



POSGRADOS

MAESTRÍA EN _____ ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-19-No.277-2018

OPCIÓN DE
TITULACIÓN:

PROYECTOS DE DESARROLLO

TEMA:

MODELAMIENTO BASADO EN DATOS PARA LA
DETECCIÓN DE FALLOS EN UNA BOMBA HIDRÁULICA
CENTRÍFUGA MULTITETAPA VERTICAL A PARTIR DE
SEÑALES MULTIMODALES EN CONDICIÓN NORMAL

AUTOR:

MAURICIO LEONARDO VILLACÍS MARÍN

DIRECTOR:

DIEGO ROMÁN CABRERA MENDIETA

CUENCA - ECUADOR
2020

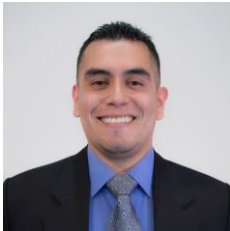
Autor:



Mauricio Leonardo Villacís Marín.

Ingeniero Electrónico.
Candidato a Magíster en Electrónica y Automatización,
Mención en Informática Industrial por la Universidad
Politécnica Salesiana - Sede Cuenca.
mvillacism@est.ups.edu.ec

Dirigido por:



Diego Román Cabrera Mendieta.

Ingeniero Electrónico
Master Universitario en Lógica, Computación e
Inteligencia Artificial
Doctor dentro del programa Ingeniería Informática
dcabrera@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

©2020 Universidad Politécnica Salesiana.
CUENCA - ECUADOR - SUDAMÉRICA
MAURICIO LEONARDO VILLACÍS MARÍN.

***MODELAMIENTO BASADO EN DATOS PARA LA
DETECCIÓN DE FALLOS EN UNA BOMBA HIDRÁULICA
CENTRÍFUGA MULTIETAPA VERTICAL A PARTIR DE
SEÑALES MULTIMODALES EN CONDICIÓN NORMAL***

Índice general

Índice de Figuras	V
Índice de Tablas	VII
Resumen	VIII
Abstract	VIII
1 Introducción	1
1.1 Descripción general del problema.....	2
1.2 Objetivos	2
1.2.1 Objetivo general	2
1.2.2 Objetivos específicos	3
1.3 Contribuciones	3
1.4 Organización del manuscrito.....	4
2 Antecedentes y fundamentación teórica	5
2.1 Antecedentes	6
2.2 Fundamentación teórica.....	7
2.2.1 Mantenimiento basado en la condición	7
2.2.2 Red generativa adversaria (GAN).....	9
2.2.3 Máquinas de soporte vectorial (SVM)	10
2.2.4 One-class SVM.....	13
2.2.5 Kernel.....	15
3 Metodología propuesta	16
3.1 Diseño de la metodología.....	17
3.2 Adquisición de las señales.....	19
3.2.1 Banco de pruebas para una bomba centrífuga multietapa de eje vertical	19

3.2.2	Bomba centrífuga multietapa de eje vertical	20
3.2.3	Instrumentos de medición	21
3.2.4	Condiciones de operación.....	21
3.2.5	Fallos implantados.....	22
3.2.6	Equipos para la adquisición de señales.....	23
3.2.7	Aplicación para adquisición y almacenamiento de señales	26
3.2.8	Proceso de adquisición de las señales	28
3.3	Generación de conjunto de datos.....	32
3.3.1	Selección de datos para entrenamiento, validación y test	33
3.3.2	Validación cruzada.....	34
3.3.3	Obtención subseñales.....	34
3.3.4	Extracción de características.....	36
3.4	Construcción de los modelos	43
3.4.1	Red generativa adversaria	43
3.4.2	WGAN-GP	43
3.4.3	Arquitectura del generador y crítico.....	46
3.4.4	Entrenamiento de la red generativa adversaria	47
3.4.5	Computo de métricas de error	47
3.4.6	Selección del mejor modelo de crítico.....	49
3.4.7	Optimización de hiperparámetros del clasificador OC-SVM	50
3.4.8	Prueba clasificador OC-SVM optimizado	50
4	Sistema de detección online de fallos en la nube	52
4.1	Desarrollo.....	53
4.2	Análisis.....	53
4.3	Diseño.....	55
4.3.1	Arquitectura.....	55
4.4	Implementación.....	56
4.4.1	Local: Adquisición y almacenamiento de datos.....	56
4.4.2	Cloud: Funciones lambda	57
4.5	Verificación.....	62
4.6	Mantenimiento.....	63
5	Análisis de resultados	64
5.1	Resultados del proceso de entrenamiento de la WGAN-GP	65
5.2	Resultados del proceso de optimización de los hiperparámetros del clasificador OC-SVM.....	70
5.3	Resultados de las pruebas con datos de prueba.....	72
5.4	Resultados de la comparación con otras metodologías para extracción de características	74
5.5	Resultados de funcionamiento del sistema de detección online	

de fallos.....	77
6 Conclusiones y recomendaciones	80
Glosario	90

Índice de figuras

2.1	Técnicas de mantenimiento [Luo et al., 2013]	8
2.2	Arquitectura GAN	10
2.3	Máquinas de soporte vectorial para la clasificación binaria.	11
2.4	SVM de una sola clase.....	14
3.1	Esquema de la metodología propuesta.	18
3.2	Diagrama de tuberías e instrumentación (P&ID) del banco de pruebas para bombas.	20
3.3	Esquema 3D del banco de pruebas de la bomba centrífuga.	20
3.4	Partes del banco de pruebas de la bomba centrífuga multietapa.	21
3.5	Corte 3/4 bomba centrífuga multietapa de eje vertical.	23
3.6	Partes principales de un impulsor de una bomba centrífuga multietapa.....	24
3.7	Distribución de instrumentos de medición.	25
3.8	Codificación tipos de fallos en impulsores.	26
3.9	Esquema de conexión del sistema de adquisición de señales.	27
3.10	Chasis cDAQ-9188.	28
3.11	Interfaz de usuario para la aplicación para adquisición y almacenamiento de las señales.	30
3.12	Diagrama de flujo de la aplicación de adquisición y almacenamiento de datos.....	31
3.13	Flujo de datos a través de pasos de la metodología propuesta.	33
3.14	Conversión de ciclos a número de muestras.	35
3.15	División de una señal.	35
3.16	Esquema de una GAN (a) vs WGAN-GP (b).	45
3.17	Arquitectura Generador (a) Arquitectura Crítico (b).	46
3.18	Arquitectura crítico modificado.....	48
3.19	Proceso de búsqueda aleatoria.	51

4.1 Fases metodología en cascada.	54
4.2 Arquitectura para el sistema de detección online de fallos.	55
4.3 Interfaz de usuario de la aplicación para el sistema de detección online de fallos.	57
4.4 Diagrama de flujo de la aplicación.	58
4.5 Diagrama de flujo de la función "Extracción de características".	60
4.6 Diagrama de flujo de la función "Detección de fallo".	61
4.7 Panel para monitoreo del estado de la bomba centrífuga multietapa.	62
5.1 Curvas de entrenamiento WGAN-GP.	66
5.2 Curva de entrenamiento WGAN-GP para la iteración 1.	67
5.3 Porcentaje de falsos negativos y falsos positivos alcanzados durante el entrenamiento para todas las iteraciones.	68
5.4 Porcentaje de falsos negativos y falsos positivos alcanzados durante el entrenamiento para la iteración 1.	69
5.5 Análisis PCA de las características extraídas durante el entrenamiento para la iteración 1.	70
5.6 Señales generadas durante el entrenamiento.	71
5.7 Comparación señales generadas y señales reales.	71
5.8 Gráfica de barras del análisis de error tipo 1 y 2.	75
5.9 Gráfica comparativa de los resultados alcanzados.	77
5.10 Funcionamiento del sistema de detección online de fallos.	78
5.11 Tiempo de ejecución de funciones lambda.	78

Índice de tablas

3.1 Principales datos técnicos de la bomba multietapa de eje vertical 3SV10GE4F20.....	22
3.2 Características sensores.....	24
3.3 Condiciones de operación.....	26
3.4 Fallos implantados en la bomba centrífuga multietapa.....	27
3.5 Conexión módulos e instrumentos de medición.....	28
3.6 Características módulos utilizados.....	29
3.7 Parámetros para realizar el proceso de adquisición de las señales.	30
3.8 Combinación utilizada para el proceso de adquisición de las señales.	32
3.9 Lista repeticiones utilizadas para cada experimento.....	34
3.10 Características en el dominio del tiempo.....	36
3.11 Características en el dominio de frecuencia.....	41
3.12 Características en el dominio tiempo-frecuencia.....	42
5.1 Mínimo porcentaje de falsos positivos alcanzados en cada iteración.....	67
5.2 Resultados obtenidos para el proceso de optimización de los hiperparámetros del clasificador OC-SVM.....	72
5.3 Resultados de exactitud, FN y FP alcanzada con la metodología propuesta.	73
5.4 Resultados FP para P2-P7.....	73
5.5 Resultados FP para P8-P13.....	74
5.6 Resultados del análisis de error tipo 1 y 2 con un nivel de significancia del 5 %.....	74
5.7 Combinaciones utilizadas para los experimentos.	75
5.8 Resultados de experimentos utilizando distintos extractores de características.....	76

Resumen

La propuesta de técnicas y metodologías enfocadas para tareas de detección y diagnóstico de fallos en máquinas constituye un reto fundamental en el área industria. Por tal motivo, el uso de nuevas tecnologías emergentes que puedan utilizarse para estas tareas es un valioso aporte para el desarrollo industrial. Con esto en cuenta, en el presente trabajo se detalla el proceso de desarrollo de una nueva metodología para la creación de detectores de fallos basados en el uso de modelos generativos obtenidos a partir de señales multimodales de una máquina funcionando en estado normal.

Para realizar esta tarea se obtienen señales de vibración de la máquina para entrenar una [WGAN-GP](#). Una vez entrenada la red se utiliza el modelo del crítico como extractor de características. A partir de las características extraídas por el modelo se entrena un clasificador basado en máquinas de soporte vectorial de una sola clase ([OC-SVM](#)) para realizar la detección de los fallos.

Esta metodología se aplicó para la detección de fallos en una bomba centrífuga multietapa, y fue comparada con otras propuestas en las cuales se utilizó métodos tradicionales para la extracción de características de las señales de vibración. Los resultados muestran que la nueva metodología puede ser utilizada para la tarea de detección de fallos en una bomba centrífuga multietapa vertical debido a que el desempeño alcanzado por esta fue superior a los métodos tradicionales.

Adicionalmente, se implementó un sistema de detección online de fallos en la nube utilizando los modelos resultantes de la aplicación de la metodología. Este sistema fue implementado en la plataforma de AWS y puesto a prueba con datos obtenidos de la bomba centrífuga trabajando en estado normal y estado con falla.

Abstract

The proposal of techniques and methodologies focused on tasks of fault detection and diagnosis in machines is a fundamental challenge in the industry area. For this reason, the use of new emerging technologies that can be used for these tasks is a valuable contribution to industrial development. With this in mind, this paper details the process of developing a new methodology for the creation of fault detectors based on the use of generative models obtained from multimodal signals of a machine operating in normal state.

To perform this task, vibration signals are obtained from the machine to train a [WGAN-GP](#). Once the network is trained, the critic model is used as a feature extractor. From the features extracted by the model, a classifier based on one-class support vector machines ([OC-SVM](#)) is trained to perform fault detection.

This methodology was applied for the fault detection in a multi-stage centrifugal pump, and was compared with other proposals in which traditional methods were used for the feature extraction of from vibration signals. The results show that the new methodology can be used for the task of fault detection in a vertical centrifugal multistage pump because the performance achieved by it was superior to traditional methods.

Additionally, an online cloud-based fault detection system was implemented using the models resulting from the application of the methodology. This system was implemented on the AWS platform and tested with data obtained from the centrifugal pump working in normal and faulty states.

Capítulo 1

Introducción

El avance tecnológico en los últimos años ha permitido el desarrollo de herramientas cada vez más avanzadas empleadas en distintos campos dentro de la industria. Uno de los campos que más se ha beneficiado con estos avances ha sido el mantenimiento basado en la condición el cual se enfoca en monitorizar la condición operativa de las máquinas, equipos y sistemas de la planta para optimizar el funcionamiento total de esta.

Para lograr este cometido una de las tareas que se realiza es la detección y diagnóstico de fallos, siendo la detección la primera tarea realizar. Una vez detectado el fallo se procede con el diagnóstico en el cual se somete las características y los cambios producidos en la planta a un procedimiento de clasificación, con el fin de determinar el tipo y severidad del fallo, la localización del mismo, así como la causa por la cual se produjo. Para lograr dichas tareas se han desarrollado distintos métodos y técnicas que combinan el procesamiento de datos, el análisis estadístico y la creación de modelos matemáticos capaces de estimar el comportamiento del sistema.

Es así como en el presente trabajo se propone una metodología para la creación de modelos de detección de fallo a partir de señales multimodales en condición normal de una bomba hidráulica centrífuga multietapa vertical. Esta metodología está enfocada a utilizar solamente señales adquiridas en estado normal de la bomba centrífuga, debido que en la mayoría de sistemas es complicado adquirir señales en estado con falla o simplemente no se puede adquirir dichas señales por la naturaleza del sistema.

1.1 Descripción general del problema

La bomba centrífuga multietapa es un equipo clave dentro de la industria de procesos. Debido a su alta eficiencia, amplio rendimiento y operación estable, se usa ampliamente en refinerías, centrales eléctricas, plantas químicas, etc. [Agüera, 2002, Kocaaslan et al., 2015]. Por tal motivo el desarrollar herramientas enfocadas a la detección y diagnóstico de fallas en este tipo de máquinas supone un gran campo de aplicación.

Trabajos previos realizados en este campo utilizan un enfoque en el cual se realiza la adquisición de señales en estado normal (sin falla) y en estado anormal (con falla implantadas). Para luego mediante un procesamiento de estas señales, se extraen indicadores de condición o características (FFT, RMS, Wavelets, etc). Con estas características obtenidas de las señales del sistema con falla y sin falla se procede a entrenar los clasificadores mediante aprendizaje supervisado. Una vez entrenados estos clasificadores se los somete a pruebas para ver el grado de acierto que tienen. Esta metodología para detección y diagnóstico de fallos es comúnmente utilizada en la mayoría de trabajos que se encuentran reportados en la literatura y son analizados en la Sección 2.1, pero su desventaja radica en disponer de señales adquiridas de la máquina sin falla y con falla.

Esto supone un gran inconveniente en sistemas en condiciones reales, en donde las máquinas operan la mayor parte del tiempo en condiciones sin fallas. Por lo tanto, obtener las señales de la condición con falla puede representar en algunos casos la espera de un tiempo prolongado y en otros simplemente no es posible su adquisición. Este inconveniente provoca que las metodologías clásicas tengan aplicabilidad limitada y no puedan ser implementadas en sistemas donde no se tiene señales adquiridas de la máquina con falla. Por esta razón el proponer metodologías con nuevos enfoques en que solamente se utilice señales provenientes de máquinas sin falla supone un gran aporte en el campo de detección y diagnóstico de fallas en máquinas.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar una metodología para la creación de modelos de detección de fallo a partir de señales multimodales en condición normal de una bomba hidráulica centrífuga multietapa vertical.

1.2.2 Objetivos específicos

- Caracterizar la condición normal de la bomba hidráulica.
- Obtener un modelado del sistema basado en datos de la distribución de probabilidad subyacente de la condición normal de la máquina.
- Crear una función discriminativa basado en one class learning.
- Evaluar el desempeño del modelo con la implementación de un sistema online de detección de fallos en maquinaria.

1.3 Contribuciones

Las contribuciones del presente trabajo son las siguientes:

1. El desarrollo de una nueva metodología a partir del entrenamiento de modelos generativos (GAN), los cuales serán utilizados para la detección de fallos. Estas señales provienen de una bomba centrífuga multietapa vertical trabajando en estado normal (sin falla) y bajo diferentes condiciones de funcionamiento (caudal, presión). El modelo resultante aproxima la distribución de probabilidad subyacente de la condición normal de la máquina, y mediante una función discriminativa basada en one-class learning se determina la condición de la máquina.
2. El desarrollo de un sistema online de detección de fallos en la nube basado en la nueva metodología generada. Este sistema online de fallos al utilizar una arquitectura sin servidor tiene las ventajas de ser fácilmente escalable, de no generar costos por mantenimiento de servidores o infraestructura y poseer una alta disponibilidad.
3. Comparativa de la nueva metodología desarrollada frente a otras metodologías clásicas. En la cual se compara el desempeño obtenido por el clasificador de una sola clase al utilizar para su entrenamiento las características extraídas por el modelo generado contra el desempeño al utilizar características extraídas en el dominio de tiempo, frecuencia, tiempo-frecuencia y distintas combinaciones de estas.

1.4 Organización del manuscrito

El presente documento se encuentra organizado de la siguiente manera:

El capítulo 2 se encuentra constituido por los antecedentes de la temática tratada en este proyecto como es la detección y diagnóstico de fallos en máquinas rotativas. Además, contiene la fundamentación teórica necesaria para el manejo de la temática abordando los conceptos sobre mantenimiento, redes generativas adversarias (GAN), máquinas de soporte vectorial (SVM) y su equivalente para one-class learning (OC-SVM).

En el capítulo 3 se describe el proceso y los procedimientos seguidos para generar la nueva metodología. Se realiza de la descripción de los componentes fundamentales del sistema. Luego se detalla cada uno de los procedimientos desarrollados para la generación de la metodología y a su vez se aborda conceptos teóricos más detallados sobre los modelos utilizados y las métricas de desempeño obtenidas.

En el capítulo 4 se describe el proceso para la creación e implementación del sistema de detección online de fallos en la nube. Se inicia el capítulo definiendo la metodología de desarrollo utilizada para el proceso de creación del sistema para luego continuar con la explicación a detalle de cada uno de las etapas para alcanzar la puesta en marcha del sistema de detección online de fallos en la nube.

El capítulo 5 contiene el análisis de los resultados obtenidos en cada una de las etapas de creación de la metodología y los resultados de la implementación del sistema de detección online de fallos en la nube. A su vez se presenta una comparación de los resultados de desempeño alcanzados con la metodología propuesta frente a metodologías clásicas para extracción de características.

Por último en el capítulo 6 se encuentran las conclusiones y recomendaciones obtenidas una vez finalizado el proyecto, además de los trabajos futuros que se podrían desarrollar.

Capítulo 2

Antecedentes y fundamentación teórica

El capítulo se encuentra conformado por los antecedentes sobre la detección y diagnóstico de fallos. Además, contiene la fundamentación teórica necesaria para el manejo de la temática abordando los conceptos sobre mantenimiento basado en la condición, [GAN](#), [SVM](#) y [OC-SVM](#).

2.1 Antecedentes

La detección y diagnóstico de fallos en máquinas es uno de los campos que ha tenido un gran crecimiento en los últimos años debido al avance de los procesos industriales que cada vez son más complejos e involucran un mayor número de componentes. Uno de los componentes más utilizados en este ámbito son las máquinas rotativas es por eso que el desarrollo de nuevas metodologías empleadas para la detección y diagnóstico de fallas en este tipo de máquinas es una gran contribución para el desarrollo industrial.

Se puede encontrar varios trabajos realizados en este campo. Por ejemplo, en [Kumar et al., 2018] se realiza una revisión del estado del arte de técnicas empleadas para el mantenimiento basado en la condición para la detección de fallas en rodamientos y engranajes, donde destaca el análisis de vibraciones como procedimiento más predominante para reconocer la naturaleza y el grado de cualquier problema en máquinas y componentes (por ejemplo, rodamientos y engranajes). Otros trabajos [Hoang and Kang, 2019, Li et al., 2016b, Bin et al., 2012, Li et al., 2015a] emplean señales de vibración y utilizan combinaciones de distintas técnicas de inteligencia artificial para realizar la detección de fallos en engranajes y rodamientos, obteniendo resultados satisfactorios en todos los casos. Por otra parte, trabajos como los presentados por [Cheng et al., 2019] se basan en muestreo comprimido e indicadores de condición para realizar la detección de fallos, de igual manera se puede ver el trabajo desarrollado por [Zhou et al., 2010] en donde se propone un sistema de detección de fallas basado en reglas jerárquicas que comprende una base de conocimiento junto con un motor de inferencia.

Otros métodos como el presentado en [Cabrera et al., 2018], se han enfocado en trabajar con datos en condición normal para la construcción de un modelo para detección de fallos, obteniendo resultados favorables. A su vez [Cabrera et al., 2019] plantea un método para el diagnóstico de fallos en una máquina reciprocante utilizando datos altamente desbalanceados. Ambos trabajos mencionados anteriormente se han enfocado en lidiar con las restricciones existentes en la fase de adquisición de datos.

Con respecto al trabajo realizado sobre bombas, [Rodríguez, 2016] presenta el diagnóstico de falla en las paletas de una bomba hidráulica de desplazamiento positivo mediante el análisis espectral de señales de vibraciones, alcanzando resultados satisfactorios en cuanto al diagnóstico del fallo estudiado. De igual manera [Liu et al., 2015] propone una metodología de diagnóstico de fallas para una bomba de engranajes basada en el método de descomposición modal empírica conjunta y la red bayesiana, determinando que la metodología implantada es eficaz y eficiente para

diagnosticar fallas basadas en información incierta e incompleta. [Zhang et al., 2017] presenta un método sobre el diagnóstico de fallos en rodamientos de una bomba multietapa basado en descomposición modal variacional, en el cual se obtiene como resultado que la propuesta puede diagnosticar con éxito la falla en rodamientos. En cuanto a bombas centrífugas [Thai and Lee, 2010, Zhang et al., 2016, Luo et al., 2013] presenta estudios de los efectos de la cavitación y las fallas que produce; y [Mousmoulis et al., 2019] realiza un sistema para la detección de cavitación en diferentes impulsores de una bomba centrífuga, utilizando señales de vibración y emisión acústica. A su vez [ALTobi et al., 2019] realiza un estudio comparativo de dos métodos basados en inteligencia artificial para detección de fallos en este tipo de bombas.

Con la revisión de lo anterior se puede notar que en cuanto a detección de fallas en máquinas rotativas la literatura es amplia, y las soluciones presentadas para resolver este problema de detección son abordadas desde distintos enfoques, tales como el análisis espectral de señales de vibración, el uso de inteligencia artificial para el análisis de señales y el uso de sistemas expertos basados en reglas. De igual manera, con respecto a bombas se encuentran algunos trabajos realizados para la detección de fallas en estos equipos.

La Universidad Politécnica Salesiana (UPS) sede Cuenca cuenta con un laboratorio de análisis de vibraciones donde el Grupo de Investigación y Desarrollo en Tecnologías Industriales (GIDTEC) trabaja en investigaciones sobre mantenimiento basado en la condición de maquinaria industrial que apoya la industria 4.0. En este grupo de investigación se han desarrollado algunos trabajos previos para detección y diagnóstico de fallos en máquinas rotativas y máquinas reciprocantes [Li et al., 2015a,b, 2016b,d,a,c, Cabrera et al., 2017, 2018, 2019]. Además recientemente se ha implementado un banco hidráulico para pruebas de bombas centrífugas multietapa [Calderón Malla and Montalván Pulla, 2019].

