



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MECÁNICA

**DISEÑO DE UN MODELO DE APRENDIZAJE DE MÁQUINAS PARA DETECCIÓN DE FALLAS
EN GENERADORES EÓLICOS EN PARQUE EÓLICO CALAMA**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

FERNANDO NICOLÁS MONÁRDEZ DELLACASA

PROFESORA GUÍA:
VIVIANA MERUANE NARANJO

MIEMBROS DE LA COMISIÓN:
SEBASTIÁN CAMPILLAY SALAZAR
RAYMI VÁSQUEZ MORENO

Este trabajo ha sido parcialmente financiado por:
ENGIE ENERGÍA CHILE

SANTIAGO DE CHILE

2024

DISEÑO DE UN MODELO DE APRENDIZAJE DE MÁQUINAS PARA DETECCIÓN DE FALLAS EN GENERADORES EÓLICOS EN PARQUE EÓLICO CALAMA

La detección temprana de fallas se vuelve una tarea esencial en la operación de parques eólicos como el Parque Eólico Calama, de la empresa Engie Energía Chile. Un porcentaje importante de los costos asociados a la operación de estos parques se debe a detenciones repentinas y mantenimientos correctivos, lo cual se desea reducir en la medida de lo posible. El presente trabajo de título busca diseñar un modelo de detección de fallas, a través de algoritmos de *aprendizaje de máquinas (machine learning)*, en turbinas eólicas de este parque con el fin de realizar mantenimiento en un tiempo oportuno y ahorrar costos.

Los equipos a estudiar corresponden a turbinas Siemens Gamesa SG4.5-145 IIB, cuyos componentes principales son el nacelle, el rotor y la torre, en los cuales se ubican diversos sensores que registran variables de operación a través de un sistema SCADA.

Se plantea el uso de algoritmos de *aprendizaje de máquinas* en base al concepto de Mantenimiento 4.0 y sistemas PHM, que utilizan herramientas de la ciencia de datos para detectar síntomas de falla, clasificarlas y predecir el tiempo de vida útil remanente de activos.

Los modelos se desarrollan mediante lenguaje de programación *Python* utilizando la plataforma de acceso libre y gratuita *Spyder*. Dentro de esta plataforma se importan las librerías *Pandas* (para procesamiento de hojas de cálculo), *NumPy* (para aplicar operaciones matemáticas a series de datos) y *Keras* (para construir, entrenar y testear redes de aprendizaje profundo).

Se evalúa el rendimiento de los algoritmos de detección de novedades *Elliptic Envelope*, *Isolation Forest*, *Local Outlier Factor* y *One-Class Support Vector Machine*, y se observa que estos no logran adaptarse al comportamiento dinámico de las turbinas eólicas, por lo que se descartan como candidatos.

Luego, se utilizan redes neuronales *Deep Autoencoder*, que entregan una reconstrucción de los datos de entrada. La forma que tienen estas redes de detectar anomalías es mediante el *error de reconstrucción*, una medida de cuánto difiere el dato real con respecto a la predicción del modelo. Adicionalmente, se incluye un umbral adaptativo basado en Teoría del Valor Extremo para determinar qué valores son considerados como anómalos. Se crean modelos para los sistemas *Gearbox*, *Grupo Hidráulico*, y *Generador* con las variables disponibles en SCADA.

Los resultados indican que con este enfoque es posible anticiparse en gran medida a las fallas que presentan las turbinas eólicas, por lo que este tipo de modelos tienen un gran potencial para ser utilizados como complemento en el monitoreo de la condición de estos equipos. Para lograr esto será necesario trabajar con mayor exhaustividad en la calidad de los datos de entrenamiento y en el registro de fallas de las turbinas para una mejor interpretación de resultados.

*A quienes están,
estuvieron y estarán.
Les tendré siempre conmigo*

Agradecimientos

Este trabajo es la culminación de un periodo tan desafiante como hermoso. La universidad me ha dejado aprendizajes que llevaré conmigo por el resto de mi vida, ya sea en lo personal como en lo académico, por lo que considero necesario hacer una reflexión, agradecer la experiencia y sobretodo dedicarles unas palabras a las personas que han sido parte del viaje.

Quiero partir dando las gracias a mi mamá Ivonne y a mi papá Osvaldo, porque cada paso que he dado no habría sido posible sin ellos a mi lado. Me dieron la vida y un hogar cálido donde pude crecer siendo un niño feliz. Gracias a ellos tengo agua y alimento para vivir, una cama donde dormir, ropa para el frío y el calor, una pieza propia, un patio donde salir a tomar aire y todo el resto de las cosas que forman parte de mi normalidad. Pero esto es tan solo lo material, que si bien ha sido esencial para que tenga una vida tranquila, no lo puedo comparar con lo humano; llevo conmigo el amor infinito, la comprensión y preocupación que recibo de su parte, y las lecciones de vida que me han llevado a ser una persona con valores y consciencia del mundo que me rodea. Dudo que sin su ejemplo y sus enseñanzas hubiera tenido la capacidad de salir con la frente en alto en el colegio y luego en la universidad. Puedo decir con mucho orgullo que tengo a la mejor mamá y el mejor papá que podría haber pedido y sólo quiero poder darles todos los agradecimientos posibles en mi futuro como ingeniero.

Por supuesto que le quiero dar las gracias también a mi hermana Paulina, por ser una de mis principales fuentes de alegría a lo largo de mi vida. Aparte de ser la única persona con edad similar a la mía dentro del núcleo familiar, creo que ha sido con la que más tiempo he compartido y la verdad no podría imaginar cómo sería haber experimentado tantos años sin ella al lado. Hemos pasado por momentos de risas y momentos tristes, y en todos ellos hemos logrado darnos apoyo a pesar de nuestras especiales formas de ser, porque tenemos la capacidad de comunicarnos sin la necesidad de las palabras. Le agradezco por estar ahí siempre y por ser así como es ella, y aceptarme a mí de la misma forma.

Le agradezco a mis abuelitas María y Rita, y a mis abuelitos Osvaldo y René por quererme con todo su corazón desde el día que nací. Me siento afortunado por haberlos tenido tan cerca de mi casa, porque he podido disfrutarlos casi todas las semanas, y lamento que la universidad me haya impedido verlos en ocasiones por la carga académica, pero me alegra que a partir de ahora nos podamos ver más. Su apoyo incondicional y deseos sinceros por mi éxito me dieron fuerza para llegar hoy a este punto, y les agradezco todas las veces que me miraron con orgullo y me imaginaron como un profesional, porque darles esa alegría ha sido una de mis principales motivaciones. A mi abuelita María le habría encantado estar presente hoy, y ojalá pudiera darle un abrazo ahora, pero sé que me sigue observando y cuidando desde el cielo, rezando por mi bienestar.

A mis tías Fabiola, Sandra y Ximena, y a mi tío Mauricio les agradezco también por acompañarme y

regalonearme durante toda la vida, por estar siempre presentes, por celebrar mis logros conmigo y darme razones para tener más confianza en mí mismo y en la persona que puedo llegar a ser.

Doy gracias a mi grupo de amigos del colegio, los Repollitos F.C., por todos los momentos que hemos compartido, por las anécdotas y chascarros de las que hemos sido protagonistas y espectadores y de las cuales conversamos hasta la actualidad. Son un pilar esencial para mi estabilidad emocional y en varias ocasiones a lo largo de la carrera me ayudaron a sacarla adelante, en particular durante la pandemia, por las conversaciones en Discord y las veces que por fortuna pudimos vernos en persona. Me pone contento pensar que, a pesar de haber salido hace 7 años ya del colegio, nos sigamos juntando y que podamos hablar como si el tiempo no hubiera pasado, porque me hace creer que pasarán otros 7 años y más y nos seguiremos viendo igual, riéndonos de cualquier estupidez.

A mis amistades de la universidad les agradezco por acompañarme en los ratos libres dentro y fuera de esta. Están quienes conozco desde el primer día de la semana de inducción, y por otro lado quienes conocí más con el pasar de los semestres, y sé que al leer esto sabrán de quienes hablo, así que quiero decirles que no puedo imaginar cómo habría sido pasar estos 7 años sin ustedes, sin alguien con quien compartir cuando la cabeza ya no daba más de tanto estudio. Gracias por las risas y las memorias que he creado con ustedes, me han hecho creer que dejé una marca en mi paso por la universidad, y estoy muy feliz por haberlos conocido. Que sepan que siempre tendrán en mí a alguien a quien acudir y que los recibirá siempre con los brazos abiertos. Sé que nos seguiremos viendo con el pasar de los años.

