LDA es un método que se clasifica tanto como un mecanismo para reducir dimensionalidad, como una técnica de aprendizaje automático supervisado. A diferencia de PCA en donde se reduce la dimensionalidad con proyecciones, en LDA se reduce la dimensionalidad maximizando la tasa de varianza entre las clases, lo que trata de maximizar una separabilidad de dichas clases [19]. El algoritmo se compone de tres etapas. En la primera se calcula la distancia entre medias de las clases, en la segunda etapa la distancia entre la media y la muestra de cada clase y en la tercera se construye el espacio de dimensiones reducida que maximiza la varianza entre clases y minimiza la varianza por clase. Por otro lado, una reducción de la dimensionalidad no lineal, se conoce como incrustación de vecinos estocásticos distribuidos en t (T-SNE, por sus siglas en inglés) [20]. Este algoritmo toma la incrustación de vecinos estocásticos (SNE) y calcula distancias entre puntos con similitudes usando la distribución t de *Student*. Este abordaje permite calcular la optimización por medio de gradientes y resolver problemas que la implementación original del algoritmo enfrenta. Es importante aclarar que T-SNE es una herramienta que sirve solo para visualizar datos con alta dimensionalidad en dos o tres dimensiones. No se recomienda como herramienta de propósito general, pero brinda una buena idea del comportamiento que se busca con una reducción no lineal. Para efectos de este trabajo, se busca hacer una reducción no lineal por medio de redes neuronales.

Existen múltiples técnicas estadísticas para determinar correlación entre las entradas y salidas. Estas dependen del tipo de datos de entrada y salida, ya sean categóricos o numéricos. Para este trabajo se planea seleccionar las características de interés por medio de la prueba F para análisis de varianza (ANOVA, por sus siglas en inglés). Esta prueba estadística permite determinar si la media de dos o más muestras proviene de una misma distribución o no. Es una prueba adecuada para la selección de características cuando las variables de entrada son valores numéricos y se desea poner a prueba su dependencia con respecto a variables categóricas. Asimismo, es una prueba que permite calcular para dos o más categorías, lo que resulta conveniente para poner a prueba un caso de clasificación binario y múltiple.

En la siguiente sección se muestran los artículos de investigación encontrados que relacionan las máquinas hidráulicas, el procesamiento digital de las señales y la inteligencia artificial.

6. Estado del arte

Para realizar esta propuesta, se realizaron dos revisiones la literatura. La primera es más general sobre el tema, se realiza en mayo del 2020 y relaciona el tema de bombas hidráulicas, su diagnóstico y prognosis utilizando inteligencia artificial. Esta revisión consistió en extraer artículos de investigación, conferencias y libros de las bases de datos ACM, IEEE ScienceDirect y Springer. Dentro de los resultados obtenidos, se encuentran revistas con prestigio en el tema que se desea desarrollar, por ejemplo: *Mechanical Systems and Signal Processing* y *Measurement*, ambas de la editorial Elsevier, que tiene artículos de cuartil Q1 y Q2 en temas de instrumentación, ingeniería electrónica, ingeniería mecánica, procesamiento de señales y computación aplicada [21, 22]. En esta primera revisión se encuentran 64 artículos relacionados. La segunda revisión es más específica, se realiza en marzo de 2021 y utiliza las mismas revistas antes consultadas, pero el análisis del estado del arte se enfoca en aprendizaje automático y el reconocimiento de cavitación en bombas hidráulicas con estas técnicas. En esta segunda revisión se encuentran 13 artículos relacionados.

6.1. Inteligencia artificial y diagnóstico de máquinas hidráulicas

Uno de los experimentos encontrados como resultado de la búsqueda descrita es la investigación del 2005, que es a su vez una de las primeras encontradas en este tipo de estudios [23]. Los autores usaron una SVM para encontrar patrones ante una serie de fallas en una turbo bomba y compararon características de forma *uno a uno y uno a muchos*. Por último, muestran la separación de los diferentes clústeres asociados a cada tipo de falla en este caso.

En el 2006 se propone usar redes neuronales artificiales para seleccionar características de vibraciones y detección de múltiples fallas en bombas de lubricación [24]. Este aporte demuestra cómo esta metodología permite obtener

conclusiones sin ser experto en análisis de vibraciones. En un caso de estudio de 2007, se propone una variación al modelo estadístico de Markov para realizar una detección temprana en fallas [25]. El modelo propuesto se diferencia del tradicional modelo de Markov en que se segmentan las observaciones. Puesto en práctica con una bomba hidráulica, se logra estimar un *remaining useful life* (RUL, por sus siglas en inglés) de elementos mecánicos como consecuencia de su desgaste. Este caso de estudio tiene resultados con una precisión del 96 %.

En el 2009 se realiza un experimento para diagnosticar fallas en elementos mecánicos por medio de reglas con lógica difusa [26]. En el experimento usaron bombas centrífugas en dos escenarios: uno de laboratorio y uno en campo. Los resultados obtenidos en el laboratorio, dependiendo de si era desbalance o desalineamiento, tuvieron precisión variable entre 54 % y 100 %. Por otro lado, con los datos obtenidos en campo la variación se redujo, pero también lo hizo la precisión máxima, de 71 % a 90 %.

Un experimento en una línea de investigación similar, pero enfocado no en fallas, sino en evaluar consumo energético, es propuesto en el 2014 [27]. Este experimento, mediante uso de vibraciones y presión del fluido, propone estimar el consumo energético de una bomba de desplazamiento positivo, mediante tres tipos distintos de redes neuronales. Sus resultados oscilaron entre 80 % y 99 % de precisión.

Cuadro 1: Distribución de los artículos encontrados, de la primera revisión de literatura general, por tipo de máquina o elemento con $N = 64$.

Tipo de máquina o elemento	Distribución
Bombas (General)	45,3 %
Máquinas rotatorias (General)	3,1 %
Engranés	9,4 %
Motor	18,8 %
Cojinetes	23,4 %

En el 2017 se propone una herramienta llamada *SpotCheck*, la cual se compara contra una SVM [28]. Este experimento es novedoso, ya que utilizan un campo magnético en vez de otras señales tradicionales para monitorear los estados operativos (no falla). Con

Cuadro 2: Distribución de los artículos encontrados, de la primera revisión de literatura general por tipo de falla con N=64.

Estado	Distribución
Normal	3,1 %
Cavitación	12,5 %
Desalineamiento	15,6 %
Falla en elemento	68,8 %

esta metodología logran obtener precisión de entre 84 % y 97 %. Por último, Siano *et al.* [29], muestran que una bomba de desplazamiento positivo cambia su respuesta en el dominio de la frecuencia ante cavitación en la máquina. Los resultados de este estudio son principalmente cualitativos, ya que no se hace una comparación o un análisis de precisión de técnicas de reconocimiento de patrones.

Además de los artículos de investigación, se encuentran dos artículos de revisión de literatura relacionados con el área de estudio, que proporcionan una guía metodológica de cómo distintos autores han llevado a cabo este tipo de experimentos y qué trabajo futuro relacionado es sugerido [12, 30]. En la revisión de diagnóstico y prognosis en bombas centrífugas, los autores agrupan sus resultados sobre el estado del arte en tres categorías: métodos de modelado físico, modelado por datos y modelos basados en conocimiento. De ellas, el modelado por datos corresponde al monitoreo por medio de análisis estadístico y técnicas de aprendizaje, donde principalmente se analizan redes neuronales y sus variantes. En la revisión de señales para el diagnóstico de fallas hidráulicas, los autores de igual forma distribuyen las metodologías en las mismas tres categorías, con la diferencia de que al modelo basado por datos se le llama métodos basados en señales. Este artículo detalla los pasos conceptuales a seguir que han usado distintos autores para extraer datos, procesar señales y tomar decisiones con base en técnicas de inteligencia artificial. Esta revisión de literatura analiza un área más grande de estudio: sistemas hidráulicos como bombas, cilindros, válvulas y motores. Finalmente, proponen un listado de problemas claves a resolver en el monitoreo de sistemas hidráulicos, en los que casos operativos particulares presentan problemas en el reconocimiento de patrones, por lo que sugieren técnicas usadas en otras aplicaciones, como reconocimiento de imágenes,

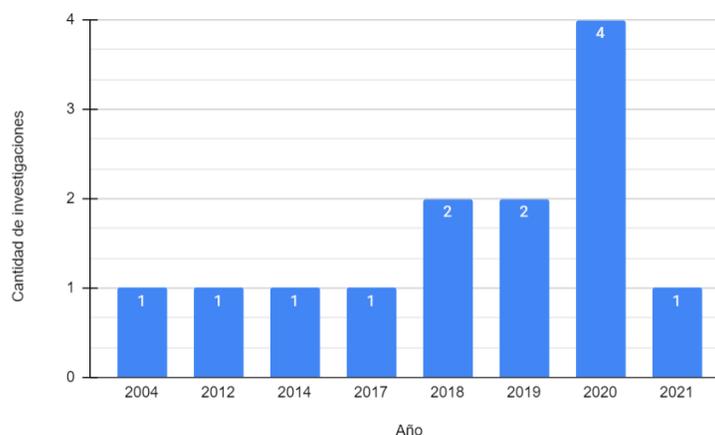


Figura 11: Distribución de las investigaciones encontradas en la segunda revisión de literatura más específica y que relacionan las palabras claves *aprendizaje automático*, *bomba* y *cavitación*. Se encuentra un total de trece investigaciones dentro de los repositorios usados desde el año 2000 hasta marzo de 2021. Se ve un incremento en este tipo de investigación en años recientes. Estos estudios fueron usados para establecer el trabajo futuro de este proyecto de investigación.

para indagar posibles soluciones.

Del ejercicio de revisión de literatura se encuentran 64 artículos relacionados con el monitoreo de bombas hidráulicas y técnicas de inteligencia artificial. Estas investigaciones se dividen en artículos de monitoreo de elementos mecánicos que podrían ser parte de una bomba y artículos exclusivamente de bombas hidráulicas. La distribución de estos experimentos por tipo de máquina o elemento se muestra en detalle en el Cuadro 1 y por tipo de falla en el Cuadro 2. Por último, dentro de algunos de los desafíos mencionados en los estudios anteriores destacan la complejidad que existe en clasificar fenómenos hidráulicos y extraer la información de interés de las señales. Para ello, se han propuesto diversos métodos para optimizar la extracción de características, algoritmos de clasificación, uso de técnicas combinadas y la exploración de técnicas utilizadas en otros contextos.

6.2. Aprendizaje automático y reconocimiento de cavitación

Se encuentran estudios que relacionan cavitación con aprendizaje automático. Un mismo equipo de investigación realizó dos experimentos relacionados con una bomba centrífuga: reconocimiento por medio de una SVM [31] y comparación entre la SVM y la técnica de K-vecinos más próximos [32]. En ambos trabajos se utilizan características tradicionales en ingeniería como lo son velocidad, presión y flujo, para el entrenamiento

supervisado de los modelos. De estos trabajos los autores concluyen que la SVM es una mejor solución, incluso con pocos datos disponibles.

Otro equipo de investigación también realizó dos experimentos con cavitación. El primero en el 2012, se realiza una comparación del algoritmo genético *Gene Expression Programming* (GEP, por sus siglas en inglés) con una SVM y una variante llamada *Proximal SVM* [33]. Esta comparación utiliza características estadísticas para entrenar los modelos: promedio, desviación estándar, curtosis y asimetría. En el 2014 realizan otra comparación, pero esta vez poniendo a prueba diferentes técnicas de reducción de la dimensionalidad con la intención de medir su efecto en la exactitud de modelos tradicionales como los árboles de decisión, el clasificador bayesiano, las redes bayesianas y los K-vecinos más próximos [34].

Un experimento propone una alternativa con un modelo no supervisado [35]. Este experimento realiza primeramente etapas de convolución 2D para procesar los espectrogramas resultantes de las vibraciones como imágenes, luego se realiza una reducción de la dimensionalidad hasta dos dimensiones y concluye con un agrupamiento K-medias para las distintas anomalías (incluyendo cavitación). Este experimento muestra de forma visual los distintos clústeres formados por el algoritmo.

Cuadro 3: Distribución de los artículos revisados por tipo de aprendizaje con $N = 19$.

Técnica	Aprendizaje	Distribución
SVM	Supervisado	42,0 %
ANN (General)	Supervisado	26,3 %
K Vecinos más cercanos	Supervisado	10,5 %
CNN (1D)	Supervisado	5,3 %
Arboles de decisión	Supervisado	5,3 %
Random forest	Supervisado	5,3 %
K-Medias	No supervisado	5,3 %

Tres experimentos se enfocan en abordar la clasificación de dos problemas concretos: el bloqueo de la bomba y cavitación. Estos experimentos fueron realizados en el mismo banco de pruebas por diversos autores [36, 37, 38] y se analizaron distintas características, entre las que destacan las siguientes: velocidad, presión y vibraciones.

Estas mediciones fueron analizadas y entrenadas en el dominio del tiempo por medio del promedio, la desviación estándar y la kurtosis. Por último, entre los modelos utilizados se encuentran las redes neuronales simples y algunas variantes de las SVM.

En resumen, del segundo ejercicio y actualización de revisión de literatura, se encuentra un total de trece investigaciones donde los autores realizaron reconocimiento de cavitación. En dos casos el estudio se desarrolló con una bomba de desplazamiento positivo, mientras que el resto fueron usando bombas centrífugas. La distribución de estas investigaciones se muestra en la Figura 11 para las últimas dos décadas. Del total, se encuentra que tres estudios utilizaron DFT para la conversión al dominio de la frecuencia, mientras que solamente en un estudio se utiliza transformada *wavelet* (CWT, por sus siglas en inglés). El resto de los estudios se enfocaron en utilizar características basadas en estadística descriptiva o variables propias de ingeniería: flujo, presión o corriente eléctrica. En cuanto al tipo de aprendizaje utilizado, se encuentra que doce investigaciones utilizaron aprendizaje supervisado y solo una investigación uso no supervisado para detectar anomalías por medio de agrupamiento. En el Cuadro 3 se muestran todas las técnicas encontradas en los trece artículos y la distribución sobre su uso en los mismos.

Con base en las revisiones de literatura, las conclusiones y la dirección del trabajo futuro propuesto, se indica a continuación la motivación para realizar este trabajo.

7. Justificación

Este trabajo de investigación propone una metodología para recolectar señales de vibración de bombas hidráulicas, extraer información, realizar un diagnóstico sobre el estado de la máquina y encontrar anomalías. En este contexto, sigue la línea que se muestra en los experimentos vistos en las revisiones de literatura, donde se evidencia el auge de la inteligencia artificial en el monitoreo de elementos mecánicos, en su gran mayoría elementos rotatorios como cojinetes y ejes.

Específicamente, en el área de aprendizaje automático y el reconocimiento de cavitación se encuentra que los distintos experimentos mencionados en la Sección 6.2 consiguen en su mayoría buena exactitud en la clasificación, aunque se omiten otras

métricas para el análisis del error como la precisión y la exhaustividad. También, no en todos se indica la naturaleza de las características usadas en detalle, principalmente cuando son en el dominio de la frecuencia y se encuentra que aproximadamente la mitad de los experimentos utilizaron características propias de la estadística descriptiva. Por otro lado, entre los modelos encontrados cabe recalcar las redes neuronales, algunas variantes como el perceptrón multicapa o redes convolucionales y mayoritariamente las SVM. Sin embargo, no se encuentran estudios detallados sobre el abordaje del problema como modelo de secuencia y con técnicas como las redes neuronales recurrentes. Tampoco se encuentran estudios implementando autocodificadores para detección de anomalías. Por lo tanto, se propone usar señales de vibración como series de tiempo en el entrenamiento de redes neuronales recurrentes con sus variantes más modernas, como las LSTM, para así generar datos y conocimiento sobre la aplicación de estas técnicas en el contexto de máquinas hidráulicas, siendo así un estudio multidisciplinario. Otros autores ya han demostrado que la cavitación cambia la forma de las señales de vibración en el dominio de la frecuencia, por lo que resulta conveniente usar este hecho como punto de partida y enfocar el trabajo de investigación en la experimentación con los algoritmos a usar en el procesamiento de las señales, la implementación de diversas arquitecturas en los modelos y el procesamiento de las características. De igual forma será valioso experimentar con técnicas de reducción de la dimensionalidad, lo cual influye en el costo computacional, con el fin de encontrar soluciones eficientes. Por último, como se comentó en la Sección 4 de este documento, al nacer este trabajo de la necesidad de bajar costos y al ser la cavitación un problema con impacto en el mantenimiento industrial, se presenta una oportunidad para realizar un aporte a las industrias que usan este tipo de máquinas.

La siguiente sección describe la intención de este trabajo de investigación y los objetivos específicos necesarios para lograrla.

8. Objetivo

8.1. Objetivo general

El objetivo general de este trabajo es desarrollar un modelo para reconocer cavitación como una anomalía en bombas hidráulicas mediante señales de vibraciones y técnicas de aprendizaje automático.

8.2. Objetivos específicos

Los objetivos específicos son los siguientes:

1. Diseñar una metodología para adquirir datos de vibraciones, procesar las señales crudas y reconocer cavitación a partir de modelos en bombas centrífugas.
2. Diseñar un experimento para simular estados operacionales en un banco de pruebas con una bomba centrífuga.
3. Construir un modelo de clasificación binario para el estado de una bomba centrífuga.
4. Construir un modelo para detección de anomalías en una bomba centrífuga.
5. Analizar la precisión, la exhaustividad y la generalización de los modelos en el contexto de clasificar cavitación y detectar anomalías en bombas centrífugas.

9. Metodología

En esta sección se describe el plan a seguir para realizar este trabajo de investigación y cumplir con cada objetivo específico. Como se indicó en la sección de marco teórico y en las revisiones de literatura relacionadas, las etapas que se van a desarrollar son adquirir los datos, procesar las señales, reducir las características y entrenar los modelos.

9.1. Adquisición de datos

La adquisición de datos en campo se realiza con sensores y convertidores análogo-digital incluidos en tarjetas electrónicas especializadas. De acuerdo con la revisión



Figura 12: Módulo de adquisición de vibraciones modelo SAB de la marca ADASH.

de literatura, los sensores más comúnmente usados son acelerómetros, sensores de presión, sensores de flujo y micrófonos especializados. Los parámetros principales que permitirán la selección apropiada de una tarjeta electrónica son la resolución, la tasa de adquisición de datos, la cantidad de sensores y el bus de transferencia de datos. La resolución del convertidor análogo-digital habitual en la medición de señales mecánicas es de 12 o 24 bits. Es importante garantizar una discretización granular de la señal como para medir cambios muy pequeños. Existe evidencia de autores que han podido recopilar información con resultados satisfactorios usando 12 bits [39]. El criterio de selección de la tasa de adquisición de datos se basa en la teoría descrita por Smith [9], en donde se indica que la tasa debe ser al menos el doble de las frecuencias de interés. Por último, experimentos anteriores muestran que algunos autores han instalado una cantidad variable desde 1 hasta 5 sensores distribuidos entre la bomba, el motor y la estructura [39].