2.2 Fundamentación teórica

2.2.1 Mantenimiento basado en la condición

Dentro del campo de mantenimiento existen distintas técnicas que se han ido desarrollando con el fin de lograr que las máquinas, equipos y sistemas utilizados en la industria obtengan la mayor eficiencia, trabajando bajo los parámetros de funcionamiento establecidos, por el mayor tiempo que sea posible. Una de los enfoques desarrollados es el mantenimiento preventivo el cual es realizado cada cierto intervalo de tiempo, el mismo que es establecido

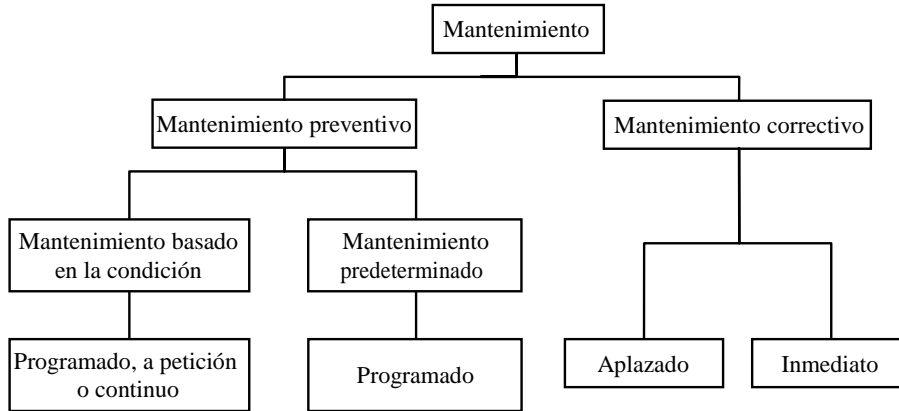


Figura 2.1: Técnicas de mantenimiento [Luo et al., 2013]

según distintos criterios. Todo esto se realiza con el propósito de reducir la probabilidad de fallas o evitar el desgaste de elementos del sistema. Otro enfoque utilizado dentro de la industria es el mantenimiento correctivo que es aplicado cuando existe una falla en una máquina o equipo del proceso. El mismo se lo realiza inmediatamente cuando la falla ocurre [García, 2013].

El mantenimiento basado en la condición es una estrategia de mantenimiento que se enfoca en utilizar la condición operativa de las máquinas, equipos y sistemas de la planta para optimizar el funcionamiento total de la planta [Tsang, 1995]. Para poder realizar esta tarea el mantenimiento predictivo utiliza las herramientas más rentables (por ejemplo, la monitorización y análisis de vibraciones, la termografía, la tribología) para obtener la condición operativa real de los sistemas críticos de la planta y, en base a estos datos reales, programa todas las actividades de mantenimiento en función de las necesidades reales. A partir del análisis y evaluación de estas variables encontrar alteraciones que permitan tener un indicio de la existencia de posibles problemas en la máquina o equipo [Moblely, 2002, García, 2013, CEN, 2010].

Para implementar el mantenimiento basado en la condición la mayoría de las industrias han optado por utilizar métodos de detección y diagnóstico de fallos en máquinas, utilizando datos históricos del funcionamiento de la máquina. Los datos obtenidos proporcionan información eficaz que permite realizar esta tarea a través del análisis de patrones.

Para el proceso de análisis y evaluación de los datos se utilizan técnicas de procesamiento para realizar la extracción de características cuantitativas o

cuantitativas. Dentro del proceso de extracción cuantitativa se puede utilizar distintos métodos como sistemas de redes neuronales y técnicas estadísticas. Una de las metodologías más utilizadas para la detección y diagnóstico de fallas en máquinas rotativas es el análisis de señales de vibración, en el cual a partir del análisis del espectro de vibración de cada máquina se puede determinar los patrones de comportamiento de la condición normal. Al presentarse una falla en la máquina analizada la forma del espectro de vibración se vería alterado. Para este tipo de metodología se utiliza acelerómetros para captar las señales de vibración provenientes de la máquina [White, 2010, Chen et al., 2015].

2.2.2 Red generativa adversaria (GAN)

Una red generativa adversaria (GAN por sus siglas en inglés) es un tipo de red neuronal en el cual se propone una arquitectura que emplea dos modelos uno denominado modelo generativo y otro denominado modelo discriminativo, los cuales se enfrentan entre sí. Durante el proceso de entrenamiento el modelo discriminativo aprende a determinar si una muestra proviene de la distribución del modelo generativo o de la distribución de los datos reales, mientras que el modelo generativo aprende a generar mejores muestras para tratar de engañar al modelo discriminativo. Esta competencia entre ambos ayuda a mejorar los modelos durante el proceso de entrenamiento.

La GAN original [Goodfellow et al., 2014] propone utilizar la arquitectura de una perceptrón multicapa tanto para el modelo generativo como para el discriminativo debido a de que su análisis es más sencillo. En el caso del modelo generativo (G) para aprender su distribución P_g sobre los datos x , este recibe como entrada z que posee una distribución conocida P_z , obteniendo $x = G(z)$. Mientras que para el modelo discriminativo (D) se tiene como salida un escalar, en el cual $D(x)$ representa la probabilidad de que x provenga de la distribución de datos reales P_{data} en lugar de P_g . Durante el entrenamiento de la GAN se trata de mejorar a D para que sea capaz de reconocer si los datos de entrada son reales o son generados por G, además de simultáneamente se entrena a G para que pueda engañar a D. Este proceso se lo conoce como el juego de mínimo-máximo descrito formalmente como sigue:

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}(x)} [\log D(x)] + E_{z \sim P_z(z)} [\log D(1 - D(G(z)))] \quad (2.1)$$

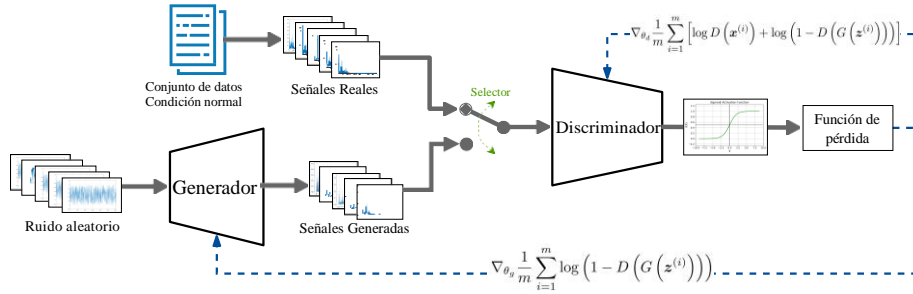


Figura 2.2: Arquitectura GAN.

Para el proceso de entrenamiento de la GAN se entrena el discriminador k veces por cada iteración de entrenamiento del generador. Esto hace que D se mantenga cerca de su solución óptima, siempre y cuando G cambie lo suficientemente lento.

En la Figura 2.2 se presenta la arquitectura de una GAN tomando como entrada la representación temporal de señales del problema particular que se aborda en este trabajo. En donde se puede observar el modelo generador (G) y el modelo discriminador (D), además de sus funciones de pérdida correspondientes para el entrenamiento de cada modelo. Como se revisó anteriormente el modelo generador se encarga de generar señales a partir de ruido aleatorio, estas señales generadas son clasificadas por el discriminador y los resultados de la clasificación sirven al generador para que este pueda ir mejorando en la tarea de generar señales que se aproximen más a las señales reales durante el proceso de entrenamiento. A su vez el discriminador al tener acceso a las señales reales puede aprender a discriminar de mejor manera las señales generadas de las reales. De esta manera durante el proceso de entrenamiento ambos modelos mejoran en su tarea correspondiente.

2.2.3 Máquinas de soporte vectorial (SVM)

Las máquinas de soporte vectorial (SVM por sus siglas en inglés) presentadas por primera vez por [Cortes and Vapnik, 1995] son un tipo de algoritmo que utiliza un tipo de aprendizaje supervisado empleado originalmente como un método de clasificación binaria, extendiendo su aplicación para clasificación múltiple y regresión. Se utiliza SVM para tareas de clasificación binaria en donde se separan un determinado conjunto de datos de entrenamiento etiquetados pertenecientes a dos clases diferentes, para esto se utiliza un hiperplano que se encuentra a la mayor distancia posible de ambas clases y

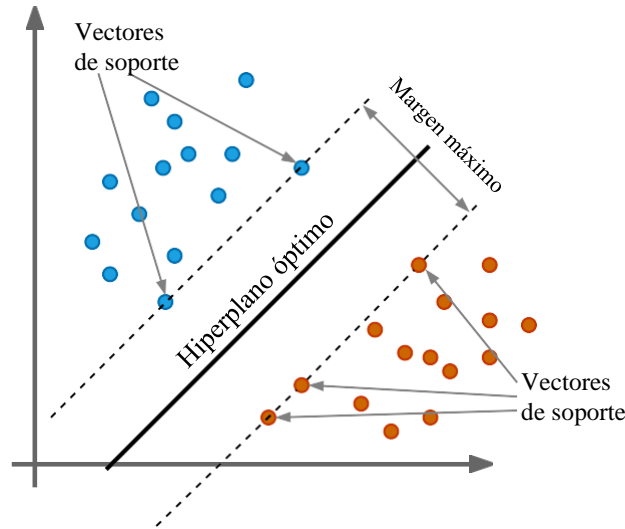


Figura 2.3: Máquinas de soporte vectorial para la clasificación binaria.

al cual se lo denomina hiperplano de margen máximo [Cortes and Vapnik, 1995, Cristianini and Shawe-Taylor, 2000].

Al tener dos clases que no son linealmente separables en el espacio de entrada original se puede utilizar la técnica de “kernels” que permite automáticamente realizar un mapeo no lineal a un espacio de características donde la separación exista. El hiperplano encontrado por SVM en el espacio de características corresponde a un límite de decisión no lineal en el espacio de entrada original. Con un mapeo no lineal adecuado a una alta dimensión los datos pertenecientes a dos clases pueden ser siempre separados por un hiperplano.

En caso de tener dos clases que son linealmente separables (Figura 2.3) en el cual se considera que se tiene un conjunto de datos $\mathcal{X} \in \mathbb{R}^n$ la ecuación del hiperplano óptimo viene dado por

$$w^T x + b = 0 \tag{2.2}$$

donde w es un vector de n dimensiones y b es el bias del hiperplano. Este hiperplano debe cumplir con las condiciones de tener el menor error posible al separar las dos clases y la distancia entre los datos más cercanos de cada clase debe ser maximizada [Steinwart and Christmann, 2008]. Bajo estas condiciones se tiene que los datos de una clase se encontraran a la derecha

del hiperplano y los datos de la otra clase se encontraran a la izquierda. Por lo tanto los dos planos están definidos por:

$$\begin{aligned} w^T x + b &> 1 \text{ para } y=1 \\ w^T x + b &\leq -1 \text{ para } y=-1 \end{aligned}$$

Sin embargo la región de generalización del hiperplano puede estar entre 1 y -1 y existen varios márgenes que pueden ser considerados como el límite para cada clase. Por lo tanto para encontrar el hiperplano óptimo la distancia (d) entre los márgenes debe ser medida y optimizada usando la siguiente ecuación:

$$d(w, b, x) = \frac{(w^T x + b - 1) - (w^T x + b + 1)}{\|w\|} = \frac{2}{\|w\|} \quad (2.3)$$

A su vez, maximizar el margen equivaldría a minimizar el vector w que puede ser escrito como $\frac{1}{2}w^T w$. Entonces el problema para determinar el hiperplano óptimo esta definido como:

$$\min_{w,b} = \frac{1}{2}w^T w \quad (2.4)$$

sujeto a

$$y(w^T x + b) \geq 1$$

Utilizando multiplicadores de Lagrange se puede hacer cumplir la restricción obteniendo:

$$L_p(w, b, a) = \frac{1}{2}w^T w - \sum_{i=1}^N a_i \cdot y_i (w^T x_i + b - 1) \quad (2.5)$$

Para encontrar el punto estacionario de la ecuación anterior, deben cumplirse las siguientes condiciones de Karush-Kuhn-Tucker (KKT):

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w_0 = \sum_{i=1}^N a_i x_i y_i \quad (2.6)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^N a_i y_i = 0 \quad (2.7)$$

Cabe señalar que a_i no será igual a cero si y sólo si sus correspondientes datos de entrada (x_i) son un vector de apoyo. Los vectores de apoyo son los datos elegidos como el límite de cada clase a través del cual se puede encontrar el margen de la clase. Finalmente, al sustituir la Ecuaciones 2.6 y 2.7 en la Ecuación 2.5 se obtiene la ecuación general del clasificador SVM para un caso linealmente separable que estaría sujeto a dos restricciones que se muestra a continuación.

$$\max_d L(a) = \sum_{i=1}^N a_i - \frac{1}{2} \sum_{i,j=1}^N y_i y_j a_i a_j x_i^T x_j \quad (2.8)$$

sujeto a

$$\sum_{i=1}^N a_i \geq 0$$

$$\sum_{i=1}^N a_i y_i = 0$$

La ecuación anterior y sus limitaciones son utilizadas por el clasificador SVM para encontrar los vectores de apoyo y sus correspondientes datos de entrada. El parámetro dos del hiperplano (función de decisión) puede entonces obtenerse a partir de la Ecuación 2.6 y el parámetro b puede calcularse a partir de la siguiente ecuación escrita en la forma promedio:

$$b = \frac{1}{N} \sum_{S=1}^N (y_S - w^T x_S) \quad (2.9)$$

2.2.4 One-class SVM

SVM de una sola clase [Schölkopf et al., 2001] (OC-SVM por sus siglas en inglés) es una extensión del algoritmo de máquinas de soporte vectorial para abordar el problema de aprendizaje de una clase. Este método tiene el objetivo de encontrar el hiperplano óptimo que separe el conjunto de datos de la clase disponible del origen. Para cumplir este objetivo el algoritmo de OC-SVM realiza un mapeo de los datos a un espacio de características de alta dimensión empleando para esto un kernel. Trabajando en el espacio de características el algoritmo de OC-SVM iterativamente encuentra el hiperplano de margen máximo que mejor separe los datos de entrenamiento del origen. Análogamente se puede considerar a OC-SVM como un SVM de clasificación binaria que toma los datos de entrenamiento como pertenecientes a la primera clase y al origen como único dato de la segunda clase [Li et al., 2020, Heller et al., 2003].

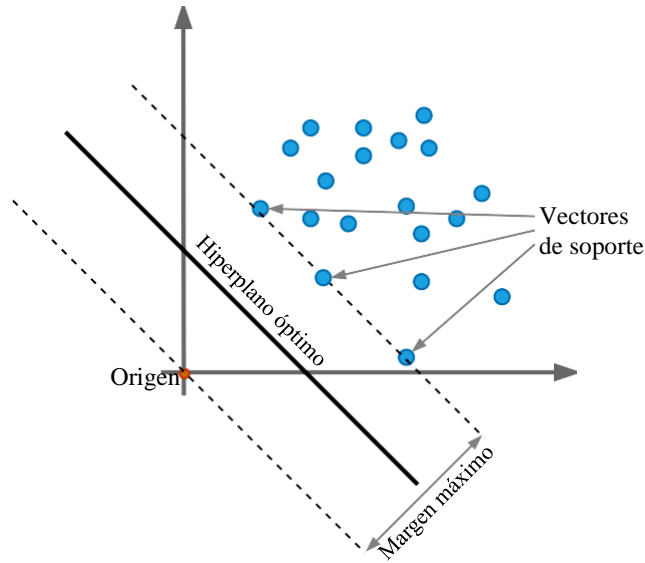


Figura 2.4: SVM de una sola clase.

Sea $x = (x_1, \dots, x_n)$ y pertenezca a un conjunto de datos $X \in \mathbb{R}^n$. Sea $\varphi: \mathbb{R}^n \rightarrow \mathbb{R}^N$ el mapeo del espacio de entrada I al espacio de características F . Así, su función de decisión esta dada por:

$$f(x) = \text{sign}(\langle w, \varphi(x) \rangle - b) \tag{2.10}$$

donde w es el vector normal y b es el bias de un hiperplano en el espacio de características.

Se utiliza esta regla de clasificación para asignar una etiqueta a un dato de prueba x . Al utilizar la función signo cuando $\langle w, \varphi(x) \rangle - b \geq 0$ la función de decisión retorna un 1, caso contrario retorna un -1.

Como se puede ver en [Heller et al., 2003] para encontrar el hiperplano óptimo que separe los datos de entrenamiento del origen se tiene que resolver el siguiente problema de optimización

$$\min_{w, b, \xi} \frac{1}{2} \|w\|_2^2 + \frac{1}{\nu} \sum_{i=1}^n \xi_i - b \tag{2.11}$$

sujeto a

$$\begin{aligned} \langle w, \varphi(x_i) \rangle &\geq b - \xi_i \text{ para } i = 1, \dots, \\ n \xi_i &\geq \text{ para } i = 1, \dots, n \end{aligned}$$

donde ξ_i es la variables de holgura que permite la presencia de ejemplos anómalos en el entrenamiento establecido a través de penalizarlos en la función objetivo y ν es el hiperparámetro que limita la fracción de ejemplos de entrenamiento clasificados como anómalos.

Usando multiplicadores de Lagrange y una función kernel para los cálculos de producto punto, la función de decisión se convierte en:

$$f(x) = \text{sign} \left(\sum_{i=1}^n a_i k(x, x^i) - b \right) \tag{2.12}$$

2.2.5 Kernel

La función $k(x, x^i) = \langle \varphi(x^i), \varphi(x) \rangle$, se conoce como la función kernel. Como el resultado de la función de decisión sólo depende del producto punto de los vectores en el espacio de características F, no es necesario realizar una proyección explícita a ese espacio, siempre que una función k pueda utilizarse en su lugar. Esto se conoce como “el truco del kernel” y es lo que da al algoritmo SVM la potencia para separar puntos de datos no separables linealmente en el espacio de entrada. El espacio de características F puede ser de dimensión infinita y por lo tanto el hiperplano que separa los datos puede ser muy complejo [Amari and Wu, 1999].

Las opciones más populares para la función del kernel son:

Lineal: $k(x, x^i) = x \cdot x^i$

Coseno: $k(x, x^i) = \frac{x \cdot x^i}{\|x\| \|x^i\|}$

Laplaciana: $k(x, x^i) = e^{-\gamma \|x - x^i\|_1}$

Función de base radial gaussiana: $k(x, x^i) = e^{-\gamma \|x - x^i\|^2}$

donde γ es el hiperparámetro del kernel y $\|x - x^i\|$ es la medida de la disimilitud.

Capítulo 3

Metodología propuesta

En este capítulo se describe el proceso y los procedimientos seguidos para generar la nueva metodología. Partiendo de la descripción de los componentes fundamentales del sistema. Luego se detalla cada uno de los procedimientos desarrollados para la generación de la nueva metodología y a su vez se aborda conceptos teóricos más detallados sobre los modelos utilizados y las métricas de desempeño obtenidas.

3.1 Diseño de la metodología

Con la revisión de la literatura se pudo notar que los métodos más utilizados para la detección de fallos en máquinas rotativas utilizan una metodología basada en la adquisición de señales en estado normal (sin falla) y en estado anormal (con falla implantadas) del funcionamiento de la máquina. Luego, mediante un posprocesado de estas señales extraer indicadores de condición o características. Con estas características obtenidas de las señales del sistema con falla y sin falla se procede a entrenar los clasificadores mediante aprendizaje supervisado. Una vez entrenados estos clasificadores se los somete a pruebas para ver el desempeño que tienen.

Esta metodología clásica o tradicional tiene algunas desventajas entre las que se encuentran:

1. Se debe disponer de señales adquiridas de la máquina funcionando sin falla y con falla, esto representa un gran problema, ya que, en condiciones reales las máquinas operan en estado sin falla, y para adquirir las señales en estado con falla supone la espera de un tiempo prolongado o simplemente no es posible su adquisición.
2. Para la extracción de características y la selección de las mismas es necesario conocer a fondo el funcionamiento de la máquina debido al desempeño del sistema de detección o diagnóstico de fallos depende en gran medida de la correcta selección de estas características, repercutiendo además en los requerimientos computacionales y en el tiempo de trabajo de entrenamiento, ejecución y prueba.

Por estas razones, se optó por proponer una nueva metodología enfocada en utilizar solamente señales provenientes de máquinas trabajando en estado normal. En la Figura 3.1 se puede observar un esquema de la metodología propuesta, que consiste en 5 etapas principales:

1. Adquisición de señales en estado normal de la máquina y generación de los conjuntos de entrenamiento, validación y test.
2. Preprocesamiento de las señales en el dominio del tiempo mediante la transformada rápida de Fourier para obtener las señales en el dominio de frecuencia.
3. Entrenamiento de una [WGAN-GP](#) para llegar a obtener un modelo del crítico que aproxima la distribución de probabilidad subyacente al estado normal de la máquina. A partir de este modelo se genera una

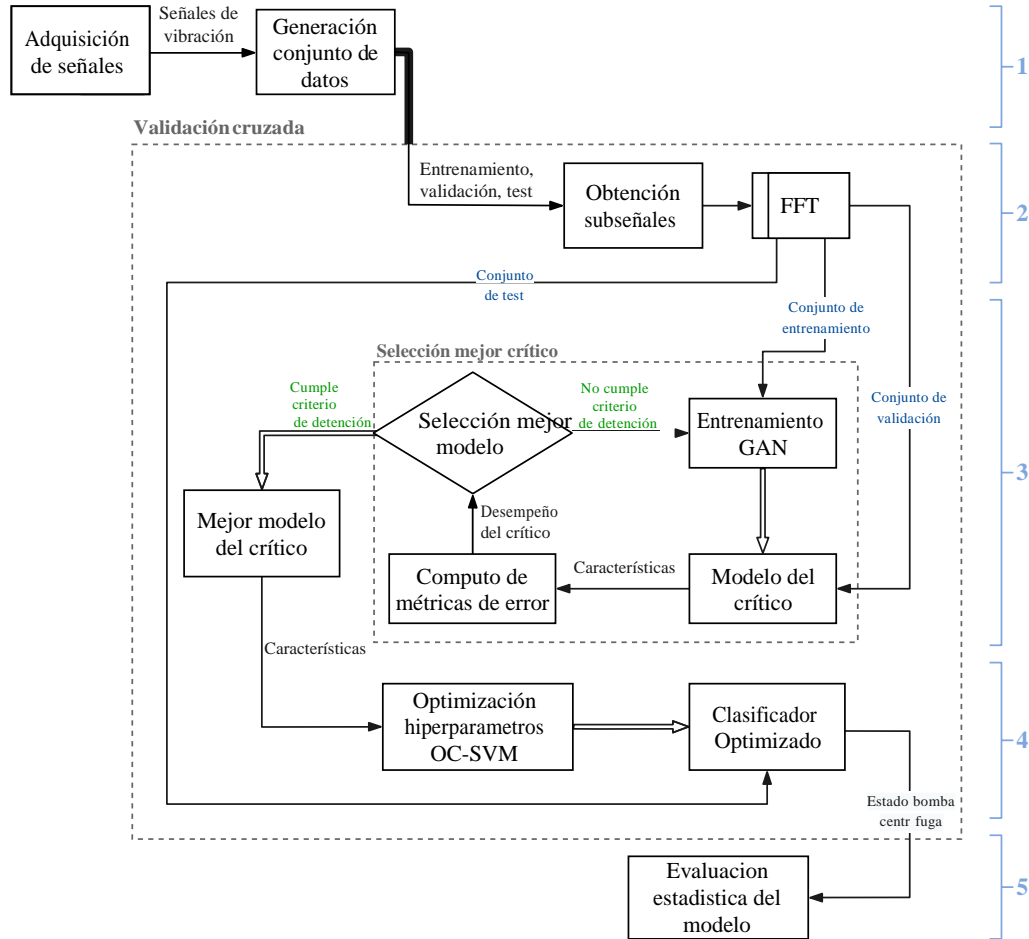


Figura 3.1: Esquema de la metodología propuesta.

función discriminativa basada en one-class learning (**OC-SVM**) capaz de realizar la detección de fallos en la máquina.