A Engie y a las personas que conocí en sus oficinas, les doy gracias por permitirme haber trabajado con ellos durante mi práctica y posteriormente la memoria, por haberme invitado en tantas ocasiones a compartir como si fuera uno más del equipo. Lo pasé muy bien durante ese año, y esos fueron días que no olvidaré nunca, al igual que sus nombres y rostros. En particular agradezco a Sebastián que fue mi tutor y co-guía de memoria, por haber depositado su confianza en mí cuando me aceptó como practicante, por darme la oportunidad de hacer este trabajo y por todos los consejos y retroalimentación que me dio.

Por último agradezco a la profesora Viviana por inspirarme con este tema de trabajo y por ser mi profesora guía, acompañándome y orientándome durante el proceso. Gracias por el tiempo, por aceptar mis reuniones y siempre responder mis dudas con la mejor voluntad del mundo; el conocimiento y material que me compartió a lo largo de estos meses fueron de vital importancia para la finalización del trabajo.

Les agradeceré por siempre el haber estado conmigo.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.2.1. Objetivo general	1
1.2.2. Objetivos específicos	2
1.3. Alcances	2
2. Antecedentes	3
2.1. Parque Eólico Calama	3
2.2. Equipo estudiado	3
2.2.1. Turbina eólica	3
2.2.2. Equipo de estudio	4
2.2.2.1. Nacelle	5
2.2.2.2. Rotor	6
2.2.2.3. Torre	7
2.2.3. Monitoreo SCADA	8
2.3. Mantenimiento	11
2.3.1. Industria 4.0	11
2.3.2. Mantenimiento 4.0	11
2.3.3. PHM	12
2.4. Machine Learning	13
2.4.1. Modelos utilizados	14
2.4.1.1. Algoritmos de Detección de Anomalías/Novedades	14
2.4.1.2. Deep Autoencoder	16
3. Metodología	23
3.1. Recursos	24
4. Desarrollo y discusión de resultados	26
4.1. Análisis de alarmas	26
4.2. Casos de estudio	31
4.3. Inicialización y preprocesamiento de datos	32
4.4. Modelos basados en algoritmos detección de novedades	36
4.4.1. Elliptic Envelope	36
4.4.2. Isolation Forest	38
4.4.3. Local Outlier Factor	40

4.4.4.	One-Class Support Vector Machine	42
4.5.	Modelo basado en Deep Autoencoder	43
4.5.1.	Arquitectura Modelo Gearbox	48
4.5.2.	Arquitectura Modelo Grupo Hidráulico	49
4.5.3.	Arquitectura Modelo Generador	51
4.5.4.	Resultados Turbina N°2	53
4.5.5.	Resultados Turbina N°9	57
4.5.6.	Resultados Turbina N°27	59
4.5.7.	Resultados Turbina N°29	63
5.	Conclusiones y trabajo futuro	68
5.1.	Conclusiones	68
5.2.	Trabajo futuro	69
Bibliografía		71
Anexos		74
A.	Resultados Turbina N°2	74
A.1.	Modelo Gearbox	75
A.2.	Modelo Grupo Hidráulico	77
A.3.	Modelo Generador	79
B.	Resultados Turbina N°9	81
B.1.	Modelo Gearbox	81
B.2.	Modelo Grupo Hidráulico	83
B.3.	Modelo Generador	84
C.	Resultados Turbina N°27	86
C.1.	Modelo Gearbox	86
C.2.	Modelo Grupo Hidráulico	88
C.3.	Modelo Generador	90
D.	Resultados Turbina N°29	92
D.1.	Modelo Gearbox	92
D.2.	Modelo Grupo Hidráulico	93
D.3.	Modelo Generador	96

Índice de Tablas

2.1.	Variables SCADA	9
2.2.	Alarmas AWS	10
4.1.	Listado de alarmas ordenadas por tiempo total	26
4.2.	Listado de turbinas ordenadas por tiempo total de alarmas.	27
4.3.	Listado de alarmas que derivan en un estado NO COMUNICA, ordenadas por tiempo acumulado.	29
4.4.	Listado de turbinas ordenadas por su 1º y 2º alarma predecesora de estado NO COMUNICA de mayor tiempo acumulado.	30
4.5.	Casos de estudio	31
4.6.	Variables de sistema gearbox	32
4.7.	Variables de sistema grupo hidráulico	32
4.8.	Variables de sistema generador	32
4.9.	Valores de outlier fraction utilizados	36
4.10.	Fallas Grupo Hidráulico Turbina Nº2	38
4.11.	Parámetros para Isolation Forest	39
4.12.	Parámetros para Local Outlier Factor	40
4.13.	Parámetros para OC-Support Vector Machine	42
4.14.	Arquitectura Modelo Gearbox	48
4.15.	Combinaciones utilizadas para Modelo Gearbox	49
4.16.	Arquitectura Modelo Grupo Hidráulico	50
4.17.	Combinaciones utilizadas para entrenamiento de Modelo Grupo Hidráulico	51
4.18.	Arquitectura Modelo Generador	52
4.19.	Combinaciones utilizadas para entrenamiento de Modelo Generador	53
4.20.	Alarmas identificadas para Turbina Nº2	54
4.21.	Anticipación de alarmas 1 y 2 de Turbina Nº2 por modelo Gearbox	55
4.22.	Anticipación de alarmas 4, 5 y 6 de Turbina Nº2 por modelo Generador	55
4.23.	Anticipación de alarma 8 de Turbina Nº2 por modelo Grupo Hidráulico	56
4.24.	Resumen de resultados de los modelos para Turbina Nº2	57
4.25.	Alarmas identificadas para Turbina Nº9	57
4.26.	Anticipación de rotura de pala por modelo Grupo Hidráulico en Turbina Nº9	59
4.27.	Resumen de resultados de los modelos para Turbina Nº9	59
4.28.	Alarmas identificadas para Turbina Nº27	60
4.29.	Anticipación de Alarmas 2 y 3 por modelo Gearbox en Turbina Nº27	61
4.30.	Anticipación de Alarma 5 por modelo Grupo Hidráulico en Turbina Nº27	61
4.31.	Anticipación de Alarmas 7 y 8 por modelo Generador en Turbina Nº27	62
4.32.	Resumen de resultados de los modelos para Turbina Nº27	63
4.33.	Alarmas identificadas para Turbina Nº29	64

4.34.	Anticipación de Alarma 1 por modelo Grupo Hidráulico en Turbina N°29	64
4.35.	Anticipación de Alarma 5 por modelo Generador en Turbina N°29	65
4.36.	Anticipación de Alarmas 15 y 16 por modelo Grupo Hidráulico en Turbina N°29	66
4.37.	Resumen de resultados de los modelos para Turbina N°29	67
A.1.	Alarmas identificadas para Turbina N°2	74
B.1.	Alarmas identificadas para Turbina N°9	81
C.1.	Alarmas identificadas para Turbina N°27	86
D.1.	Alarmas identificadas para Turbina N°29	92

Índice de Ilustraciones

2.1.	Siemens Gamesa SG 4.5-145 IIB	5
2.2.	Componentes principales del nacelle	5
2.3.	Componentes principales del rotor	7
2.4.	WindNet Pro ®	8
2.5.	Disposición reporte AWS	10
2.6.	Estructura de una red DAE [17]	17
2.7.	Estructura de una RBM	17
2.8.	R_e de una gearbox	20
4.1.	Componentes principales del grupo hidráulico. 1: Unidad hidráulica. 2: Sistema de acondicionamiento térmico. 3: Circuito hidráulico. 4: Sistema de control pitch	31
4.2.	Resultados Elliptic Envelope para Turbina N°2. Sistema Grupo Hidráulico	37
4.3.	Alarmas Grupo Hidráulico detectadas con modelo Elliptic Envelope	38
4.4.	Resultados Isolation Forest para Turbina N°2. Sistema Grupo Hidráulico, con outlier_fraction = 0.001	39
4.5.	Alarmas Grupo Hidráulico detectadas con modelo Isolation Forest, con outlier_fraction = 0.001	40
4.6.	Resultados Local Outlier Factor para Turbina N°2. Sistema Grupo Hidráulico, con outlier_fraction = 0.001	41
4.7.	Resultados Local Outlier Factor para Turbina N°2. Sistema Grupo Hidráulico, con outlier_fraction = 0.0001	42
4.8.	Resultados One-Class SVM para Turbina N°2. Sistema Grupo Hidráulico, con $\nu = 0.001$	43
4.9.	Ejemplo Gráfico R_e v/s Fecha + Umbral DSPOT	47
4.10.	Curvas de aprendizaje del modelo Gearbox	49
4.11.	Curvas de aprendizaje del modelo Grupo Hidráulico	51
4.12.	Curvas de aprendizaje del modelo Generador	53
4.13.	Detección de alarmas 1 y 2 por modelo Gearbox en Turbina N°2	54
4.14.	Detección de alarmas 4, 5 y 6 por modelo Generador en Turbina N°2	55
4.15.	Detección de alarma 8 por modelo Grupo Hidráulico en Turbina N°2	56
4.16.	Detección de rotura de pala por modelo Gearbox en Turbina N°9	58
4.17.	Detección de rotura de pala por modelo Grupo Hidráulico en Turbina N°9	58
4.18.	Detección de Alarmas 2 y 3 por modelo Gearbox en Turbina N°27	60
4.19.	Detección de Alarma 5 por modelo Grupo Hidráulico en Turbina N°27	61
4.20.	Detección de Alarmas 7 y 8 por modelo Generador en Turbina N°27	62
4.21.	Detección de Alarma 1 por modelo Grupo Hidráulico en Turbina N°29	64
4.22.	Detección de Alarma 5 por modelo Generador en Turbina N°29	65
4.23.	Detección de Alarmas 15 y 16 por modelo Grupo Hidráulico en Turbina N°29	66
A.1.	Resultado General Modelo Gearbox en Turbina N°2	75