En este experimento se usará un módulo de adquisición de vibraciones modelo SAB de la marca ADASH, el cual cumple con los requerimientos descritos. La Figura 12 muestra una fotografía del módulo. Algunos autores han demostrado que la cavitación podría estar presente en ámbitos bajos del espectro como 10-500 Hz [40, 41]. Otros autores coinciden, pero incluso amplían el ámbito hasta 20 kHz inclusive [42]. Para efectos de este ejercicio se ajusta la tasa de adquisición de datos a 8192 Hz, ya que es una potencia de dos lo que facilita su procesamiento en las etapas siguientes y como se verá en la sección 10.1 de análisis de datos, es lo suficientemente elevada como para adquirir el espectro de frecuencias deseado en el banco de pruebas del experimento. El banco de pruebas a usar es un modelo HM 150.04, de la marca GUNT Hamburg y propiedad



Figura 13: La bomba centrífuga y el banco de pruebas a usar en la adquisición de datos. Este equipo es propiedad de la Escuela de Ingeniería Mecánica de la Universidad de Costa Rica.

de la escuela de ingeniería mecánica de la Universidad de Costa Rica. Este banco de pruebas tiene instalada una bomba centrífuga modelo JESX6 de la marca EBARA. Esta bomba tiene una sola entrada en la succión y una sola salida en la descarga. El motor se encuentra acoplado al mismo nivel que la bomba, por lo que se considera una bomba *horizontal*, el eje que une al rotor del motor es el mismo que está unido al impulsor, por lo que no hay acople entre ejes, y el impulsor mide 104 mm con una cantidad de 6 álabes. La Figura 13 muestra una fotografía del sistema completo junto con su bomba. El fabricante de la bomba centrífuga proporciona una curva de funcionamiento, la cual se muestra en la Figura 14 y se convierte al sistema internacional de unidades (SI) para este documento. Esta curva es el resultado de pruebas de laboratorio realizadas por el fabricante y solo aplica para una velocidad de giro de 2800 revoluciones por minuto (rev/min).

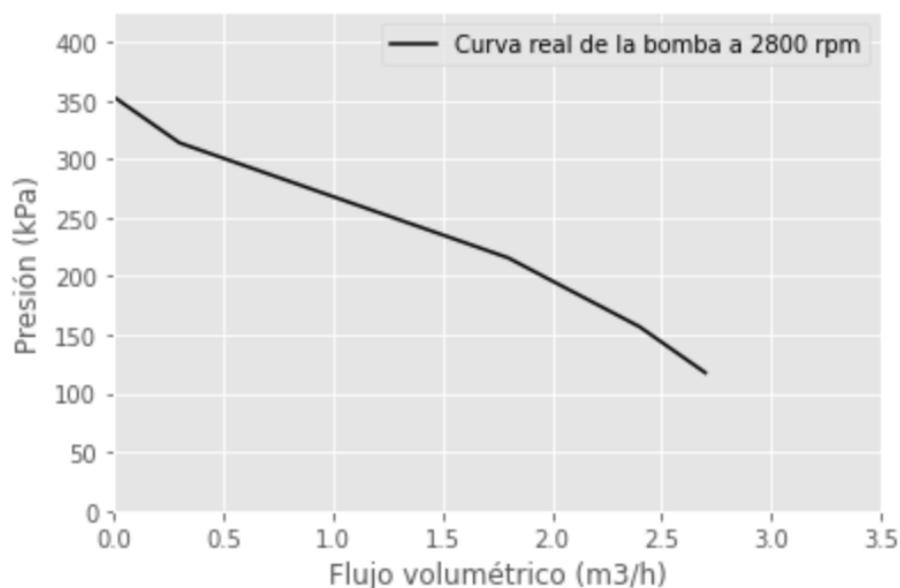


Figura 14: La bomba centrífuga usada en el experimento es una Ebara JESX 6 con seis álabes. Su curva de operación presión-flujo volumétrico sigue el patrón esperado de una bomba centrífuga. La curva es dependiente de la velocidad de giro. Curva adaptada de [43].

9.2. Cavitación de la bomba hidráulica

Para cumplir con el segundo objetivo específico se debe recolectar datos de la bomba en su estado operativo normal y bajo los efectos de la cavitación. Se define el estado normal como un punto en la curva donde el NPSHr es potencialmente bajo y en donde usualmente las bombas hidráulicas tienen su punto operativo con una alta eficiencia. Para este ejercicio, este punto se define aproximadamente a 200 kPa y 2 m³/h. Puesto que es un banco de pruebas, la tubería no puede ser modificada y por lo tanto el NPSHa es un valor fijo. Esto nos limita a solo dos mecanismos posibles para potencialmente aumentar el NPSHr de la bomba y aumentar la probabilidad de que la bomba entre en cavitación. La primera alternativa es aumentar la velocidad de giro de la bomba, para así desplazar la curva del fabricante "hacia arriba y extendiendo hacia la derecha". El otro método es abrir más la válvula de descarga, tal que el flujo volumétrico aumente, disminuya la presión de descarga y aumente el NPSHr. En la Figura 15 se muestran los mecanismos concretamente para este banco de pruebas. Asimismo, en la Figura 16 se ilustra como cambian las curvas y los puntos de operación con estas alternativas. Se toma la decisión de mantener la velocidad de giro de la bomba como constante en 2800 rev/min y se abre la válvula para inducir cavitación. El punto elegido como estado

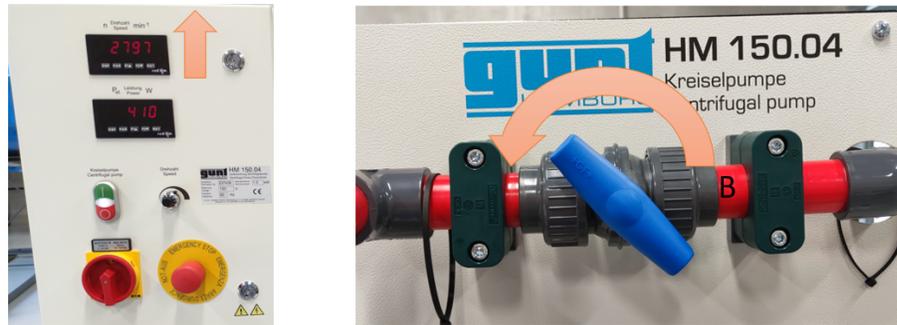


Figura 15: Los dos mecanismos disponibles para modificar el estado operativo de la bomba hidráulica. En la fotografía de la izquierda se muestra un variador de frecuencia con el cual es posible aumentar o disminuir la velocidad de giro de la bomba. En la fotografía de la derecha se muestra la válvula de descarga que regula el flujo de agua a la salida de la bomba.

de cavitación corresponde aproximadamente a 110 kPa y 2,7 m³/h. En la Figura 17 se muestran ambos puntos en la curva.

En vista de que la tasa de adquisición es de 8192 muestras cada segundo, se decide recolectar durante tramos cortos de tiempo. Los pasos a seguir para la toma de datos fueron los siguientes:

- Se inicia el banco de pruebas a 2800 rev/min. Se abre la apertura de la válvula al punto definido como operación normal.
- Se recolectan datos por 1 minuto. En todas las recolecciones se confirma que la cantidad de muestras coincida con la tasa de adquisición antes de proceder.
- Se recolectan datos por 1 minuto adicional en estado normal.
- Luego se ajusta la apertura de la válvula a la posición definida como operación con cavitación.
- Se recolectan datos por 1 minuto. Se confirman los datos.
- Se recolectan datos por 1 minuto adicional.
- Se ajusta la válvula nuevamente a la posición normal. Se inicia la recolección por 20 segundos. Luego se abre la válvula a la posición de cavitación por 20 segundos. Por último, se regresa la válvula a la posición normal por otros 20 segundos. Esta prueba se denomina combinada y el tiempo de cada estado es aproximado. El objetivo es poder visualizar fácilmente el cambio de estado en una misma toma de datos.

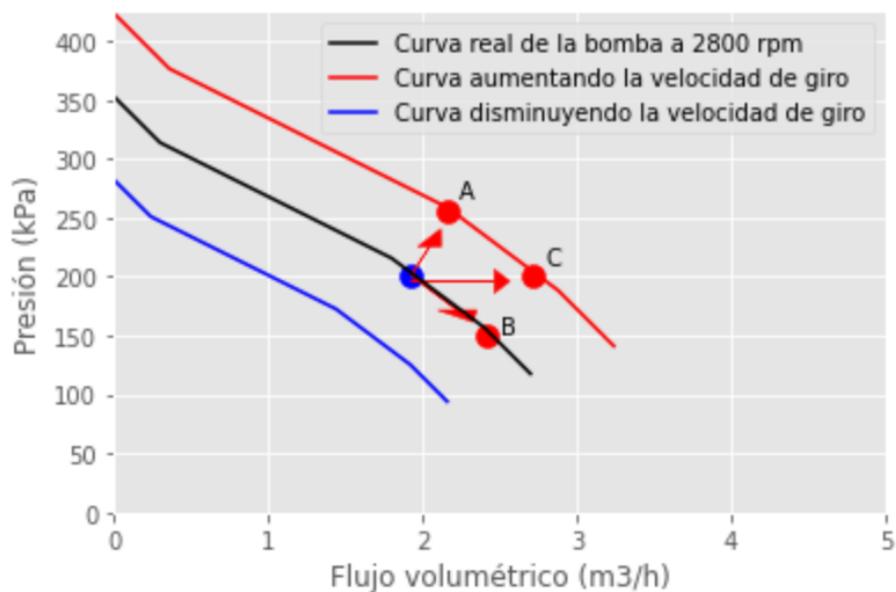


Figura 16: Distintas curvas de operación. Se muestra la curva operativa de la bomba a distintas revoluciones de giro. El punto A corresponde al traslado del punto operativo al aumentar la velocidad de giro. El punto B corresponde a abrir más la válvula para aumentar el flujo. El punto C es la combinación de ambos mecanismos descritos.

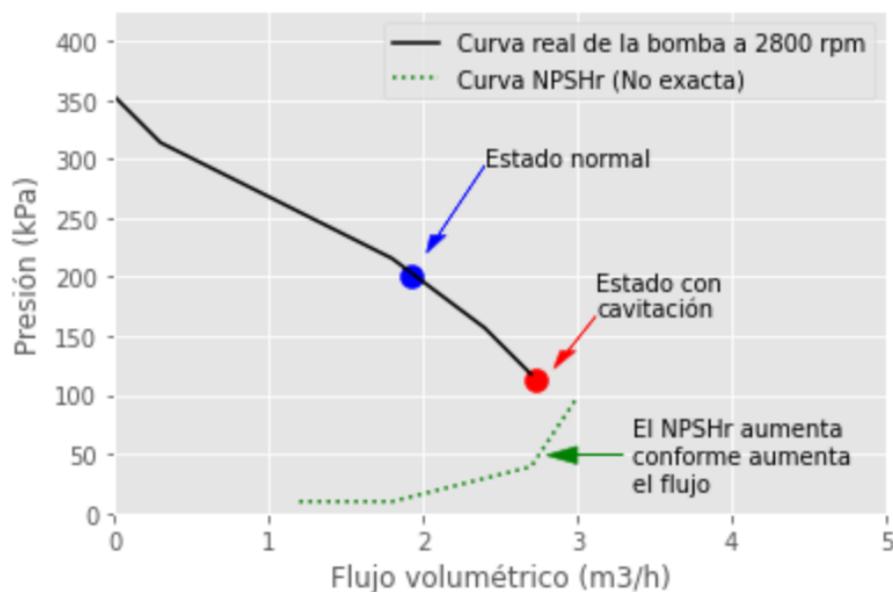


Figura 17: Puntos de operación escogidos para este experimento. El punto normal corresponde a 200 kPa y 2 m³/h. El punto de cavitación corresponde a 110 kPa y 2,7 m³/h. Estos puntos son válidos únicamente para una velocidad de giro de 2800 rev/min. El fabricante no proporciona una curva exacta de NPSHr, por lo que se estima con fines ilustrativos de como aumenta su valor requerido.

En total se realizan cinco tomas de datos en momentos distintos, para un total de 5 minutos y lo que se traduce en 2 457 600 de muestras por eje. La toma de datos se distribuyó de acuerdo con el Cuadro 4.

Cuadro 4: Distribución de la recolección de datos

	Estado	Tiempo (min)	Muestras
1	Normal	1	491 520
2	Normal	1	491 520
3	Cavitación	1	491 520
4	Cavitación	1	491 520
5	Combinado	1	491 520

9.3. Procesamiento de las señales y selección de las características

Este trabajo de investigación aborda el problema desde el dominio del tiempo y el dominio de la frecuencia. Algunos modelos procesaran los datos de una forma u la otra. Para preparar los datos con el fin de extraer características del dominio de la frecuencia primero se recorre la serie de tiempo con una ventana deslizante de 512 muestras, luego a cada ventana se le aplica un filtro con una ventana de Hamming para aplanar los extremos, seguido por un filtro de media para centrar con respecto a cero y por último se transforma al dominio de la frecuencia usando el algoritmo FFT. Esto da como resultado una matriz de 2048 muestras con 4096 frecuencias discretas por cada categoría. Como el problema de clasificación es binario, la matriz resultante es de 4096×4096 . El siguiente paso es seleccionar aquellas frecuencias que aportan información que permita clasificar estas señales, por lo que cada frecuencia se somete a una prueba-F ANOVA. Como regla general en el abordaje de selección de características se consideran como las más relevantes para la resolución del problema aquellas que tienen un valor-F alto y se descarta el resto. Con el fin de no perder generalidad en el modelo, se decide continuar con al menos mil características, lo que permite conservar información de las frecuencias ubicadas a lo largo de varios ámbitos de interés del espectro. Esta herramienta es de gran ayuda para descartar componentes sinusoidales

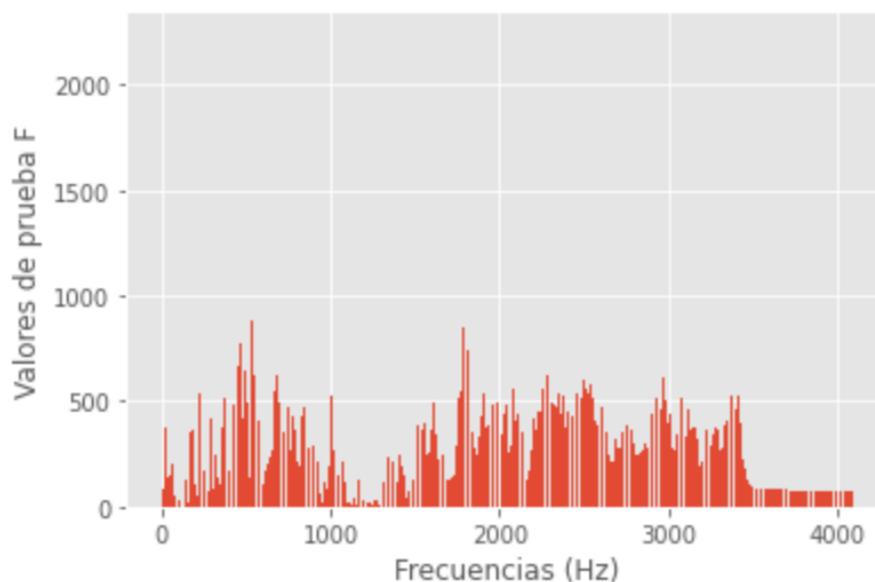


Figura 18: El resultado del análisis de varianza le asigna un valor-F alto a aquellas frecuencias del espectro que permiten diferenciar si una bomba se encuentra en estado de cavitación o no.

cuyas frecuencias se encuentran presentes en ambos estados y que no proporcionan información para separar una categoría de la otra. De igual forma permite reducir el costo computacional al eliminar cientos de frecuencias innecesarias para este análisis. La Figura 18 muestra el resultado del proceso de selección de características. El resultado de este procedimiento es una matriz de 4096×1000 . Con el fin de visualizar este procedimiento se presenta un diagrama de concepto en la Figura 19.

El procesamiento de la señal en el dominio del tiempo se realiza de forma distinta. Se tienen tres señales de entrada, una por cada eje x , y , z . Cada una de estas señales podría pasar o no por una capa de convolución 1D, para luego pasar por una red LSTM. El objetivo es conseguir una reducción de la dimensionalidad hasta obtener un vector reducido que representa cada categoría. Seguidamente este vector de características podría ser usado en un modelo de clasificación con LSTM o un decodificador para el caso de los modelos para detección de anomalías. En la Figura 20 se ilustra este procedimiento en detalle.

En las siguientes dos subsecciones se detallan las arquitecturas, los parámetros y los hiperparámetros elegidos para los modelos.

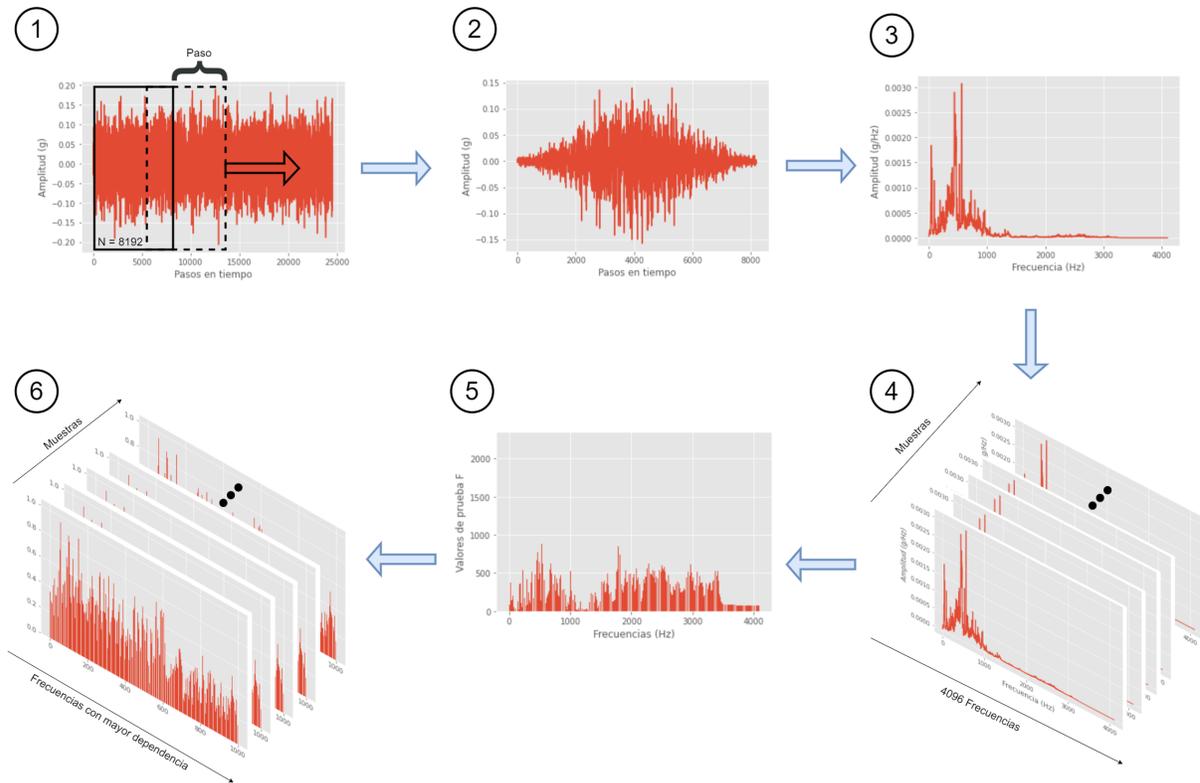


Figura 19: Metodología para la preparación de datos en el dominio de la frecuencia. En el primer paso, una ventana deslizante recorre la totalidad de los datos y los separa. En el segundo paso se aplica un filtro con una ventana de Hamming y un filtro de media. En el tercer paso se muestra el resultado de los pasos anteriores. En el cuarto paso se agrupan para someter los datos a una prueba-F ANOVA. El quinto paso muestra el resultado de la prueba. En el sexto paso se toman las mil frecuencias más relevantes de cada muestra para formar el conjunto de datos de entrenamiento y validación cruzada.