4. Optimización de los modelos del clasificador de una sola clase.
5. Evaluación estadística del modelo mediante el método de validación cruzada.

La metodología se diseñó para realizar la detección de fallos en una bomba centrífuga multietapa de eje vertical. Concretamente se trabajó con

señales de vibración adquiridas por medio de un acelerómetro. Con las señales adquiridas se realizó el preprocesamiento de las mismas para ser utilizadas en el entrenamiento y posterior optimización de los modelos. Luego estos modelos del crítico y del clasificador obtenidos se pusieron a prueba en una primera instancia en un sistema offline con datos previamente adquiridos, y luego sobre un sistema online de detección de fallos en la nube.

A continuación se indicará a detalle cada uno de los pasos que componen la metodología propuesta.

3.2 Adquisición de las señales

Los datos para el conjunto original fueron adquiridos de un banco de pruebas para una bomba centrífuga multietapa de eje vertical trabajando en estado normal y en estado con fallo, bajo diferentes condiciones de funcionamiento (caudal, presión). Para la adquisición de los datos se utilizó distintos instrumentos de medición entre los cuales se encuentran un tacómetro, cuatro acelerómetros, dos sensores de emisión acústica, tres pinzas de corriente, dos micrófonos capacitivos, dos sensores de presión y un medidor de flujo, que a su vez se encontraban conectados a una tarjeta de adquisición de datos (NI cDAQ). A través de un programa desarrollado en el software LabView se realizó la adquisición de las señales provenientes de los instrumentos de medición. Estas señales son almacenadas en archivos con formato .mat para posteriormente poder ser utilizadas en distintos procesos de experimentación.

A continuación se describe a detalle la configuración de los elementos utilizados en el proceso de obtención de datos.

3.2.1 Banco de pruebas para una bomba centrífuga multietapa de eje vertical

La Universidad Politécnica Salesiana Sede Cuenca, en su laboratorio de vibraciones mecánicas cuenta con un banco de pruebas para una bomba centrífuga multietapa de eje vertical, el cual se encuentra instrumentado para poder realizar el control y la adquisición de distintas señales procedentes del sistema de bombeo.

Además, cuenta con distintos repuestos y accesorios para poder implantar diferentes fallos mecánicos en la bomba centrífuga. En la Figura 3.2 se puede observar el diagrama de tuberías e instrumentación del banco de pruebas. Mientras que en la Figura 3.3 y Figura 3.4 se observa un modelo 3D del banco y la distribución física de los elementos que lo conforman.

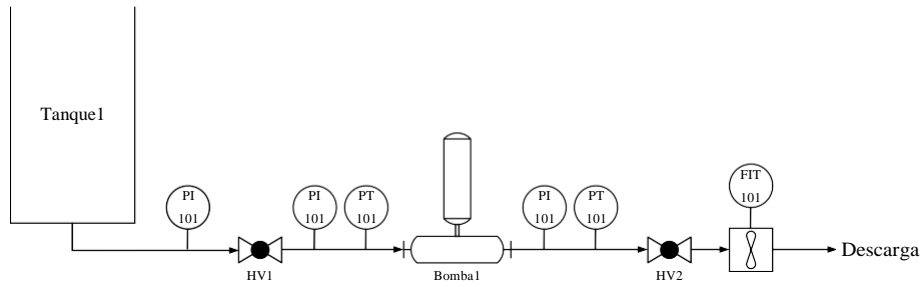


Figura 3.2: Diagrama de tuberías e instrumentación (P&ID) del banco de pruebas para bombas.

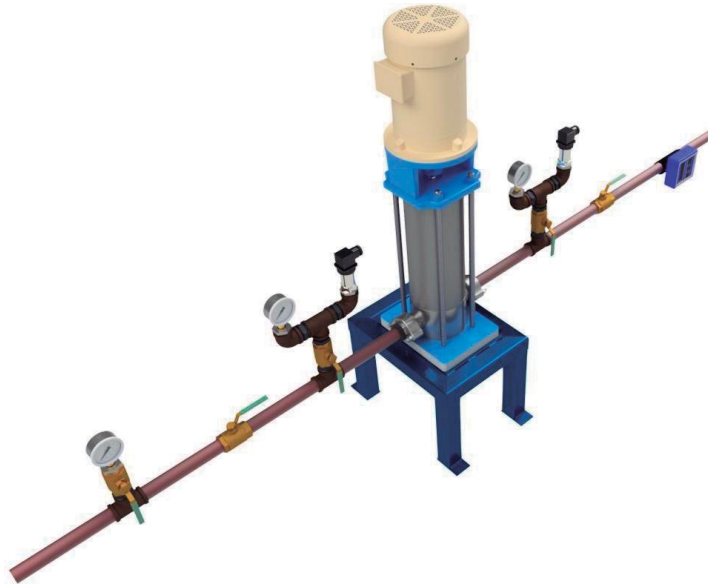


Figura 3.3: Esquema 3D del banco de pruebas de la bomba centrífuga.

3.2.2 Bomba centrífuga multietapa de eje vertical

Como componente principal del banco se encuentra la bomba centrífuga multietapa de eje vertical 3SV10GE4F20 ITT- GOULDS PUMPS, cuyos datos técnicos se encuentran descritos en la Tabla 3.1, mientras que en la Figura 3.5 se observa un corte en 3/4 donde se especifica cada una de sus

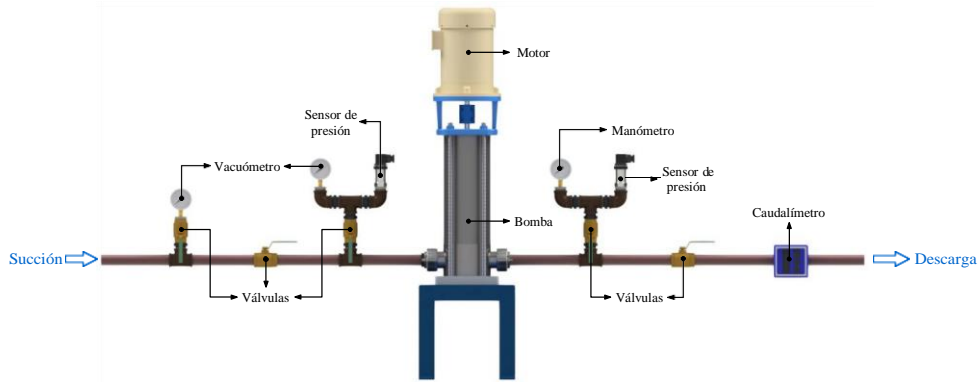


Figura 3.4: Partes del banco de pruebas de la bomba centrífuga multietapa.

partes internas además de los niveles.

Una de las partes primordiales de una bomba centrífuga multietapa son los impulsores que se encuentran en cada una de las etapas. En la Figura 3.6 se puede observar las partes principales de un impulsor.

3.2.3 Instrumentos de medición

El banco de pruebas consta de varios sensores y transductores empleados para captar distintas señales provenientes del sistema de bombeo. En la Tabla 3.2 se detalla cada uno de los instrumentos de medición utilizados, así como algunas de sus características. Mientras que en la Figura 3.7 se observa la distribución de los instrumentos de medición en el banco de bombas.

3.2.4 Condiciones de operación

Para el proceso de adquisición de las señales se plantearon 6 condiciones de operación del sistema de bombeo, que tienen que ver con el porcentaje de apertura de la válvula de descarga (HV2). Inicialmente el porcentaje de apertura se regula con la presión obtenida a la descarga de la bomba, la apertura de la válvula se ha marcado cuando coincide con las presiones indicadas en la Tabla 3.3. Es importante tener en cuenta que esta regulación y determinación del porcentaje de apertura de la válvula se la realizó al momento de adquirir las señales del sistema de bombeo trabajando en estado normal (sin fallos implantados en la bomba), cuando se adquirió las señales

Tabla 3.1: Principales datos técnicos de la bomba multietapa de eje vertical 3SV10GE4F20.

Modelo	3SV10GE4F20
Tipo	centrífuga flujo axial – eje vertical
Potencia	2 HP (1.5 KW)
Alimentación del motor	Trifásico 208-230/460 V
Velocidad del motor	3500 rpm
Tipo de transmisión	Acople de aluminio
Numero de etapas	10 etapas
Flujo nominal	15 GPM
Rango de flujo	3-22 GPM
Máxima cabeza	1085 pies
Máxima presión de trabajo	580 PSIG
Presión en la descarga	10.7 BAR
Rango de temperatura	-30°C-121°C
Eficiencia máxima de la bomba	60 %

con el sistema trabajando en estado con falla (con fallos implantados en la bomba), estas presiones varían debido a la presencia de estos fallos.

3.2.5 Fallos implantados

Para el experimento propuesto se utilizaron 4 diferentes tipos de fallos: picadura a la entrada de los álabes, picadura a la salida de los álabes, canal del impulsor bloqueado e impulsor desbalanceado, que ocurren en los impulsores de una bomba multietapa, cada uno de estos con 8 niveles de severidad. Estos fallos se distribuyeron en las últimas 5 etapas de la bomba (etapas 5-10) bajo diferentes combinaciones, mientras que el resto de etapas se encuentran en estado normal. En la Figura 3.8 se muestra la codificación utilizada para los fallos y en la Tabla 3.4 se describe la combinación de fallos que se utilizaron en el experimento.

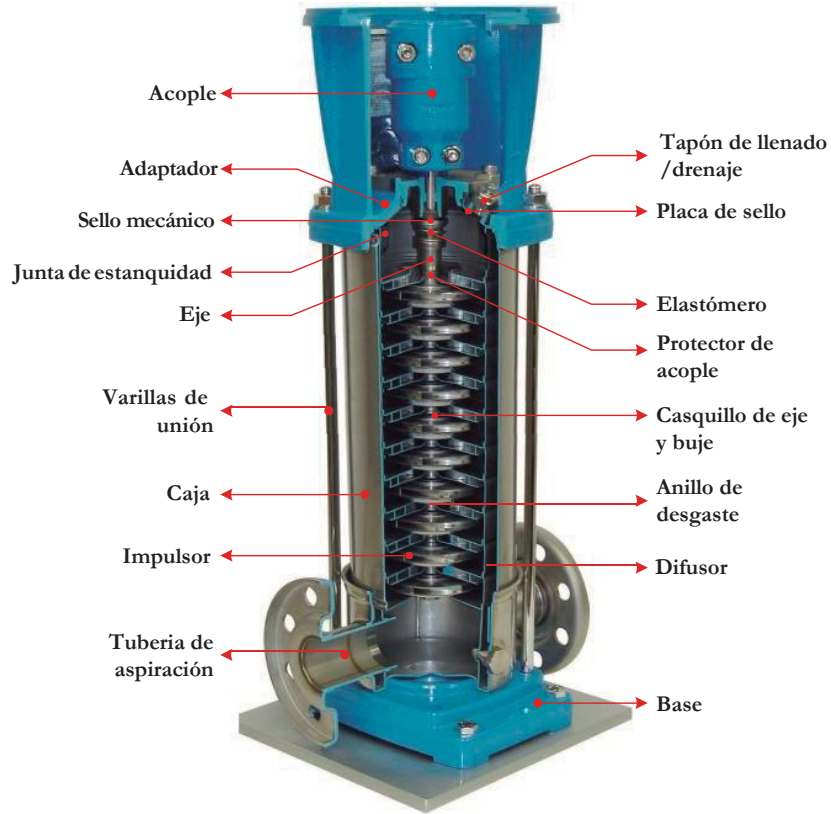


Figura 3.5: Corte 3/4 bomba centrífuga multietapa de eje vertical.

3.2.6 Equipos para la adquisición de señales

Par el proceso de adquisición de las señales se utilizó como componente principal un chasis cDAQ-9188 que permite acoplar distintos módulos de entrada y salida a los cuales se conectan los diferentes instrumentos de medición y a su vez esta tarjeta se conecta al computador a través de una conexión Ethernet. En la Figura 3.9 se puede apreciar un esquema de conexión del sistema de adquisición.

El cDAQ-9188 que se muestra en la Figura 3.10 es un chasis CompactDAQ Ethernet con soporte para 8 módulos de E/S, está diseñado para ser utilizado en sistemas pequeños y distribuidos empleados para realizar mediciones con sensores. El chasis es capaz de controlar la temporización, la sincronización



Figura 3.6: Partes principales de un impulsor de una bomba centrífuga multietapa.

Tabla 3.2: Características sensores.

Ident.	Instrumento de medición	Marca	Modelo	Serie	Rango
P1	Sensor de presión 1	SENDO SENSOR	SS302+	SSB15B05	-1 - 0.6 bar
P2	Sensor de presión 2	SENDO SENSOR	SS302+	SSB15B05	0 - 25 bar
F1	Caudalímetro 1	BURKERT	-	11267	
Mic1	Micrófono 1	PCB	HT378B02	135419	0 - 138,5dB
Mic2	Micrófono 2	PCB	HT378B02	135419	0 - 138,5dB
CVC1	Pinza amperométrica 1	FLUKE	i30s	34340055	5mA - 30A
CVC2	Pinza amperométrica 2	FLUKE	i30s	34340055	5mA - 30A
CVC3	Pinza amperométrica 3	FLUKE	i30s	34340055	5mA - 30A
A1	Acelerómetro 1	IMI SENSORS	603C01	LW321689	±50g
A2	Acelerómetro 2	IMI SENSORS	603C01	LW321689	±50g
A3	Acelerómetro 3	IMI SENSORS	603C01	LW321689	±50g
A4	Acelerómetro 4	IMI SENSORS	603C01	LW321689	±50g

y la transferencia de datos entre los módulos de E/S y un servidor externo, además también tiene cuatro contadores/temporizadores de 32 bits de uso general.

En el sistema de adquisición propuesto se utilizaron seis módulos de cuatro tipos diferentes (NI 9234, NI9223, NI 9401 y NI 9205). En la Tabla 3.5 se detalla los módulos empleados para la adquisición de señales, los instrumentos conectados a cada uno de estos, y la frecuencia de muestreo

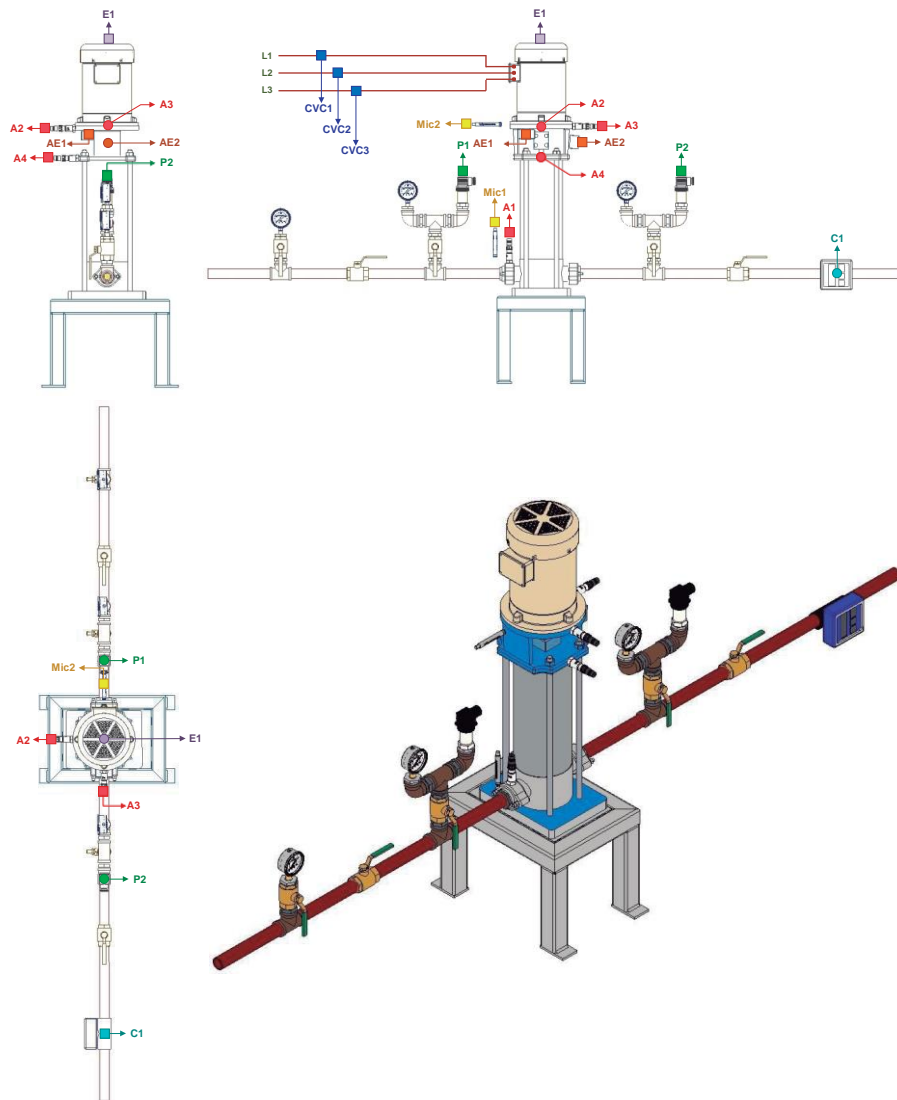


Figura 3.7: Distribución de instrumentos de medición.

utilizada durante el proceso de adquisición.

En la Tabla 3.6 se describe los 4 tipos de módulos utilizados, detallando algunas de las características más relevantes de los mismos.

Tabla 3.3: Condiciones de operación.

<i>Condiciones de operación</i>	<i>Unidades</i>	<i>Presión en la descarga</i>
C1	bar	5.5 ± 0.2
C2	bar	6.5 ± 0.2
C3	bar	7.5 ± 0.2
C4	bar	8.5 ± 0.2
C5	bar	9.5 ± 0.2
C6	bar	10.4 ± 0.2

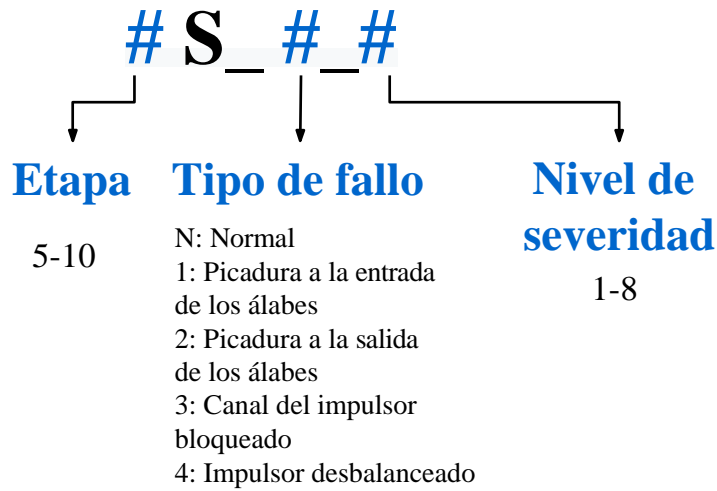


Figura 3.8: Codificación tipos de fallos en impulsores.

3.2.7 Aplicación para adquisición y almacenamiento de señales

Para realizar el proceso de adquisición y almacenamiento de las señales se desarrolló una aplicación que permite la comunicación con el chasis cDAQ-9188 y de esta manera realizar la adquisición de las señales provenientes de los instrumentos de medición. Además, entre otras cosas

Tabla 3.4: Fallos implantados en la bomba centrífuga multietapa.

Código del fallo (clases)	Etapa 10	Etapa 9	Etapa 8	Etapa 7	Etapa 6	Etapa 5
P1	N	N	N	N	N	N
P2	10S_1_2	9S_1_2	N	N	N	N
P3	10S_1_5	9S_1_4	8S_1_3	7S_1_2	6S_1_1	N
P4	10S_1_8	9S_1_7	8S_1_6	7S_1_5	6S_1_4	5S_1_3
P5	10S_2_2	9S_2_1	N	N	N	N
P6	10S_2_5	9S_2_4	8S_2_3	7S_2_2	6S_2_1	N
P7	10S_2_8	9S_2_7	8S_2_6	7S_2_5	6S_2_4	5S_2_3
P8	10S_3_1	N	N	N	N	N
P9	10S_3_4	9S_3_3	8S_3_2	7S_3_1	N	N
P10	10S_3_6	9S_3_5	8S_3_4	7S_3_3	6S_3_2	5S_3_1
P11	10S_4_1	N	N	N	N	N
P12	10S_4_4	9S_4_3	8S_4_2	7S_4_1	N	N
P13	10S_4_6	9S_4_5	8S_4_4	7S_4_3	6S_4_2	5S_4_1

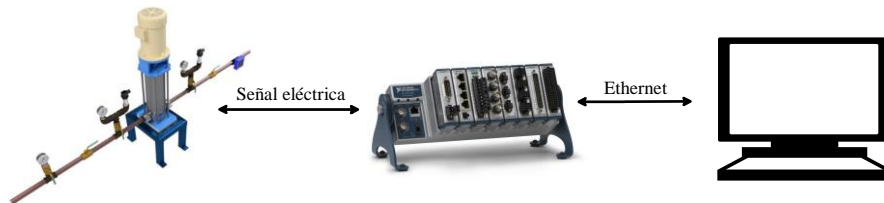


Figura 3.9: Esquema de conexión del sistema de adquisición de señales.

permite configurar la frecuencia de muestreo con la que se desea adquirir cada señal, seleccionar el tiempo de toma de los datos del proceso, realizar la adecuación de los datos para almacenarlos, seleccionar la ubicación para el almacenamiento de los datos y visualizar las señales adquiridas. En la Figura 3.11 se puede observar la interfaz de usuario para la aplicación de adquisición y almacenamiento de las señales, mientras que en la Figura 3.12 se puede observar el diagrama de flujo del funcionamiento de la aplicación.



Figura 3.10: Chasis cDAQ-9188.

Tabla 3.5: Conexión módulos e instrumentos de medición.


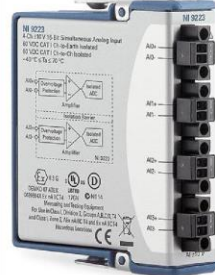


Módulo	Instrumento de medición	Frecuencia de muestreo [kS/s]
NI 9234	CVC1, CVC2, CVC3	50
NI 9234	A1, A2, A3, A4	50
NI 9223	AE1-AE2	1000
NI 9234	Mic1, Mic2	50
NI 9401	E1	500
NI 9205	P1, P2, F1	25

3.2.8 Proceso de adquisición de las señales

Una vez establecido el hardware (Banco de prueba de bombas, Instrumentos de medición y Hardware para la adquisición de señales) y el software (aplicación para adquirir y almacenar las señales) para el sistema se procedió a realizar la tarea de la adquisición de las señales multimodales provenientes del Banco de pruebas de bombas. Para realizar esta tarea se establecieron los parámetros a utilizarse durante la adquisición, lo cuales se muestran en la Tabla 3.7.

Con estos parámetros establecidos se realizó la adquisición de las señales para generar el conjunto de datos. Para lo cual se empezó por tomar las señales del sistema de bombeo trabajando en condiciones normales (clase P1) y bajo cada una de las seis condiciones de operación (C1-C6)

Tabla 3.6: Características módulos utilizados.