A.2.	Resultados específicos de Alarmas Modelo Gearbox en Turbina N°2	76
A.3.	Resultado General Modelo Grupo Hidráulico en Turbina N°2	77
A.4.	Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°2	78
A.4.	Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°2	79
A.5.	Resultado General Modelo Generador en Turbina N°2	79
A.6.	Resultados específicos de Alarmas Modelo Generador en Turbina N°2	80
B.1.	Residuales Modelo Gearbox en Turbina N°9	81
B.2.	Resultado General Modelo Gearbox en Turbina N°9	82
B.3.	Resultados específicos de Alarmas Modelo Gearbox en Turbina N°9	82
B.4.	Residuales Modelo Grupo Hidráulico en Turbina N°9	83
B.5.	Resultado General Modelo Grupo Hidráulico en Turbina N°9	83
B.6.	Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°9	84
B.7.	Residuales Modelo Generador en Turbina N°9	84
B.8.	Resultado General Modelo Generador en Turbina N°9	85
B.9.	Resultados específicos de Alarmas Modelo Generador en Turbina N°9	85
C.1.	Residuales General Modelo Gearbox en Turbina N°27	86
C.2.	Resultado General Modelo Gearbox en Turbina N°27	87
C.3.	Resultados específicos de Alarmas Modelo Gearbox en Turbina N°27	87
C.4.	Resultado General Modelo Grupo Hidráulico en Turbina N°27	88
C.5.	Resultado General Modelo Grupo Hidráulico en Turbina N°27	88
C.6.	Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°27	89
C.7.	Residuales Modelo Generador en Turbina N°27	90
C.8.	Resultado General Modelo Generador en Turbina N°27	90
C.9.	Resultados específicos de Alarmas Modelo Generador en Turbina N°27	91
D.1.	Residuales Modelo Gearbox en Turbina N°29	92
D.2.	Resultado General Modelo Gearbox en Turbina N°29	93
D.3.	Residuales Modelo Grupo Hidráulico en Turbina N°29	93
D.4.	Resultado General Modelo Grupo Hidráulico en Turbina N°29	94
D.5.	Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°29	95
D.5.	Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°29	96
D.6.	Residuales Modelo Generador en Turbina N°29	96
D.7.	Resultado General Modelo Generador en Turbina N°29	97
D.8.	Resultados específicos de Alarmas Modelo Generador en Turbina N°29	98
D.8.	Resultados específicos de Alarmas Modelo Generador en Turbina N°29	99

Capítulo 1

Introducción

1.1. Motivación

Una de las principales preocupaciones para los equipos de Operación y Mantenimiento (O&M) de las empresas que requieran de esta área, es la reducción de los costos de mantenimiento, en particular para Engie Energía Chile (ENGIE), una de las empresas más importantes de generación de energía en el mundo y en Chile.

Dentro de su plan de transformación energética, ENGIE ha incorporado grandes parques de generación de energía a través de la energía eólica. Los elementos esenciales de estos parques, las turbinas eólicas, presentan elevados costos de reparación, y tasas de falla que no son menores. Se estima que entre el 30% y el 60% de los costos de O&M están asociados a mantenimientos no programados, es decir, que ocurrieron a causa de eventos repentinos o inesperados [1]. Como ejemplo de esto, se sabe que el reemplazo de grandes componentes de turbinas eólicas, como las palas, implica gastos en despacho desde el origen del repuesto (que casi siempre debe ser por vía marítima), gastos en gestión de camiones y grúas, y por supuesto el costo asociado a la pérdida de generación de energía. Todo lo anterior, eleva hasta un 400% el costo de un proyecto de reemplazo de componente con respecto al valor de este por sí solo [2]. Añadido a estos costos, debe considerarse que los operadores de los parques eólicos deben invertir un porcentaje importante de su tiempo en viajes a la turbina defectuosa, ya sea para identificar la falla, diagnosticarla y finalmente repararla. Por lo tanto, detectar con anticipación las fallas incipientes se vuelve esencial para la reducción de costos, ya que permitiría aplicar un mantenimiento preventivo que es menos costoso que un mantenimiento correctivo.

Resulta que, mediante análisis de variables de monitoreo, es posible llevar a cabo esta tarea. Algoritmos de *Machine Learning* pueden aprender a identificar patrones en estas variables que indican la falla de algún componente. El presente trabajo busca aplicar estos algoritmos a uno de los modelos de turbina eólica con los que cuenta ENGIE.

1.2. Objetivos

1.2.1. Objetivo general

El objetivo de este trabajo es diseñar un modelo capaz de detectar fallas en componentes de turbinas eólicas con anticipación.

1.2.2. Objetivos específicos

Para cumplir el objetivo general se deben cumplir los siguientes objetivos específicos:

- Identificar sub-sistemas de turbinas eólicas y sus modos de falla críticos, y determinar el conjunto de variables necesarias para detectarlas.
- Obtener un set de datos apropiado y etiquetado, que contenga las variables identificadas.
- Implementar y validar un algoritmo de *Machine Learning* para la detección de fallas críticas en turbinas eólicas.

1.3. Alcances

- ENGIE cuenta con 3 parques eólicos con los que es posible trabajar. Este trabajo se centra sólo en uno. Dentro de la misma línea, el trabajo se enfoca en un modelo en particular de turbina eólica.
- El estado de salud de una turbina eólica se estudia por medio de modelos construidos para distintos sub-sistemas de esta. El entregable no corresponde a un modelo único.
- Este es un modelo piloto para ENGIE; si la empresa decide que es conveniente adoptar el modelo dentro de su sistema de monitoreo, el trabajo puede no representar la versión final.

Capítulo 2

Antecedentes

2.1. Parque Eólico Calama

El Parque Eólico Calama (PE Calama), ubicado a 15 km de la ciudad de Calama, en la Región de Antofagasta, pertenece a la empresa Engie Energía Chile (ENGIE), dedicada a la generación de energía eléctrica. PE Calama corresponde a una planta de generación de energía en base a energía eólica, y cuenta con una capacidad instalada de 151.2 MW por medio de 36 turbinas eólicas (WTG, por sus siglas en inglés: Wind Turbine Generator). El parque comenzó su construcción el 22 de octubre del año 2019, y la fecha de energización al 100% del este fue el 29 de octubre de 2021.

2.2. Equipo estudiado

El proceso de generación eléctrica en un parque eólico es un proceso continuo en el que la detención debido a fallas se traduce en pérdidas monetarias significativas y problemas de confiabilidad con el ente regulador del mercado eléctrico chileno, el Coordinador Eléctrico Nacional. Este proceso inicia en los WTG con la conversión de la energía eólica en energía eléctrica, y finaliza subestación eléctrica donde se realiza la transformación de tensión y se envía la energía a las redes de transmisión. Dentro de este proceso existen diversos sub-procesos donde hay equipos críticos cuyo funcionamiento debe ser continuo y predecible. En esta sección se describe el equipo de estudio para el desarrollo del modelo de Aprendizaje de Máquinas.

2.2.1. Turbina eólica

Una turbina eólica se define como un elemento que transforma la energía cinética del viento en energía mecánica, la cual puede ser posteriormente transformada en energía eléctrica. A través de la historia, la energía eólica se ha utilizado para molienda o levantamiento de agua, pero a partir del siglo XX comenzó a utilizarse para la generación de potencia eléctrica. Actualmente los generadores eólicos pueden clasificarse en dos tipos: de eje horizontal (HAWT, por sus siglas en inglés: Horizontal Axis Wind Turbine) y de eje vertical (VAWT: Vertical Axis Wind Turbine).

Turbinas eólicas de eje horizontal:

Las HAWT son las más comunes y representan el mayor porcentaje de la capacidad instalada de plantas de energía eólica. Como su nombre indica, tienen su rotor paralelo al suelo. En general están compuestas de múltiples álabes (típicamente 3), con forma de perfiles alares. Estos perfiles generan fuerzas aerodinámicas

al cruzar el viento por ellos, provocando así torque y giro.