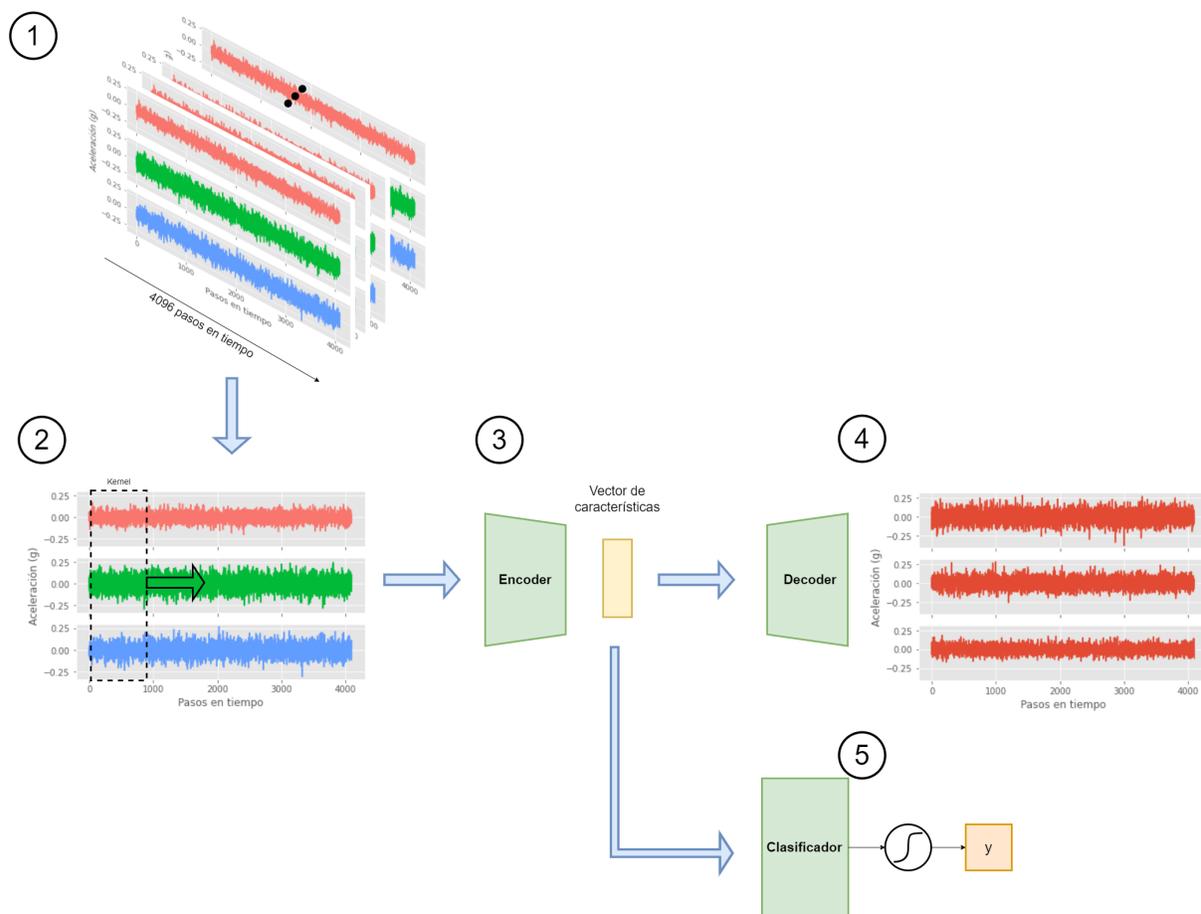


Figura 20: Metodología para la preparación de datos en el dominio del tiempo. En el primer paso los datos de los tres ejes se fraccionan en muestras de 4096 pasos cada una. En el segundo paso, para algunos modelos se aplica una ventana deslizante con un Kernel para convolución. En el tercer paso se reduce la dimensionalidad hasta obtener un vector de características. En el cuarto paso se reconstruye las señales originales. Finalmente, en el quinto paso se usa el mismo vector de características obtenido en el tercer paso y se suministran a una red neuronal como clasificador.

9.4. Modelos de clasificación

Para cumplir el tercer objetivo específico, se construyeron cuatro modelos para clasificación: dos modelos que procesan datos en el dominio de la frecuencia y dos modelos que procesan datos en el dominio del tiempo.

El primer modelo que se construye es una SVM (máquina de soporte vectorial) para analizar los datos en el dominio de la frecuencia resultado de la prueba-F ANOVA. Este modelo se elige principalmente para realizar el ejercicio encontrado en la revisión de literatura, donde hay extensa documentación del uso de SVM para máquinas en general. Como parámetro se elige un kernel lineal y como hiperparámetro el factor de regularización se conserva por defecto igual a la unidad. En la Sección 10.1, con el análisis exploratorio, se encuentran indicios de que el problema de clasificación binaria puede ser resuelto con la separación del hiperplano.

El segundo modelo es un LDA (análisis discriminante lineal) y también recibe como datos de entrada las frecuencias procesadas. En el caso de clasificación binaria, el LDA requiere de un solo discriminante para separar las clases y, por lo tanto, la proyección final es de una dimensión. En este caso, el único parámetro de entrada es la cantidad de características para el entrenamiento del modelo. A diferencia del primer modelo, se encuentra que la proyección LDA se beneficia de una mayor cantidad de características provenientes del paso anterior con la prueba-F de ANOVA, por lo que se aumenta a 3000 características, distinto de las 1000 características mencionadas en la Sección 9.3 sobre la selección de frecuencias.

El tercer modelo corresponde a una red neuronal recurrente. Este modelo recibe como entrada los datos en el dominio del tiempo de los tres ejes, por lo que es multivariante. Primeramente, inicia con capas de convolución 1D de las secuencias originales. La cantidad de filtros para el kernel y su tamaño son respectivamente de 64 y 32 en la primera capa y 32 y 16 en la segunda capa. Estos parámetros se eligen con base en buenas prácticas exploradas por otros autores [44], en donde se indica que los valores deben ser proporcionales a la tasa de adquisición de datos y no menor que 10 cuando Δf es igual a 1 Hz. Se mantienen valores con potencias de dos por consistencia con otros valores elegidos. El paso de la ventana deslizante (en este caso refiriéndose al parámetro conocido como *stride*) se mantiene fijo en cada capa e igual a 2. Esto permite

que, a la salida de cada capa, el nuevo vector resultante se reduzca a la mitad. El vector resultante ingresa a una capa LSTM para resolver el problema de clasificación donde y cuya salida es una capa densa con un solo nodo con función sigmoide para clasificación binaria. Se elige como función de pérdida entropía cruzada binaria por recomendación de la biblioteca usada para este tipo de problemas. Entre los hiperparámetros más destacados, se añade una capa de *dropout* igual a 0,5 que permite regularización y que obliga a la red a solo retener el 50 % de la capa anterior. Este paso tiene como objetivo disminuir la probabilidad de tener un modelo con alta varianza. Se introduce un mecanismo para detener el entrenamiento temprano, por lo que si el aprendizaje no produce una mejoría en la exactitud por al menos 10 iteraciones, entonces se detendrá. Esta estrategia no solo reduce el tiempo de entrenamiento, sino que además se considera como otro mecanismo de regularización [45]. Por último, la tasa de aprendizaje se conserva con el valor por defecto de la biblioteca e igual a 0,001, ya que resulta un paso adecuado para la optimización. En la Figura 21 se muestra esta arquitectura.

El cuarto modelo es similar al tercer modelo, pero en las primeras capas se reemplaza una capa de convolución por una capa LSTM que reduce la salida hasta un vector de tan solo 64 pasos. Las siguientes capas profundas son igualmente LSTM anidados, los cuales mejoran el aprendizaje durante el entrenamiento. Todos los demás parámetros e hiperparámetros se conservan igual al del segundo modelo, exceptuando la tasa de aprendizaje, ya que después de prueba y error, el valor por defecto impide que la optimización alcance el mínimo local, por lo que se elige un valor diez veces menor: 0,0001. En la Figura 22 se muestra esta arquitectura.

9.5. Modelos de detección de anomalías

Para cumplir con el cuarto objetivo específico se entrenaron un total de dos modelos para detección de anomalías. Ambos modelos son autocodificadores con una etapa de reducción de la dimensionalidad hasta un vector de características y una etapa de reconstrucción de la señal.

El primer modelo, en la Figura 23, inicia con dos capas de convolución 1D exactamente igual que el primer modelo de clasificación hasta la capa con el vector de

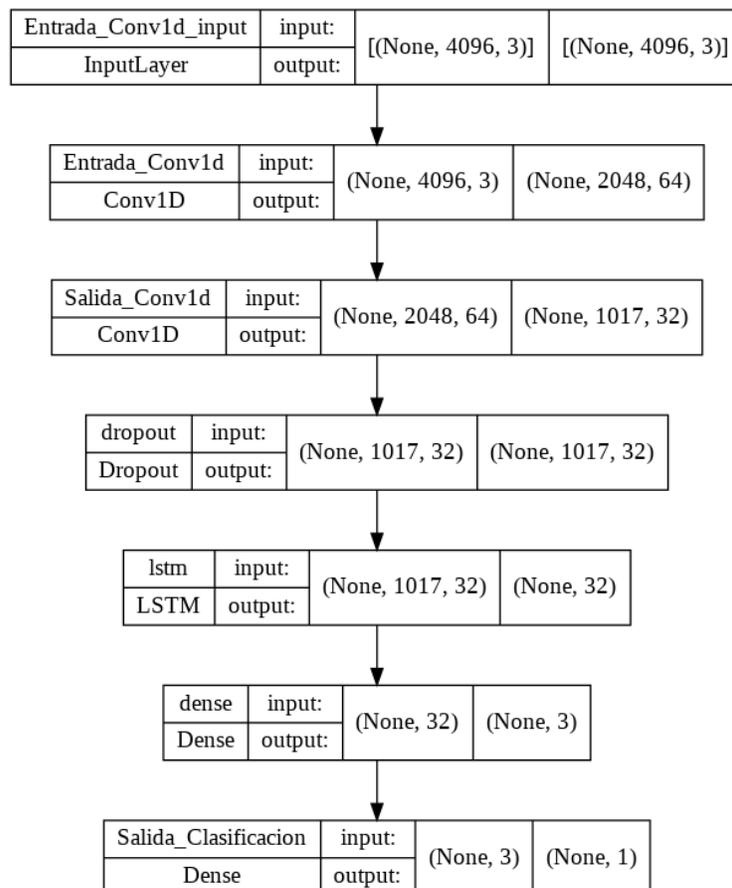


Figura 21: Arquitectura del modelo de clasificación usando convolución 1D. En las primeras capas se procesa por medio de convoluciones y una ventana deslizante. Luego es seguido por una red LSTM que se encarga de resolver el problema de clasificación.

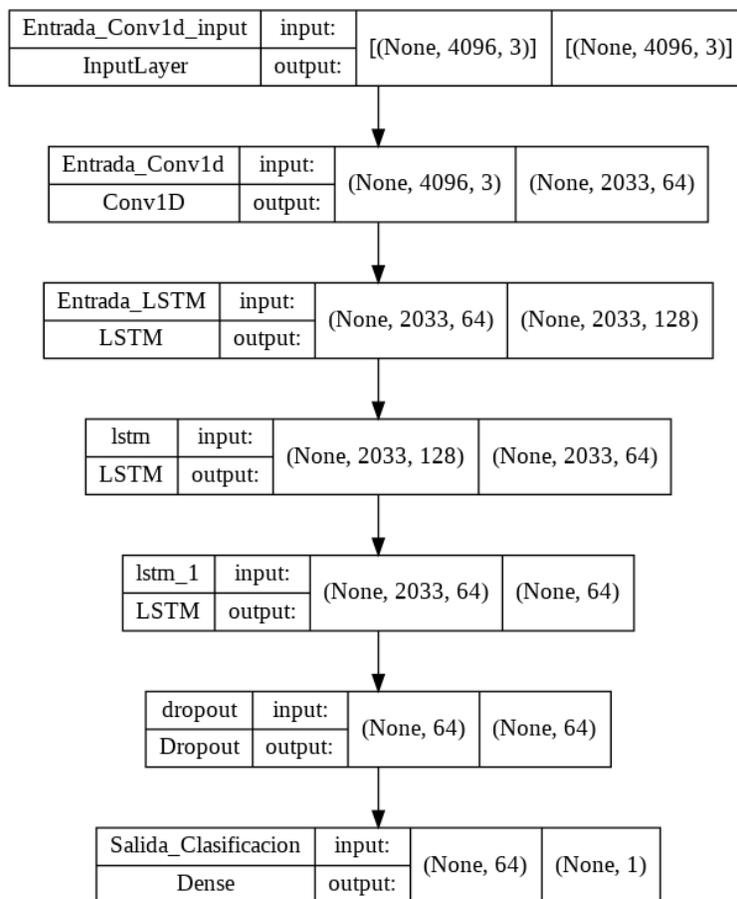


Figura 22: Arquitectura del modelo de clasificación usando convolución 1D y LSTM anidados. En las primeras capas se procesan las secuencias por medio de una convolución 1D y luego una red LSTM anidados hasta culminar en una capa densa para la clasificación.

características. Luego, en la etapa de reconstrucción, se invierten las convoluciones por medio de convoluciones transpuestas 1D hasta llegar al tamaño original de la serie. Los parámetros de kernel y tamaño de kernel se eligen con los mismos criterios antes expuestos. Considerando que este problema no es una de clasificación, si no que de regresión, entonces se elige como función de pérdida el error absoluto medio (MAE, por sus siglas en inglés) y es quien mide el error entre lo que entra y sale del autodecodificador.

El segundo modelo de detección de anomalías se muestra en la Figura 24. Tiene etapas de reducción de la dimensionalidad exclusivamente con redes LSTM anidadas y reduciendo con potencias de dos. Finalizada la salida del codificador, por particularidad de la biblioteca, se requiere de una capa intermedia que una ambos LSTM con una función *RepeatVector*. Esto permite que la entrada del decodificador inicie con una matriz de 3 dimensiones. Luego, la etapa de reconstrucción corresponde nuevamente a redes LSTM anidadas, pero con la cantidad de nodos en orden inverso, lo que da como salida series de tiempo con la misma cantidad de datos que los originales. Nuevamente se elige MAE como función de pérdida. Todos los demás parámetros e hiperparámetros se mantienen como los modelos anteriormente descritos.

Cualquier parámetro o hiperparámetro no explícitamente mencionado en esta sección, se elige su valor por defecto de acuerdo con las bibliotecas de Keras y TensorFlow en sus versiones 2.9.0 y 2.7.0 respectivamente.

9.6. Modelo con simulaciones

Una estrategia recomendada por algunos autores es preparar el conjunto de datos para que reflejen los escenarios que esperamos ver en el futuro y en los que se busca que los modelos vayan a tener un buen desempeño [45]. Por esta razón se trata de simular otros problemas no vistos en nuestra recolección de datos, pero que se describen en la literatura [46] y usando como base los datos normales recolectados en campo. Para simular el problema de desalineamiento, se le agrega a la señal original un componente sinusoidal a exactamente $2\times$ la frecuencia fundamental, lo que resulta en un armónico en esta frecuencia. También se simula el problema de soltura entre distintos elementos, por lo que se añaden otros componentes sinusoidales de $3\times$ a $9\times$

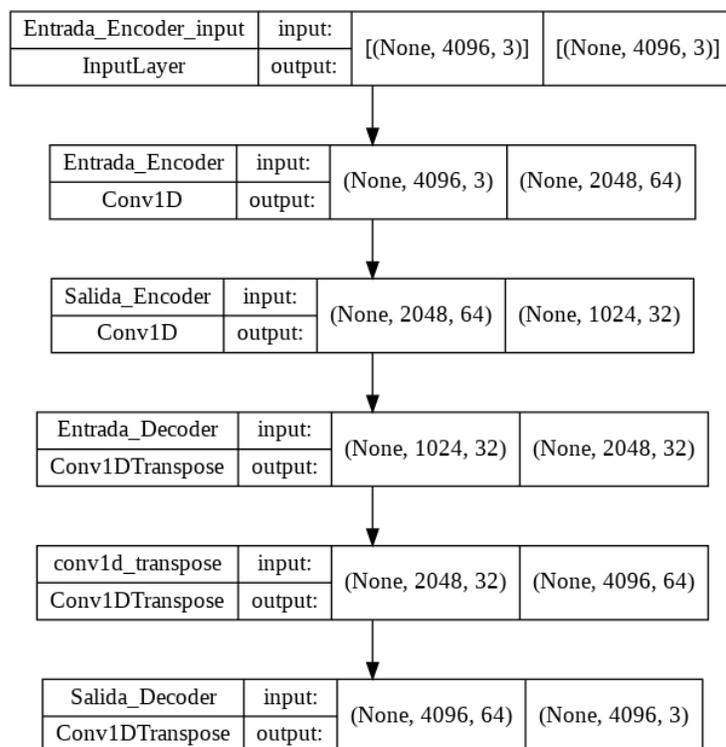


Figura 23: Arquitectura del modelo de detección de anomalías con reducción usando convolución 1D. En las capas del codificador se reduce la dimensionalidad hasta el vector de características. Luego en las capas del decodificación se intenta reproducir las series de tiempo originales con las convoluciones transpuestas.

de la frecuencia fundamental vista en los datos reales. Una vez introducidos estos nuevos problemas al conjunto de datos, el problema deja de ser un problema de clasificación binaria y pasa a ser un problema de clasificación múltiple, por lo que el modelo debe adaptarse para esta situación. Para hacer esta demostración se entrena un único modelo de un autocodificador variacional adaptado de la literatura de la biblioteca [47]. Este modelo, similar a otros autodecodificadores mencionados previamente tiene una etapa de reducción de la dimensionalidad con el codificador y una de reconstrucción con el decodificador. La principal diferencia yace en que a la salida del decodificador se añade una capa compuesta de un elemento de media y otro de varianza, que busca ubicar cada categoría de datos dentro de una distribución normal. Esto resulta conveniente para mostrar el vector de características y poder compararlo con otras reducciones mencionadas en este documento. Por último, este vector de características es usado como entrada para un clasificador como el que muestra la Figura 25. La capa de salida de esta red usa una función *softmax* para estimar la probabilidad de cada una de las cuatro categorías.

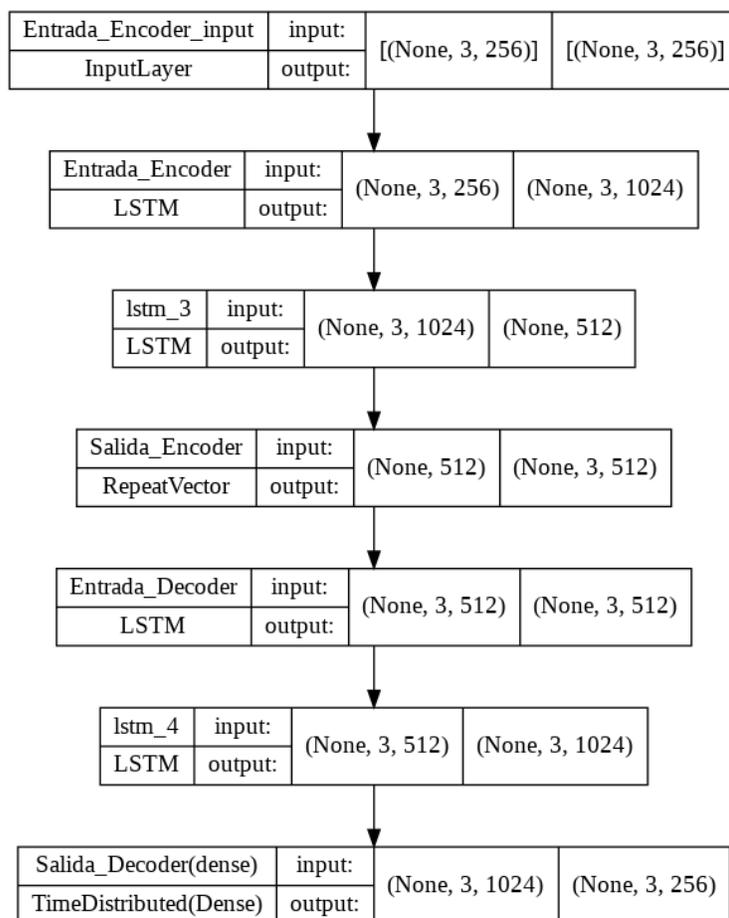


Figura 24: Arquitectura del modelo de detección de anomalías con reducción de la dimensionalidad usando LSTM. En las capas del codificador se reduce la dimensionalidad con redes LSTM anidadas. Luego se une a un decodificador con redes LSTM anidadas que intentan reproducir las series de tiempo originales.