Módulo	Foto	Características
NI 9234		<p>Canales: 4 canales de entrada de $\pm 5V$</p> <p>Resolución: 24 Bits</p> <p>Frecuencia de muestreo: Hasta 51.2 kS/s por cada canal simultáneamente</p>
NI 9223		<p>Canales: 4 canales de entrada de $\pm 10V$</p> <p>Resolución: 16 Bits</p> <p>Frecuencia de muestreo: Hasta 1 MS/s por cada canal simultáneamente</p>
NI 9401		<p>Canales: 8 canales de E/S configurables a 5 V/TTL</p> <p>Velocidad de respuesta: 100ns</p>
NI 9205		<p>Canales: 32 canales de E/S de $\pm 10V$</p> <p>Resolución: 16 Bits</p> <p>Frecuencia de muestreo: Hasta 250 kS/s multiplexados</p>

propuestas para el experimento. Además, bajo cada condición de operación se repitió el proceso de adquisición diez veces (R1-R10) obteniendo 60 señales

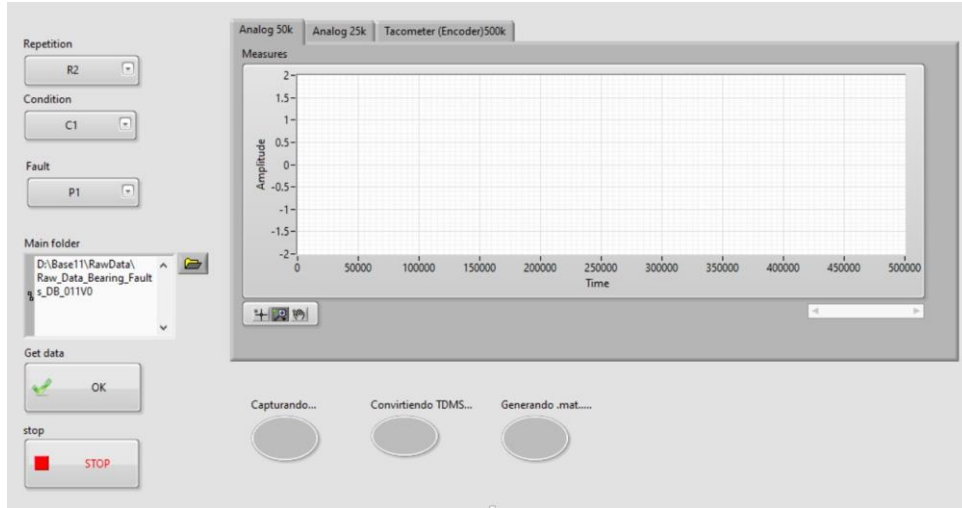


Figura 3.11: Interfaz de usuario para la aplicación para adquisición y almacenamiento de las señales.

Tabla 3.7: Parámetros para realizar el proceso de adquisición de las señales.

Duración de la toma de datos:	10 segundos
Numero de repeticiones:	10 repeticiones por cada condición de operación y bajo cada tipo fallo (R1-R10)
Condiciones de operación (presión y caudal):	6 condiciones: C1 (5,5 bar), C2 (6,5 bar), C3 (7,5 bar), C4 (8,5 bar), C5 (9,5 bar) y C6 (10,4 bar)
Alimentación de la bomba:	220VAC-60 Hz (Se tiene velocidad constante en el motor de la bomba)

multimodales. Luego de obtener las señales de la clase P1 se procedió a implantar cada uno de los fallos descritos anteriormente en la Tabla 3.4 (P2-P13) y se volvió a realizar el mismo proceso de adquisición. Al finalizar con el proceso para cada clase se obtiene un conjunto de datos compuesto por 780 señales multimodales (60 por cada clase). En la Tabla 8 se detalla la combinación utilizada para el proceso de adquisición de las señales

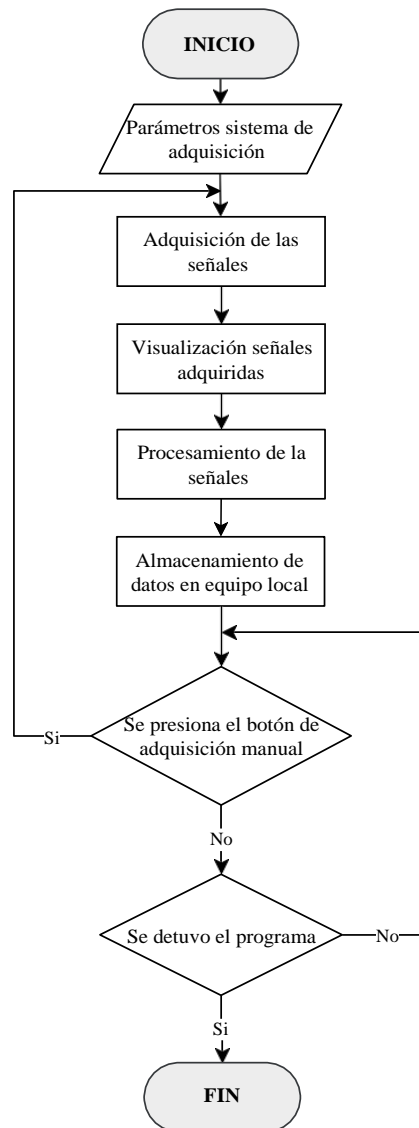


Figura 3.12: Diagrama de flujo de la aplicación de adquisición y almacenamiento de datos.

Tabla 3.8: Combinación utilizada para el proceso de adquisición de las señales.

<i>Clases</i>	<i>Condición de operación</i>	<i>Repetición</i>
P1	C1	R1
P2	C2	R2
P3	C3	R3
P4	C4	R4
P5	C5	R5
P6	C6	R6
P7		R7
P8		R8
P9		R9
P10		R10
P11		
P12		
P13		

3.3 Generación de conjunto de datos

Una vez realizada la adquisición de las señales se procedió a generar los conjuntos de datos a ser utilizados en cada uno de los pasos de la metodología propuesta. Para realizar esta tarea se tomó el conjunto de datos original obtenido en el paso anterior y se seleccionó solamente los datos provenientes del sensor acelerómetro 1 (A1). Con la selección de un solo sensor el conjunto de datos quedó compuesto por 60 señales de 500000 muestras en el dominio del tiempo muestreados a una frecuencia de 50kS/s, por cada una de las clases (P1-P13). Con este conjunto de datos se realizaron los pasos siguientes de la metodología. En la Figura 3.13 se puede observar el flujo de los conjuntos de datos a través de cada uno de los pasos de la metodología propuesta.

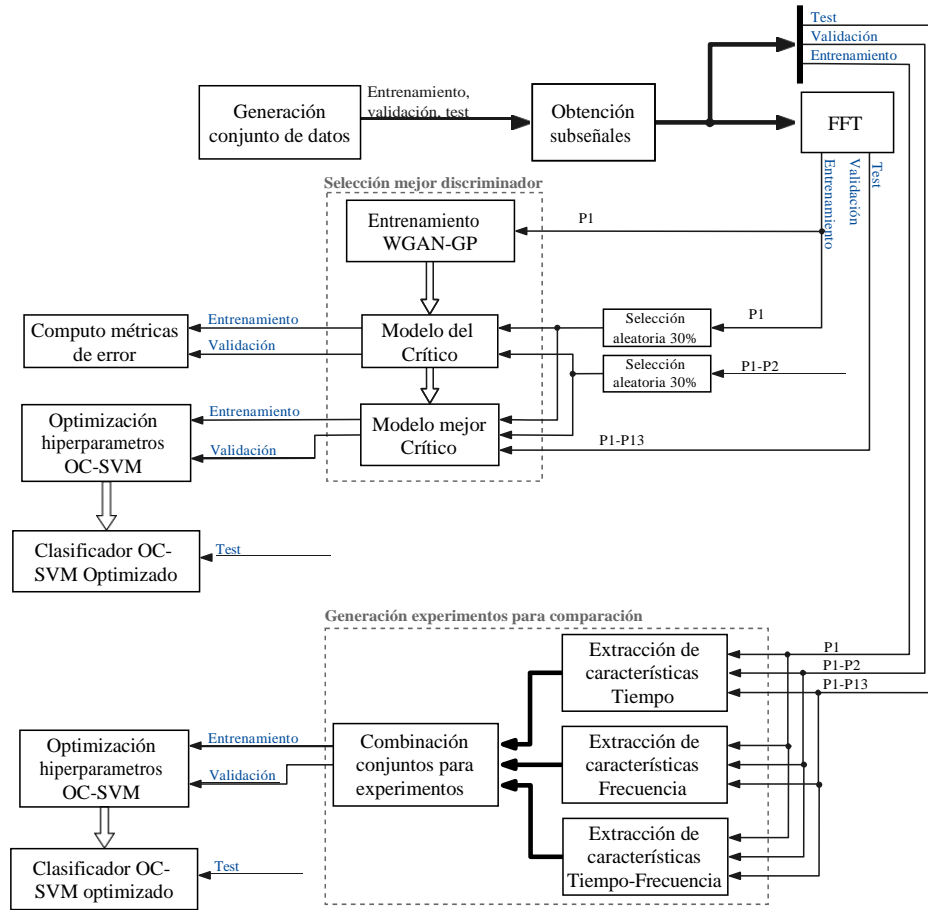


Figura 3.13: Flujo de datos a través de pasos de la metodología propuesta.

3.3.1 Selección de datos para entrenamiento, validación y test

Una vez establecido el conjunto de datos para usarlos en la metodología se realizó el proceso de selección de datos para entrenamiento validación y test. Para realizar la división de los datos se utilizó como parámetro las repeticiones de cada clase (R1-R10). Seleccionando 6 repeticiones para entrenamiento (60 % del conjunto de señales), 2 repeticiones para validación (20 % del conjunto de señales) y 2 repeticiones para test (20 % del conjunto de señales).

Tabla 3.9: Lista repeticiones utilizadas para cada experimento.

<i>Iteración</i>	<i>Entrenamiento</i>	<i>Validación</i>	<i>Test</i>
1	R1, R2, R3, R4, R5, R6	R7, R8	R9, R10
2	R1, R2, R4, R6, R7, R8	R3, R9	R5, R10
3	R1, R5, R7, R8, R9, R10	R4, R6	R2, R3
4	R1, R2, R6, R7, R8, R10	R3, R4	R5, R9
5	R1, R4, R5 R6, R8, R9	R2, R7	R3, R10
6	R1, R2, R4, R7, R9, R10	R3, R6	R5, R8
7	R1, R3, R4, R5, R7, R8	R2, R9	R6, R10
8	R1, R5, R6, R7, R9, R10	R3, R4	R2, R8
9	R2, R5, R6, R7, R8, R9	R1, R10	R3, R4
10	R3, R4, R5, R6, R7, R10	R8,R9	R1, R2

3.3.2 Validación cruzada

Debido a que en la metodología propuesta se utilizó la técnica de validación cruzada para evaluar los resultados obtenidos y garantizar que estos son independientes de la selección de datos de entrenamiento validación y test, se procedió a seleccionar aleatoriamente las repeticiones que componen la selección de cada conjunto de entrenamiento, validación y test. En la Tabla 3.9 se presenta la distribución utilizada para cada iteración del proceso de validación cruzada.

3.3.3 Obtención subseñales

Para este proceso se tomó a cada conjunto de señales de entrenamiento, valida y test, compuestas por señales de 500000 muestras y se las dividió en subseñales más pequeñas para lo cual se utilizó una ventana de 8600 muestras y un desplazamiento de 512 muestras.

La selección de la ventana de 8600 muestras fue realizada a partir de la revisión de la literatura donde se presenta que para trabajar con señales de vibración es necesario que el tiempo de duración de la señal recojan la información de 10 ciclos del proceso, es decir, que la duración de la señal

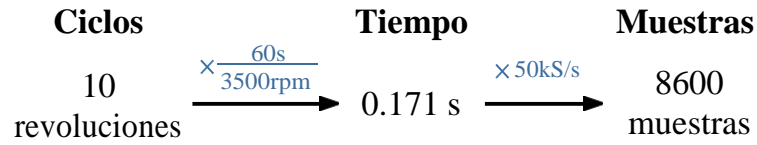


Figura 3.14: Conversión de ciclos a número de muestras.

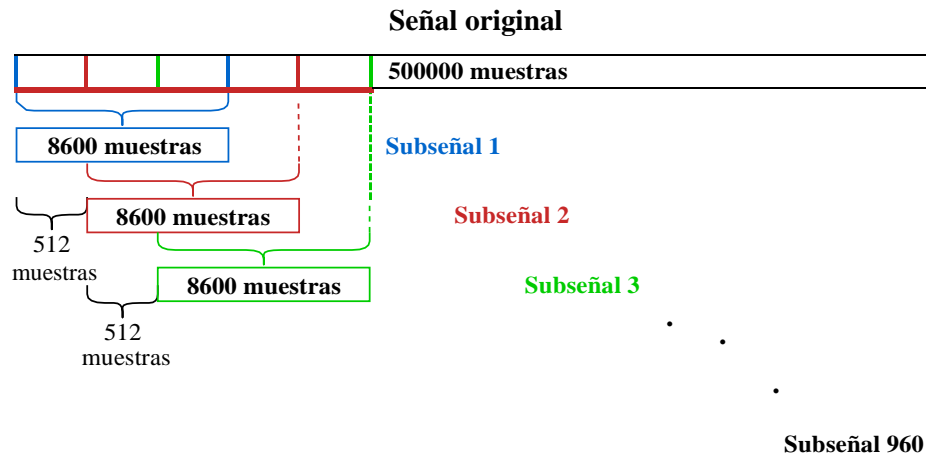


Figura 3.15: División de una señal.

para trabajar no debe ser menor al tiempo en que se cumplen 10 ciclos de rotación del motor. Debido a que la velocidad de rotación del motor de la bomba es de 3500 RPM, para el caso propuesto se cumple con 10 ciclos de rotación en un tiempo de 0.171 segundos. Además, al utilizar una frecuencia de muestro de 50kS/s para la adquisición de las señales de vibración, esto se traduce a utilizar 8600 muestras de la señal. En la Figura 3.14 se muestra el calculo realizado para obtener el número de muestras. Mientras que el desplazamiento de 512 muestras fue seleccionado empíricamente en función de la cantidad de muestras que se tenía por cada señal original. En la Figura 3.15 se observa un esquema del proceso de división de una señal.

Una vez realizado el proceso de división de las señales se obtiene en total 57600 subseñales de 8600 muestras divididas en los tres conjuntos de entrenamiento, validación y test. Este proceso se lo realiza a cada una de las señales de cada clase. Este conjunto de señales obtenido mantiene la

combinación original de repeticiones y condiciones de operación del conjunto original de datos (R1-R10 y C1-C6).

Como siguiente paso a las subseñales obtenidas se le aplicó la transformada rápida de Fourier obteniendo un nuevo conjunto de datos compuesto por 57600 subseñales de 4300 muestras en el dominio de frecuencia, por cada clase, siendo este conjunto utilizado para trabajar en los pasos siguientes de la metodología propuesta.

3.3.4 Extracción de características

De igual manera a las subseñales obtenidas con el proceso de división de las señales se la procesó para realizar la extracción de características en el dominio de tiempo (64 características), dominio de frecuencia (24 características) y dominio de tiempo-frecuencia (96 características)[Sánchez Loja et al., 2018]. Siendo luego estas señales empleadas para construir distintos experimentos para comparar el desempeño de la metodología propuesta.

En la Tabla 3.10 se presenta el nombre de las características en el dominio de tiempo acompañadas de la ecuación empleada para su cálculo. A su vez en la Tabla 3.11 se presenta el nombre de las características en el dominio de frecuencia y la ecuación para su cálculo. En cuanto a las características en el dominio de tiempo-frecuencia se empleó la transformada Wavelet para descomponer las señales empleando distintas Wavelets madre y diferentes niveles. Luego se realizó el cálculo de la energía de cada nodo obtenido con la descomposición, obtenido de esta manera las características para las señales. En la Tabla 3.12 se especifica las Wavelets utilizadas, el nivel y el número de características obtenidas.

Tabla 3.10: Características en el dominio del tiempo.

Nombre de la característica	Fórmula
Media	$T 1 = \frac{1}{N} \sum_{i=1}^N X_i$
Varianza	$T 2 = \frac{1}{N} \sum_{i=1}^N (x_i - T1)^2$
Desviación estándar	$T3 = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - T1)^2}$
Media cuadrática	$T 4 = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i)^2}$
Valor máximo	$T 5 = \max(x_i)$

Table 3.10 Continuación de la tabla anterior

Nombre de la característica	Fórmula
Curtosis	$T6 = \frac{N}{\sum_{i=1}^N (x_i - T1)^4}$
Asimetría	$T7 = \frac{N}{\sum_{i=1}^N (x_i - T1)^3}$
Operador de energía	$T8 = \frac{N^2 \sum_{i=1}^N (\Delta y_i - \Delta \bar{y})^4}{[\sum_{i=1}^N (\Delta y_i - \Delta \bar{y})^2]^2}$ $\Delta \bar{y} = \text{media de } \Delta y$ $\Delta y_i = x_{i+1}^2 - x_i^2$
Media de valores absolutos	$T9 = \frac{1}{N} \sum_{i=1}^N x_i $
Raíz cuadrada de la amplitud	$T10 = \left(\frac{\sum_{i=1}^N x_i }{N} \right)^2$
Factor de forma	$T11 = \frac{T4}{T9}$
Factor de impulso	$T12 = \frac{T5}{T9}$
Factor de cresta	$T13 = \frac{T5}{T4}$
Factor de holgura	$T14 = \frac{\sum_{i=1}^N T5}{\sum_{i=1}^N (x_i)^2}$
CPT1	$T15 = \frac{\sum_{i=1}^N \log(x_i + 1)}{\sum_{i=1}^N \log(T3 + 1)}$
CPT2	$T16 = \frac{\sum_{i=1}^N \exp(x_i)}{N \cdot \exp(T3)}$
CPT3	$T17 = \frac{\sum_{i=1}^N x_i }{N}$
Error cuadrático medio	$T18 = \frac{1}{N} \sum_{i=1}^N (x_i - T1)^2$
Relación log-log	$T19 = \frac{\sum_{i=1}^N \log(x_i + 1)}{N}$
SDIF	$T20 = \frac{\log(T3)}{T9}$
Quinto momento estático	$T21 = \sum_{i=1}^N (x_i - T1)^5$
Sexto momento estático	$T22 = \sum_{i=1}^N (x_i - T1)^6$
Quinto momento normal	$T23 = \frac{\sum_{i=1}^N (x_i - T1)^5}{\left(\frac{1}{N} \sum_{i=1}^N (x_i - T1)^2 \right)^{5/2}}$
k momento central	$T24 = \text{mean}[(x_i - T1)^k]$

Table 3.10 Continuación de la tabla anterior

Nombre de la característica	Fórmula
	<i>k es igual a 3</i>
Indicadores de pulso	$T 25 = \frac{T5}{T1}$
Índice de margen	$\left(\frac{\frac{T5}{N} - \sqrt{\frac{T1}{N}}}{\frac{T5}{N}} \right)^2$
MDR	$T 27 = \frac{T1}{T3}$
Diferencia absoluta de la varianza	$T 28 = \frac{1}{N-2} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2$
Valor mínimo	$T 29 = \min(x_i)$
Valor pico	$T 30 = \frac{1}{2} [T5 - T30]$
Valor pico-pico	$T 31 = T5 - T30$
Límite inferior del histograma	$T 32 = T30 - \frac{1}{2} \frac{T5 - T30}{N-1}$
Límite superior del histograma	$T 33 = T5 + \frac{1}{2} \frac{T5 - T30}{N-1}$
Factor de latitud	$T 34 = \frac{2 \sqrt{N} T1}{\sqrt{\sum_{i=1}^N x_i ^2}}$
Valor n.n de probabilidad negativa	$T 35 = \ln T4^3$
Indicador de forma de onda	$T 36 = \frac{T4}{T1}$
Entropía de Shannon	$T 37 = - \sum_{i=1}^N x_i^2 \log(x_i^2)$
Log. entropía energética (LEE)	$T 38 = \sum_{i=1}^N \log(x_i^2)$ <i>donde $\log(0) = 0$</i>
Umbral de entropía	$T 39 = \text{Umbral} \begin{cases} \square 1, & \text{Si } x_i > p, y \\ \square 0, & \text{cualquier otro caso} \end{cases}$ <i>p es igual a 0,2</i>
Entropía de Sure	$T 40 = n - \#\{\text{ital que } x_i \leq p\}$ $+ \sum_i \min(x_i^2, p^2);$ <i>p es igual a 0,2</i>

Table 3.10 Continuación de la tabla anterior

<i>Nombre de la característica</i>	<i>Fórmula</i>
Norma de la entropía	$T 41 = \sum_{i=1}^N x_i ^p$ <p><i>p es igual a 0,2</i></p>
Cambio de signo de la pendiente	$T 42 = \sum_{i=2}^N f[(x_i - x_{i-1}) * (x_i - x_{i+1})]$ $f = \begin{cases} 1, & \text{if } x \geq \text{umbral} \\ 0, & \text{cualquier otro caso} \end{cases}$
Cruce por cero	$T 43 = \sum_{i=1}^N \text{step}[\text{sign}(-x_i * x_{i+1})]$ $\text{step} = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases}$ $\text{sign} = \begin{cases} 1, & \text{if } x > 0 \\ \frac{1}{2}, & \text{if } x = 0 \\ 0, & \text{if } x < 0 \end{cases}$
Amplitud de Wilson	$T 44 = \sum_{i=1}^N f(x_i - x_{i+1} - T)$ $f = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$ <p><i>T = umbral puesto a 0,2</i></p>
Tasa de porcentaje de miopulso	$T 45 = \frac{1}{N} \sum_{i=1}^N [f(x_i)];$ $f(x) = \begin{cases} 1, & \text{if } x \geq \text{umbral} \\ 0, & \text{cualquier otro caso} \end{cases}$ <p><i>the umbral puesto a 0,2</i></p>
Longitud de onda	$T 46 = \sum_{i=1}^N x_{i+1} - x_i $
Detector logarítmico	$T 47 = e^{\frac{1}{N} \sum_{i=1}^N \log x_i }$

Table 3.10 Continuación de la tabla anterior

Nombre de la característica	Fórmula
Media de la amplitud	$T 48 = \sum_{i=1}^N (x_{i-1} - x_i)$
Energía	$T 49 = \sum_{i=1}^N x_i ^2$
Integral de la señal	$T 50 = \sum_{i=1}^N x_i $
Valor absoluto medio modificado 1	$T 51 = \frac{1}{N} \sum_{i=1}^N W_i x_i $ $W_i = 1; \text{ if } 0,25N \leq n \leq 0,75N$ $W_i = 0,5; \text{ cualquier otro caso}$
Valor absoluto medio modificado 2	$T 52 = \frac{1}{N} \sum_{i=1}^N W_i x_i $ $W_i = 1; \text{ if } 0,25N \leq n \leq 0,75N$ $W_i = \frac{4n}{N}; \text{ if } n < 0,25N$ $W_i = \frac{4(n-N)}{N}; \text{ if } n > 0,75N$
Pendiente del valor absoluto medio	$T 53 = T 9_{i+1} - T 9_i$
Delta de la media cuadrática	$T 54 = T 4_{i+1} - T 4_i$
suma de raíces de los cuadrados	$T 55 = \frac{\sqrt{\sum_{i=1}^n x_i ^2}}{\sqrt{n}}$
Ponderación SSR absoluta	$T 56 = \frac{1}{N} \left(\sum_{i=1}^N x_i \right)^2$
Logaritmo de la media cuadrática	$T 57 = \log(T 4)$
Velocidad de conducción de la señal	$T 58 = \left(\frac{1}{N} \sum_{i=1}^N x_i^2 \right)$
Promedio de cambio de la amplitud	$T 59 = \frac{1}{N} \sum_{i=1}^{N-1} x_{i+1} - x_i $
Weibull logaritmo negativo de probabilidad	$T 60 = - \sum_{i=1}^N \log[(T 11 * \eta)^{-T 1}$ $* x_i ^{T 1-1} \exp \frac{ x_i ^{T 1}}{\eta^{T 1}}]$ <i>donde η es el factor de escala</i>
V-orden 3	$T 61 = \frac{1}{N} \sum_{i=1}^N x_i^3$

Table 3.10 Continuación de la tabla anterior

Nombre de la característica	Fórmula
Longitud máxima de los fractales	$T62 = \log_{10} \frac{\sum_{i=1}^{N-1} (x_i - x_{i+1})^2}{\dots}$
Diferencia absoluta de STD	$T63 = \dots \frac{\sum_{i=1}^{N-1} (x_i - x_{i+1})^2}{\dots}$
Momentos de orden superior	$T64 = \left \frac{1}{N} \sum_{i=1}^N x_i^m \right $

donde m es 3 por defecto

Tabla 3.11: Características en el dominio de frecuencia.