Entre las características de estas turbinas, se tiene que logran giros muy estables, pueden llegar a altas velocidades de punta de álabe, y en general están más estudiadas, por lo que hay mayor certeza al diseñar. Para su funcionamiento es esencial que estén direccionadas al viento; luego se hace necesario un mecanismo de orientación para direccionar la turbina frente a cambios en la dirección del viento. Además, el generador eléctrico debe ser puesto en altura para estar en línea con el giro sin tener reducciones mecánicas (grandes fuentes de pérdida). Esto último dificulta la mantención [3].

Turbinas eólicas de eje vertical:

Las VAWT son aquellas que tienen su rotor perpendicular al suelo y giran verticalmente. Al contrario de las de eje horizontal, son mucho menos comunes y probadas. Poseen varias desventajas que han desacelerado su desarrollo : tienen problemas para comenzar su rotación (self-starting) sin ayuda externa [4], y además las cargas dinámicas que se generan sobre los perfiles alares y el eje son significativas, por lo que la falla por fatiga ha sido un problema frecuente en instalaciones anteriores.

Sin embargo, estas turbinas poseen ciertas características que las de eje horizontal no pueden cumplir. Primero, son omnidireccionales; pueden girar sin importar desde qué dirección venga el viento, por lo que no son necesarios los mecanismos de orientación que poseen las HAWT [5]. Por esta razón son de gran utilidad en sitios donde la dirección del viento no es constante o donde hay alta turbulencia [4].

En PE Calama, las turbinas son HAWT en su totalidad, por lo que no se entra en mayor detalle sobre las VAWT.

2.2.2. Equipo de estudio

El modelo de turbina presente en PE Calama es el *Siemens Gamesa SG 4.5-145 IIB*, de 3 álabes y rotor de cara al viento, con potencia nominal **4.5 MW**.

Estas turbinas tienen un diámetro de rotor de 145 m (posición d en la Figura 2.1) y altura del eje de 90 m, 107.5 m y 127.5 m (posición h en la Figura 2.1), lo cual depende de la altura de la torre.

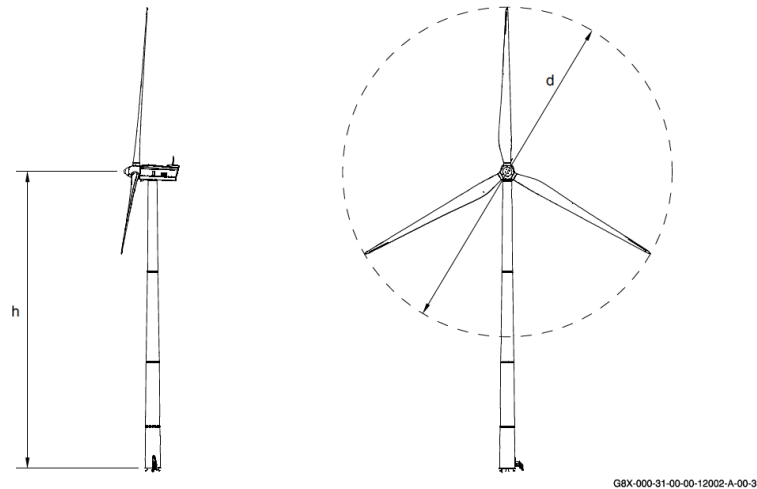


Figura 2.1: Siemens Gamesa SG 4.5-145 IIB

A continuación se presenta una descripción de los componentes principales de la turbina **SG 4.5-145 IIB**.

2.2.2.1. Nacelle

El nacelle o góndola es el espacio, típicamente detrás del rotor, que alberga la gran mayoría de los componentes mecánicos, hidráulicos y eléctricos del WTG, siendo estos los enumerados en la Figura 2.2.

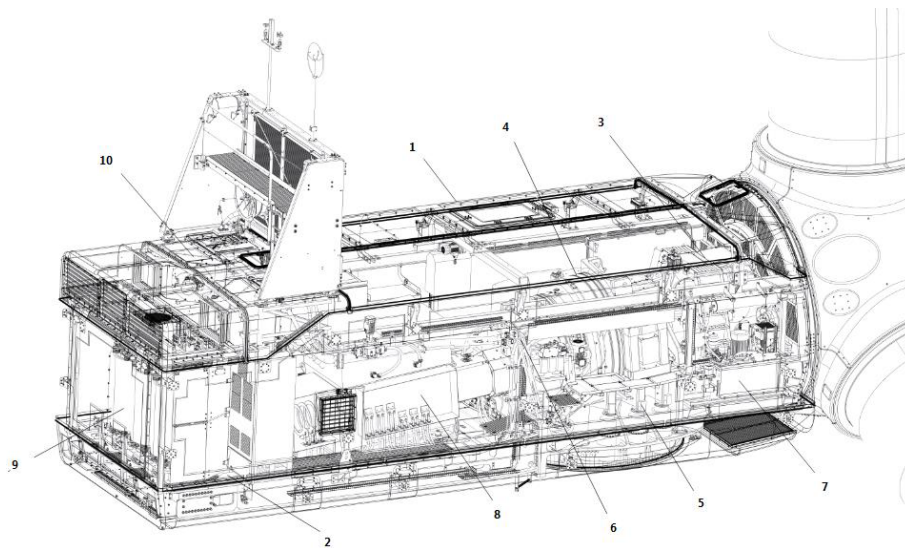


Figura 2.2: Componentes principales del nacelle

1. Cubierta

La cubierta protege los componentes de la turbina dentro del nacelle de la exposición a eventos meteorológicos y condiciones ambientales externas.

2. Armazón

Los armazones de estas turbinas han sido diseñados utilizando criterios de simplicidad mecánica y robustez para soportar adecuadamente los elementos del nacelle y transmitir cargas a la torre.

3. Eje principal

El torque motriz producido por el viento en el rotor es transmitido a la caja multiplicadora a través del eje principal.

4. Caja multiplicadora

Esta transmite la potencia del eje principal al generador. La caja multiplicadora consiste de 3 etapas combinadas, dos engranajes planetarios y un engranaje de eje paralelo.

5. Sistema yaw

El sistema yaw le permite al nacelle rotar en torno al eje de la torre. Este es un sistema activo y tiene 7 engranajes yaw operados eléctricamente por el sistema de control de la turbina de acuerdo a la información recibida de los anemómetros.

6. Freno mecánico

El freno mecánico consiste en un freno de disco hidráulicamente activado, el cual está montado en el eje de alta velocidad de la caja multiplicadora. Este freno mecánico solo es utilizado como freno de estacionamiento o si un botón de emergencia es presionado.

7. Sistema hidráulico

El sistema hidráulico provee aceite presurizado a los 3 actuadores de control de *pitch* independientes, al freno mecánico del eje de alta velocidad y al sistema de freno del sistema yaw.

8. Generador

El generador es una unidad asíncrona de doble alimentación con 6 polos (50 Hz/60 Hz), rotor de bobina y anillos colectores. Es altamente eficiente y es enfriado por un intercambiador agua-glicol/aire.

9. Transformador

El transformador es de 3 fases, encapsulado seco, con diferentes opciones de voltaje de salida entre 6.6 kV y 35 kV. Se encuentra ubicado en la parte trasera del nacelle, en un compartimiento separado por una pared de metal, la cual entrega aislación térmica y eléctrica del resto de los componentes.

10. Gabinetes eléctricos para control y potencia

El hardware del sistema eléctrico está distribuido en cuatro gabinetes, ubicados en el nacelle, en el transformador, en el buje y en las fundaciones.

2.2.2.2. Rotor

El rotor de la turbina consiste de tres álabes unidos al buje por rodamientos de álabe. El buje tiene un ángulo cónico de 4° en las bridas uniéndolo con los álabes, lo cual mantiene las puntas de los álabes lejos de la torre. Los componentes del rotor son los enumerados en la Figura 2.3.