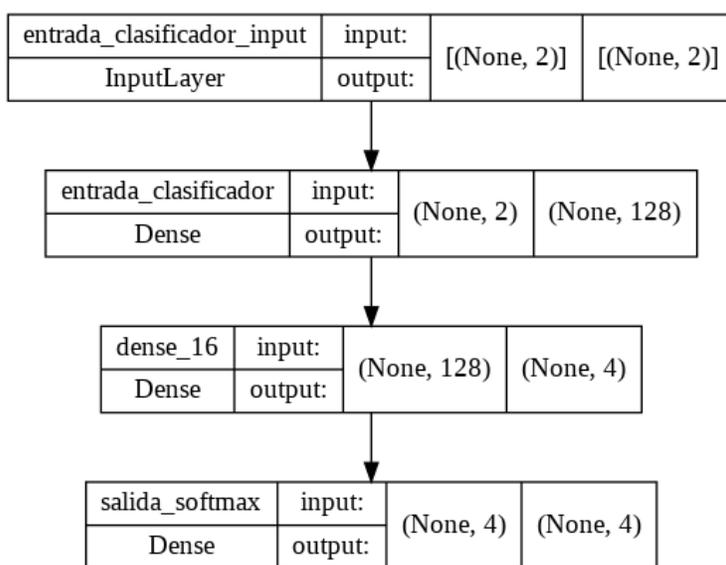


Figura 25: Arquitectura del modelo clasificador para múltiples categorías. Esta red tiene como entrada un vector de características y como salida una capa *softmax* para clasificar cuatro categorías.

9.7. Validación y análisis de error

El quinto y último objetivo específico de este trabajo de investigación requiere de un análisis del error en cada modelo construido. Este análisis permite entender que tan efectiva resulta una hipótesis H en estimar la función f . La métrica más intuitiva es la *exactitud*, donde se mide la relación entre las predicciones que son correctas (verdaderos positivos más verdaderos negativos) y la totalidad de ejemplos. La exactitud se define como

$$\text{Exactitud} = \frac{VP+VN}{VP+FP+VN+FN}, \quad (11)$$

donde VP representa los verdaderos positivos, VN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos. Esta métrica es útil, pero es dependiente de un buen balance entre verdaderos positivos y verdaderos negativos, por lo que, por sí misma podría no ser suficiente para juzgar el verdadero desempeño de un modelo [48]. Por lo tanto, se definen otras dos métricas que ayudarán a evaluar el desempeño de forma más general: *precisión* y *exhaustividad*. Se define la precisión como la relación entre los verdaderos positivos y la cantidad total clasificada como positivo:

$$\text{Precisión} = \frac{VP}{VP+FP}. \quad (12)$$

En palabras más simples, la precisión indica una tasa de cuántos verdaderos positivos en realidad resultaron ser ciertos. Por otro lado, se define exhaustividad como la relación de los clasificados como verdaderos positivos y la cantidad total que en realidad es positiva:

$$\text{Exhaustividad} = \frac{VP}{VP+FN}. \quad (13)$$

En otras palabras, la exhaustividad es un indicador de que tan bueno será un modelo encontrando esos verdaderos positivos. El desempeño de un modelo será mejor cuando mayor y cercano al valor de 1 sean tanto la precisión como la exhaustividad. En la literatura algunos autores combinan estas dos métricas en una sola y es así como se define el *Valor F1*. Solamente conforme la precisión y la exhaustividad se acerquen ambas a la unidad, entonces así lo hará el valor F1. Se define el valor F1 como

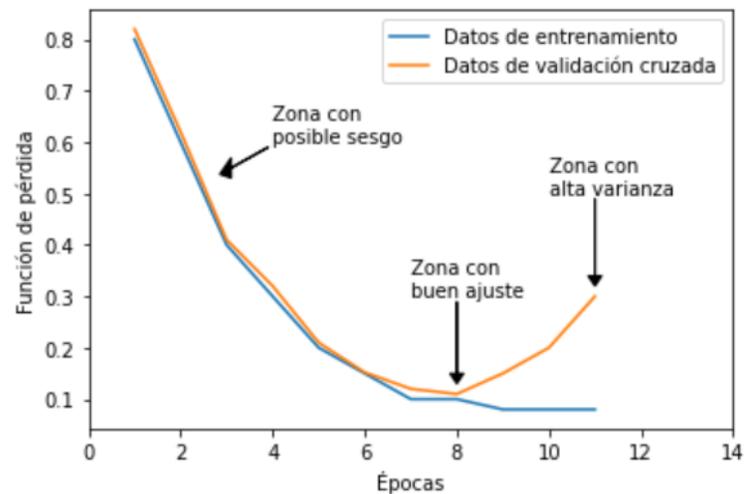


Figura 26: Ejemplo de distintos ajustes usando una curva de aprendizaje.

$$\text{Valor F1} = 2 \times \frac{\text{Precisión} \times \text{Exhaustividad}}{\text{Precisión} + \text{Exhaustividad}} \quad (14)$$

Se calcula la precisión y la exhaustividad para cada modelo de clasificación construido. Asimismo, estas métricas son usadas como ayuda para ajustar los modelos y así encontrar los parámetros e hiperparámetros que permiten resolver el problema. Además, para demostrar que estos modelos implementados no presentan sesgo ni alta varianza, se construyen las curvas de aprendizaje de las funciones de pérdida para los datos de entrenamiento y los datos de validación cruzada. El comportamiento esperado de un modelo sin sesgo, sin alta varianza y por ende con buen ajuste es aquel donde ambas curvas de aprendizaje decrecen, donde en cada iteración el algoritmo se acerca a un valor mínimo local y en donde una curva sigue de cerca a la otra. Finalizar este entrenamiento tempranamente podría generar sesgo en el modelo. Por otro lado, sobreentrenar el modelo con muchas iteraciones podría inducir a un sobreajuste, lo que se vería reflejado en una separación entre ambas curvas (usualmente siendo la curva de validación cruzada la que crece). La Figura 26 muestra un ejemplo de estos escenarios.

Por último, en el caso de los detectores de anomalías, se estimó su umbral de sensibilidad usando la distribución del error esperado en estado normal. El objetivo fue identificar cuál es el porcentaje de error mínimo que pueden detectar sin generar falsos positivos. En la siguiente sección se discuten los resultados de esta metodología

propuesta.

10. Resultados y discusión

A continuación se presentan los resultados. Se inicia con un análisis exploratorio de los datos para entender la naturaleza del problema. Seguidamente se muestran los resultados obtenidos en los modelos de clasificación, empezando por sus curvas de entrenamiento y seguido por un cuadro resumen con las métricas propuestas para el análisis del error. Luego con los modelos de detección de anomalías también se muestran las curvas de entrenamiento, seguidos por la distribución del error esperado para cada modelo y, por último, una curva con cada detector de anomalías sometido a datos reales de cavitación.

10.1. Análisis exploratorio de datos

Los datos crudos corresponden a series de tiempo de las vibraciones en tres coordenadas distintas (x , y , z). Debido a que la tasa de adquisición es de 8192 Hz, existen exactamente 8192 puntos cada segundo. En la Figura 27 se muestra un ejemplo de los datos. Se recolectaron datos tanto para el estado normal como para el de cavitación. En la Figura 28 se comparan estas dos series de tiempo, una seguida de la otra y se evidencia que tienen un comportamiento similar, con la única diferencia cuando la bomba se encuentra en cavitación, donde sus vibraciones parecen aumentar en amplitud. Sin embargo, esto no es criterio suficiente como para clasificar el estado de la bomba. Por esta razón, se trasladan estas señales al dominio de la frecuencia usando el algoritmo FFT. Esta conversión da como resultado un espectro con los componentes sinusoidales distribuidos en 4096 frecuencias distintas, tanto para señales correspondientes al estado normal, como para señales correspondientes al estado de cavitación.

La Figura 29 muestra un ejemplo del resultado de la transformación por medio de FFT. Como se evidencia en dicha figura, la mayor parte de la información se encuentra dentro de un ámbito entre los 0-800 Hz aproximadamente, por lo que se expande el gráfico en esta zona en la Figura 30, para los datos en estado normal y con el fin de

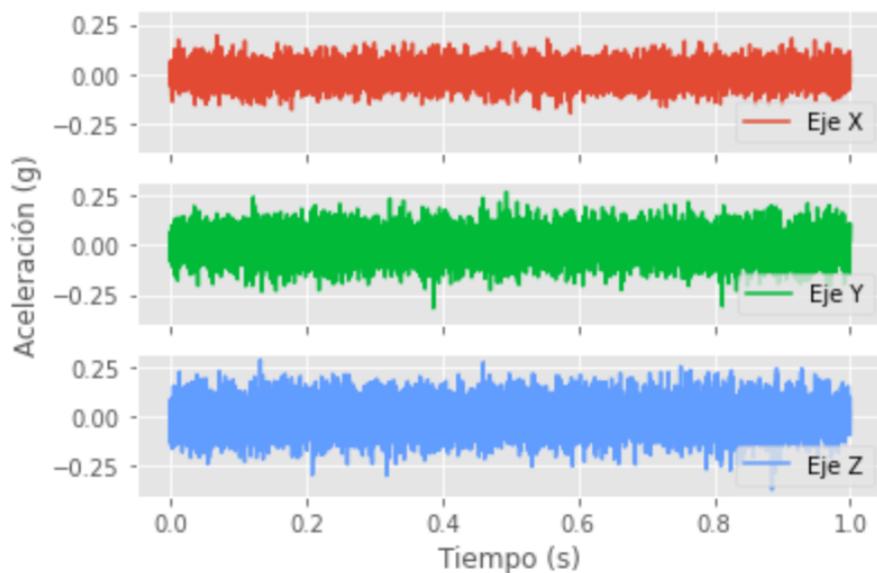


Figura 27: Un segundo de datos de vibraciones en las tres coordenadas. La unidad de vibración usada en este trabajo corresponde a “g”, por la fuerza gravitacional y es equivalente a $9,8 \text{ m/s}^2$.

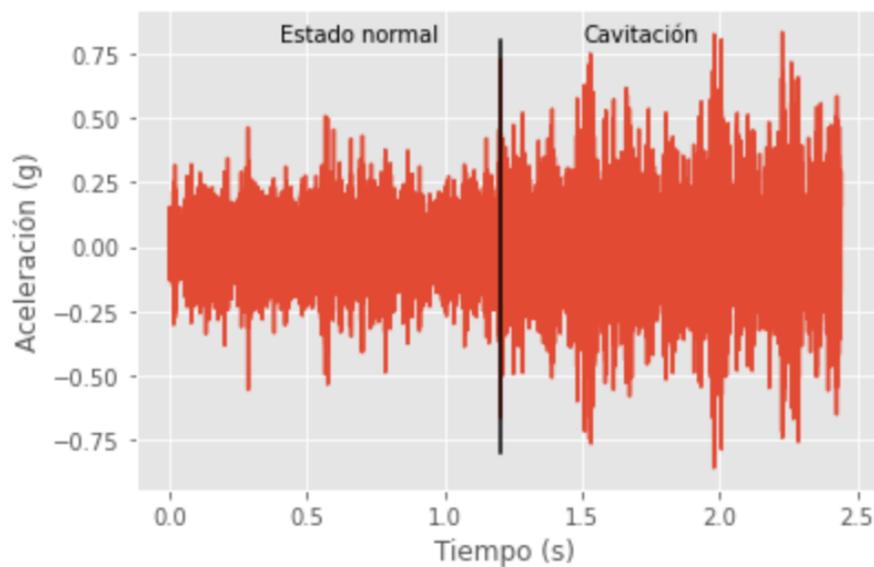


Figura 28: Se comparan las señales normales y las de cavitación usando los datos de una única coordenada. La única diferencia notoria a simple vista es la magnitud de la amplitud de las vibraciones.

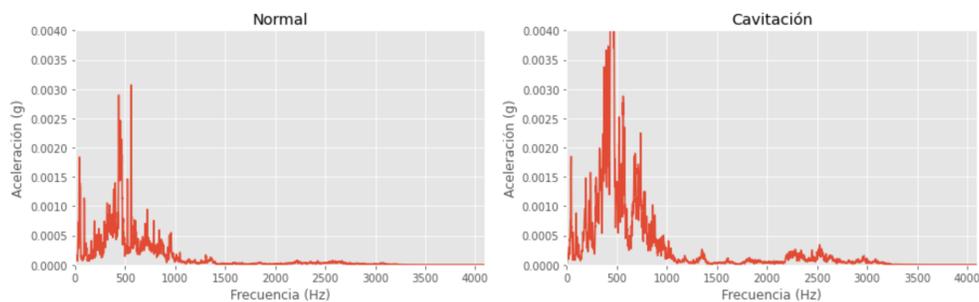


Figura 29: Comparación del espectro normal y del de cavitación. Ambos espectros siguen una tendencia similar, pero hay diferencias en las magnitudes en distintos ámbitos de frecuencia.

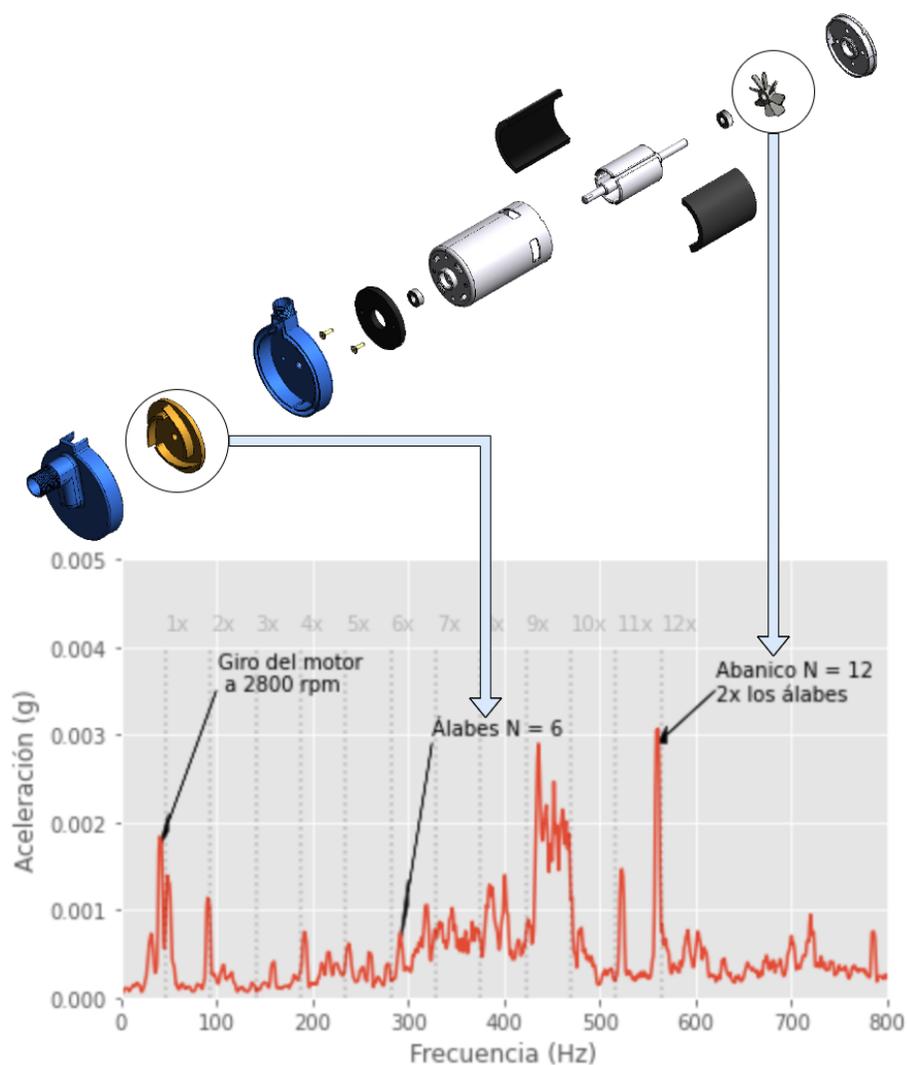


Figura 30: Frecuencia fundamental y armónicos esperados. En la parte posterior se muestra un isométrico de ensamble con los elementos más usuales que componen una bomba. El armónico a 6 veces la frecuencia fundamental corresponde a la cantidad de álabes del impulsor, mientras que el armónico a 12 veces coincide con el abanico del motor eléctrico. Imagen CAD de uso libre obtenida de <https://grabcad.com/library/self-priming-pump-1> y adaptada para propósitos ilustrativos.

visualizar mejor los elementos del espectro que se consideran esperados. Se define el primer pico de interés como la frecuencia fundamental de este sistema, se ubica a 47 Hz y coincide con la velocidad de giro del motor a 2800 rev/min. Los elementos subsecuentes son los armónicos, los cuales son múltiplos de esta frecuencia fundamental y por conveniencia se señalan con las etiquetas de 2x, 3x, 4x, etc. Algunos armónicos tienen su explicación en el elemento mecánico que las origina, por ejemplo, en este caso el impulsor de la bomba tiene 6 álabes y da como resultado un armónico en 6x. Lo mismo ocurre con el abanico del motor eléctrico, que tiene 12 álabes y da como resultado otro armónico, pero en 12x. La teoría muestra que a veces se presenta de igual forma armónicos a 2 veces la cantidad de álabes del impulsor, los cuales también pueden coincidir con una frecuencia natural del sistema, lo que podría explicar también la magnitud del mismo en este caso [46]. Puesto que ninguna toma de datos es perfecta, es de esperar que aparezcan elementos no esperados en las señales, los cuales se consideran factores de ruido en este ejercicio. En la Figura 31 se señalan los armónicos que son indicios de otros problemas no relacionados con la cavitación y que hay que tomar en consideración en este ejercicio. La mayoría de los armónicos (3.º, 4.º, 5.º, 7.º, 8.º, 9.º, 10.º y 11.º) se pueden explicar por una soltura entre los elementos mecánicos, los cuales no se encuentran perfectamente unidos o amortiguados y vibran en proporción con la velocidad de giro durante la operación de la máquina. Otro elemento de interés es el ruido generalizado a lo largo del espectro, el cual podría ser normal durante la conversión análogo-digital electrónica, pero principalmente resulta interesante el ruido localizado entre los armónicos 9.º y 10.º. Se considera que el análisis de frecuencias naturales y resonancia está fuera del alcance de este trabajo de investigación, pero es importante mencionar que algunos de los aumentos en el ruido, a raíz de las vibraciones, podrían estar explicados por las distintas frecuencias naturales que posea este sistema.