Nombre de la característica	Fórmula
Frecuencia media	$F1 = \frac{\sum_{k=1}^K X(k)}{\sum_{k=1}^K K}$
Varianza	$F2 = \frac{\sum_{k=1}^K (X(k) - F1)^2}{K-1}$
Asimetría	$F3 = \frac{\sum_{k=1}^K (X(k) - F1)^3}{(\sum_{k=1}^K (X(k) - F1)^2)^{3/2}}$
Curtosis	$F4 = \frac{\sum_{k=1}^K (X(k) - F1)^4}{K(F2)^2}$
Frecuencia central	$F5 = \frac{\sum_{k=1}^K k X(k)}{\sum_{k=1}^K X(k)}$
Desviación estándar	$F6 = \sqrt{\frac{\sum_{k=1}^K X(k)}{K-1}}$
Media cuadrática	$F7 = \frac{\sum_{k=1}^K f^2 X(k)}{\sum_{k=1}^K K}$
CP1	$F8 = \frac{\sum_{k=1}^K k^2 X(k)}{\sum_{k=1}^K X(k)}$
CP2	$F9 = \frac{F6^6}{F5^6}$
CP3	$F10 = \frac{\sum_{k=1}^K (f_k - F5)^2 X(k)}{\sum_{k=1}^K (f_k - F5)^2 X(k)}$
CP4	$F11 = \dots$
CP5	$F12 = \frac{\sum_{k=1}^K f^k X(k)}{\sum_{k=1}^K X(k)}$
Dispersión espectral	$F13 = \frac{\sum_{k=1}^K k}{K}$
Centroide espectral	$F14 = \dots$

Table 3.10 Continuación de la tabla anterior

Nombre de la característica

Fórmula

$$\sum_{k=1}^{(k-F-13)^2} X(k)$$

$k=1$

$$\sum_k$$

Table 3.10 Continuación de la tabla anterior

<i>Nombre de la característica</i>	<i>Fórmula</i>
Entropía espectral	$F15 = - \sum_{k=1}^{K-1} P_n(k) \log_2[P_n(k)]$

Table 3.11 Continuación de la tabla anterior

Nombre de la característica	Fórmula
	donde P_n es energía espectral total normalizada $P_n(k) = \frac{X(k)}{\sum_{k=1}^K X(k)}$
Poder total	$F 16 = \sum_{k=1}^M P(k)$
Frecuencia media	$F 17 = \frac{1}{2} \sum_{k=1}^M P(k)$
Frecuencia pico	$F18 = \max(P(k)), \quad k = 1, \dots, M.$
Primer momento espectral	$F 19 = \sum_{k=1}^M P(k)f_k$
Segundo momento espectral	$F 20 = \sum_{k=1}^M P(k)f_k^2$
Tercer momento espectral	$F 21 = \sum_{k=1}^M P(k)f_k^3$
Cuarto momento espectral	$F 22 = \sum_{k=1}^M P(k)f_k^4$
Varianza de la frecuencia central	$F23 = \frac{F20}{F16} - (F19)^2$
Relación de frecuencia	$F 25 = \frac{\sum_{LHC=f_{min}}^{ULC=f_{max}/2} P(k)}{\sum_{UHC=f_{max}} P(k)}$
	$LLC=f_{min} \quad \quad \quad LHC=f_{max} + 1$ Donde m es 3 por defecto

Tabla 3.12: Características en el domino tiempo-frecuencia.

Wavelet madre	Niveles	Número de características
Daubechies 3	3	8 características
Daubechies 5	3	8 características
Daubechies 7	4	16 características
Symlet 5	4	16 características
Coiflet 4	4	16 características
Biorotogonal 6.8	4	16 características
Biorotogonal reversa 6.8	4	16 características

Algunas de estos métodos de extracción de características descritos anteriormente fueron utilizados en los trabajos presentados en [Sánchez et al., 2017, 2018, 2019] y desarrollados por el grupo de investigación GIDTEC de la Universidad Politécnica Salesiana.

3.4 Construcción de los modelos

Para la metodología propuesta se generaron dos modelos, el primero a partir del uso del discriminador de una GAN utilizado como extractor de características de las señales y otro modelo que es un clasificador de una sola clase utilizado para clasificar entre clase sin falla y con falla. A continuación se detalla a profundidad el procedimiento realizado para la obtención de los dos modelos.

3.4.1 Red generativa adversaria

Como se vio en la Sección 2.2.2 una red generativa adversaria es un sistema que emplea dos modelos uno denominado generador y otro discriminador. Estos dos modelos compiten entre sí; el modelo generador se encarga de generar muestras las cuales son evaluadas por el discriminador que aprende a determinar si la muestra procede del modelo generador o del conjunto de muestras reales [Goodfellow et al., 2014]. Esta competencia entre ambos modelos lleva a que estos mejoren en generar muestras (modelo generador) y para diferenciar las muestras entre reales y falsa (modelo discriminador).

Para la metodología propuesta se utilizó una Wasserstein GAN con penalidad de gradiente (WGAN-GP) debido a que mejora el entrenamiento de la GAN original que tiene entre sus mayores problemas la inestabilidad durante el entrenamiento provocado por la competencia entre el generador y el discriminador [Arjovsky et al., 2017], y el problema denominado mode collapse o colapso modal que se presenta cuando el generador solamente es capaz de producir un pequeño conjunto de tipos de salidas sin abarcar todos los modos posibles de los datos multimodales con las que se entrena la GAN [Thanh-Tung and Tran]. Al utilizar una arquitectura de una WGAN-GP ayuda a remediar el problema del colapso modal, aunque aún pudiera producirse el problema de inestabilidad.

3.4.2 WGAN-GP

WGAN-GP es una mejora de la GAN original que a diferencia de esta utiliza la distancia de Wasserstein como función de pérdida. La distancia

de Wasserstein ($W(P_r, P_g)$) se puede definir como el costo mínimo de transportar "masa" al convertir la distribución de datos P_r en la distribución de datos P_g . La distancia de Wasserstein para la distribución de datos reales P_r y la distribución de datos generados P_g se define matemáticamente como el mayor límite inferior (ínfimo) para cualquier plan de transporte [Gulrajani et al., 2017]:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] \quad (3.1)$$

donde $\Pi(P_r, P_g)$ denota el conjunto de todas las distribuciones conjuntas $\gamma(x, y)$ cuyos márgenes son respectivamente P_r y P_g . Intuitivamente, $\gamma(x, y)$ indica cuánta "masa" debe ser transportada de x a y para transformar las distribuciones P_r en la distribución P_g .

Sin embargo, la ecuación para la distancia de Wasserstein (Ecuación 3.1) es altamente intratable. Por tal motivo se usa la dualidad Kantorovich-Rubinstein [Hanin, 1992] para obtener la función de valor de la WGAN:

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{x \sim P_r} [D(x)] - \mathbb{E}_{\tilde{x} \sim P_r} [D(\tilde{x})] \quad (3.2)$$

donde \mathcal{D} es el conjunto de funciones de 1-Lipschitz [Cobzaş et al., 2019]. Al poseer un modelo de discriminador óptimo, al minimizar la función de valor con respecto a los parámetros del generador se minimiza $W(P_r, P_g)$.

En el caso de una WGAN el modelo conocido como discriminador pasa a llamarse crítico debido a que ya no realiza la función de clasificar, sino que produce una puntuación escalar en lugar de una probabilidad, pudiendo ser este puntaje interpretado como cuan reales son las señales de entrada. Como dos de las mayores ventajas de la WGAN frente a una GAN se tiene que:

- La función de valor de la WGAN da como resultado una función crítica cuyo gradiente con respecto a su entrada se comporta mejor que su contraparte GAN, esto facilita la optimización del generador.
- Se genera mayor estabilidad del proceso de optimización.

A pesar de tener ventajas frente a una GAN, la WGAN-GP presenta un problema el cual es la dificultad de hacer cumplir la restricción de Lipschitz. Al utilizar el recorte de pesos (Ecuación 3.3) para hacer cumplir la restricción se puede generar que el entrenamiento no converja cuando el hiperparámetro no se encuentra correctamente seleccionado.

$$w \leftarrow w + a \cdot \text{RMSProp}(w, g_w) \tag{3.3}$$

$$w \leftarrow \text{clip}(w, -c, c)$$

Para lidiar con este problema [Thanh-Tung and Tran] propone una mejora de la WGAN la cual es introducir la penalidad de gradiente en lugar del recorte de pesos para cumplir con la restricción de Lipschitz.

$$L = \underbrace{\mathbb{E}_{\tilde{x} \sim P_g} [D(\tilde{x})]}_{\text{Función de pérdida original}} - \underbrace{\mathbb{E}_{x \sim P_r} [D(x)]}_{\text{Función de pérdida original}} + \lambda \underbrace{\mathbb{E}_{\tilde{x} \sim P_x} (\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2}_{\text{Penalidad de gradiente}} \tag{3.4}$$

Al incluir la penalidad de gradiente provoca que el entrenamiento sea más estable y por lo tanto sea más fácil de entrenar los modelos.

En la Figura 3.16 se puede observar un esquema de una GAN original (a) y una WGAN-GP (b), en las cuales se puede notar sus diferencias.

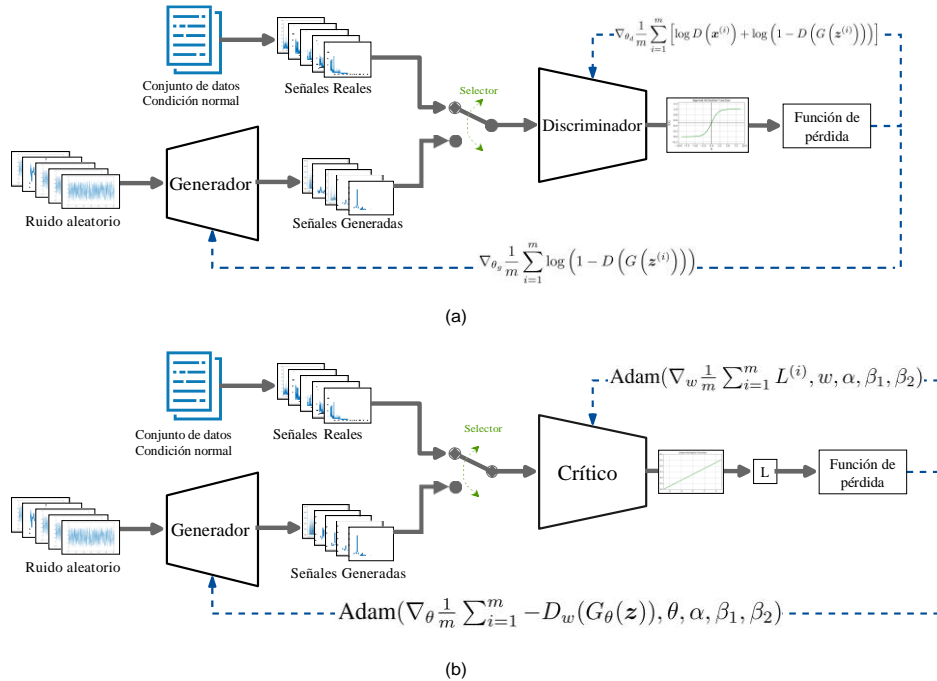


Figura 3.16: Esquema de una GAN (a) vs WGAN-GP (b).

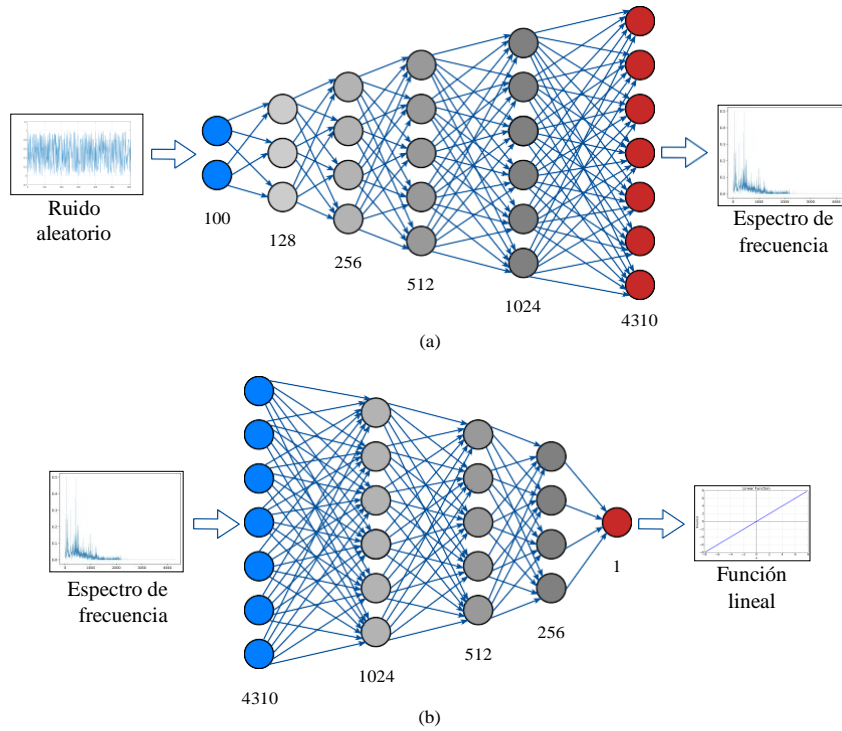


Figura 3.17: Arquitectura Generador (a) Arquitectura Crítico (b).

Una diferencia que se puede notar en ambas redes está en el diseño del discriminador/crítico, en efecto uno posee una función sigmoidea y el otro posee una función lineal como salida. Aunque, la diferencia más notable radica en la función de pérdida para cada red tal como se puede observar en la Figura 3.16.

3.4.3 Arquitectura del generador y crítico

Para la arquitectura de las redes se utilizó una red neuronal perceptrón multicapa tanto para el Generador como para el crítico. En la Figura 3.17 se presenta un esquema de la arquitectura utilizada en estas dos redes.

Como se puede observar el Generador (a) tiene una capa de entrada con 100 neuronas y una capa de salida con 4310 neuronas, siendo esta seleccionada por motivo del tamaño de las señales utilizadas. Mientras que el crítico (b) tiene una capa de entrada con 4310 neuronas y una capa de salida de 1 neurona.

3.4.4 Entrenamiento de la red generativa adversaria

Al tener generada la arquitectura de la **WGAN-GP** (Figura 3.16 b) se procedió con el entrenamiento de la misma. Para este proceso se tomó el conjunto de entrenamiento del cual se seleccionó aleatoriamente un lote o conjunto de 64 señales para cada época del entrenamiento de la **WGAN-GP**. Para el experimento propuesto se generaron 540 lotes de 64 subseñales, y se estableció como número de épocas de entrenamiento 300.

Durante cada época del entrenamiento de la **WGAN-GP** se procedió a realizar las acciones descritas en los siguientes puntos:

Entrenamiento del crítico

Para realizar el entrenamiento del crítico se toma el lote de entrenamiento seleccionado aleatoriamente (señales reales), luego se genera un conjunto de 64 señales aleatorias que se utilizan como entrada para el generador, con el cual se obtiene un lote de 64 señales (señales generadas).

Con los lotes de las señales reales y señales generadas se procede a evaluarlos con el crítico, obteniendo un lote de resultados, con estos resultados se calcula la penalidad del gradiente y luego se optimiza al crítico.

Este proceso de entrenamiento del crítico se lo realiza un número establecido de veces (para el experimento propuesto fue de 20 veces). Al concluir con el número de entrenamientos del crítico se procede al entrenamiento del generador.

Entrenamiento generador

Utilizando como entrada para el generador las 64 señales aleatorias se obtiene 64 señales generadas. El lote de señales generadas se las evalúa con el crítico y los resultados obtenidos se los utiliza para optimizar el generador.

Este ciclo de entrenamiento del crítico y generador se lo repite un total de 27 veces, esto debido al tamaño del conjunto de datos de entrenamiento. Al realizar el proceso 27 veces se utilizaron los 540 lotes disponibles esto debido a que se utilizan 20 lotes por cada ciclo de entrenamiento. Al concluir los 27 ciclos de entrenamiento se considerara como concluida una época.

3.4.5 Computo de métricas de error

Luego de terminar cada época de entrenamiento de la **WGAN-GP** se procedió a realizar el cómputo del score de un clasificador de una sola clase con el fin evaluar el desempeño del crítico como extractor de características y poder

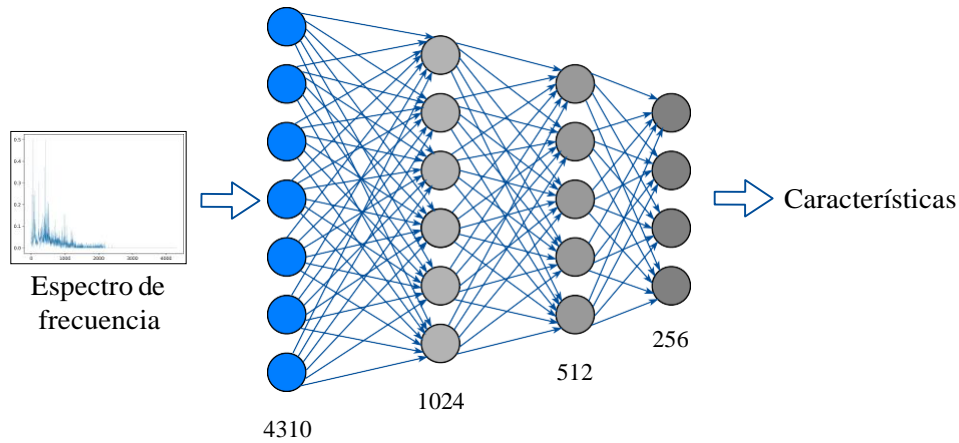


Figura 3.18: Arquitectura crítico modificado.

seleccionar el mejor modelo. Para este proceso se utilizó un clasificador **OC-SVM**, el mismo fue entrenado con el conjunto de entrenamiento, y evaluado con los conjuntos de validación con el fin de obtener las métricas de error del clasificador para datos en estado sin falla y con falla.

En los siguientes puntos se detalla el procedimiento realizado para la obtención de las métricas.

- Se procede a modificar el crítico de la **WGAN-GP** al cual se le elimina su última capa o capa de salida de la red, obteniendo una nueva red con una capa de entrada de 4310 neuronas y una capa de salida de 256 neuronas. A través de esta red se procesa los subconjuntos de las señales de entrenamiento y validación para extraer las características de estas subseñales, obteniendo tres nuevos subconjuntos con subseñales de 256 características. En la Figura 3.18 se puede observar la arquitectura del crítico modificado.
- Luego se realiza el proceso de normalización de las características, para lo cual se toma el subconjunto de entrenamiento y se calcula un factor de escalamiento para normalizar las subseñales en un rango 0 a 1. Con este factor de escalamiento se procede a normalizar los subconjuntos para validación.
- Una vez procesados los conjuntos se genera un clasificador **OC-SVM**, estableciendo como hiperparámetros del clasificador un valor de $\nu=0,1$

y $\gamma = 0,05$, los cuales son seleccionados empíricamente. Realizado este proceso se procede a entrenar al clasificador utilizando el subconjunto de entrenamiento normalizado.

- Con el clasificador entrenado se procede a evaluarlo con los subconjuntos de validación normalizados. Con los resultados de la clasificación de los subconjuntos de validación se generan las métricas de falsos negativos y falsos positivos.

Normalización de las características

Para realizar el proceso de normalización de las características se utilizó el método Min-Max con un rango entre 0-1. El método utiliza la Ecuación 3.5 para realizar la normalización.

$$X^j = a + \frac{(X - X_{min})(b - a)}{X_{max} - X_{min}} \quad (3.5)$$

donde X^j son las características normalizadas, X son las características sin normalizar, los valores a y b son los límites entre los que se quiere normalizar, X_{min} y X_{max} son los valores mínimo y máximo respectivamente de las características del subconjunto de entrenamiento. Este proceso se realizó para cada característica por separado, obteniendo 256 factores de escalamiento.

Calculo falsos negativos y falsos positivos

Para realizar el cálculo del porcentaje de falsos positivos y falsos negativos se utilizaron las ecuaciones 3.6 y 3.7 respectivamente

$$\% FN = \frac{FN}{FN + FP} \quad (3.6)$$

$$\% FP = \frac{FP}{FN + FP} \quad (3.7)$$

donde, FN es el número de falsos positivos y FP es el número de falsos negativos.

3.4.6 Selección del mejor modelo de crítico

Para realizar la selección del mejor modelo de crítico, durante cada época de entrenamiento se evaluó los resultados de las métricas de error. Seleccionando como mejor crítico al modelo que alcanzo el mínimo porcentaje error de falsos positivos durante el entrenamiento.

3.4.7 Optimización de hiperparámetros del clasificador OC-SVM

Una vez obtenido el mejor modelo para el crítico se procedió a optimizar los hiperparámetros para el clasificador OC-SVM. Para este proceso se utilizaron los subconjuntos de entrenamiento y validación.

Como primer paso para el proceso de optimización se procedió a modificar el modelo del mejor crítico para lo cual se le eliminó su última capa, obteniendo un nuevo crítico con una capa de salida de 256 neuronas tal como se puede observar en la Figura 3.18. A través de esta red se procesó los subconjuntos, obteniendo tres nuevos subconjuntos con subseñales de 256 características.

Como siguiente paso se realizó el proceso de normalización de las características para lo cual se tomó el subconjunto de entrenamiento generado en el paso anterior y a partir de este se generó un escalador para normalizar las subseñales entre 0 y 1. Con este escalador se procedió a normalizar los subconjuntos para validación.

Con las señales de entrenamiento y validación normalizadas se utilizó el método de búsqueda aleatoria para optimizar los hiperparámetros del clasificador OC-SVM. En la Figura 3.19 se puede observar un esquema del proceso de búsqueda aleatoria, el cual consiste en asignar valores aleatorios para los hiperparámetros ν y γ del clasificador OC-SVM. Luego se entrena el clasificador con el subconjunto de entrenamiento normalizado y se lo evalúa con los subconjuntos de validación normalizados, obteniendo una métrica de la exactitud del clasificador. Este proceso se lo realiza k veces cambiando los parámetros ν y γ por distintos valores aleatorios, y se almacena los resultados de la exactitud para cada prueba. Concluido el proceso de búsqueda aleatoria se seleccionó los mejores hiperparámetros para el clasificador en función de con que parámetros se obtuvo la mejor exactitud.

3.4.8 Prueba clasificador OC-SVM optimizado

Una vez concluido el proceso de optimización de los hiperparámetros del clasificador OC-SVM se tomó los conjuntos de señales de test de todas las clases, a la cual se procesó con el modelo del mejor crítico eliminada su última capa obteniendo.