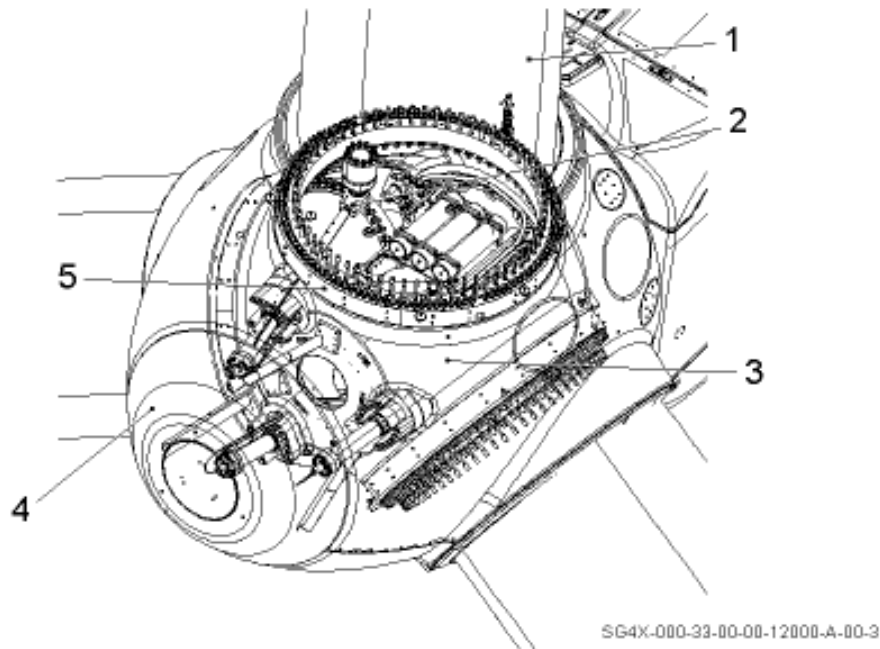


Figura 2.3: Componentes principales del rotor

1. Álabes

Los álabes tienen control de *pitch* a través de todo el largo de estos, así maximizando la producción de energía y reduciendo cargas y ruido.

2. Sistema de control *pitch*

Consiste de actuadores hidráulicos independientes para cada álabes, que proveen capacidad de rotación de entre -5° y 90° , y un sistema de acumuladores que aseguran estabilización en caso de emergencia.

3. Buje

Se encuentra unido a la pista de rodadura exterior de los tres rodamientos de álabes, y al eje principal con uniones atornilladas.

4. Cono

El cono protege el buje y los rodamientos de álabes de la atmósfera. El cono está atornillado al frente del buje y está diseñado para permitir acceso al buje para mantenimiento.

5. Rodamiento de álabes

Los rodamientos de álabes son la interfaz entre el álabes y el buje, y permiten el movimiento de control de *pitch*.

El álabes está unido a la pista de rodadura interior por medio de tornillos tensionados para facilitar inspección y retiro.

2.2.2.3. Torre

La torre de las turbinas es una estructura de acero tubular, en forma cónica truncada, dividida en secciones dependiendo de la altura de la torre.

Es provista con las correspondientes plataformas, escaleras e iluminación de emergencia.

2.2.3. Monitoreo SCADA

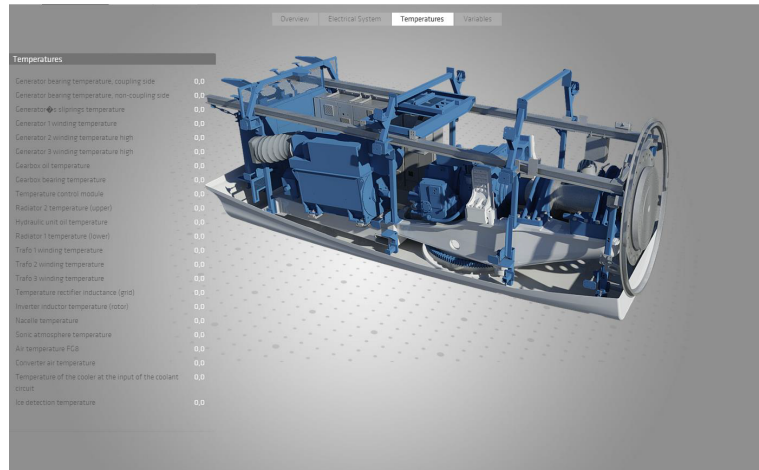
Las turbinas **SG 4.5-145 IIB** poseen un sistema propio de *supervisión, control y adquisición de datos* (SCADA, por sus siglas en inglés), denominado **Siemens Gamesa WindNet®**, el cual permite que la información del parque eólico sea de acceso fácil e intuitivo a través de un navegador.

Para este trabajo, las tareas más importantes que este sistema puede cumplir son:

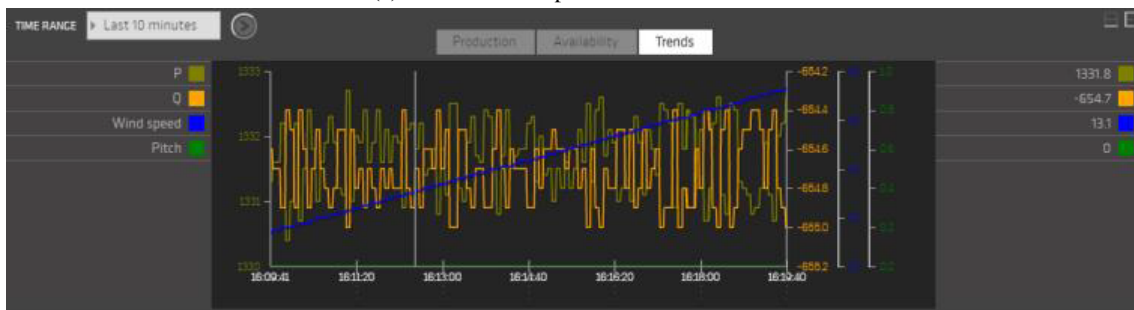
- Monitorear las alarmas para diferentes elementos del parque eólico en tiempo real y desplegar el registro de alarmas.
- Analizar la evolución de variables a través del tiempo.

Esto es posible gracias a sensores que monitorean continuamente diferentes parámetros de operación de las turbinas, tales como temperaturas, niveles de presión, vibraciones de álabes o posicionamiento.

WindNet registra las variables de operación de forma diezminutal y quinceminutal, las cuales se pueden visualizar en las pantallas con las que dispone el programa, mostradas en la Figura 2.4. El sistema SCADA entrega varias métricas de los datos en los periodos de tiempo mencionados, como el promedio, la desviación estándar, el máximo y el mínimo, entre otras.



(a) Pantalla de temperaturas monitoreadas



(b) Detalle de la pantalla

Figura 2.4: WindNet Pro ®

Las variables a las cuales se tiene acceso en este sistema SCADA son resumidas en la Tabla 2.1 agrupadas según los subsistemas a las cuales pertenecen.

Tabla 2.1: Variables SCADA

Sistema	Variable
Ambiente	Velocidad del viento (m/s)
Ambiente	Temperatura ambiente (°C)
Ambiente	Voltaje de red (V)
Sistema pitch	Ángulo pitch (°)
Sistema yaw	Posición nacelle (°)
Grupo hidráulico	Presión grupo hidráulico (bar)
Gearbox	Velocidad rotor (rpm)
Gearbox	Velocidad generador (rpm)
Gearbox	Temperatura rodamiento gearbox (°C)
Gearbox	Temperatura aceite gearbox (°C)
Generador	Potencia activa (kW)
Generador	Potencia reactiva (kW)
Generador	Potencia activa del estator (kW)
Generador	Temperatura rodamiento drive-end (°C)
Generador	Temperatura rodamiento non-drive-end (°C)
Generador	Temperatura devanados 1 (°C)
Generador	Temperatura devanados 2 (°C)
Generador	Temperatura devanados 3 (°C)
Generador	Temperatura anillos colectores (°C)
Transformador	Temperatura devanados 1 (°C)
Transformador	Temperatura devanados 2 (°C)
Transformador	Temperatura devanados 3 (°C)

Además de entregar las variables anteriores, el SCADA permite descargar un reporte de las alarmas activadas en las turbinas en periodos de tiempo que define el usuario. Este reporte se denomina *AWS*, cuya disposición puede verse en la Figura 2.5 y presenta las siguientes columnas:

Planta	Dispositivo	Categoría	Evento	Subevento / Categorización	Fecha Inicio	Fecha Fin	Duración	Descripción de la categorización
CALAMA	C11	Alarma	222 Muy baja presión grupo hidráulico		2/15/2022 7:48	2/15/2022 8:14	00:25:28	
		Estado	Aerogenerador en EMERGENCIA		2/15/2022 7:48	2/15/2022 8:14	00:25:33	
		Aviso	17545 Reset por comando OPC		2/15/2022 8:14	2/15/2022 8:14	00:00:39	
		Estado	Aerogenerador en STOP		2/15/2022 8:14	2/15/2022 8:15	00:01:45	
		Estado	Aerogenerador en LISTO		2/15/2022 8:15	2/15/2022 8:24	00:08:29	
		Estado	Aerogenerador en MARCHA		2/15/2022 8:24	2/15/2022 8:44	00:20:26	
		Alarma	203 Baja presión grupo hidráulico		2/15/2022 8:44	2/15/2022 8:49	00:04:59	
		Alarma	222 Muy baja presión grupo hidráulico		2/15/2022 8:44	2/15/2022 9:05	00:20:11	
		Estado	Aerogenerador en EMERGENCIA		2/15/2022 8:44	2/15/2022 9:05	00:20:15	
		Aviso	17545 Reset por comando OPC		2/15/2022 9:05	2/15/2022 9:05	00:00:40	
		Estado	Aerogenerador en STOP		2/15/2022 9:05	2/15/2022 9:06	00:01:44	
		Estado	Aerogenerador en PAUSA		2/15/2022 9:06	2/15/2022 9:07	00:00:20	