Por otro lado, cuando analizamos el espectro con cavitación en la Figura 32, se evidencia un aumento generalizado del ruido a lo largo de la mayoría de las frecuencias hasta los 800 Hz. Esto da como resultado que incluso algunos armónicos que con anterioridad eran claros (por ejemplo 11.º y 12.º) desaparezcan entre el ruido. Como prueba del aumento del ruido, se estima que la relación de señal a ruido decrece de 1,4 a 1,2 para el ámbito de frecuencias de 250-500 Hz. Es importante mencionar que la

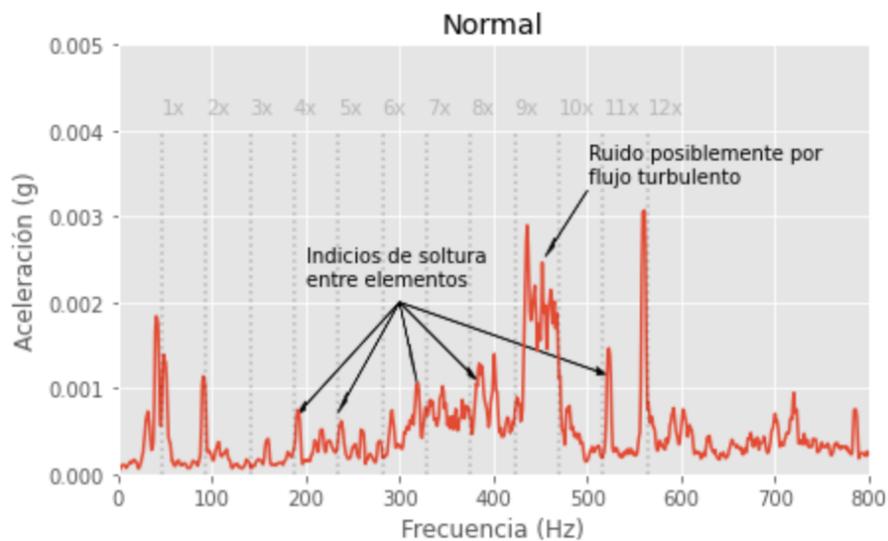


Figura 31: Un factor de ruido es la soldadura que existe entre los elementos mecánicos, los cuales también vibran y generan picos en el espectro. Otro factor de ruido es la posible presencia de frecuencias naturales que intensifican las magnitudes por resonancia. Asimismo, no se descarta que la turbulencia del flujo del agua ocasione parcialmente ruido localizado.

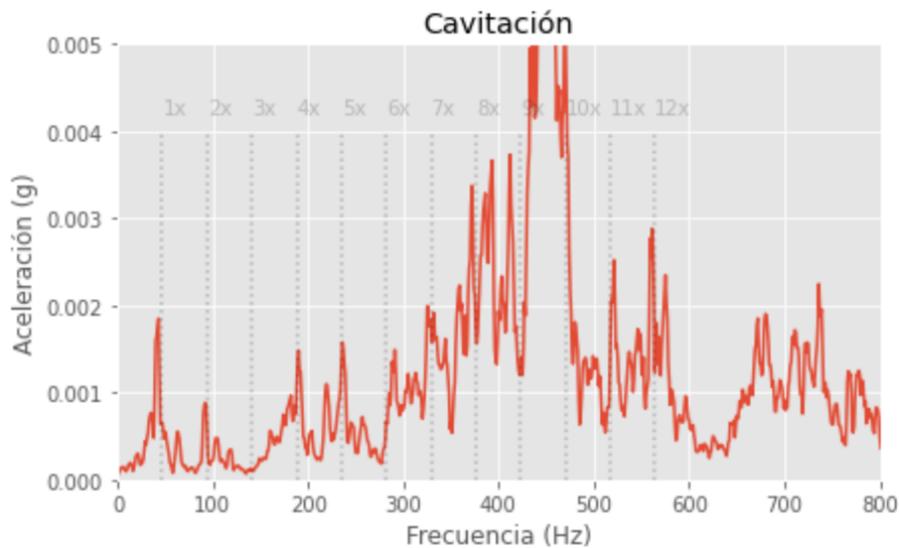


Figura 32: Se evidencia un aumento del ruido aleatorio a lo largo de la mayoría de frecuencias. Muchos de los armónicos que eran visibles en estado normal, ahora se mezclan con el ruido. La información a baja frecuencia, como la frecuencia fundamental y 2.º se mantienen casi que inalterados. El aumento más significativo del ruido se ubica entre 9.º y 10.º armónico.

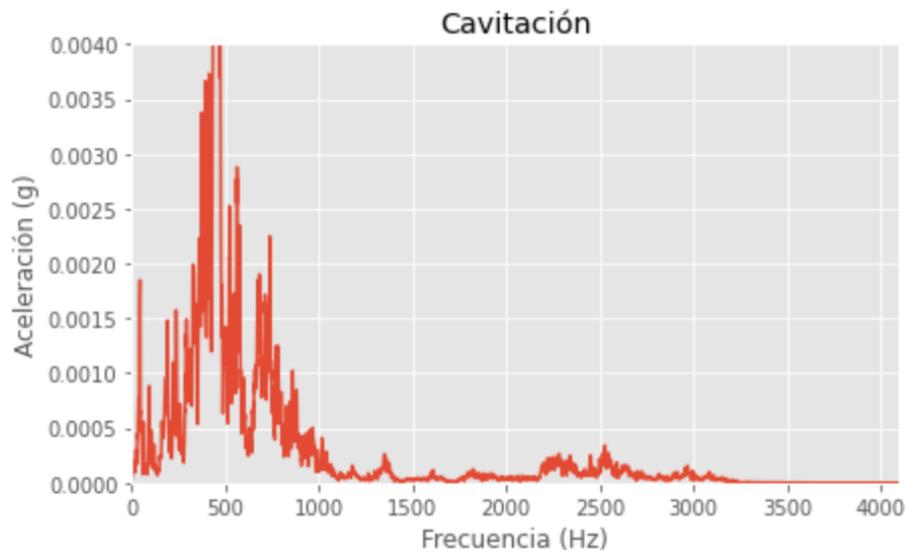


Figura 33: Hay evidencia de ruido aleatorio en altas frecuencias, entre 2000 Hz y 3200 Hz. Se asocia estas frecuencias con el fenómeno de cavitación, ya que se solo se visualizan claramente bajo esta condición. De igual forma, no se descarta que este ámbito de frecuencias no se esté amplificando por la presencia de alguna frecuencia natural y resonancia.

frecuencia fundamental permanece muy similar al ejemplo con el espectro normal, lo cual es esperado, ya que es un elemento sinusoidal que está presente en todo momento y además que el fenómeno de la cavitación se espera en ámbitos de frecuencia usualmente más altos. Nuevamente, después de ampliar todo el espectro de frecuencias como se muestra en la Figura 33, se puede apreciar también un aumento de ruido aleatorio a lo largo del ámbito de frecuencias entre los 2000-3200 Hz aproximadamente. Este patrón también se considera como un indicio de la presencia del fenómeno de cavitación. Las diferencias vistas en los espectros representan información útil para ser usadas como características en modelos de aprendizaje automático.

En la siguiente sección se usa la información vista en esta sección y la metodología propuesta para entrenar modelos de aprendizaje supervisado y no supervisado.

10.2. Modelos de clasificación

Con el modelo SVM se reduce la dimensionalidad por medio de PCA, como se muestra en la Figura 34, lo que permite que clasificar ambas categorías por medio de un hiperplano sea una tarea sencilla. Sin embargo, una consideración importante es la que se muestra en la Figura 35, donde se grafica el porcentaje de retención de la información original en función de los distintos tamaños en los que se ha reducido la

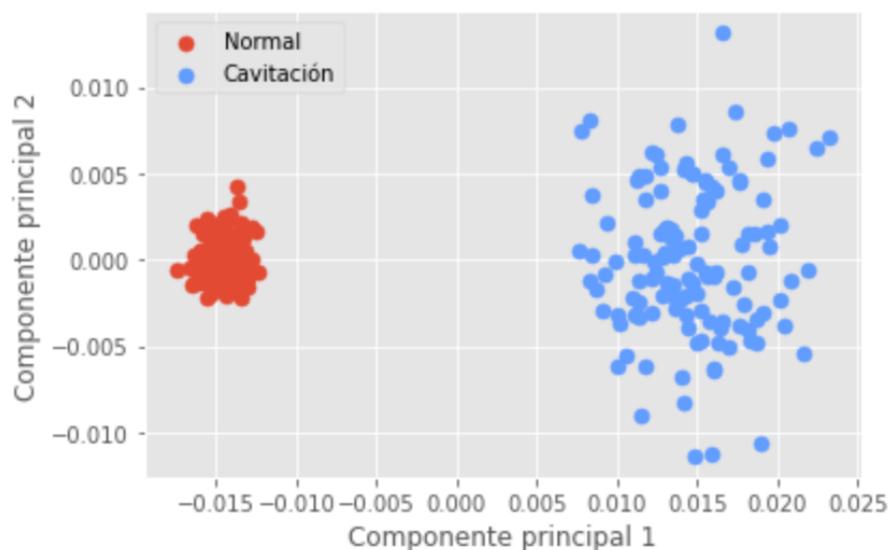


Figura 34: Reducción de la dimensionalidad por medio de PCA. Con dos componentes principales es fácilmente diferenciable el estado normal con respecto al estado de cavitación. Se aprecia además la aleatoriedad de la cavitación.

dimensionalidad. Aquí se evidencia que el mínimo al que se puede reducir la dimensionalidad de las características de entrenamiento, con el fin de mantener al menos un 99 % de la información original, es con 63 dimensiones.

El modelo LDA es un algoritmo de aprendizaje automático que maximiza la distancia entre las medias de las categorías. Esto lo hace mediante una proyección a una dimensión reducida. Como se mencionó con anterioridad, en el caso binario, solo se requiere de una dimensión para separar una clase de la otra. Con el fin de visualizar este concepto mejor, la Figura 36 muestra la distribución de las muestras en cada categoría. Es importante aclarar que este ejercicio es posible ya se dispone de un buen balance de datos, con una buena representación del estado normal y de cavitación. De lo contrario, al ser LDA un modelo supervisado, podría generarse un sesgo. Por otro lado, se encontró que reducir en gran medida la selección de características para el entrenamiento del modelo no beneficia al modelo. En la Figura 37 se muestra que a menos de 3000 frecuencias de entrenamiento, el desempeño del modelo cae y su tendencia se torna incluso impredecible.

El modelo de clasificación con convolución 1D y ventana deslizante muestra un entrenamiento esperado donde la función de pérdida avanza hacia un mínimo local, como se muestra en la Figura 38. El detenimiento prematuro hace que el entrenamiento

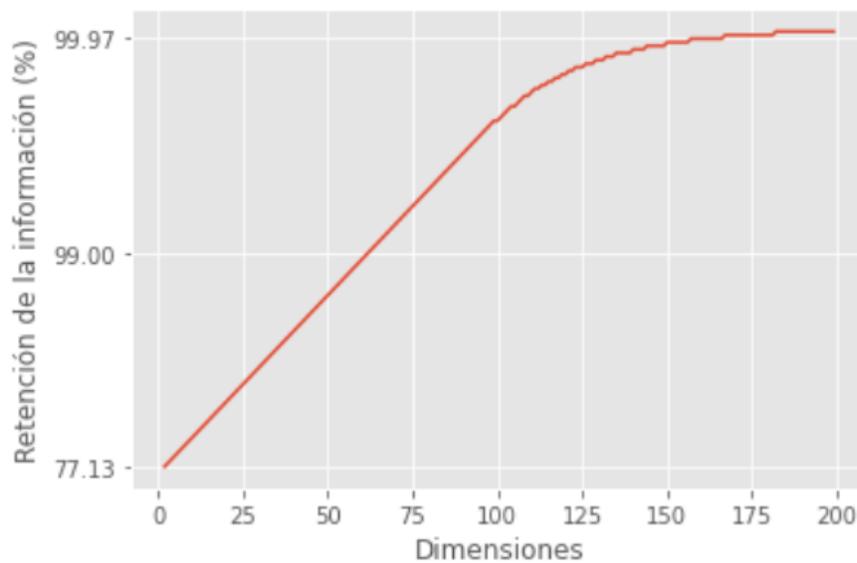


Figura 35: Retención de la información a distintas reducciones. El objetivo es mantener al menos un 99 % de la información original.

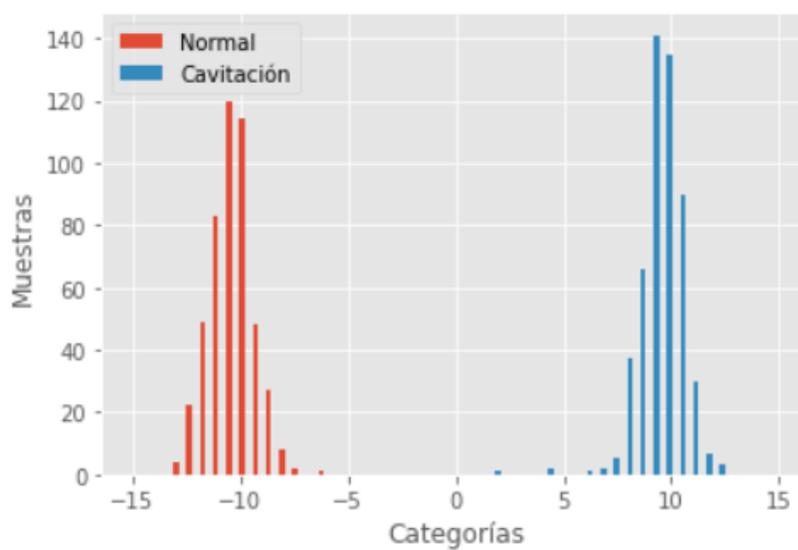


Figura 36: Reducción de la dimensionalidad y clasificación por medio de LDA. Distribución de las muestras de entrenamiento y su clasificación. El algoritmo de LDA maximiza la distancia entre las medias de cada categoría.

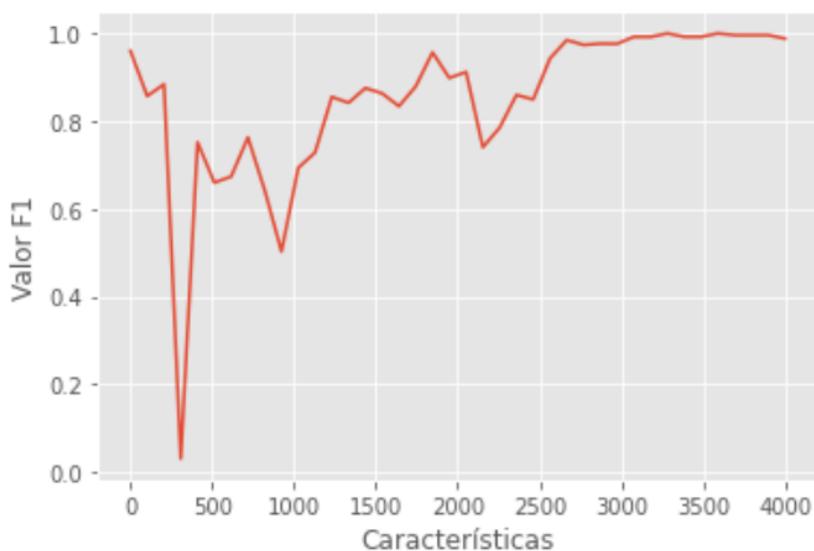


Figura 37: Relación entre el desempeño del modelo LDA y la cantidad de características de entrada. Con menos de 3000 características la tendencia del desempeño es impredecible.

finalice en tan solo la *época*¹ 14. Por otro lado, el modelo con la arquitectura que adiciona capas LSTM se muestra en la Figura 39. Este entrenamiento también muestra un avance hacia un mínimo local con avance lento en las primeras iteraciones solamente. La función de detenimiento prematuro hace que el entrenamiento concluya con en la época 23, donde ya las métricas no muestran cambios significativos. En ambos casos, la similitud y cercanía de la curva de entrenamiento y validación cruzada evidencian un buen ajuste y generalización del modelo.

Finalmente, los resultados de clasificar con el conjunto de pruebas se muestra en el Cuadro 5. La cantidad de muestras de prueba varía entre los modelos debido al tamaño de la ventana de datos usada para fraccionar los datos, donde particularmente la SVM y el LDA se vieron beneficiados de tener una ventana más pequeña y por lo tanto una mayor cantidad de datos de entrenamiento. Todos los modelos son capaces de clasificar correctamente todos los ejemplos de prueba: ya sea catalogarlos como cavitación (verdadero positivo) o como estado normal (verdadero negativo). Este resultado no es de sorprenderse, ya que como se evidenció durante el análisis exploratorio de datos y la reducción de la dimensionalidad, hay información suficiente para que el preprocesamiento de los datos y la estadística ayuden a separar las

¹Una *época* es una iteración y representa un paso completo del conjunto de datos de entrenamiento por el algoritmo.

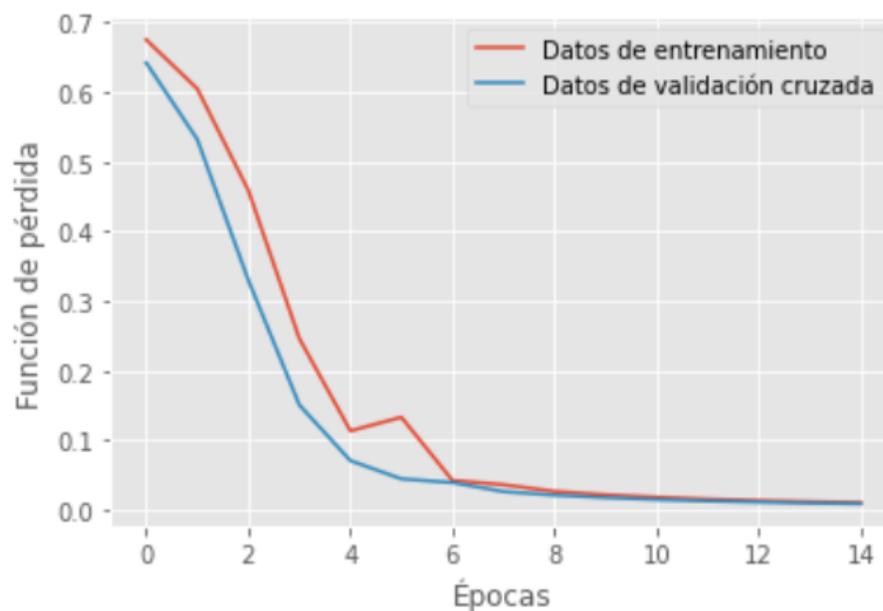


Figura 38: Entrenamiento del modelo de clasificación con convolución 1D. Conforme avanza el entrenamiento el algoritmo avanza en busca de un mínimo local. Estas curvas permiten entender si el modelo sufre de alta varianza. La alta varianza ocurre cuando la validación cruzada y el entrenamiento difieren y se alejan uno del otro. El entrenamiento se detiene cuando el algoritmo no detecta mejoría en la optimización.

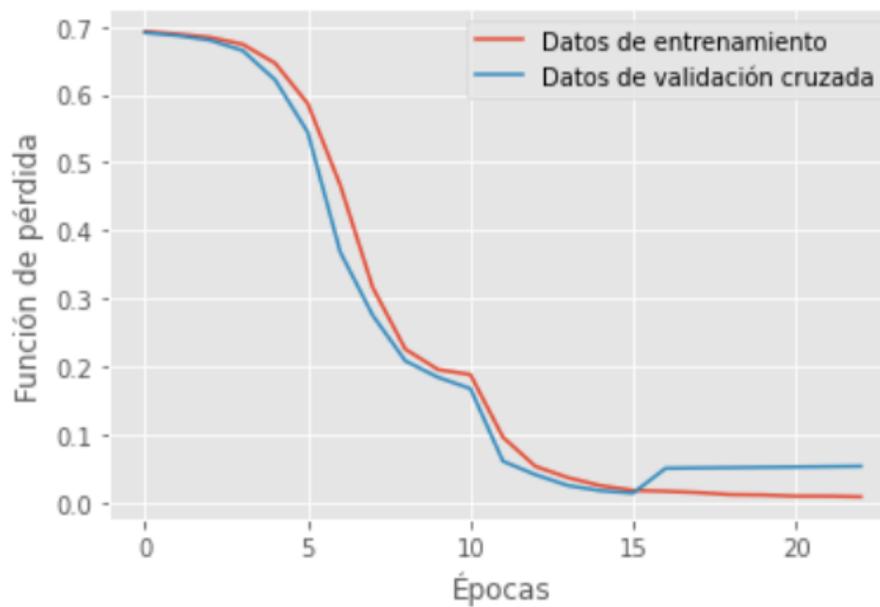


Figura 39: Entrenamiento del modelo de clasificación con convolución 1D y LSTM.