Con este conjunto de test obtenido se procedió a realizar el proceso de normalización para lo cual se utilizó el mismo escalador generado durante el proceso de optimización. Una vez normalizados los conjuntos de test se realizó la clasificación utilizando el clasificador OC-SVM optimizado. Con

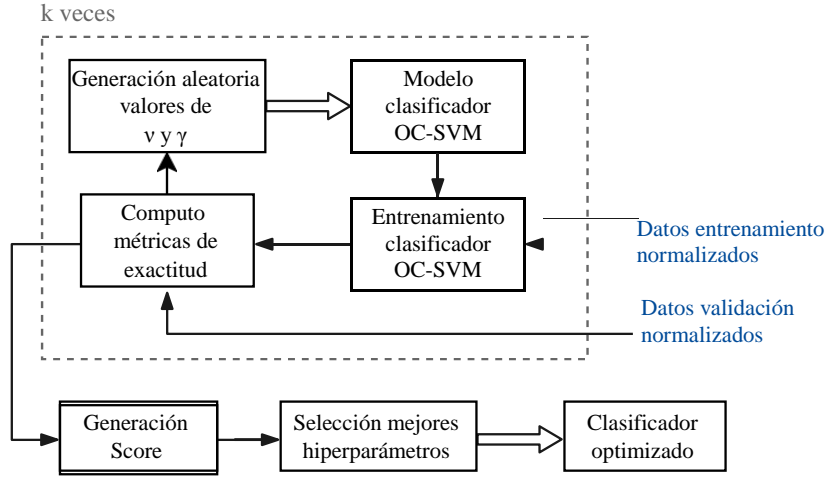


Figura 3.19: Proceso de búsqueda aleatoria.

los resultados de clasificación obtenidos se generó las métricas de error para la clase normal (P1) y las clases con falla (P2-P13). De esta manera se probó el desempeño del clasificador con datos nuevos para el mismo.

Capítulo 4

Sistema de detección online de fallos en la nube

En este capítulo se describe el proceso para la creación e implementación del sistema de detección online de fallos en la nube. Se inicia el capítulo definiendo la metodología de desarrollo utilizado para el proceso de creación del sistema para luego continuar con la explicación a detalle de cada uno de las etapas para alcanzar la implementación y pruebas del sistema.

4.1 Desarrollo

Luego de tener lista la nueva metodología y haber sido puesta a prueba en un sistema offline se realizó la puesta en marcha de un sistema de detección online de fallos en la nube. Este sistema tiene como función principal realizar la detección de fallos en una bomba centrífuga multietapa de eje vertical que se encuentra trabajando en un banco de pruebas de bombas.

Para el desarrollo del sistema se utilizó la metodología de desarrollo en cascada que es un procedimiento lineal que se caracteriza por dividir el proceso de desarrollo en etapas o fases. Además, tiene como característica que cada una de las fases se ejecuta una vez y los resultados de cada fase sirven como hipótesis de partida para la fase siguiente [Ruparelia, 2010].

Para el caso propuesto se utilizó una metodología en cascada cuyo esquema se puede observar en la Figura 4.1 y esta compuesta por cinco fases descritas a continuación:

1. Análisis: Planificación, análisis y especificación detallada de los requerimientos.
2. Diseño: Diseño y especificación del sistema (diseño de la arquitectura)
3. Implementación: Programación y configuración del software y hardware, búsqueda de errores y pruebas unitarias.
4. Verificación: Integración de sistemas, pruebas del sistema completo (versión beta) por parte de los usuarios.
5. Mantenimiento: Entrega, mantenimiento y mejora del sistema.

A continuación se detalla cada una de las fases para generar el sistema de detección online de fallos en la nube.

4.2 Análisis

En la fase de análisis se generaron los requerimientos para el sistema a partir de las necesidades del usuario. Para el caso propuesto se establecieron los siguientes requerimientos detallados a continuación:

- El sistema deberá ser capaz de realizar la tarea de detección de fallos en una bomba centrífuga multietapa de eje vertical.
- El sistema utilizará la metodología propuesta para realizar la detección de fallos en la máquina.

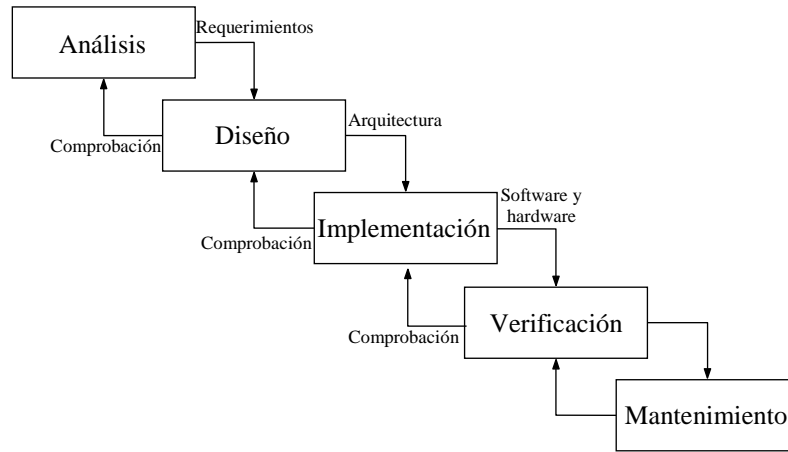


Figura 4.1: Fases metodología en cascada.

- El sistema deberá ser capaz de realizar la detección de fallos de manera online.
- El sistema mostrará el estado de funcionamiento de la bomba centrífuga en tiempo real.
- El sistema deberá realizar la notificación al usuario a través de correo electrónico cuando ocurra un fallo en la bomba.
- El sistema tendrá que ser escalable, es decir, deberá permitir integrar más máquinas al sistema en procesos futuros.
- El subsistema de detección de fallos deberá ser implementado en la nube.
- El subsistema de adquisición de señales deberá estar montado de manera local.
- Para realizar la detección de fallos se deberá utilizar señales de vibración
- El intervalo mínimo de tiempo para el monitoreo de estado de la bomba centrífuga deberá ser de 1 minuto.

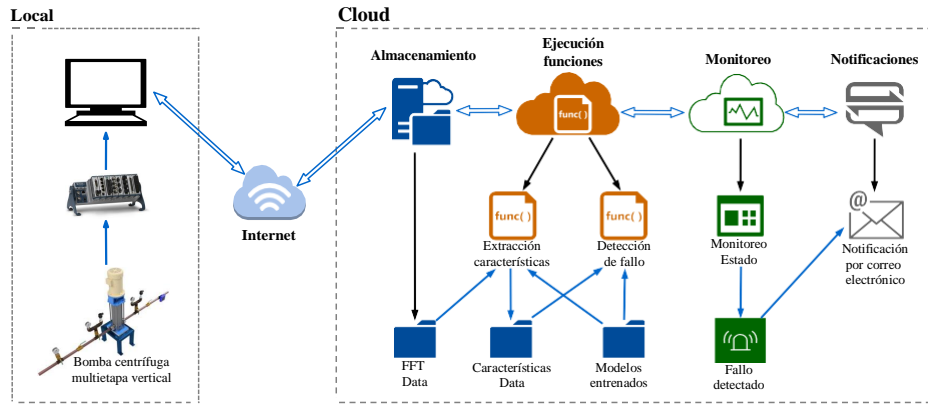


Figura 4.2: Arquitectura para el sistema de detección online de fallos.

4.3 Diseño

Una vez establecido los requerimientos para el sistema se procedió a la fase de diseño. En un inicio de la fase de diseño se estableció el tipo de arquitectura para implementar el sistema, la cual es una arquitectura sin servidor, debido a que este tipo de arquitectura permite crear y ejecutar aplicaciones y servicios sin tener que administrar infraestructura, es decir sin tener que aprovisionar, escalar ni mantener servidores. Esto genera una gran ventaja al momento de desarrollar sistemas o aplicaciones que necesitan ser escalables [McGrath and Brenner, 2017].

4.3.1 Arquitectura

A partir de los requerimientos del sistema se desarrolló la arquitectura, la cual es capaz de cumplir con las funcionalidades requeridas. En la Figura 4.2 se puede observar la arquitectura propuesta para el sistema de detección online de fallos en la nube, la cual se compone por dos grupos principales el local y el remoto (cloud), que se conectan a través de una conexión de internet. En el caso del grupo local se tiene el subsistema encargado de realizar la adquisición de las señales de vibración provenientes de la bomba centrífuga multietapa de eje vertical, mientras que el grupo cloud se encuentra compuesto por los subsistemas de almacenamiento, ejecución de funciones, monitoreo y notificaciones. Cada uno de estos subsistemas trabajan en conjunto para cumplir con todos los requerimientos de funcionalidad establecidos.

4.4 Implementación

A partir de la arquitectura generada en la fase de diseño se procedió a realizar la fase de implementación. Para la cual se realizó primero la implementación de subsistemas local para luego continuar con la implementación de los subsistemas que se ejecutan en la nube, siendo Amazon Web Services la plataforma utilizada para la implementación del sistema en la nube. A continuación se detalla la composición de cada subsistema y su proceso de implementación.

4.4.1 Local: Adquisición y almacenamiento de datos

El subsistema de adquisición de señales se encuentra compuesto por el banco de pruebas de una bomba centrífuga multietapa de eje vertical, los instrumentos de medición y equipos de adquisición detallados en el capítulo 3. Utilizando estos equipos en conjunto se realizó la adquisición de las señales de vibración provenientes del sensor acelerómetro 1 (A1) utilizando una frecuencia de muestreo de 50 kS/s.

Para adquirir estas señales se desarrolló una aplicación que se encarga de realizar el proceso de adquisición automáticamente que consiste en:

- Realizar la adquisición de las señales cada cierto intervalo de tiempo asignado por el usuario de la aplicación, y durante un tiempo predefinido de 10 segundos obteniendo una señal de 500000 muestras
- Luego la señal adquirida es procesada para extraer 10 subseñales, para este proceso se utiliza una ventana de 8600 muestras y un desplazamiento de 50000 muestras.
- Al terminar de realizar este proceso las subseñales adquiridas son procesadas para cambiarlas del dominio de tiempo al dominio de la frecuencia, para lo cual se emplea la transformada rápida de Fourier obteniendo un conjunto de 10 subseñales de 4300 muestras cada una.
- Utilizando el conjunto de subseñales se procede a generar una matriz de datos para almacenarla en el equipo local y a su vez este archivo se envía a través de internet a una carpeta denominada "FFT data" que se encuentra en un repositorio de almacenamiento en la nube.

Este proceso de adquisición y almacenamiento de datos en la nube se lo realiza continuamente según la selección de tiempo realizada por el usuario o también puede ser ejecutado de manera manual.

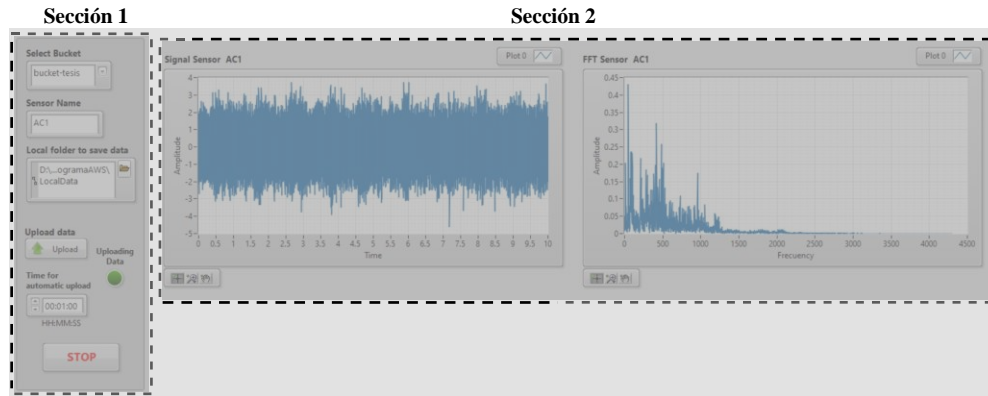


Figura 4.3: Interfaz de usuario de la aplicación para el sistema de detección online de fallos.

En la Figura 4.3 se puede observar la interfaz de usuario de la aplicación que está compuesta por dos secciones, en la Sección 1 se tienen los controles para seleccionar la ruta para el almacenamiento de los archivos en la nube, asignar un nombre al sensor, seleccionar la ruta para almacenamiento local de los archivos, iniciar el proceso de adquisición de manera manual, asignar un tiempo para el intervalo de adquisición de datos y un control para detener la ejecución de la aplicación. Mientras que en la Sección 2 tenemos los visualizadores gráficos de las señales adquiridas y de la señal procesada.

En la Figura 4.4 se puede observar el diagrama de flujo del funcionamiento de la aplicación para adquisición y almacenamiento de datos en la nube.

4.4.2 Cloud: Funciones lambda

Luego de realizar la tarea de adquisición, procesamiento y almacenamiento de las señales se procedió a desarrollar el subsistema encargado de la extracción de características y detección de fallos, para lo cual se empleó dos funciones tipo lambda. Para el sistema propuesto se crearon las funciones lambda, denominadas “Extracción de características” y “Detección de fallos”, cada una de las cuales son ejecutadas al ocurrir dos eventos específicos.

A continuación se describe detalladamente el proceso de desarrollo de cada una de las funciones.

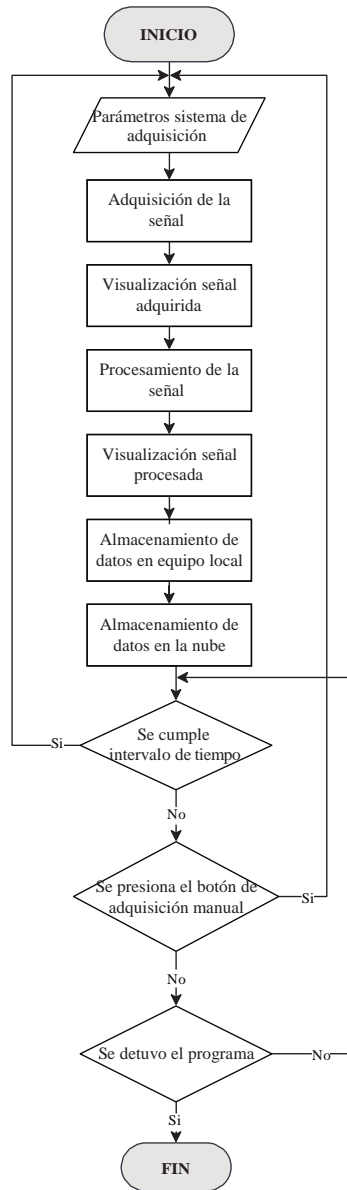


Figura 4.4: Diagrama de flujo de la aplicación.

Función “Extracción de características”

Esta función tiene como objetivo principal realizar la extracción de las características de las señales provenientes del subsistema de adquisición y almacenamiento de datos y almacenar los resultados en una carpeta en la nube. Para caso propuesto la función se creó para que se ejecute al momento de que un archivo es almacenado en la carpeta “FFT data”.

Al ejecutarse la función esta toma los datos del archivo almacenado recientemente en la carpeta “FFT data”, el mismo que contiene las 10 señales en el dominio de frecuencia. Luego carga el modelo del mejor crítico modificado que se encuentra almacenado en la carpeta “Modelos entrenados” y utiliza este modelo para procesar las señales para extraer las 256 características.

Una vez realizado el proceso de extracción de características se obtiene 10 señales de 256 características cada una, las cuales son almacenadas en un archivo que es cargado en la carpeta “Características data”. En la Figura 4.5 se puede observar un diagrama de flujo del funcionamiento de la función lambda “Extracción de características”.

Función “Detección de fallos”

La función lambda denominada “Detección de fallos”, tiene como objetivo realizar el proceso de detección de un fallo en la bomba y publicar los resultados en un visualizador gráfico. Esta función fue creada para ejecutarse al momento que un archivo es almacenado en la carpeta “Características data”.

Al realizarse la ejecución de la función esta toma los datos del archivo almacenado recientemente en la carpeta “Características data” el cual contiene los datos de las 10 señales de 256 características. Luego se cargan los modelos del escalador y del clasificador [OC-SVM](#) optimizado obtenidos al finalizar el proceso de validación cruzada los cuales se encuentran almacenados en la carpeta “Modelos entrenados”.

Como siguiente paso se normalizan las características de las señales en un rango de 0 a 1 utilizando el escalador previamente cargado. Luego se toma las señales normalizadas y se las clasifica con el mejor clasificador [OC-SVM](#) obteniendo 10 resultados de clasificación. Estos resultados son sometidos a un proceso de conteo de votos para determinar si la bomba se encuentra en estado con fallo o estado sin fallo. En caso de producirse un empate se prioriza el estado con fallo.

A partir de los resultados del conteo de votos se realiza la publicación del

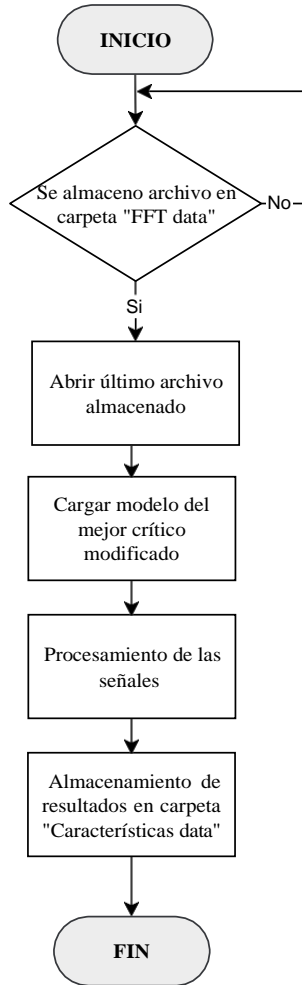


Figura 4.5: Diagrama de flujo de la función “Extracción de características”.

resultado del estado de la máquina en un visualizador gráfico denominado “Condición bomba centrífuga multietapa”. Para el caso propuesto se envía un 1 al visualizador para el estado sin fallo y un 0 para el estado con fallo. En la Figura 4.6 se puede observar un diagrama de flujo del funcionamiento de la función lambda “Detección de fallos”.

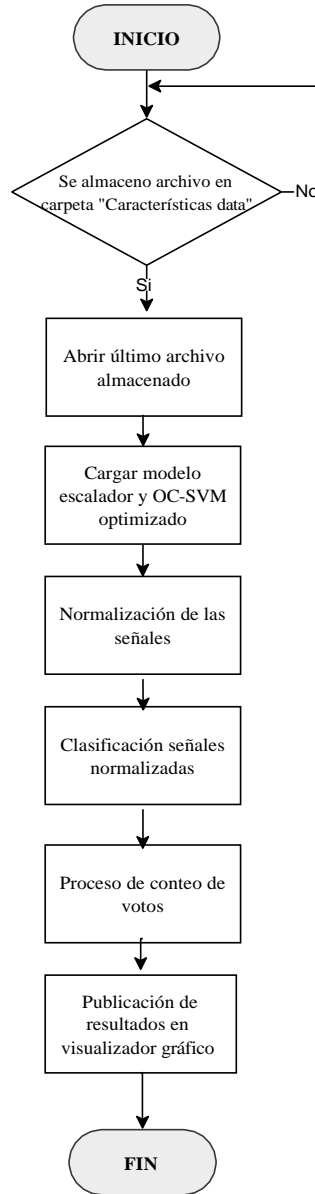


Figura 4.6: Diagrama de flujo de la función "Detección de fallo".

Cloud: Monitoreo y notificaciones

Como siguiente paso para el desarrollo del sistema de detección online de fallos fue la creación de un panel para visualización del estado de la bomba

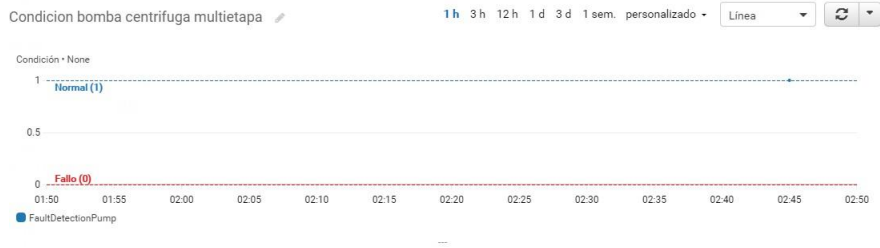


Figura 4.7: Panel para monitoreo del estado de la bomba centrífuga multietapa.

centrífuga multietapa. El cual permite a los usuarios monitorizar es estado de la máquina además de generar alarmas al momento de detectarse un fallo.

Para el sistema propuesto se desarrolló un panel denominado “Monitoreo de estado” al cual contiene un visualizador gráfico denominado “Condición bomba centrífuga multietapa” empleado para monitorizar el estado de la bomba. La métrica monitorizada es la generada por la función lambda “Detección de fallo” que envía un 1 o un 0.

Una vez creado el visualizador para el panel se realizó la creación de una alarma denominada “Fallo detectado” la cual se activa al recibir un valor de 0 en la métrica de estado. Al ser activada la alarma realiza el proceso de notificación al usuario del sistema a través de un correo electrónico.

Con la finalización de la creación de la alarma el panel de visualización quedo terminado obteniendo el resultado que se muestra en la Figura 4.7.

4.5 Verificación

Al tener implementado todo el sistema de detección online de fallos en la nube se procedió a verificar el funcionamiento y la integración de todos los elementos del sistema. Para la fase de verificación se generó una versión beta del sistema el cual fue probado por los usuarios finales. Con estas pruebas de aceptación se pudo determinar que el sistema cumple con los requerimientos establecidos en la fase de análisis.

4.6 Mantenimiento

Para la fase de mantenimiento se planteó la posibilidad de poder probar el sistema por un periodo prolongado de tiempo, hasta que se generen requerimiento de escalabilidad del sistema en el cual se tenga que agregar más máquinas para monitorizar su estado. Esto daría una retroalimentación para poder realizar mejoras en el sistema de detección online de fallos en la nube.

Capítulo 5

Análisis de resultados

El presente capítulo contiene el análisis de los resultados obtenidos en cada una de las etapas de creación de la metodología y los resultados de la implementación del sistema de detección online de fallos en la nube. A su vez se presenta una comparación de los resultados de desempeño alcanzados con la metodología propuesta frente a metodologías clásicas para extracción de características.

Durante el desarrollo de la metodología propuesta se obtuvieron varios resultados durante cada una de las etapas planteadas. Estos resultados fueron obtenidos mediante distintos procesos de experimentación y empleando diferentes métricas. A continuación se detalla cada uno de los resultados alcanzados durante cada etapa.

5.1 Resultados del proceso de entrenamiento de la WGAN-GP

Para la etapa de entrenamiento de los modelos, se generó las curvas de entrenamiento de la [WGAN-GP](#) para cada una de las iteraciones de la validación cruzada. Estos resultados se muestran en la [Figura 5.1](#) donde se puede observar que las curvas de entrenamiento convergen a valores similares para cada iteración lo que nos da a entender que para la metodología propuesta los resultados obtenidos son independientes de la selección de datos de entrenamiento, validación y prueba.

Para proceso de análisis del entrenamiento se tomó la iteración 1 por motivo de que para el resto de iteraciones el comportamiento es similar. Realizando el análisis de la curva de entrenamiento ([Figura 5.2](#)) se puede observar que el comportamiento del entrenamiento del generador desde un inicio es oscilante, luego su oscilación sufre un decremento hasta estabilizarse al final del entrenamiento. Este comportamiento en el entrenamiento del generador es el esperado en este tipo de arquitectura por motivo de que se encuentra compitiendo con el crítico que de igual manera va mejorando conforme el entrenamiento avanza, siendo su comportamiento mucho más estable que el generador. Con el análisis de las curvas de entrenamiento se puede notar que el entrenamiento de la [WGAN-GP](#) es el esperado y no se presentan los problemas de inestabilidad o el problema del colapso modal.