Figura 2.5: Disposición reporte AWS

- **Planta:** Calama.
- **Dispositivo:** Número de la turbina del 1 al 36 (ejemplo C01).
- **Categoría:** Puede ser Alarma, Aviso o Estado.
- **Evento:** En caso de ser Alarma o Aviso indica *Código + Nombre de Alarma/Aviso*. En caso de que la Categoría sea Estado puede indicar *Aerogenerador en :{LISTO, MARCHA, STOP, PAUSA, EMERGENCIA, NO COMUNICA}*. En LISTO indica que la turbina está lista para funcionar después de una detención, en MARCHA indica que la turbina está operando con normalidad. En STOP indica que la turbina fue detenida por comando del operador del SCADA. En PAUSA indica que la turbina se detuvo de forma automática por una alarma. En EMERGENCIA indica que la turbina alcanzó niveles críticos de operación y fue detenida. Por último, en NO COMUNICA indica que la comunicación entre la turbina y el SCADA fue cortada, comúnmente para realizar una intervención en el equipo.
- **Subevento/Categorización:** Corresponde a un grupo de eventos que no tienen relación la operación interna de las máquinas. Un ejemplo de esto es el Subevento *17516 Limitando potencia activa*, que corresponde a un estado inducido por el operador, por orden del Coordinador Eléctrico Nacional, que fuerza a limitar la generación de potencia.
- **Fecha inicio:** En formato MM/DD/YYYY HH:mm
- **Fecha fin:** En formato MM/DD/YYYY hh:mm
- **Duración:** En formato hh:mm:ss

Las alarmas de la columna Evento pueden ser clasificadas según el subsistema al cual pertenecen mediante códigos, los cuales que se detallan en la Tabla 2.2:

Tabla 2.2: Alarmas AWS

Sistema	Códigos
---------	---------

Sistema yaw	[100,200)
Grupo hidráulico	[200,300)
Ambiente	[300,400)
Gearbox	[400,500)
Generador	[500,600)
Control	[600,700)
Comunicaciones	[700,800)
Pitch	[800,900)
Estado de Operación	[900,1000)
Conexión a Red	[1800,1900)
Software	[2100,2200)
Convertidor	[6000,7000)
SCADA	[17000,...]

2.3. Mantenimiento

2.3.1. Industria 4.0

El concepto de Industria 4.0 hace referencia a la cuarta revolución industrial que se avecina con la introducción de tecnologías del internet en la industria. El término fue usado por primera vez en 2011 por líderes de la industria principalmente de Alemania. Desde ese entonces se ha tratado de clarificar cuáles son los límites, beneficios y tecnologías habilitadoras de la Industria 4.0. Algunos principios fundamentales de la cuarta revolución industrial ya han sido introducidos como el uso extensivo del internet, creando nuevas funciones en los productos que usan el internet como fuente de información, la flexibilidad de la producción, relacionado con la manufactura aditiva y trazabilidad de productos, y la virtualización de los procesos con los sistemas ciberfísicos. [6]

2.3.2. Mantenimiento 4.0

Uno de los campos de aplicación de la industria 4.0 es el mantenimiento. Las estrategias de mantenimiento predictivo, en particular las basadas en condición (CBM, por sus siglas en inglés: Condition Based Maintenance) son más fáciles de implementar utilizando las herramientas tecnológicas de la industria 4.0. Con la gran cantidad de información generada por la infraestructura del Internet Industrial de las Cosas (IIoT) [7], que incluye sensores, dispositivos, histórico de datos, entre otros, es posible monitorear la condición de los activos de manera continua, pero también es posible aplicar herramientas de Inteligencia Artificial, particularmente desarrollando algoritmos de *Machine Learning* que aprenden de estos datos generando nueva información asociada a la detección de anomalías, reconocimiento de fallas o modos de operación de un sistema, estimación de vida útil remanente de un activo, entre otras. Lo anteriormente descrito corresponde específicamente a la utilización de Modelos Basados en Datos (*Data-Driven Models*). Otros modelos

utilizados en el contexto de mantenimiento 4.0 son los *Physical Models* y los *Knowledge-Based Models* [8].

2.3.3. PHM

Dentro del concepto de mantenimiento 4.0 y los *Data-Driven Models* surgen los sistemas PHM: *Prognostic and Health Management*, cuyo objetivo es entregar un análisis exhaustivo de la condición de salud de un activo y sus componentes [9]. Los sistemas PHM ocupan herramientas de la Ciencia de Datos, la estadística, la física y la Inteligencia Artificial con el objetivo de: detectar síntomas de una eventual falla en el sistema (Detección de Anomalías), clasificar la falla de acuerdo con su tipo (Diagnóstico), y finalmente predecir por cuánto tiempo puede seguir operando la máquina en presencia de esta falla (Pronóstico) [10].

A continuación, se pasa a describir más en detalle los pasos principales de un proceso típico de PHM [11].

1. **Adquisición de datos:** Para la adquisición de datos en primer lugar es necesario definir los sistemas críticos que se quieren monitorear. Lo siguiente es identificar los componentes críticos de ese sistema en base a un análisis de fallas del sistema. Luego se debe seleccionar las variables físicas que sea pertinente monitorear en los componentes críticos ya identificados. Finalmente seleccionar los sensores y dispositivos adecuados para la medición de los parámetros físicos, ubicarlos en la posición más apropiada y configurarlos en una frecuencia óptima de muestreo. Es importante la implementación de un sistema de comunicación que permita manejar los datos de manera *offline* y también monitorear en tiempo real.
2. **Pre-Procesamiento:** Los datos adquiridos por lo general no están en un formato que sea compatible con los *inputs* de los algoritmos inteligentes que se quieran aplicar, por tanto, es necesario aplicar métodos para limpiar los datos, mitigar efectos inducidos por el ruido, o reestructurar la forma de los datos.
3. **Procesamiento:** Los datos aún pueden contener información redundante, lo que motiva el uso de métodos de extracción de parámetros para reducir la dimensionalidad de los datos y retener solamente la información más importante. De esta manera, los datos ya están listos para ser recibidos por algoritmos inteligentes que ayuden en la toma de decisiones y que aporten información relevante sobre el estado o condición de un sistema.
4. **Detección de Anomalías:** Los modelos de detección de anomalías establecen de manera automática si la información que está siendo recibida presenta alguna discrepancia en comparación con el funcionamiento normal de la máquina. Los algoritmos aprenden a partir de información recopilada de activos sanos, es decir, cuando estos operan bajo condiciones normales. Cabe mencionar que una anomalía no necesariamente es una falla, a veces puede representar un nuevo modo de operación de la máquina que no estaba presente en los datos históricos.
5. **Diagnóstico de Fallas:** Estos modelos, además de detectar una anomalía permiten identificar la causa de esta. Están basados en datos históricos que contienen eventos de distintas condiciones defectuosas, y permiten clasificar nuevas fallas en las clases previamente definidas.
6. **Pronóstico de Fallas:** Los modelos de pronóstico de fallas entregan una predicción de la vida útil remanente (RUL, por sus siglas en inglés: Remaining Useful Life) de un determinado componente. La estimación de la RUL comienza con la identificación del intervalo de tiempo en el cual se comienzan

a observar los efectos de una falla, para luego inferir por cuánto tiempo puede seguir operando la máquina y establecer el intervalo de tiempo en que ocurriría la falla.

2.4. Machine Learning

El Aprendizaje Automático, o *Machine Learning*, es una rama de la Inteligencia Artificial. La Inteligencia Artificial a su vez, es una rama de la ciencia computacional que corresponde al estudio y desarrollo de máquinas inteligentes y *softwares* que pueden razonar, aprender, recopilar conocimiento, comunicar, manipular y percibir objetos [12]. El *Machine Learning* corresponde al desarrollo de algoritmos y modelos estadísticos para realizar tareas específicas sin el uso explícito de instrucciones para su programación. Estos algoritmos son capaces de aprender de la experiencia; pueden reconocer patrones y generalizar comportamientos a partir de un conjunto de datos, y en base a esto tomar decisiones o hacer predicciones con precisión. Según el método de aprendizaje, los modelos y algoritmos de *Machine Learning* son clasificados en 4 grupos [13]:

- **Supervisado**

En este tipo de aprendizaje, los datos de entrenamiento están etiquetados, es decir, el *output* asociado a cada *input* es conocido de antemano. A través de la optimización iterativa de una función objetivo el algoritmo aprende; encuentra una función que dados los *input*, asigna una etiqueta o *output* adecuado. Dentro de esta categoría se encuentran los métodos de clasificación, donde la salida corresponde a clases discretas, y los métodos de regresión, donde la salida es un valor continuo.