Cuadro 5: Resultados con modelos de clasificación binaria: estado normal y estado de cavitación

Modelo	Dominio	Muestras	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos negativos	Precisión	Exhaustividad
SVM	Frecuencia	250	52,0 %	48,0 %	0 %	0 %	100 %	100 %
LDA	Frecuencia	250	52,0 %	48,0 %	0 %	0 %	100 %	100 %
Conv1D	Tiempo	103	50,5 %	49,5 %	0 %	0 %	100 %	100 %
Conv1DLSTM	Tiempo	103	50,5 %	49,5 %	0 %	0 %	100 %	100 %

categorías. De igual forma, procesar las series de tiempo con convoluciones 1D permite extraer información de distintas perspectivas, lo que facilita también su clasificación, tal y como lo indica la teoría. A raíz de estos resultados, la tarea de clasificación binaria con cavitación recae en mayor medida en el correcto procesamiento de la señal y de una adecuada selección de frecuencias, ya sea con métodos estadísticos o con otras herramientas. Por lo tanto, el ejercicio sugiere que no es necesario recurrir a modelos complejos y potencialmente costosos, como lo son las redes neuronales, para diagnosticar que una bomba centrífuga se encuentra cavitando.

10.3. Modelos de detección de anomalías

En los modelos de detección de anomalías se muestra primero el modelo con convolución 1D. La Figura 40 muestra en entrenamiento con la función de pérdida MAE y evidencia un decrecimiento en busca de un mínimo local. Este mínimo se alcanza aproximadamente en la época 60. Nuevamente, la similitud entre ambas curvas evidencia un buen ajuste del modelo y por lo tanto generalización. Por otro lado, cuando se grafica la distribución del error como se aprecia en la Figura 41, se muestra que el error esperado entre la entrada y la señal reconstruida es menor de 0,005 o, en otras palabras, 0,5 %. Por último, la Figura 42 muestra el escenario del detector de anomalías ante la presencia de cavitación. En aumento abrupto del error MAE marca el inicio de la cavitación, donde muy fácilmente supera el umbral del 1 % potencialmente sin riesgo de falsas alarmas.

El siguiente modelo para detección de anomalías es el LSTM. En la Figura 43 se muestra como similar a todos los otros modelos vistos la función de pérdida decrece, sin embargo conforme avanza el entrenamiento, la validación cruzada tiende a alejarse y subir levemente con respecto al entrenamiento. Esto quiere decir, que de continuar

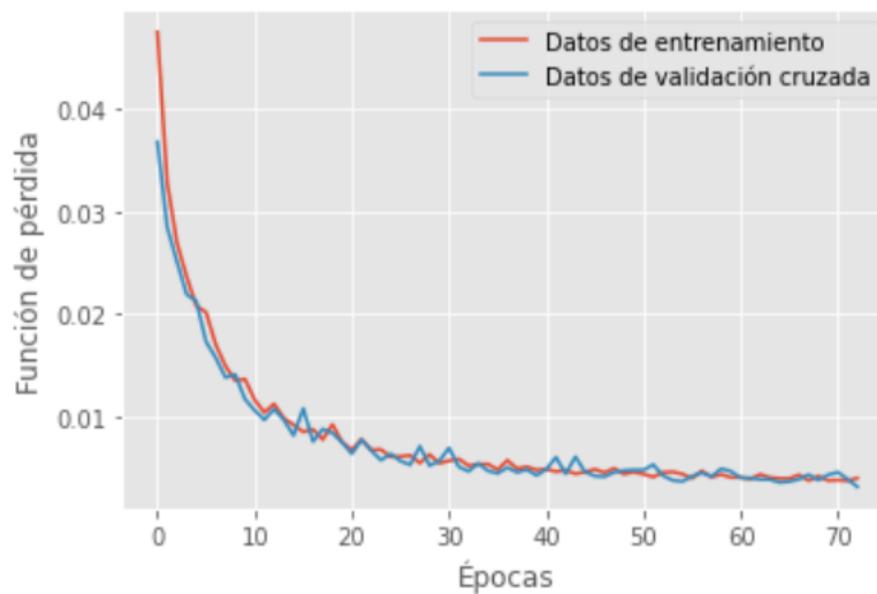


Figura 40: Entrenamiento del modelo de detección de anomalías con convolución 1D.

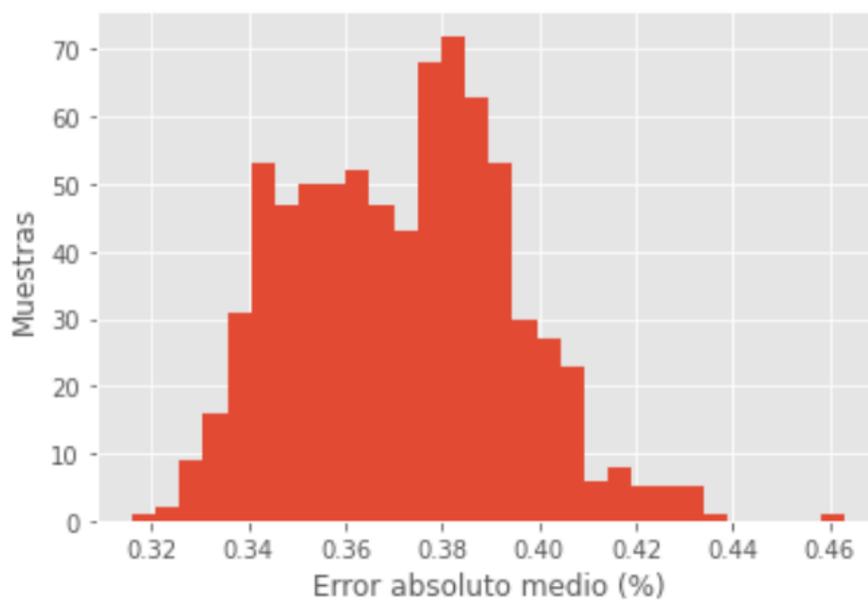


Figura 41: Distribución del error esperado con el modelo con convolución 1D. La distribución muestra que el error (MAE) es menor a 0,5%

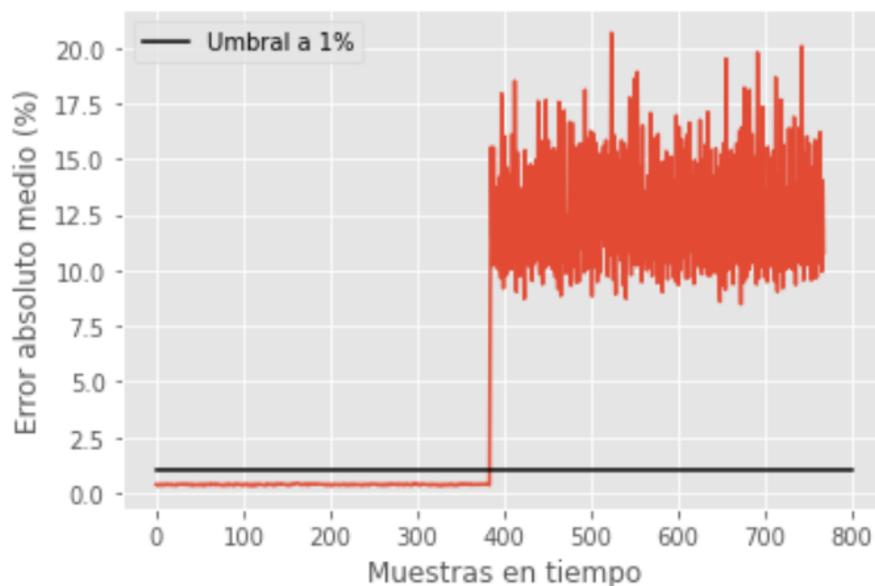


Figura 42: Detector de anomalías con convolución 1D detectando cavitación. Se agrega un umbral a 1% para ejemplificar que por encima de este valor las anomalías son detectables.

entrenando este modelo por muchas más iteraciones, se podría caer en un caso de alta varianza y poca generalidad. Potencialmente, este modelo se podría ver beneficiado aún más de un detenimiento prematuro más agresivo o alguna técnica adicional de regularización. En la Figura 44 la distribución del error esperado está por debajo de 0,06 o 6% y sigue una distribución normal con sesgo a valores aún más pequeños, en su mayoría cercanos al 1% aproximadamente. De igual forma, en la Figura 45 se ilustra como se comportaría el detector de anomalías ante un escenario con cavitación. Nuevamente existe un incremento abrupto del error MAE, sin embargo habría que elevar el umbral hasta un 10% o se correría riesgo de algunas falsas alarmas. En este sentido, parece que iniciar el autocodificador con una capa de convolución 1D ayuda a ajustar la sensibilidad de estos modelos.

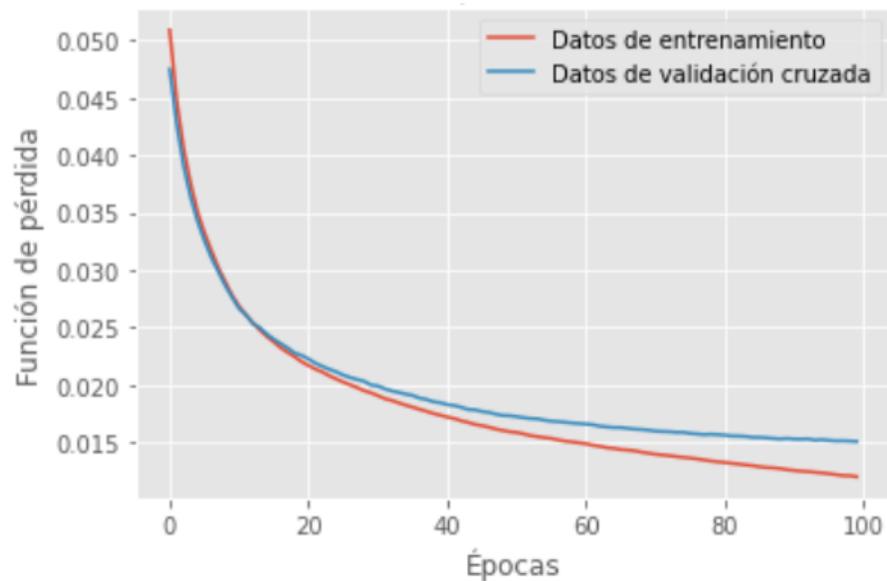


Figura 43: Entrenamiento del modelo de detección de anomalías con LSTM.

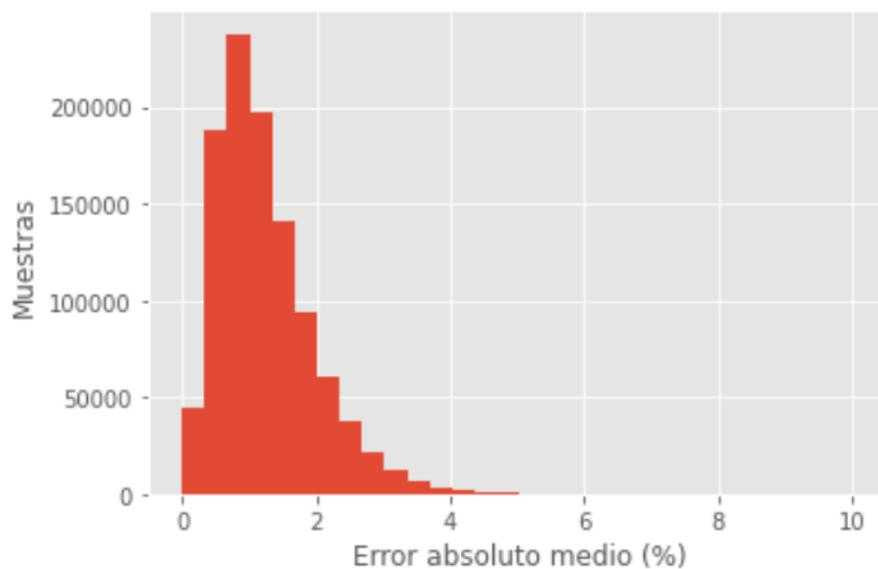


Figura 44: Distribución del error esperado con el modelo con LSTM. La distribución muestra que el error (MAE) es menor a 6%.

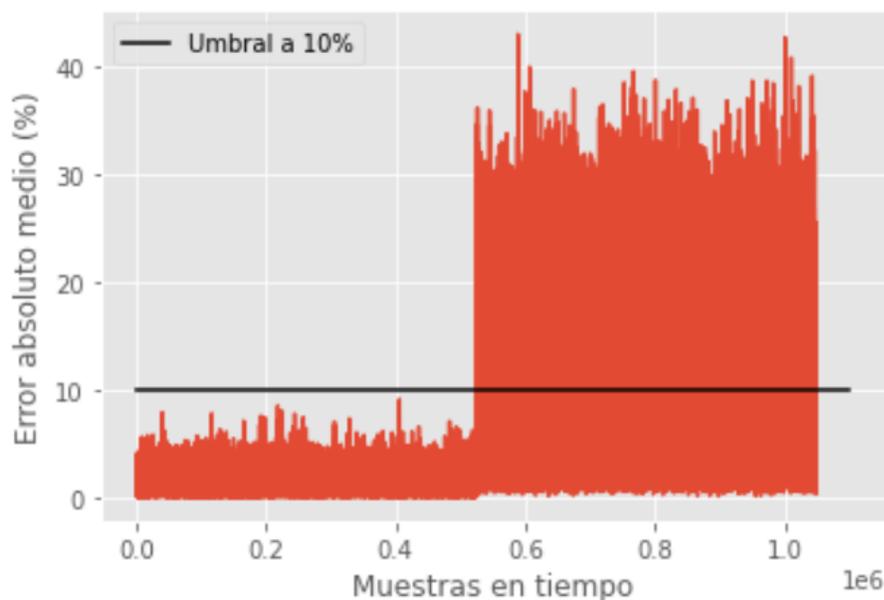


Figura 45: Detector de anomalías con LSTM detectando cavitación. Se agrega un umbral a 10% para ejemplificar que por encima de este valor las anomalías son detectables.

10.4. Problemas adicionales simulados

Como se mencionó en la Sección 9.6 y en preparación del trabajo futuro, es importante entender como se comportaría el abordaje planteado frente a un escenario más real de la industria, donde podrían existir más problemas que solo la cavitación. La Figura 46 muestra un ejemplo de problemas adicionales simulados en el dominio del tiempo. Recordando el proceso descrito en la Figura 19, los datos se convierten al dominio de la frecuencia, luego se someten a un proceso de selección con prueba-F ANOVA y estos vectores resultantes son los usados en el entrenamiento. Si volvemos a reducir la dimensionalidad por medio de PCA, el escenario se vuelve más complejo. Los espectros de algunos problemas podrían tener mucha similitud, siendo la diferencia la presencia o ausencia de algunos armónicos únicamente. La Figura 47 muestra un ejemplo de esto, donde PCA no puede separar linealmente por medio de proyecciones únicamente. Alternativamente, en este caso podría usarse LDA, que sigue siendo una herramienta lineal, pero que al ser supervisada puede maximizar la separación entre los clústeres de las categorías. Otra opción es usar una reducción no lineal. El algoritmo T-SNE permite justamente visualizar el comportamiento que se podría conseguir por medio de una reducción no lineal usando este mismo conjunto de datos. Se construye el autocodificador, se reduce la dimensionalidad no linealmente

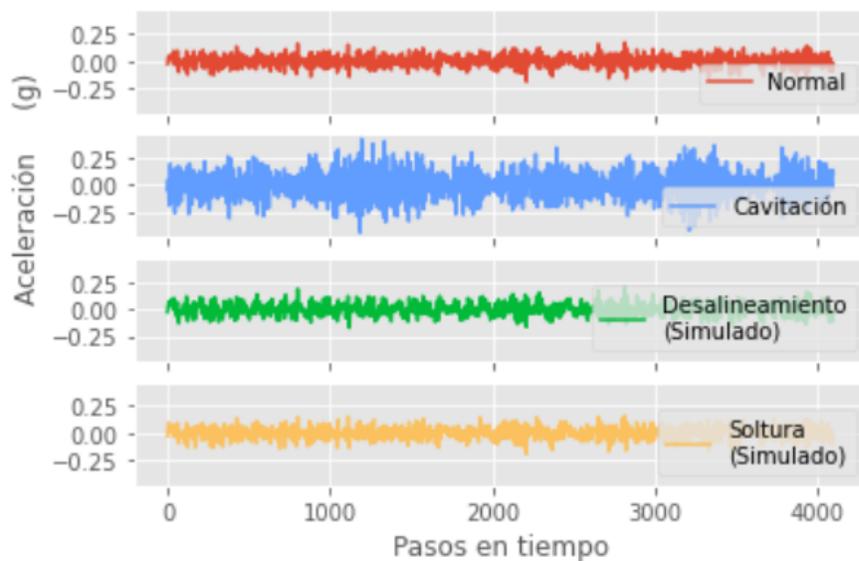


Figura 46: Problemas adicionales simulados. La imagen muestra cuatro series de tiempo. Las primeras dos son los datos reales usados en otras secciones de este trabajo. Las últimas dos son datos sintéticos generados a partir de los considerados normales.

y se confirma esta intuición provista por T-SNE, tal y como se aprecia en la figura. Por último, se someten los datos obtenidos del vector de características de este autocodificador a prueba. Los resultados de la Figura 48 son prometedores para el caso de clasificación múltiple, donde se presenta un ejemplo de como el LDA y el autocodificador permitieron reconocer la presencia de armónicos, y por lo tanto, potencialmente podrían contribuir a clasificar problemas adicionales en una bomba hidráulica.

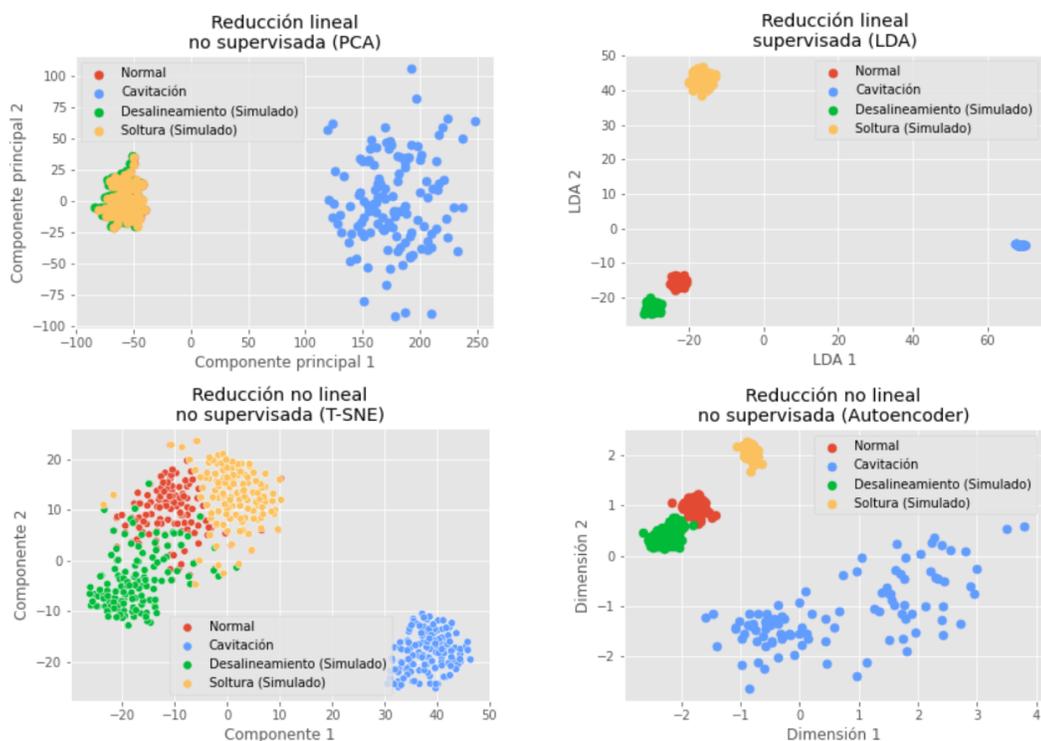


Figura 47: Comparación de distintos abordajes para la reducción de la dimensionalidad. Todos los gráficos de dispersión se generaron a partir de los mismos datos, pero la reducción de la dimensionalidad se hace con técnicas distintas. El primer gráfico es una reducción lineal por medio de PCA. El segundo gráfico es una reducción lineal supervisada por medio de LDA. El tercer gráfico es una visualización con reducción no lineal por medio de T-SNE. El cuarto gráfico es una reducción no lineal por medio de un autocodificador variacional.