De la misma manera durante esta etapa se generó las métricas para el clasificador [OC-SVM](#) utilizado para obtener el mejor crítico. En la [Figura 5.3](#) se muestra las métricas de porcentaje de falsos positivos para la clase P1 y falsos negativos para la clase P2 alcanzados durante el entrenamiento y para cada iteración de la validación cruzada. Como se puede observar el comportamiento es similar para cada una de las iteraciones por tal motivo para el análisis se tomó los resultados alcanzados durante la iteración 1.

En la [Figura 5.4](#) se observa los resultados de falsos positivos y falsos negativos alcanzados durante cada época de entrenamiento, siendo la métrica de falsos positivos utilizada para seleccionar el mejor crítico para la metodología propuesta. En el caso del comportamiento de la curva de falsos

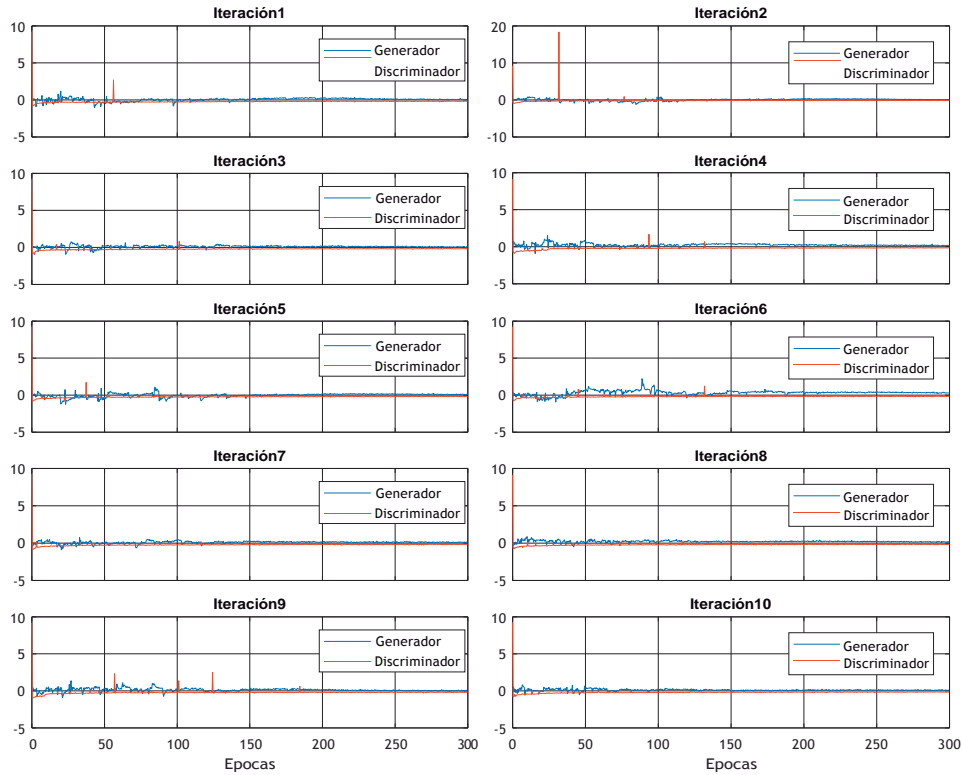


Figura 5.1: Curvas de entrenamiento WGAN-GP.

positivos se puede observar que en un inicio el porcentaje de error es alto y conforme el entrenamiento avanza este porcentaje disminuye hasta alcanzar un mínimo y luego empezar a crecer. En el caso de la metodología propuesta se tomó como mejor crítico cuando se alcanza el mínimo porcentaje de error de falsos positivos. En la Tabla 5.1 se muestra el número de época en la que se alcanzó el mínimo porcentaje de falsos positivos en cada iteración y su valor.

Durante el entrenamiento de la WGAN-GP también se generaron gráficas de las de las dos componentes principales de las características extraídas utilizando el modelo del crítico modificado, esto con el fin de observar la distribución de los datos de la clase sin falla (P1) y la clase con falla (P2) y poder verificar si los mismos forman grupos que puedan ser separados.

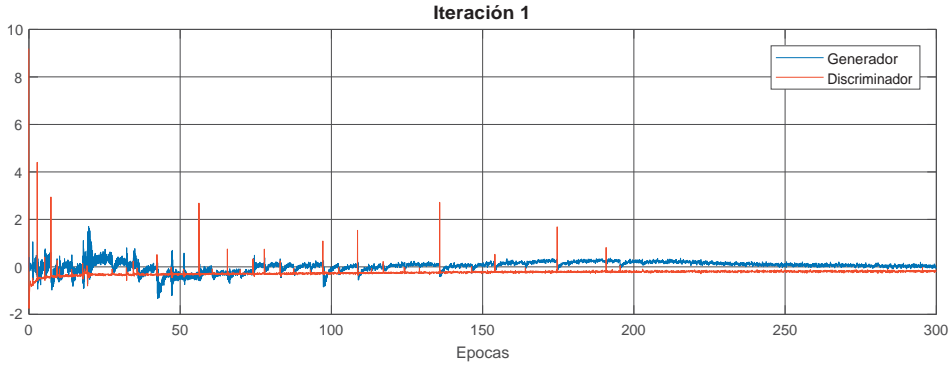


Figura 5.2: Curva de entrenamiento **WGAN-GP** para la iteración 1.

Tabla 5.1: Mínimo porcentaje de falsos positivos alcanzados en cada iteración.

<i>Iteración</i>	<i>Época</i>	<i>Mínimo porcentaje de falsos positivos alcanzado [%]</i>
1	197	2.7199
2	197	5.9606
3	178	2.662
4	149	3.2118
5	143	2.4884
6	279	1.3021
7	140	2.4306
8	151	10.1563
9	190	5.0347
10	208	0.463

Para el análisis de las componentes principales se utilizaron los mismos subconjuntos de señales empleados para la generación de las métricas del clasificador **OC-SVM**. A estos subconjuntos de señales que contenían cada

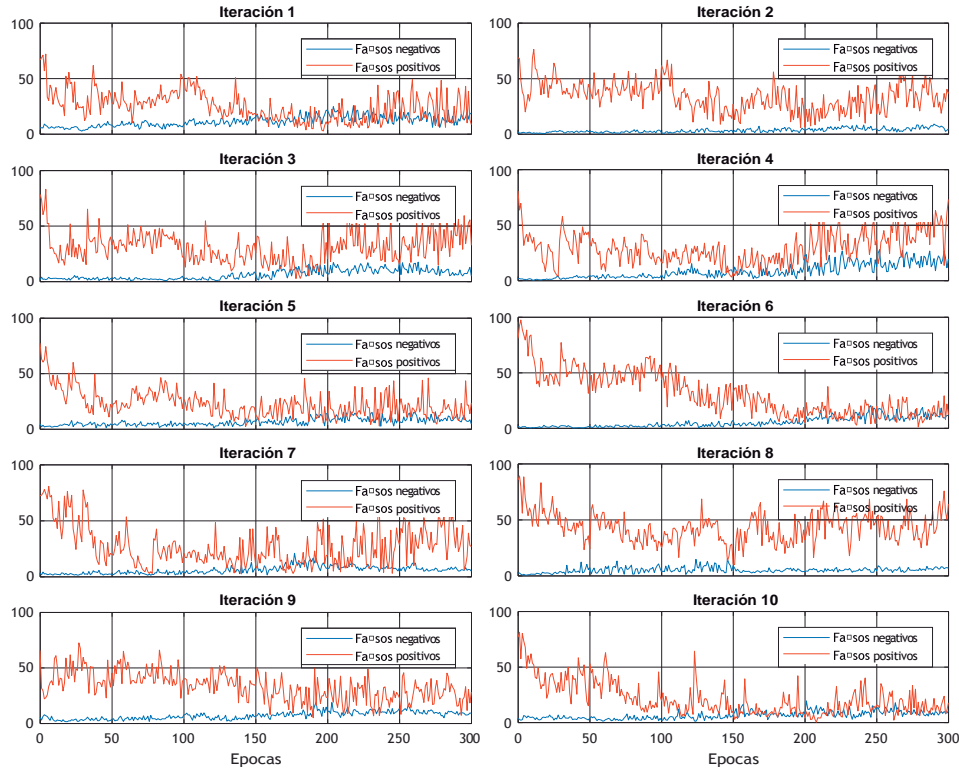


Figura 5.3: Porcentaje de falsos negativos y falsos positivos alcanzados durante el entrenamiento para todas las iteraciones.

una 256 características se les extrajo las dos componentes principales y se generaron las gráficas en dos dimensiones para poder observar la distribución de los subconjuntos de datos conforme avanzaba el entrenamiento del crítico. En la Figura 5.5 se puede observar la distribución de los datos cada 15 épocas de entrenamiento del crítico para la iteración 1 del proceso de validación cruzada, en la misma se observa que el comportamiento de los conjuntos de datos en un inicio cuando el crítico no está entrenado (época 0) se encuentran agrupados sin distinción de la clase sin falla (P1) y de la clase con falla (P2), conforme avanza el entrenamiento del crítico, los datos comienzan a agruparse según su clase, formando grupos separados. Lo cual quiere decir que conforme el crítico es entrenado mejora en la capacidad de extraer las

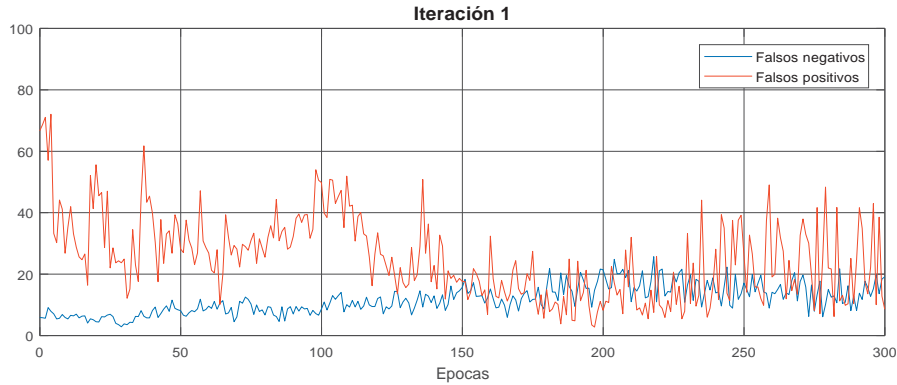


Figura 5.4: Porcentaje de falsos negativos y falsos positivos alcanzados durante el entrenamiento para la iteración 1.

características de las señales. Esto a su vez se traduce que el clasificador **OC-SVM** obtenga un mejor desempeño en la clasificación de las clases con falla y sin falla, lo que se traduce en generar menos falsos positivos como se pudo observar en la Figura 5.4.

Además de evaluar el comportamiento del crítico como extractor de características durante cada época del entrenamiento de la **WGAN-GP** también se evaluó el comportamiento del generador para generar señales del espectro de frecuencia. En la Figura 5.6 se puede observar los espectros de frecuencias resultantes del generador para cada 15 épocas de entrenamiento. En la época 0 cuando el generador no ha sido entrenado el resultado de la señal generada no se aproxima en ninguna medida a las señales reales, mientras que el entrenamiento avanza las señales generadas se aproximan cada vez más a un espectro de frecuencia. A su vez en la Figura 5.7 se puede observar una comparación entre señales generadas al finalizar el proceso de entrenamiento de la **WGAN-GP** y las señales reales. En esta imagen podemos notar que las señales generadas y señales reales se aproximan en gran medida lo que da como resultado que el entrenamiento del generador se ha realizado correctamente y ha sido capaz de aprender a generar señales del espectro de frecuencia.

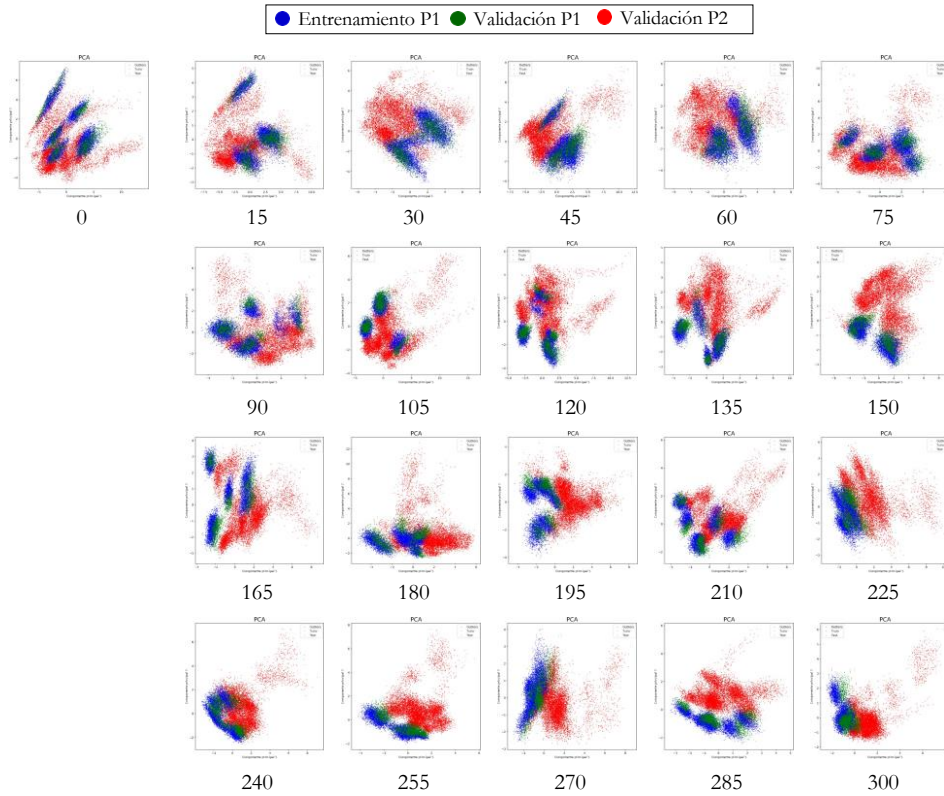


Figura 5.5: Análisis PCA de las características extraídas durante el entrenamiento para la iteración 1.

5.2 Resultados del proceso de optimización de los hiperparámetros del clasificador OC-SVM

Para el proceso de optimización de los hiperparámetros del clasificados [OC-SVM](#) se utilizó el método de búsqueda aleatoria con el cual se obtuvieron los resultados presentados en la Tabla 5.2, en la cual constan los valores de ν y γ obtenidos en cada iteración del proceso de validación cruzada, además se muestra el valor de exactitud alcanzada durante el proceso de evaluación de la búsqueda aleatoria. Como se puede observar la selección de los hiperparámetros para el clasificador [OC-SVM](#) no son independientes del conjunto de datos seleccionados para entrenamiento, validación y test,

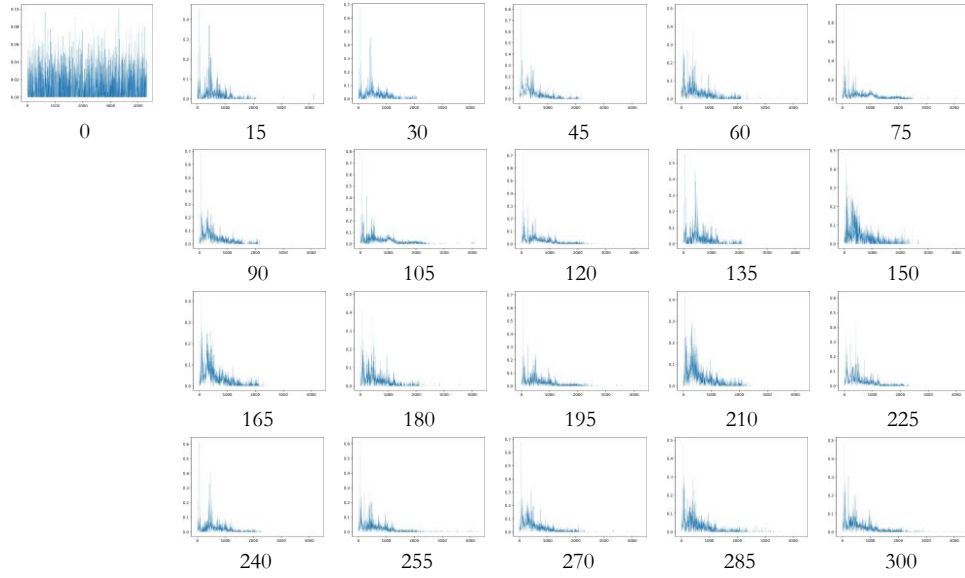


Figura 5.6: Señales generadas durante el entrenamiento.

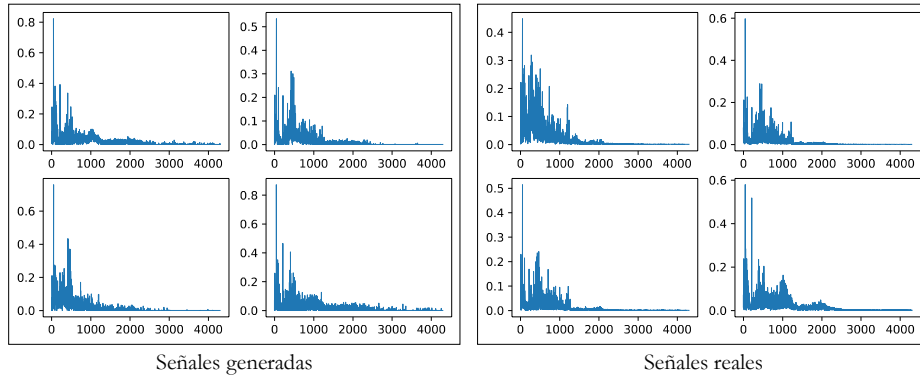


Figura 5.7: Comparación señales generadas y señales reales.

obteniendo valores distintos para cada iteración de la validación cruzada. Si se analiza los resultados de la exactitud alcanzada se puede notar que está entre un rango de 91.47 % y 95.60 %.

Tabla 5.2: Resultados obtenidos para el proceso de optimización de los hiperparámetros del clasificador OC-SVM.

<i>Iteración</i>	ν	γ	<i>Exactitud [%]</i>
1	0.0012	0.0926	92.8240
2	0.0033	0.189	95.6018
3	0.069	0.0384	92.6504
4	0.0058	0.1423	94.1406
5	0.0023	0.0976	94.6035
6	0.0093	0.0822	94.4155
7	0.003	0.1003	94.7337
8	0.0205	0.0936	91.4785
9	0.0033	0.1365	93.4172
10	0.0019	0.0337	94.9363

5.3 Resultados de las pruebas con datos de prueba

Luego de tener los hiperparámetros del clasificador optimizados se puso a prueba el sistema con el conjunto de datos de test de todas las clases, obteniendo las métricas de exactitud y de porcentaje de falsos negativos y falsos positivos. Los resultados alcanzados son presentados en la Tabla 5.3 que contiene la exactitud, los falsos negativos y falsos positivos y la Tabla 5.4 y Tabla 5.5 que contiene el porcentaje de falsos positivos por cada clase con falla (P2-P13).

Con los resultados obtenidos luego de las pruebas con el conjunto de test se realizó el análisis de error tipo 1 y tipo 2 y se calculó el margen de error con un intervalo de confianza al 95 % para comprobar que los resultados del experimento son independientes del conjunto de entrenamiento y validación que se utilice. En la Tabla 5.6 se presentan los resultados obtenidos mientras que en la Figura 5.8 se muestra su representación en un gráfico de barras.

Tabla 5.3: Resultados de exactitud, FN y FP alcanzada con la metodología propuesta.

<i>Iteración</i>	<i>Exactitud [%]</i>	<i>Falsos negativos [%]</i>	<i>Falsos positivos [%]</i>
1	93.5974	12.0746	0.7303
2	96.7605	5.4253	1.0536
3	92.0128	11.8315	4.1427
4	97.3073	5.026	0.3591
5	96.126	6.5625	1.1853
6	93.9659	9.6614	2.4066
7	96.7896	4.2534	2.1671
8	95.509	8.3593	0.6226
9	94.0225	11.0329	0.9219
10	97.8129	3.2725	1.1015

Tabla 5.4: Resultados FP para P2-P7.

<i>Iteración</i>	<i>P2 [%]</i>	<i>P3 [%]</i>	<i>P4 [%]</i>	<i>P5 [%]</i>	<i>P6 [%]</i>	<i>P7 [%]</i>
1	5.7471	0.862	0	2.0114	0	0
2	5.0287	2.7298	0	0.2873	0	0
3	10.9195	4.0229	0	4.5977	0	0
4	1.4367	1.1494	0	0.1436	0	0
5	2.1551	1.0057	0	0.5747	0	0
6	3.3045	7.9022	0.431	0.2873	0.1436	0
7	1.5804	8.6206	0	0.431	0.431	0
8	5.0287	1.4367	0	0.1436	0	0
9	4.0229	5.1724	0	0	0.1436	0
10	6.3218	2.0114	0	0.2873	0	0

Tabla 5.5: Resultados FP para P8-P13.

<i>Iteración</i>	<i>P8 [%]</i>	<i>P9 [%]</i>	<i>P10 [%]</i>	<i>P11 [%]</i>	<i>P12 [%]</i>	<i>P13 [%]</i>
1	0	0	0	0	0	0.1436
2	0	0	0	1.2931	1.2931	2.0114
3	3.7356	0	0	0.2873	20.2586	5.8908
4	0	0	0	0	1.2931	0.2873
5	0	0	0	0	10.4885	0
6	0.1436	1.0057	0.2873	2.8735	1.2931	11.2068
7	0.862	0.7183	1.8678	0	11.4942	0
8	0	0	0	0	0.431	0.431
9	0	0	0	0	0.1436	1.5804
10	0	2.0114	0	2.2988	0.2873	0

Tabla 5.6: Resultados del análisis de error tipo 1 y 2 con un nivel de significancia del 5 %.

<i>Error tipo 1</i> [%]	<i>Margen de error</i> <i>(Error tipo 1) [%]</i>	<i>Error tipo 2</i> [%]	<i>Margen de error</i> <i>(Error tipo 2) [%]</i>
7.7500	2.0314	1.4691	0.7066

5.4 Resultados de la comparación con otras metodologías para extracción de características

Para comparar el desempeño de la metodología propuesta frente a otras metodologías existentes para extracción de características se realizaron 7 experimentos más. Para realizar estos experimentos se utilizaron los conjuntos de datos que contienen las características de las señales en el dominio de tiempo, frecuencia y tiempo-frecuencia. En la Tabla 5.7 se muestra las combinaciones utilizadas para los experimentos.

Una vez establecidos las combinaciones de datos para cada experimento se realizó la división de los conjuntos de entrenamiento, validación y test siguiendo las mismas pautas que se tomaron en la metodología propuesta.

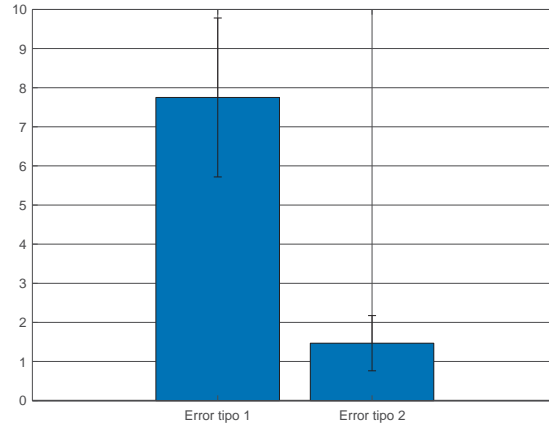


Figura 5.8: Gráfica de barras del análisis de error tipo 1 y 2.

Tabla 5.7: Combinaciones utilizadas para los experimentos.