- **No-Supervisado**

En este caso los datos de entrenamiento no están etiquetados o no presentan una estructura conocida. Este método permite explorar la estructura de estos datos para extraer información importante sin la ayuda de variables de resultado conocidas. Parte de este tipo de aprendizaje son los métodos de agrupamiento y de reducción de dimensionalidad.

- **Semi-Supervisado**

Combina aspectos de los métodos anteriores. Los métodos de detección de novedades son parte de este grupo, donde los datos de entrenamiento están etiquetados mientras que los de prueba no lo están.

- **Refuerzo**

Este es un método basado en ensayo y error, donde el sistema entrega una señal de recompensa cuando el modelo tiene buen rendimiento, o una señal de castigo en caso contrario. El objetivo es desarrollar un sistema que mejore el rendimiento del algoritmo basado en interacciones con el entorno.

Los algoritmos y modelos de Aprendizaje Automático no solamente se clasifican según el tipo de aprendizaje; también pueden ordenarse de acuerdo con su aplicación, como se describe a continuación [13].

- **Clasificación**

Corresponden a modelos de aprendizaje supervisado. Su objetivo es identificar la clase de una nueva observación a partir de un conjunto de datos cuya categoría es conocida (en este caso las etiquetas de clase son discretas). Algunos ejemplos reconocidos de algoritmos de clasificación son: K-Nearest Neighbors, Decision Forest, Random Forest y Support Vector Machine.

- **Regresión**

Estos modelos permiten predecir valores continuos en base a una o más variables predictoras o explicativas, mediante un ajuste a los datos de entrada. El método de regresión lineal es el más básico; en

este el conjunto de datos se ajusta mediante una combinación lineal de estos, buscando minimizar el error cuadrático medio.

- **Agrupamiento**

Esta técnica exploratoria no supervisada permite organizar los datos en subgrupos o *clusters* sin conocer las etiquetas o clases. Los datos son agrupados en base a criterios de distancia o similitud, y de esta manera, los métodos de agrupamiento permiten estructurar la información y derivar relaciones significativas de los datos. Algoritmos de agrupamiento conocidos son : K-Means, Mean Shift Clustering, DBSCAN, Gaussian Mixture Models y Agglomerative Hierarchical Clustering.

- **Reducción de Dimensionalidad**

En reiteradas ocasiones, los datos poseen una alta dimensionalidad, lo cual resulta ser condicionante en términos de almacenamiento y capacidad computacional. Este tipo de algoritmos se utiliza por lo general en etapas de procesamiento de datos y permite reducir la cantidad de los datos conservando gran parte de la información y eliminando redundancia en los datos. Lo anterior permite disminuir los tiempos de entrenamiento y mejorar el desempeño de los algoritmos. También puede ser útil para la visualización de los datos. Entre los métodos de reducción más usados se tienen: Principal Component Analysis, Kernel PCA, Factor Analysis, Linear Discriminant Analysis, Multidimensional Scaling, ISOMAP y Local Linear Embedding.

- **Detección de Anomalías/Novedades**

La detección de anomalías consiste en identificar elementos que parecen desviarse notoriamente de los otros miembros de un conjunto de datos [14]. Dependiendo de las características de los datos de entrenamiento estos métodos pueden clasificarse en dos categorías:

- Detección de anomalías, donde los datos de entrenamiento contienen datos anómalos y se debe estimar la proporción de estos con respecto a los datos normales
- Detección de novedades, donde los datos de entrenamiento son exclusivamente datos etiquetados como "sanos" mientras los datos de prueba no están etiquetados.

Algunos algoritmos de detección conocidos son: Robust Covariance, One Class Support Vector Machine, Local Outlier Factor y Isolation Forest.

2.4.1. Modelos utilizados

Debido a la naturaleza temporal de las variables de operación de las turbinas eólicas a estudiar, y a la necesidad de detectar anomalías con respecto al funcionamiento normal de los activos durante su ciclo de vida, se estudian únicamente los algoritmos de detección de anomalías/novedades.

2.4.1.1. Algoritmos de Detección de Anomalías/Novedades

1. Elliptic Envelope

Una forma de llevar a cabo la detección de anomalías/novedades es asumir que los datos de entrada vienen de una distribución de probabilidad dada (por ejemplo, distribución Gaussiana). En base al supuesto anterior, trata de definir la forma de los datos, y así puede encontrar observaciones que se alejan

del límite de estos.

El algoritmo Elliptic Envelope primero ajusta una estimación de covarianza robusta a los datos, y así acomoda una elipse a los puntos de datos centrales, ignorando puntos del modo central.

Por ejemplo, al asumir que los datos de entrada siguen una distribución Gaussiana, va a estimar la ubicación y la covarianza de estos de una forma robusta, es decir, sin la influencia de datos anómalos. El grado de anomalía de los datos se mide mediante la distancia de Mahalanobis obtenida de esta estimación [15].

$$M_d(x, \mu, \Sigma) = \sqrt{(x - \mu)\Sigma^{-1}(x - \mu)^T} \quad (2.1)$$

donde μ es la media de los datos y Σ es la matriz de covarianzas.

2. Isolation Forest

Otra forma de efectuar detección de anomalías/novedades en sets de datos de alta dimensionalidad es utilizando *bosques aleatorios*. El algoritmo Isolation Forest aísla observaciones al seleccionar aleatoriamente un parámetro y luego seleccionando un valor de división entre el máximo y mínimo del parámetro en cuestión.

Ya que la partición recursiva puede ser representada como una estructura de árbol, el número de divisiones requeridas para aislar una muestra es equivalente al largo del camino entre el nodo raíz y el nodo terminal. Este largo, promediado sobre un bosque de árboles aleatorios, es una medida de normalidad y la función de decisión.

La partición aleatoria genera caminos notablemente más cortos para anomalías. Por lo tanto, cuando un bosque de árboles aleatorios produce de forma colectiva caminos cortos para una muestra en particular, lo más probable es que sea una anomalía [15].

3. Local Outlier Factor

El algoritmo Local Outlier Factor (LOF) calcula un puntaje que refleja el grado de anormalidad de las observaciones. Mide la desviación de densidad local de un dato dado con respecto a sus vecinos. La idea es detectar las muestras que tienen una densidad sustancialmente menor que la de sus vecinos

En la práctica, la densidad local es obtenida de los k -vecinos más cercanos. El puntaje LOF de una observación es igual a la razón entre la densidad local promedio de los k -vecinos más cercanos y su propia densidad; se espera que un ejemplo normal tenga una densidad local similar a la de sus vecinos, mientras que los datos anormales tendrían una densidad local mucho menor.

La fortaleza de este algoritmo radica en que toma en consideración tanto propiedades locales como globales de un set de datos; puede rendir bien incluso en sets de datos donde las anomalías tienen densidades subyacentes distintas. La pregunta es no es qué tan aislada está la muestra, sino qué tan aislada está con respecto a sus vecinos [15].

4. One-Class Support Vector Machine

El algoritmo One-Class Support Vector Machine (OC SVM) define una frontera que delimita el contorno de la distribución de observaciones iniciales, trazada en un espacio p -dimensional. Entonces, si las siguientes observaciones caen dentro del subespacio delimitado por la frontera, se considera como que vienen de la misma población que las observaciones iniciales. De lo contrario, si caen fuera de la frontera, se puede decir que son anormales.[15]

OC SVM requiere de la elección de un *kernel* y de un parámetro escalar ν para definir una frontera. Comúnmente se escoge el *kernel* RBF, que construye una hiper-esfera de radio mínimo r que se ajusta a las muestras de una única clase, con la idea de minimizar r , considerando como anómalas las muestras que quedan fuera de la hiper-esfera [16]. El parámetro ν , también conocido como el margen de OC SVM, corresponde a la probabilidad de encontrar una nueva observación, pero normal, fuera de la frontera [15].

2.4.1.2. Deep Autoencoder

Un Deep Autoencoder (DAE) es un tipo de red de aprendizaje profundo, las cuales consisten en una red neuronal profunda con múltiples capas ocultas entre las capas de entrada y de salida, que puede modelar relaciones no-lineales complejas entre varios tipos de variables [17].