Figura 48: La matriz de confusión en clasificación múltiple es cuadrada con N igual a la cantidad de categorías. Entre más el mapa de calor se aproxime a la matriz identidad, mejor es el desempeño del modelo, ya que es cuando las categorías coinciden con las predicciones. La categoría 0 es normal, 1 es cavitación, 2 es desalineamiento y 3 soldadura.

11. Conclusiones

Se construyeron modelos capaces de identificar cavitación y detectar anomalías en una bomba hidráulica. Algunos modelos abordan el problema usando características en el dominio del tiempo y otros en el dominio de la frecuencia. A pesar de que los modelos muestran buenos valores de precisión y exhaustividad durante las pruebas, estos tienen limitantes que hay que aclarar. Primeramente, el entrenamiento de los modelos está estrechamente relacionado con el tipo de bomba usado durante el experimento, con la velocidad de giro y la frecuencia fundamental del sistema. Por lo tanto, los modelos construidos son solo generalizables para bombas centrífugas de una sola descarga que se encuentran operando con la misma velocidad de giro de 2800 rev/min y con características similares a las descritas en la Sección 9 con la metodología. Segundo, el análisis estadístico de ANOVA podría perder relevancia conforme se aumenta el problema a clasificación de múltiples categorías, ya que hay una mayor cantidad de frecuencias de interés. En general, se sugiere evaluar todo el espectro primero antes de reducirlo para no perder generalidad y ya después aplicar métodos estadísticos para descartar información que no aporta al problema concretamente. Tercero, no hay garantía de que el desempeño sea extrapolable para tasas de adquisición menores a las usadas en el entrenamiento de este trabajo, ya que esto implicaría una reducción del espectro de frecuencias a analizar y del que los modelos originales fueron entrenados y puestos a prueba. Las pruebas realizadas muestran que las características en el dominio de la frecuencia facilitan en gran medida la separación de las categorías, ya que proporcionan información más identificable y son más fáciles de simplificar con técnicas estadísticas o reducciones de la dimensionalidad. Por estas razones, se recomienda este abordaje para modelos futuros. Este proyecto muestra que el fenómeno de cavitación puede ser identificable con una combinación de un adecuado procesamiento de las señales, acompañada de métodos estadísticos y reducciones de la dimensionalidad. Por ejemplo, la proyección PCA muestra que es posible separar en clústeres excluyentes al estado normal y al de cavitación, por lo que un clasificador podría resolver el problema con buen desempeño. Esto sugiere que no es necesario recurrir a modelos complejos y costosos para resolver el problema planteado.

En el caso de los modelos de detección de anomalías, usar capas de convolución 1D permite procesar mejor las señales al extraer características y así alcanzar un detector más sensible. El detector de anomalías es un complemento a un clasificador, ya que al ser un modelo no supervisado, frente a una anomalía desconocida no va a brindar una etiqueta. De usarse el detector de anomalías se sugiere extraer los datos al momento del evento, analizarlo con la metodología vista en la sección del análisis exploratorio, someterlo ante la prueba-F ANOVA y la reducción de la dimensionalidad no lineal e idealmente lograr generar un nuevo clúster que lo represente. Este apoyo de detector de anomalías y modelo de clasificación presenta una buena oportunidad para continuar generando conocimiento en esta área de la ingeniería.

Desde el punto de vista de eficiencia, algunas de las estrategias mostradas permiten reducir el costo computacional, como por ejemplo, el uso de algoritmos eficientes (FFT), la selección adecuada de frecuencias por medio de ANOVA y la reducción de la dimensionalidad con los distintos algoritmos. En general reducir la cantidad de características conlleva a un problema más pequeño de resolver, pero es de suma importancia mantener un balance entre costo y desempeño.

Este trabajo comparte similitudes con otros trabajos vistos en la revisión de literatura, como por ejemplo con el uso de datos de vibración, la clasificación mediante una SVM y redes convolucionales 1D. Sin embargo, a pesar de que sigue un esquema similar, también es importante mencionar que se diferencia por las técnicas de reducción de la dimensionalidad escogidas, como por ejemplo el autocodificador, y también por la reducción de frecuencias seleccionadas en vez de características de la estadística descriptiva.

Como trabajo futuro se sugiere como primer paso ampliar el experimento para recolectar datos reales de otros problemas adicionales a la cavitación. Las pruebas realizadas con datos sintéticos sugieren que una proyección como la obtenida por medio de PCA podría no ser suficiente para clasificar un problema del otro cuando la únicamente diferencia es la presencia de un armónico adicional. Idealmente hay que someter esta hipótesis a prueba con datos reales e identificar si usando otro tipo de reducción lineal, pero supervisada, como ocurre con LDA podría efectivamente ser una mejor alternativa, o si por el contrario se requiere definitivamente una reducción no

lineal por medio de autocodificadores. Recolectar datos de desalineamiento o soldadura no es una tarea trivial, ya que implica un ambiente controlado y seguro. También es parte del trabajo futuro abordar las amenazas contra la validez de este proyecto. Por ejemplo, en este experimento no se profundizó el impacto de la resonancia del sistema, por lo que otro experimento sería identificar estos ámbitos y su influencia en las características. Asimismo, este experimento idealmente debe repetirse con otros ámbitos de velocidad de giro, con la idea de aumentar los ejemplos y construir un clasificador más generalizable al problema. Por último, recordando la Figura 1, este experimento debe repetirse con otro tipo de bombas hidráulicas. Se sugiere del tipo recíprocante, ya que siguen una curva operativa distinta y la cavitación ocurre bajo circunstancias distintas a las aquí mostradas.

Referencias

- [1] G.-Y. Lee, M. Kim, Y.-J. Quan, M.-S. Kim, T. J. Y. Kim, H.-S. Yoon, S. Min, D.-H. Kim, J.-W. Mun, J. W. Oh, *et al.*, "Machine health management in smart factory: A review," *Journal of Mechanical Science and Technology*, vol. 32, no. 3, pp. 987–1009, 2018.
- [2] S. Gowid, R. Dixon, and S. Ghani, "A novel robust automated fft-based segmentation and features selection algorithm for acoustic emission condition based monitoring systems," *Applied Acoustics*, vol. 88, pp. 66–74, 2015.
- [3] D. S. Thomas and D. S. Thomas, *The costs and benefits of advanced maintenance in manufacturing*. US Department of Commerce, National Institute of Standards and Technology, 2018.
- [4] D. Pandya, S. Upadhyay, and S. P. Harsha, "Fault diagnosis of rolling element bearing with intrinsic mode function of acoustic emission data using apf-knn," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4137–4145, 2013.
- [5] J. Dai, C. P. Chen, X.-Y. Xu, Y. Huang, P. Hu, C.-P. Hu, and T. Wu, "Machinery vibration signals analysis and monitoring for fault diagnosis and process control," in *International Conference on Intelligent Computing*, pp. 696–703, Springer, 2008.
- [6] R. L. Mott, F. M. Noor, and A. A. Aziz, "Applied fluid mechanics," 2006.
- [7] A. Azadeh, V. Ebrahimipour, and P. Bavar, "A fuzzy inference system for pump failure diagnosis to improve maintenance process: The case of a petrochemical industry," *Expert Systems with Applications*, vol. 37, no. 1, pp. 627–639, 2010.
- [8] J.-P. Franc and J.-M. Michel, *Fundamentals of cavitation*, vol. 76. Springer science & Business media, 2006.
- [9] S. Smith, *Digital signal processing: a practical guide for engineers and scientists*. Elsevier, 2013.
- [10] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

- [11] K. Kido, *Digital Fourier analysis: fundamentals*. Springer, 2014.
- [12] J. Dai, J. Tang, S. Huang, and Y. Wang, "Signal-based intelligent hydraulic fault diagnosis methods: Review and prospects," *Chinese Journal of Mechanical Engineering*, vol. 32, no. 1, p. 75, 2019.
- [13] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.
- [14] J. Brownlee, *Long short-term memory networks with python: develop sequence prediction models with deep learning*. Machine Learning Mastery, 2017.
- [15] T. G. Dietterich, "Machine learning for sequential data: A review," in *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*, pp. 15–30, Springer, 2002.
- [16] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.
- [17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [18] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.
- [19] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI communications*, vol. 30, no. 2, pp. 169–190, 2017.
- [20] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [21] *Mechanical Systems and Signal Processing*, 2020.
- [22] *Measurement: Journal of the International Measurement Confederation*, 2020.
- [23] S.-F. Yuan and F.-L. Chu, "Support vector machines-based fault diagnosis for turbo-pump rotor," *Mechanical Systems and Signal Processing*, vol. 20, no. 4, pp. 939–952, 2006.

- [24] K. Wang, "Neural network approach to vibration feature selection and multiple fault detection for mechanical systems," in *First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06)*, vol. 3, pp. 431–434, IEEE, 2006.
- [25] M. Dong and D. He, "Hidden semi-markov model-based methodology for multi-sensor equipment health diagnosis and prognosis," *European Journal of Operational Research*, vol. 178, no. 3, pp. 858–878, 2007.
- [26] M. Tsunoyama, H. Jinno, M. Ogawa, and T. Sato, "An application of fuzzy measure and integral for diagnosing faults in rotating machines," in *Tools and Applications with Artificial Intelligence*, pp. 121–133, Springer, 2009.
- [27] P. Pławiak, "An estimation of the state of consumption of a positive displacement pump based on dynamic pressure or vibrations using neural networks," *Neurocomputing*, vol. 144, pp. 471–483, 2014.
- [28] R. DiBiano and S. Mukhopadhyay, "Automated diagnostics for manufacturing machinery based on well-regularized deep neural networks," *Integration*, vol. 58, pp. 303–310, 2017.
- [29] D. Siano and M. Panza, "Diagnostic method by using vibration analysis for pump fault detection," *Energy Procedia*, vol. 148, pp. 10–17, 2018.
- [30] K. McKee, G. Forbes, I. Mazhar, R. Entwistle, and I. Howard, "A review of machinery diagnostics and prognostics implemented on a centrifugal pump," in *Engineering asset management 2011*, pp. 593–614, Springer, 2014.
- [31] N. Dutta, S. Umashankar, V. A. Shankar, S. Padmanaban, Z. Leonowicz, and P. Wheeler, "Centrifugal pump cavitation detection using machine learning algorithm technique," in *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pp. 1–6, IEEE, 2018.
- [32] N. Dutta, U. Subramaniam, P. Sanjeevikumar, S. C. Bharadwaj, Z. Leonowicz, and J. B. Holm-Nielsen, "Comparative study of cavitation problem detection in

- pumping system using svm and k-nearest neighbour method," in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pp. 1–6, IEEE, 2020.
- [33] N. Sakthivel, B. B. Nair, and V. Sugumaran, "Soft computing approach to fault diagnosis of centrifugal pump," *Applied Soft Computing*, vol. 12, no. 5, pp. 1574–1581, 2012.
- [34] N. Sakthivel, B. B. Nair, M. Elangovan, V. Sugumaran, and S. Saravanmurugan, "Comparison of dimensionality reduction techniques for the fault diagnosis of mono block centrifugal pump using vibration signals," *Engineering Science and Technology, an International Journal*, vol. 17, no. 1, pp. 30–38, 2014.
- [35] I. Griffiths, "Automated condition monitoring using artificial intelligence," in *Proceedings of the 2020 4th International Conference on Vision, Image and Signal Processing*, pp. 1–7, 2020.
- [36] R. Tiwari, D. Bordoloi, and A. Dewangan, "Blockage and cavitation detection in centrifugal pumps from dynamic pressure signal using deep learning algorithm," *Measurement*, vol. 173, p. 108676, 2021.
- [37] A. K. Panda, J. S. Rapur, and R. Tiwari, "Prediction of flow blockages and impending cavitation in centrifugal pumps using support vector machine (svm) algorithms based on vibration measurements," *Measurement*, vol. 130, pp. 44–56, 2018.
- [38] D. Bordoloi and R. Tiwari, "Identification of suction flow blockages and casing cavitations in centrifugal pumps by optimal support vector machine techniques," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 39, no. 8, pp. 2957–2968, 2017.
- [39] M. Milovančević, V. Nikolić, D. Petkovic, L. Vracar, E. Veg, N. Tomic, and S. Jović, "Vibration analyzing in horizontal pumping aggregate by soft computing," *Measurement*, vol. 125, pp. 454–462, 2018.

- [40] N. Zhang, M. Yang, B. Gao, and Z. Li, "Vibration characteristics induced by cavitation in a centrifugal pump with slope volute," *Shock and Vibration*, vol. 2015, 2015.
- [41] B. Gao, P. Guo, N. Zhang, Z. Li, and M. Yang, "Experimental investigation on cavitating flow induced vibration characteristics of a low specific speed centrifugal pump," *Shock and Vibration*, vol. 2017, 2017.
- [42] S. A. Al-Hashmi, "Spectrum analysis of acoustic signals for cavitation detection," in *2012 IEEE Symposium on Industrial Electronics and Applications*, pp. 321–325, IEEE, 2012.
- [43] "Bomba ebara jex 6," *Catalogo de productos Ebara*, Recuperado de <http://www.ebara.es/productos/bombas-centrifugas-horizontales/jex-jesjes-je-autoaspirante-jet-ac-inox-aisi-304/>, Julio 2022.
- [44] R. S. Srinivasamurthy, *Understanding 1D Convolutional Neural Networks Using Multiclass Time-Varying Signalss*. PhD thesis, Clemson University, 2018.
- [45] A. Ng, "Machine learning yearning: Technical strategy for ai engineers in the era of deep learning," Recuperado de <https://www.mlyearning.org>, 2019.
- [46] R. Birajdar, R. Patil, and K. Khanzode, "Vibration and noise in centrifugal pumps-sources and diagnosis methods," in *3rd International conference on Integrity, Reliability and Failure*, pp. 20–24, 2009.
- [47] F. Chollet, "The keras blog," *The Keras Blog ATOM*, Recuperado de <https://blog.keras.io/building-autoencoders-in-keras.html>, Mayo 2016.
- [48] P. Flach and M. Kull, "Precision-recall-gain curves: Pr analysis done right," *Advances in neural information processing systems*, vol. 28, 2015.

Apéndice A

Implementación de los modelos

```

1 import numpy as np
2 import math
3 from sklearn import metrics
4 import seaborn as sns
5 import tensorflow as tf
6
7 ENTRENAMIENTO_BINARIO = "C:\\PATH\\Binario\\"
8 ENTRENAMIENTO_MULTIPLE = "C:\\PATH\\Multiple\\"
9 PRUEBA_BINARIO = "C:\\PATH\\Binario\\"
10 PRUEBA_MULTIPLE = "C:\\PATH\\Multiple\\"
11
12 def cargar_entrenamiento_binario():
13     return (np.genfromtxt(ENTRENAMIENTO_BINARIO+'Datos.csv', delimiter=','),
14           np.genfromtxt(ENTRENAMIENTO_BINARIO+'Etiquetas.csv', delimiter=','))
15
16 def cargar_entrenamiento_multiple():
17     return (np.genfromtxt(ENTRENAMIENTO_MULTIPLE+'Datos.csv', delimiter=','),
18           np.genfromtxt(ENTRENAMIENTO_MULTIPLE+'Etiquetas.csv', delimiter=','))
19
20 def cargar_prueba_binario():
21     return (np.genfromtxt(PRUEBA_BINARIO+'Datos.csv', delimiter=','),
22           np.genfromtxt(PRUEBA_BINARIO+'Etiquetas.csv', delimiter=','))
23
24 def cargar_prueba_multiple():
25     return (np.genfromtxt(PRUEBA_MULTIPLE+'Datos.csv', delimiter=','),
26           np.genfromtxt(PRUEBA_MULTIPLE+'Etiquetas.csv', delimiter=','))
27
28 def convertir_dominio_frecuencia(serie, promedio=1, tasa=8192, hamming=True):
29     if(hamming):
30         serie = serie*np.hamming(tasa)
31
32         serie_fft = abs(np.fft.fft(serie))
33         kernel = np.ones(promedio)/promedio
34         serie_fft = np.convolve(serie_fft, kernel, mode='valid')
35         serie_fft = serie_fft[0:int(tasa/2)]
36
37     return serie_fft
38
39 def recortar_multiplo(serie, multiplo=8192):
40     # Metodo para recortar una secuencia de datos para ser una cantidad multiplo de n
41     multiplo_cercano = multiplo*math.floor(len(serie)/multiplo)
42     particiones = multiplo_cercano/multiplo
43

```

```

44     return serie[0:multiplo_cercano], int(particiones)
45
46 def ventana_deslizante(serie, tamano=512, tasa=8192):
47     # Metodo para recorrer datos con una ventana deslizante
48     arreglo = np.empty
49
50     for i in range(int(len(serie)/tamano)):
51         inicio = i*tamano
52
53         if i == 0:
54             arreglo = serie[inicio:inicio+tasa]
55         else:
56             arreglo = np.concatenate((arreglo, serie[inicio:inicio+tasa]))
57
58     return arreglo
59
60 def generar_desalineamiento(serie, tasa=8192, tiempo_final=128, frecuencia_fundamental=47):
61     # Metodo para generar datos simulados de soltura
62     tiempo = np.arange(0, tiempo_final, 1/tasa)
63     magnitud_aleatoria = 0.03
64     senoide = (magnitud_aleatoria
65 * np.sin(2 * np.pi * 2*frecuencia_fundamental * tiempo + 0))
66
67     if(serie.ndim > 1):
68         for i in range(serie.shape[1]):
69             serie[:,i] = serie[:,i] + senoide
70     else:
71         serie = serie + senoide
72
73     return serie
74
75 def generar_soltura(serie, tasa=8192, tiempo_final=128, frecuencia_fundamental=47):
76     # Metodo para generar datos simulados de soltura
77     tiempo = np.arange(0, tiempo_final, 1/tasa)
78     magnitud_aleatoria = 0.01
79
80     if(serie.ndim > 1):
81         for i in range(serie.shape[1]):
82             for j in range(3,9):
83                 senoide = (magnitud_aleatoria
84 * np.sin(2 * np.pi * j * frecuencia_fundamental * tiempo + 0))
85
86                 serie[:,i] = serie[:,i] + senoide
87     else:
88         for i in range(3,9):
89             senoide = (magnitud_aleatoria
90 * np.sin(2 * np.pi * i * frecuencia_fundamental * tiempo + 0))