<i>Experimento</i>	<i>Características</i>		
	<i>Tiempo</i>	<i>Frecuencia</i>	<i>Tiempo-frecuencia</i>
1	C		
2		C	
3			C
4	C	C	
5	C		C
6		C	C
7	C	C	C

Una vez realizada la división se genera un clasificador de una sola clase (OC-SVM) y se realiza el proceso de optimización de hiperparámetros utilizando el método de búsqueda aleatoria, siguiendo los procedimientos visto en el capítulo 3. Una vez realizado esto se procede a implementar la técnica de validación cruzada para garantizar que los resultados obtenidos sean independientes del conjunto selecciona para entrenamiento, validación y test.

Tabla 5.8: Resultados de experimentos utilizando distintos extractores de características.

<i>Experimento</i>	<i>Error tipo 1</i> [%]	<i>Margen de error</i> <i>(Error tipo 1) [%]</i>	<i>Error tipo 2</i> [%]	<i>Margen de error</i> <i>(Error tipo 2) [%]</i>	<i>Exactitud</i>
WGAN-GP	7.7500	2.0314	1.4691	0.7066	95.3905
1	9.6224	5.0745	11.8411	0.5512	89.2683
2	16.1597	7.7214	10.3022	1.1169	86.7691
3	10.4184	3.8154	19.3768	3.5647	85.1024
4	15.3411	5.3698	5.5230	0.6961	89.5680
5	12.7917	3.0446	3.8680	0.7775	91.6702
6	8.7405	5.5413	2.6889	0.9562	94.2853
7	8.4670	6.5656	0.9103	0.3914	95.3114

Este proceso descrito es realizado para cada experimento propuesto con el cambio de que para cada experimento se utilizan distintas combinaciones de características.

Con los resultados obtenidos luego de finalizar los 7 experimentos se realizó el análisis de error tipo 1 y tipo 2 y se calculó del margen de error con un intervalo de confianza al 95 % verificando que para todos los experimentos propuestos los resultados son independientes del conjunto de entrenamiento, validación y test que se utilice. Los resultados del análisis de error tipo 1 y 2 son mostrados en la Tabla 5.8. A su vez también se presenta un gráfico de barras de los resultados alcanzados.

Como se puede observar en la Tabla 5.8 en el caso del error tipo 1 (Falsos negativos) y la exactitud la metodología propuesta supera a todos los experimentos alcanzando el menor porcentaje de error. Mientras que el caso del error tipo 2 (Falsos positivos) la metodología propuesta supera a seis de los siete experimentos que utilizan metodologías tradicionales para extracción de características, solo siendo superada por el experimento que combina todas las características extraídas en los dominios de tiempo, frecuencia y tiempo-frecuencia. Estos resultados alcanzados dan a entender que la metodología propuesta puede ser aplicada para la detección de fallos en una bomba centrífuga multietapa y supera en términos generales a los enfoques clásicos.

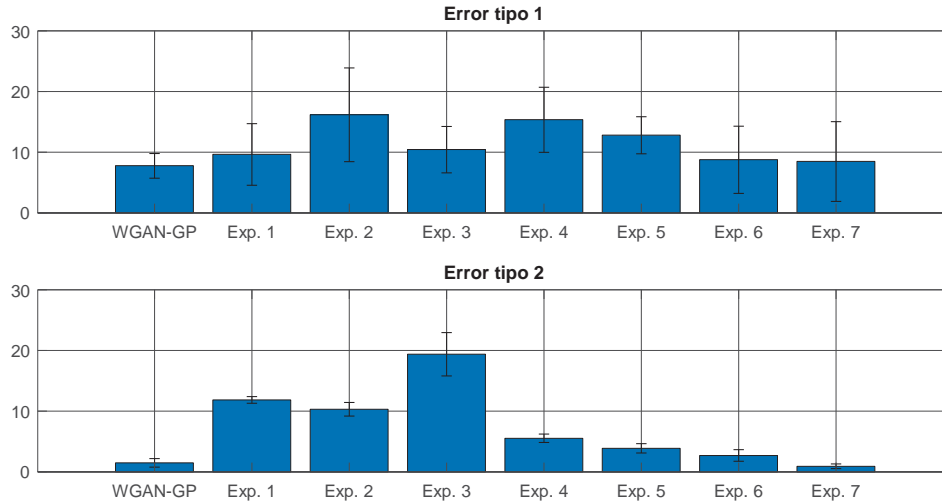


Figura 5.9: Gráfica comparativa de los resultados alcanzados.

5.5 Resultados de funcionamiento del sistema de detección online de fallos.

La etapa final del proyecto fue la de implementar un sistema de detección online de fallos en la nube, el cual sea capaz de detectar satisfactoriamente un fallo en una bomba centrífuga multietapa de eje vertical, empleando para esta tarea los modelos generados con la nueva metodología propuesta.

Para el sistema propuesto se utilizó una arquitectura sin servidor por las ventajas que esta ofrece, obteniendo como resultado un sistema de detección online de fallos que se ejecuta en la nube. A este sistema se lo puso a prueba para verificar el funcionamiento. Para la prueba del sistema se envió señales del sensor A1 recolectadas del banco de pruebas de bomba trabajando en estado sin falla, en este caso el sistema de detección online de fallos paso la prueba sin cometer ningún error en la clasificación del estado del sistema. Luego se envió señales recolectadas del banco de prueba trabajando en estado con fallo y se comprobó que el mismo era detectado por el sistema de detección online de fallos, a su vez al momento de detectar el fallo se comprobó que la alarma implementada en el sistema realizaba la notificación al usuario a través de correo electrónico. En la Figura 5.10 se puede observar el panel de visualización de la condición de la bomba centrífuga multietapa.

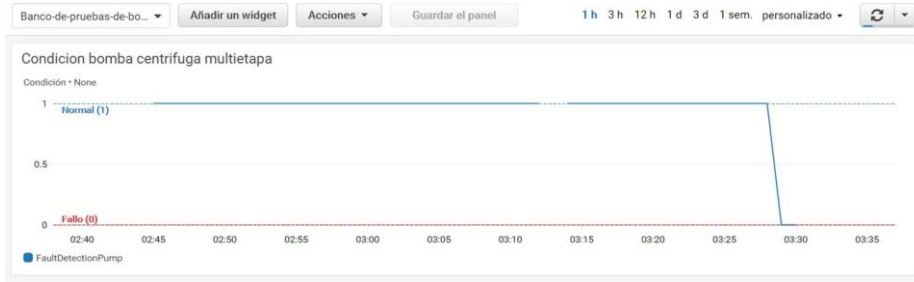


Figura 5.10: Funcionamiento del sistema de detección online de fallos.



Figura 5.11: Tiempo de ejecución de funciones lambda.

Además de realizar pruebas de funcionamiento del sistema también se generó métricas para verificar el tiempo de ejecución de las funciones lambda creadas. En la Figura 5.11 se puede observar el tiempo que tarda en ejecutarse cada función lambda.

Realizando el análisis del tiempo de ejecución de la función lambda utilizada para la extracción de características (línea azul) se puede observar que al momento de ejecutarse la función por primera vez el tiempo de ejecución es de aproximadamente 17 segundos. Luego de un minuto al ser ejecutada nuevamente el tiempo de ejecución es menor a 1 segundo. Esto es debido a que al momento de ejecutarse la función por primera vez la misma

tiene que realizar la carga de los modelos y de las librerías necesarias. Al ser ejecutada la función luego de un intervalo corto de tiempo (menor a 5 minutos) las librerías y los modelos se encuentran almacenados en el cache de la función por lo cual el tiempo de ejecución disminuye. De igual manera sucede con el tiempo de ejecución de la función lambda utilizada para la detección de fallos (línea amarilla). En este caso la ejecución de la función por primera vez es de aproximadamente de 16 segundos y para procesos posteriores disminuye hasta alcanzar un tiempo de ejecución menor a 1 segundo. Estos tiempos de ejecución de las funciones no afectan el desempeño del sistema de detección online de fallos debido a que el intervalo mínimo en el que pueden ser adquiridas las señales del banco de bombas se estableció en un minuto.

Capítulo 6

Conclusiones y recomendaciones

En el presente capítulo se encuentran las conclusiones y recomendaciones obtenidas a partir de la realización del presente proyecto, además de los trabajos futuros que se podrían desarrollar.

En este trabajo se desarrolló una metodología para la creación de modelos de detección de fallo basado en modelos generativos obtenidos a partir de señales multimodales en condición normal de una bomba hidráulica centrífuga multietapa vertical. Como primera etapa se realizó la adquisición de señales de vibración proveniente de la bomba en estado normal y estado con falla (12 tipos de fallos) trabajando bajo 5 distintas condiciones de funcionamiento (caudal, presión). Utilizando los datos del estado normal de la bomba se entrenó una [WGAN-GP](#). Luego del entrenamiento el modelo crítico obtenido se utilizó como extractor de características. Utilizando las características extraídas por el modelo crítico se generó un clasificador de una sola clase ([OC-SVM](#)) usando los datos en estado normal de bomba para su entrenamiento. Finalmente, se evaluó el clasificador entrenado con los datos de la bomba en estado con falla y en estado normal para determinar su desempeño.

Además se realizó la extracción de características de las señales en estado con falla y sin falla utilizando métodos tradicionales (extracción de características en el dominio del tiempo, frecuencia y tiempo-frecuencia) y se generó 7 experimentos diferentes combinando distintas características en cada uno de ellos. Para cada experimento se utilizó las características para entrenar un [OC-SVM](#) para realizar la clasificación entre clase con falla y sin falla. Los resultados de desempeño obtenidos en los 7 experimentos se utilizaron para comparar con el desempeño obtenido utilizando la metodología propuesta. Además, se implementó un sistema de detección online de fallos en la nube con el modelo que presentó mejor desempeño, el cual fue puesto a prueba con datos de la bomba trabajando en estado normal y en estado con falla.

Los resultados obtenidos en cada una de las etapas de desarrollo permiten llegar a las siguientes conclusiones:

- La metodología propuesta logra superar en desempeño a todos los experimentos propuestos, alcanzando una mayor exactitud en cuanto a la detección de fallos. Por lo tanto, puede ser utilizada favorablemente en el campo de la detección de fallos en bombas.
- Al utilizar [OC-SVM](#) como método para detección de fallos es importante una correcta selección de los hiperparámetros ν y γ . En el caso de la metodología propuesta estos parámetros fueron optimizados utilizando búsqueda aleatoria, con lo cual se pudo encontrar los hiperparámetros que obtuvieron el mejor desempeño para el clasificador. En base a los resultados obtenidos durante

los experimentos realizados se puede notar que la selección de estos hiperparámetros influye significativamente en el desempeño del clasificador. Siendo el parámetro ν el que tiene mayor repercusión en los resultados por su influencia en la cantidad de vectores de apoyo que se utilice para el clasificador.

- El sistema de detección online de fallos en la nube que fue desarrollado e implementado en el presente proyecto puede ser utilizado satisfactoriamente en el campo de detección de fallos. Además, al tener un sistema que utiliza una arquitectura implementada en la nube permite que el mismo pueda ser fácilmente escalable y sea posible utilizar otros modelos para detección de fallos ya sea en máquinas similares o distintas.

A continuación se mencionan algunas recomendaciones para el desarrollo de trabajos similares o futuros:

- En cuanto a la generación del conjunto de datos para el desarrollo de la metodología es recomendable que los mismos sean adquiridos del proceso bajo distintas condiciones de funcionamiento. Esto con el fin de poseer una mayor diversidad de datos que puedan utilizarse para generar un modelo con capacidad de generalización ante distintos escenarios de funcionamiento.
- Es recomendable utilizar arquitecturas distintas a la de la [GAN](#) original ya que la misma presenta algunos problemas vistos en la Sección 3.4.2. En el caso del trabajo presentado se utilizó una [WGAN-GP](#) que es una mejora de la [GAN](#) original.
- Para el entrenamiento de los modelos se recomienda hacer uso de sistemas que utilicen GPUs para los cálculos computacionales, ya que las mismas se encuentran optimizadas para realizar cálculos matriciales, reduciendo en gran medida los tiempos de entrenamiento de los modelos.

El enfoque utilizado para la metodología propuesta es el de utilizar solamente datos del estado normal de la máquina para generar los modelos del crítico, pero para el proceso entrenamiento de la [WGAN-GP](#) se utilizó una detención temprana de entrenamiento para la cual se buscó una métrica a evaluar como parámetro de detención. Para esta métrica se utilizó los falsos positivos alcanzados por una sola de las clases con falla, de las 12 que se

disponía, provocando que para implementar la metodología sea necesario tener datos del estado con falla de la máquina. En trabajos futuros se pretende abordar esta problemática y generar una métrica para la detención temprana del entrenamiento en la cual no sea necesario tener datos del estado con falla de la máquina.

Bibliografía

- J. Agüera. *Mecánica de los fluidos incompresibles y turbo máquinas hidráulicas*. 5ta edition, 2002.
- M. A. S. ALTobi, G. Bevan, P. Wallace, D. Harrison, and K. P. Ramachandran. Fault diagnosis of a centrifugal pump using MLP-GABP and SVM with CWT. *Engineering Science and Technology, an International Journal*, 22(3):854–861, 2019. ISSN 22150986. doi: 10.1016/j.jestch.2019.01.005. URL <https://doi.org/10.1016/j.jestch.2019.01.005>.
- S.-i. Amari and S. Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.
- M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.
- G. F. Bin, J. J. Gao, X. J. Li, and B. S. Dhillon. Early fault diagnosis of rotating machinery based on wavelet packets - Empirical mode decomposition feature extraction and neural network. *Mechanical Systems and Signal Processing*, 27(1):696–711, 2012. ISSN 08883270. doi: 10.1016/j.ymssp.2011.08.002.
- D. Cabrera, F. Sancho, C. Li, M. Cerrada, R. V. Sánchez, F. Pacheco, and J. V. de Oliveira. Automatic feature extraction of time-series applied to fault severity assessment of helical gearbox in stationary and non-stationary speed operation. *Applied Soft Computing Journal*, 58: 53–64, 2017. ISSN 15684946. doi: 10.1016/j.asoc.2017.04.016. URL <http://dx.doi.org/10.1016/j.asoc.2017.04.016>.
- D. Cabrera, F. Sancho, M. Cerrada, R.-V. Sánchez, and F. Tobar. Echo state network and variational autoencoder for efficient one-class learning on dynamical systems. *Journal of Intelligent & Fuzzy Systems*, 34(6): 3799–3809, 2018.

- D. Cabrera, F. Sancho, J. Long, R.-V. Sánchez, S. Zhang, M. Cerrada, and C. Li. Generative adversarial networks selection approach for extremely imbalanced fault diagnosis of reciprocating machinery. *IEEE Access*, 7: 70643–70653, 2019.
- J. C. Calderón Malla and F. I. Montalván Pulla. Adquisición de señales acústicas y vibratorias para el diagnóstico de fallos en una bomba centrífuga multietapa de eje vertical. B.S. thesis, 2019.
- CEN. *BSI Standards Publication Maintenance – Maintenance terminology*. European Standard, Brussels, 2010.
- Z. Chen, C. Li, and R.-V. Sanchez. Gearbox fault identification and classification with convolutional neural networks. *Shock and Vibration*, 2015, 2015.
- Z. Cheng, N. Hu, Y. Zhang, and X. Zhang. Fault Detection Method Based on the Condition Indicators of Compressive Sampling for the Gearbox in Helicopter. *Proceedings - 2018 Prognostics and System Health Management Conference, PHM-Chongqing 2018*, pages 333–337, 2019. doi: 10.1109/PHM-Chongqing.2018.00063.
- Ş. Cobzaş, R. Miculescu, and A. Nicolae. *Lipschitz functions*, volume 2241. Springer, 2019.
- C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000. doi: 10.1017/CBO9780511801389.
- S. García. *Ingeniería de Mantenimiento, manual práctico para la gestión eficaz del mantenimiento industrial*. 2013.
- I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. arxiv e-prints (june 2014). *arXiv preprint stat.ML/1406.2661*, 2014.
- I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

- L. G. Hanin. Kantorovich-rubinstein norm and its application in the theory of lipschitz spaces. *Proceedings of the American Mathematical Society*, 115 (2):345–352, 1992.
- K. Heller, K. Svore, A. D. Keromytis, and S. Stolfo. One class support vector machines for detecting anomalous windows registry accesses. 2003.
- D. T. Hoang and H. J. Kang. Rolling element bearing fault diagnosis using convolutional neural network and vibration image. *Cognitive Systems Research*, 53:42–50, 2019. ISSN 13890417. doi: 10.1016/j.cogsys.2018.03.002. URL <https://doi.org/10.1016/j.cogsys.2018.03.002>.
- O. Kocaaslan, K. M. Guleren, O. Babayigit, M. Ozgoren, and M. H. Aksoy. Numerical identification of blade exit angle effect on the performance for a multistage centrifugal pump impeller. *EPJ Web of Conferences*, 92:02003, 2015. doi: 10.1051/epjconf/20159202003.
- S. Kumar, D. Goyal, R. K. Dang, S. S. Dhami, and B. S. Pabla. Condition based maintenance of bearings and gears for fault detection-A review. *Materials Today: Proceedings*, 5(2):6128–6137, 2018. ISSN 22147853. doi: 10.1016/j.matpr.2017.12.219. URL <https://doi.org/10.1016/j.matpr.2017.12.219>.
- C. Li, R. V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez. Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis. *Neurocomputing*, 168: 119–127, 2015a. ISSN 18728286. doi: 10.1016/j.neucom.2015.06.008. URL <http://dx.doi.org/10.1016/j.neucom.2015.06.008>.
- C. Li, R. V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez. Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis. *Neurocomputing*, 168: 119–127, 2015b. ISSN 18728286. doi: 10.1016/j.neucom.2015.06.008. URL <http://dx.doi.org/10.1016/j.neucom.2015.06.008>.
- C. Li, J. V. De Oliveira, R. V. Sanchez, M. Cerrada, G. Zurita, and D. Cabrera. Fuzzy determination of informative frequency band for bearing fault detection. *Journal of Intelligent and Fuzzy Systems*, 30(6): 3513–3525, 2016a. ISSN 18758967. doi: 10.3233/IFS-162097.
- C. Li, R. V. Sánchez, G. Zurita, M. Cerrada, and D. Cabrera. Fault diagnosis for rotating machinery using vibration measurement deep statistical

- feature learning. *Sensors (Switzerland)*, 16(6), 2016b. ISSN 14248220. doi: 10.3390/s16060895.
- C. Li, R. V. Sánchez, G. Zurita, M. Cerrada, and D. Cabrera. Fault diagnosis for rotating machinery using vibration measurement deep statistical feature learning. *Sensors (Switzerland)*, 16(6), 2016c. ISSN 14248220. doi: 10.3390/s16060895.
- C. Li, R. V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mechanical Systems and Signal Processing*, 76-77: 283–293, 2016d. ISSN 10961216. doi: 10.1016/j.ymsp.2016.02.007. URL <http://dx.doi.org/10.1016/j.ymsp.2016.02.007>.
- C. Li, D. Cabrera, F. Sancho, M. Cerrada, R.-V. Sánchez, and E. Estupinan. From fault detection to one-class severity discrimination of 3d printers with one-class support vector machine. *ISA Transactions*, 2020.
- Z. Liu, Y. Liu, H. Shan, B. Cai, and Q. Huang. A fault diagnosis methodology for gear pump based on EEMD and bayesian network. *PLoS ONE*, 10(5): 1–15, 2015. ISSN 19326203. doi: 10.1371/journal.pone.0125703.
- X. Luo, W. Wei, B. Ji, Z. Pan, W. Zhou, and H. Xu. Comparison of cavitation prediction for a centrifugal pump with or without volute casing. *Journal of Mechanical Science and Technology*, 27(6):1643–1648, 2013. ISSN 1738494X. doi: 10.1007/s12206-013-0411-5.
- G. McGrath and P. R. Brenner. Serverless computing: Design, implementation, and performance. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 405–410. IEEE, 2017.
- R. K. Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.
- G. Mousmoulis, N. Karlsen-Davies, G. Aggidis, I. Anagnostopoulos, and D. Papantonis. Experimental analysis of cavitation in a centrifugal pump using acoustic emission, vibration measurements and flow visualization. *European Journal of Mechanics, B/Fluids*, 75:300–311, 2019. ISSN 09977546. doi: 10.1016/j.euromechflu.2018.10.015. URL <https://doi.org/10.1016/j.euromechflu.2018.10.015>.
- N. Rodríguez. Diagnóstico de falla en las paletas de una bomba de desplazamiento positivo por análisis de vibraciones. pages 1–7, 2016.

- N. B. Ruparelia. Software development lifecycle models. *SIGSOFT Softw. Eng. Notes*, 35(3):8–13, May 2010. ISSN 0163-5948. doi: 10.1145/1764810.1764814. URL <https://doi.org/10.1145/1764810.1764814>.
- R.-V. Sánchez, P. Lucero, J.-C. Macancela, M. Cerrada, R. E. Vásquez, and F. Pacheco. Multi-fault diagnosis of rotating machinery by using feature ranking methods and svm-based classifiers. In *2017 international conference on sensing, diagnostics, prognostics, and control (SDPC)*, pages 105–110. IEEE, 2017.
- R.-V. Sánchez, P. Lucero, R. E. Vásquez, M. Cerrada, J.-C. Macancela, and D. Cabrera. Feature ranking for multi-fault diagnosis of rotating machinery by using random forest and knn. *Journal of Intelligent & Fuzzy Systems*, 34(6):3463–3473, 2018.
- R.-V. Sánchez, P. Lucero, J. C. Macancela, M. Cerrada, D. Cabrera, and R. Vásquez. Gear crack level classification by using knn and time-domain features from acoustic emission signals under different motor speeds and loads. In *Proceedings-2018 International Conference on Sensing, Diagnostics, Prognostics, and Control, SDPC 2018*, 2019.
- R. V. Sánchez Loja et al. Diagnóstico de fallos en cajas de engranajes con base en la fusión de datos de señales de vibración, corriente y emisión acústica. 2018.
- B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
- I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008.
- Q. Thai and C. Lee. The cavitation behavior with short length blades in centrifugal pump. *Journal of Mechanical Science and Technology*, 24(10): 2007–2016, 2010. ISSN 1738494X. doi: 10.1007/s12206-010-0705-9.
- H. Thanh-Tung and T. Tran. On catastrophic forgetting and mode collapse in gans.
- A. H. Tsang. Condition-based maintenance: tools and decision making. *Journal of quality in maintenance engineering*, 1995.
- G. White. Introducción al análisis de vibraciones. USA, Azima DLI, pages 22–29, 2010.

- M. Zhang, Z. Jiang, and K. Feng. Research on variational mode decomposition in rolling bearings fault diagnosis of the multistage centrifugal pump. *Mechanical Systems and Signal Processing*, 93(15): 460–493, 2017. ISSN 10961216. doi: 10.1016/j.ymssp.2017.02.013. URL <http://dx.doi.org/10.1016/j.ymssp.2017.02.013>.
- S. Zhang, R. Zhang, S. Zhang, and J. Yang. Effect of Impeller Inlet Geometry on Cavitation Performance of Centrifugal Pumps Based on Radial Basis Function. *International Journal of Rotating Machinery*, 2016, 2016. ISSN 15423034. doi: 10.1155/2016/6048263.
- J. H. Zhou, L. Wee, and Z. W. Zhong. A knowledge base system for rotary equipment fault detection and diagnosis. *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, pages 1335–1340, 2010. doi: 10.1109/ICARCV.2010.5707843.

Glosario

FN Falsos Negativos.

FP Falsos Positivos.

GAN *Generative Adversarial Network* .

GIDTEC Grupo de Investigación y Desarrollo en Tecnologías Industriales.

OC-SVM *One class - Support Vector Machines* .

SVM *Support Vector Machines* .

UPS Universidad Politécnica Salesiana.

WGAN-GP *Wasserstein Generative Adversarial Network with Gradient Penalty* .