```

```

91
92     serie = serie + sinusoide
93
94     return serie
95
96 def matriz_confusion_binario(y_prueba, prediccion):
97     # Metodo para imprimir una matriz de confusion en clasificacion binaria
98     exactitud = metrics.accuracy_score(y_prueba, prediccion)
99     precision = metrics.precision_score(y_prueba, prediccion)
100    exhaustividad = metrics.recall_score(y_prueba, prediccion)
101    valor_f1 = metrics.f1_score(y_prueba, prediccion)
102    resultado_matriz = metrics.confusion_matrix(y_prueba, prediccion)
103
104    group_names = ['Verdadero negativo',
105                  'Falso positivo',
106                  'Falso negativo',
107                  'Verdadero positivo']
108    group_counts = ["{0:0.0f}".format(valor) for valor in
109                  resultado_matriz.flatten()]
110    group_percentages = ["{0:.2%}".format(valor) for valor in
111                        resultado_matriz.flatten()/np.sum(resultado_matriz)]
112    labels = [f"{v1}\n{v2}\n{v3}" for v1, v2, v3 in
113             zip(group_names, group_counts, group_percentages)]
114    labels = np.asarray(labels).reshape(2,2)
115
116    sns.heatmap(resultado_matriz, annot=labels, fmt='', cmap='Blues')
117
118    print('Clasificador binario')
119    print('Exactitud: '+str("{:.2f}".format(exactitud)))
120    print('Precision: '+str("{:.2f}".format(precision)))
121    print('Exhaustividad: '+str("{:.2f}".format(exhaustividad)))
122    print('Valor F1: '+str("{:.2f}".format(valor_f1)))
123
124 def matriz_confusion_multiple(y_prueba, prediccion):
125     # Metodo para imprimir una matriz de confusion en clasificacion multiple
126     resultado_matriz = metrics.confusion_matrix(y_prueba, np.argmax(prediccion, axis=-1))
127
128     group_counts = ["{0:0.0f}".format(value) for value in
129                   resultado_matriz.flatten()]
130     group_percentages = ["{0:.2%}".format(value) for value in
131                         resultado_matriz.flatten()/np.sum(resultado_matriz)]
132     labels = [f"{v1}\n{v2}" for v1, v2 in
133             zip(group_counts, group_percentages)]
134     labels = np.asarray(labels).reshape(4,4)
135
136     sns.heatmap(resultado_matriz,
137                 annot=labels,

```

```

138         fmt='',
139         cmap='Blues').set(xlabel='Predicciones', ylabel='Categorías')
140
141 def guardar_modelo(model, output_path, nombre):
142     # Metodo para guardar un modelo
143     model_json = model.to_json()
144
145     with open(output_path+nombre+'.json', 'w') as json_file:
146         json_file.write(model_json)
147
148     model.save_weights(output_path+nombre+'.h5')
149
150 def cargar_modelo(input_path, nombre):
151     # Metodo para cargar un modelo
152     json_file = open(input_path+nombre+'.json', 'r')
153     loaded_model_json = json_file.read()
154     json_file.close()
155     model = tf.keras.models.model_from_json(loaded_model_json)
156     model.load_weights(input_path+nombre+'.h5')
157
158     return model
159
160 def sampling(distribucion):
161     # Metodo para forzar la salida de un encoder a una distribucion normal
162     encoder_media, encoder_varianza = distribucion
163     epsilon = tf.keras.backend.random_normal(
164         shape=(tf.keras.backend.shape(encoder_media)[0], 2),
165         mean=0,
166         stddev=0.1)
167
168     return encoder_media + tf.keras.backend.exp(encoder_varianza) * epsilon

```

```

1 import tfia_helper as tfia
2 import os
3 import math
4 import matplotlib.pyplot as plt
5 import numpy as np
6 import tensorflow as tf
7 import pandas as pd
8 from keras.models import Sequential
9 from keras.layers import Dense
10 from keras.layers import LSTM
11 from keras.layers import Conv1D
12 from keras.layers import Conv1DTranspose
13 from keras.layers import TimeDistributed
14 from keras.layers import RepeatVector
15 from keras.layers import Dropout
16 from sklearn.model_selection import train_test_split
17 from sklearn.feature_selection import SelectKBest
18 from sklearn.feature_selection import f_classif
19 from sklearn.preprocessing import MinMaxScaler
20 from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
21 from sklearn.svm import SVC
22 from sklearn import metrics
23
24 # Usar el CPU en vez del GPU
25 os.environ["CUDA_VISIBLE_DEVICES"] = "-1"
26 print(tf.config.list_physical_devices('GPU'))
27
28 # Estilo
29 plt.style.use('ggplot')
30
31 # Constantes
32 tasa = 8192 #Hz
33 top_caracteristicas = 1000
34 f = 47 #Frecuencia fundamental
35 output_path = 'C:\\\\OUTPUT_PATH\\\\'
36 path_prueba = 'C:\\\\PATH\\\\prueba_completo.csv'
37
38 # MODELO CLASIFICACION SVM
39 normal_data_path = 'C:\\\\PATH\\\\normal_x_completo.csv'
40 cav_data_path = 'C:\\\\PATH\\\\cavitacion_x_completo.csv'
41 normal_data = np.genfromtxt(normal_data_path, delimiter=';')
42 cav_data = np.genfromtxt(cav_data_path, delimiter=';')
43
44 # Determinacion de la dependencia de las variables (frecuencias)
45 # y las respuestas (categorias)
46 normal = tfia.ventana_deslizante(normal_data[:,1], tamano=2048)
47 cavitacion = tfia.ventana_deslizante(cav_data[:,1], tamano=2048)

```

```

48 x_normal, normal_particiones = tfia.recortar_multiplo(normal)
49 x_cavitacion, cavitacion_particiones = tfia.recortar_multiplo(cavitacion)
50 x_total = np.concatenate((x_normal, x_cavitacion))
51 particiones = normal_particiones+cavitacion_particiones
52 y_normal = np.zeros(normal_particiones)
53 y_cavitacion = np.ones(cavitacion_particiones)
54 y_entrenamiento = np.concatenate((y_normal, y_cavitacion))
55 x_tiempo_total = np.concatenate((x_normal, x_cavitacion))
56 x_tiempo_total = x_total.reshape(particiones, tasa)
57
58 for i in range(len(x_tiempo_total)):
59     if i == 0:
60         x_frecuencia = tfia.convertir_dominio_frecuencia(x_tiempo_total[i, :], promedio=5)
61     else:
62         temp = tfia.convertir_dominio_frecuencia(x_tiempo_total[i, :], promedio=5)
63         x_frecuencia = np.concatenate((x_frecuencia, temp))
64
65 x_frecuencia = x_frecuencia.reshape(particiones, int(tasa/2))
66
67 # Prueba-F ANOVA
68 idx = np.random.permutation(particiones)
69 seleccion_caracteristicas = SelectKBest(score_func=f_classif, k=top_caracteristicas)
70 seleccion_caracteristicas.fit(x_frecuencia, y_entrenamiento)
71
72 x_frecuencias_dependientes = seleccion_caracteristicas.fit_transform(x_frecuencia,
73                                                                     y_entrenamiento)
74
75 # Normalizacion
76 scaler = MinMaxScaler()
77 scaler.fit(x_frecuencias_dependientes)
78 x_frecuencias_dependientes = scaler.transform(x_frecuencias_dependientes)
79
80 # Shuffle
81 x_entrenamiento, x_prueba, y_entrenamiento, y_prueba = train_test_split(
82     x_frecuencias_dependientes,
83     y_entrenamiento,
84     test_size=0.20,
85     random_state=1,
86     stratify=y_entrenamiento)
87
88 contador_entrenamiento = np.bincount(y_entrenamiento.astype(int))
89 contador_prueba = np.bincount(y_prueba.astype(int))
90
91 clf = SVC(kernel='linear')
92 clf.fit(x_entrenamiento, y_entrenamiento)
93 y_prediccion = clf.predict(x_prueba)
94

```

```

95 # Plots
96 print("Exactitud: ", metrics.accuracy_score(y_prueba, y_prediccion))
97 print("Precision: ", metrics.precision_score(y_prueba, y_prediccion))
98 print("Exhaustividad: ", metrics.recall_score(y_prueba, y_prediccion))
99 print(metrics.classification_report(y_prueba, y_prediccion))
100
101 # MODELO CLASIFICACION LDA
102 lda = LinearDiscriminantAnalysis()
103 lda_componentes_principales = lda.fit_transform(x_entrenamiento, y_entrenamiento)
104 y_dataframe = pd.DataFrame(y_entrenamiento, columns = ['categoria'])
105 principal_df = pd.DataFrame(data=lda_componentes_principales, columns=['LDA 1'])
106 final_df = pd.concat([principal_df, y_dataframe['categoria']], axis=1)
107 normal = final_df.where(final_df['categoria'] == 0)['LDA 1'].dropna()
108 cavitacion = final_df.where(final_df['categoria'] == 1)['LDA 1'].dropna()
109
110 # Plots
111 plt.hist([normal.to_list(), cavitacion.to_list()],
112         np.linspace(-30, 30, 100),
113         label=['Normal', 'Cavitacion'])
114 plt.legend(loc='upper left')
115 plt.xlabel('Categorias')
116 plt.ylabel('Muestras')
117
118 # MODELO CLASIFICACION CONV1D
119 x_train, y_train = tfia.cargar_entrenamiento_binario()
120 x_train = x_train.reshape(409,4096,3)
121
122 model = Sequential()
123 model.add(Conv1D(64,32, activation='relu', padding='same', strides=2,
124               input_shape=x_train.shape[1:], name='Entrada_Conv1d'))
125 model.add(Conv1D(32,16, activation='relu', padding='valid', strides=2,
126               name='Salida_Conv1d'))
127 model.add(Dropout(0.5))
128 model.add(LSTM(32))
129 model.add(Dense(3, activation="relu"))
130 model.add(Dense(1, activation="sigmoid", name='Salida_Clasificacion'))
131 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
132
133 callback = tf.keras.callbacks.EarlyStopping(monitor='accuracy', patience=10)
134 history = model.fit(x_train, y_train, validation_split=0.25,
135                   epochs=100, callbacks=[callback], verbose=0)
136 print(model.summary())
137
138 # Guardar el modelo
139 tfia.guardar_modelo(model, output_path, 'clasificacion_CONV1D')
140
141 # Plots

```

```

142 plt.plot(history.history['accuracy'], label='Datos de entrenamiento')
143 plt.plot(history.history['val_accuracy'], label='Datos de validacion cruzada')
144 plt.xlabel("Epoch")
145 plt.ylabel("Exactitud")
146 plt.title("Evolucion de la exactitud durante el entrenamiento")
147 plt.legend()
148 plt.show()
149
150 plt.plot(history.history['loss'], label='Datos de entrenamiento')
151 plt.plot(history.history['val_loss'], label='Datos de validacion cruzada')
152 plt.xlabel("Epoch")
153 plt.ylabel("Funcion de perdida")
154 plt.title("Evolucion de la funcion de perdida durante el entrenamiento")
155 plt.legend()
156 plt.show()
157
158 # Analisis del error
159 x_prueba, y_prueba = tfia.cargar_prueba_binario()
160 x_prueba = x_prueba.reshape(103,4096,3)
161 prediccion = model.predict(x_prueba)
162
163 # Valor AUC
164 auc_score=metrics.roc_auc_score(y_prueba , prediccion)
165
166 # Plot
167 fpr, tpr, thresholds = metrics.roc_curve(y_prueba , prediccion)
168 plt.plot(fpr,tpr, label='Modelo - AUC: '+str("{0:.2%}".format(auc_score)))
169 plt.plot([0,1], [0,1],linestyle='dashed', color='grey')
170 plt.title("Curva ROC")
171 plt.xlabel('Tasa de falsos positivos')
172 plt.ylabel('Tasa de verdaderos positivos')
173 plt.legend()
174 prediccion = np.round(prediccion)
175 tfia.matriz_confusion_binario(y_prueba,prediccion)
176
177 # MODELO CLASIFICACION LSTM
178 model = Sequential()
179 model.add(Conv1D(64,32, activation='relu',padding='valid', strides=2,
180     input_shape=(x_train.shape[1],x_train.shape[2]), name='Entrada_Conv1d'))
181 model.add(LSTM(128, return_sequences = True, name='Entrada_LSTM'))
182 model.add(LSTM(64, return_sequences = True))
183 model.add(LSTM(64))
184 model.add(Dropout(0.5))
185 model.add(Dense(1, activation="sigmoid", name='Salida_Clasificacion'))
186 model.compile(loss='binary_crossentropy',
187     optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001), metrics=['accuracy'])
188

```

```

189 callback = tf.keras.callbacks.EarlyStopping(monitor='accuracy', patience=10)
190 history = model.fit(x_train, y_train, validation_split=0.25,
191     epochs=100, callbacks=[callback], verbose=0)
192 print(model.summary())
193
194 # Guardar el modelo
195 tfia.guardar_modelo(model, output_path, 'clasificacion_LSTM')
196
197 # Plots
198 plt.plot(history.history['accuracy'], label='Datos de entrenamiento')
199 plt.plot(history.history['val_accuracy'], label='Datos de validacion cruzada')
200 plt.xlabel("Epoch")
201 plt.ylabel("Exactitud")
202 plt.title("Evolucion de la exactitud durante el entrenamiento")
203 plt.legend()
204 plt.show()
205
206 plt.plot(history.history['loss'], label='Datos de entrenamiento')
207 plt.plot(history.history['val_loss'], label='Datos de validacion cruzada')
208 plt.xlabel("Epoch")
209 plt.ylabel("Funcion de perdida")
210 plt.title("Evolucion de la funcion de perdida durante el entrenamiento")
211 plt.legend()
212 plt.show()
213
214 # Analisis del error
215 x_prueba, y_prueba = tfia.cargar_prueba_binario()
216 x_prueba = x_prueba.reshape(103,4096,3)
217 prediccion = model.predict(x_prueba)
218
219 # Valor AUC
220 auc_score=metrics.roc_auc_score(y_prueba, prediccion)
221
222 # Plot
223 fpr, tpr, thresholds = metrics.roc_curve(y_prueba, prediccion)
224 plt.plot(fpr, tpr, label='Modelo - AUC: '+str("{0:.2%}".format(auc_score)))
225 plt.plot([0,1], [0,1], linestyle='dashed', color='grey')
226 plt.title("Curva ROC")
227 plt.xlabel('Tasa de falsos positivos')
228 plt.ylabel('Tasa de verdaderos positivos')
229 plt.legend()
230 prediccion = np.round(prediccion)
231 tfia.matriz_confusion_binario(y_prueba, prediccion)
232
233 # MODELO DE DETECCION DE ANOMALIAS CON CONVID
234 normal_data_path = 'C:\\\\PATH\\normal_completo.csv'
235 normal_data = np.genfromtxt(normal_data_path, delimiter=',')

```

```

236 data = normal_data.reshape(256, 4096, 3) # particiones , pasos , características
237
238 # Encoder
239 model = Sequential()
240 model.add(Conv1D(64,32, activation='relu',padding='same',
241     strides=2,input_shape=data.shape[1:],name='Entrada_Encoder'))
242 model.add(Conv1D(32,16, activation='relu', padding='same', strides=2, name='Salida_Encoder'))
243 # Decoder
244 model.add(Conv1DTranspose(32,16,
245     activation='relu',padding='same', strides=2, name='Entrada_Decoder'))
246 model.add(Conv1DTranspose(64,32, activation='relu',padding='same', strides=2))
247 model.add(Conv1DTranspose(data.shape[2],32,padding='same', name='Salida_Decoder'))
248 model.compile(loss='mae', optimizer='adam', metrics=['accuracy'])
249
250 callback = tf.keras.callbacks.EarlyStopping(monitor='accuracy', patience=10)
251 history = model.fit(data, data, validation_split=0.2,
252     epochs=100, callbacks=[callback], verbose=0)
253 print(model.summary())
254
255 # Guardar el modelo
256 tfia.guardar_modelo(model, output_path, 'detector_anomalias_CONV1D')
257
258 # Plots
259 plt.plot(history.history['loss'], label='Datos de entrenamiento')
260 plt.plot(history.history['val_loss'], label='Datos de validacion cruzada')
261 plt.xlabel("Epochs")
262 plt.ylabel("Funcion de perdida")
263 plt.title("Evolucion de la funcion de perdida durante el entrenamiento")
264 plt.legend()
265
266 prediccion = model.predict(data)
267 error = np.mean(np.abs(prediccion-data) , axis=1)
268 error_distribucion = error.reshape(error.shape[0]*error.shape[1])
269
270 plt.plot(error_distribucion)
271 plt.ylabel('Error absoluto medio (MAE)')
272 plt.xlabel('Tiempo (s)')
273
274 plt.hist(error_distribucion ,bins=30)
275 plt.xlabel('Error absoluto medio (MAE)')
276 plt.ylabel('Muestras')
277
278 data_prueba = np.genfromtxt(path_prueba, delimiter=',')
279 data_prueba = data_prueba.reshape(256, 4096, 3)
280 detector_prueba = model.predict(data_prueba)
281 error = np.mean(np.abs(detector_prueba-data) , axis=1)
282 error_distribucion_prueba = error.reshape(error.shape[0]*error.shape[1])

```

```

283
284 plt.plot(error_distribucion_prueba)
285 plt.plot([0,800],[0.01,0.01], color='black', label='Umbral a 1%')
286 plt.xlabel("Muestras en tiempo")
287 plt.ylabel("Error absoluto medio (MAE)")
288 plt.title("Detector de anomalias")
289 plt.legend()
290
291 # MODELO DE DETECCION DE ANOMALIAS CON LSTM
292 data = normal_data.reshape(4096, 3, 256) # pasos , características , particiones
293
294 # Encoder
295 model = Sequential()
296 model.add(LSTM(1024, return_sequences = True,
297     input_shape=(data.shape[1],data.shape[2]), name='Entrada_Encoder'))
298 model.add(LSTM(512, return_sequences = False))
299 model.add(RepeatVector(data.shape[1], name='Salida_Encoder'))
300 # Decoder
301 model.add(LSTM(512, return_sequences = True, name='Entrada_Decoder'))
302 model.add(LSTM(1024, return_sequences = True))
303 model.add(TimeDistributed(Dense(data.shape[2]), name='Salida_Decoder'))
304 model.compile(loss='mae', optimizer='adam', metrics=['accuracy'])
305
306 callback = tf.keras.callbacks.EarlyStopping(monitor='accuracy', patience=10)
307 history = model.fit(data, data, validation_split=0.2,
308     epochs=100, callbacks=[callback], verbose=0)
309 print(model.summary())
310
311 # Guardar el modelo
312 tfia.guardar_modelo(model, output_path, 'detector_anomalias_LSTM')
313
314 # Plots
315 plt.plot(history.history['loss'], label='Datos de entrenamiento')
316 plt.plot(history.history['val_loss'], label='Datos de validacion cruzada')
317 plt.xlabel("Epochs")
318 plt.ylabel("Funcion de perdida")
319 plt.title("Evolucion de la funcion de perdida durante el entrenamiento")
320 plt.legend()
321
322 prediccion = model.predict(data)
323 error = np.mean(np.abs(prediccion-data), axis=1)
324 error_distribucion = error.reshape(error.shape[0]*error.shape[1])
325
326 plt.plot(error_distribucion)
327 plt.ylabel('Error absoluto medio (MAE)')
328 plt.xlabel('Tiempo (s)')
329

```

```
330 plt.hist(error_distribucion , bins=30)
331 plt.xlabel('Error absoluto medio (MAE)')
332 plt.ylabel('Muestras')
333
334 data_prueba = np.genfromtxt(path_prueba , delimiter=',')
335 data_prueba = data_prueba.reshape(4096, 3, 256)
336 detector_prueba = model.predict(data_prueba)
337 error = np.mean(np.abs(detector_prueba-data) , axis=1)
338 error_distribucion_prueba = error.reshape(error.shape[0]*error.shape[1])
339
340 plt.plot(error_distribucion_prueba)
341 plt.plot([0,1100000],[0.1,0.1], color='black', label='Umbral a 10%')
342 plt.xlabel("Muestras en tiempo")
343 plt.ylabel("Error absoluto medio (MAE)")
344 plt.title("Detector de anomalias")
345 plt.legend()
```

Apéndice B

Datos recolectados

Los datos usados en este trabajo de investigación se encuentran disponibles en

<https://github.com/lsanchezc14/tfia>