El DAE está compuesto de varias pilas de *Restricted Boltzmann Machines* (RBM). En el DAE, la salida de cada RBM es considerada como la nueva entrada de una RBM de mayor nivel para alcanzar la transmisión de los resultados del aprendizaje capa por capa. Este procedimiento es repetido varias veces para inicializar los parámetros de cada capa oculta

El mecanismo de una red DAE incluye dos procesos, codificado (*encoding*) y decodificado (*decoding*). En el proceso de codificado, la entrada X primero es transformada para producir un conjunto de características para posteriores transformaciones por capas, y las características más complejas son obtenidas en las capas más altas. Finalmente, el código Y es adquirido a través del proceso de codificado. Similarmente, el código Y es iterativamente transformado de vuelta a la entrada original por medio de RBMs y la reconstrucción de X , \hat{X} , es generada en el proceso de decodificado. La Figura 2.6 muestra los mecanismos de codificado y decodificado.

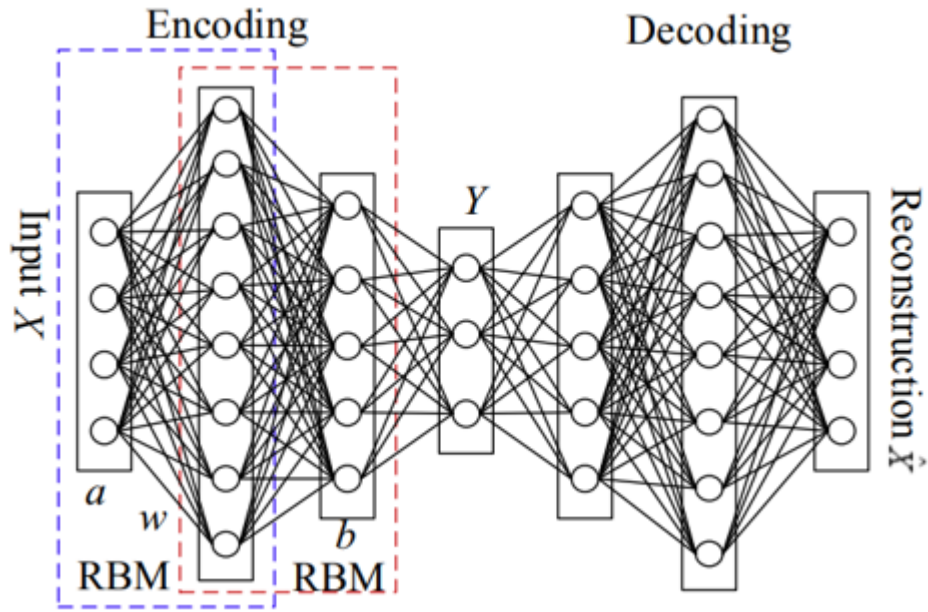


Figura 2.6: Estructura de una red DAE [17]

Una RBM es una red de dos capas de un tipo Markov aleatorio con N unidades visibles $v_i = \{0, 1\}^N$ y M unidades ocultas $h_j = \{0, 1\}^M$, como se muestra en la Figura 2.7. Se introduce el modelo de energía para describir la energía de las unidades de configuración conjunta en la RBM, mediante:

$$E(v, h; \theta) = - \sum_{i=1}^N a_i v_i - \sum_{j=1}^M b_j h_j - \sum_{i=1}^N \sum_{j=1}^M w_{ij} v_i h_j \quad (2.2)$$

donde $\theta = \{w_{ij}, a_j, b_i\}$, w_{ij} es el peso entre la unidad visible i y la unidad oculta j , y b_i y a_j son los sesgos (*bias*) de las unidades visibles y ocultas, respectivamente. Estos son los parámetros del modelo.

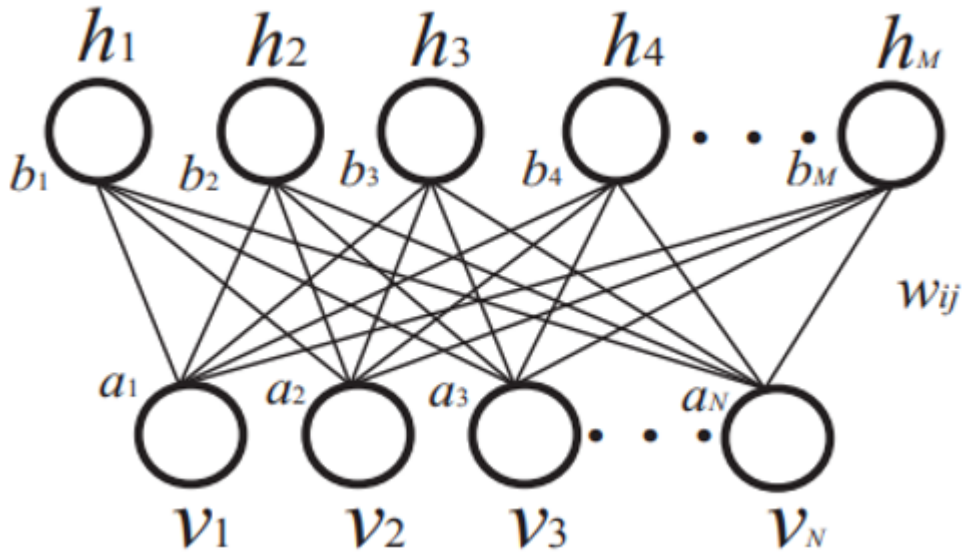


Figura 2.7: Estructura de una RBM

En la RBM, la distribución conjunta entre unidades basada en el modelo de energía está descrita como:

$$P(v, h; \theta) = \frac{1}{Z(\theta)} \exp\{-E(v, h; \theta)\} \quad (2.3)$$

$$Z(\theta) = \sum_v \sum_h E(v, h; \theta) \quad (2.4)$$

donde $Z(\theta)$ es la constante de normalización. La red entrega los valores de probabilidad de cada vector de entrada por medio de la función de energía, y la probabilidad puede ser elevada cambiando el parámetro θ para ajustar el valor de la energía en la ecuación 2.2.

Las distribuciones condicionales de las unidades ocultas h y del vector de entrada v en la RBM están dadas por:

$$P(h_j = 1|v) = f\left(\sum_{i=1}^N w_{ij}v_i + b_j\right) \quad (2.5)$$

$$P(v_i = 1|h) = f\left(\sum_{j=1}^M w_{ij}h_j + a_i\right) \quad (2.6)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.7)$$

donde $f(x)$ es la función de activación, seleccionada como *sigmoid* en el trabajo de Zhao et al. [17]. La esencia de la función de activación es retener las características de la neurona activada y mapearla. Se opta por utilizar *sigmoid* debido a que su valor cambia sensiblemente cuando las entradas están entre $\{-1, 1\}$. Una vez que el valor está cerca o fuera de los límites el valor de *sigmoid* pierde su sensibilidad, y la precisión de la red será reducida cuando el valor de *sigmoid* esté en condición de saturación [18].

En cada red RBM, la información de activación de las unidades de capas ocultas es representada como las características extraídas de los datos de entrada. Dicho de otra manera, el objetivo de aprendizaje de las RBM es obtener el parámetro θ para recuperar los datos de entrada originales a la perfección. Por lo tanto, la función de verosimilitud de la capa visible v para obtener el parámetro θ es construida como:

$$L(\theta; v) = \prod_v P_\theta(v) = \prod_v \frac{\sum_h e^{-E_\theta(v, h)}}{\sum_{v, h} e^{-E_\theta(v, h)}} \quad (2.8)$$

Luego, en la ecuación 2.2, se toma el logaritmo por los dos lados, y la derivada de la función logaritmo con respecto a $\theta = \{w, b, a\}$ se presenta como:

$$\frac{\partial \ln L(\theta, v)}{\partial \theta} = \frac{\partial \sum \ln P_\theta(v)}{\partial \theta} = \sum \left\{ E_{P_\theta(h, v)} \left[-\frac{\partial E_\theta(v, h)}{\partial \theta} \right] - E_{P_\theta(v, h)} \left[-\frac{\partial E_\theta(v, h)}{\partial \theta} \right] \right\} \quad (2.9)$$

Luego, se aplica el algoritmo de divergencia contrastiva (CD) para estimar el gradiente. Se utiliza el aprendizaje CD de un paso para actualizar los parámetros θ [19]. Para reducir la pérdida de información de la entrada X y obtener parámetros certeros para la red DAE, la función de pérdida (*loss function*) puede ser definida como:

$$J_{AE}(\theta) = \frac{1}{N} \sum_{x \in \mathcal{X}} R_e(X, f(\hat{X})) \quad (2.10)$$

El parámetro θ se obtiene mediante el método del gradiente descendente para minimizar la función de pérdida. Así, la regla de actualización se describe como:

$$w^{k+1} = w^k + \varepsilon \frac{\partial J_{AE}(\theta)}{\partial w} \quad (2.11)$$

$$a^{k+1} = a^k + \varepsilon \frac{\partial J_{AE}(\theta)}{\partial a} \quad (2.12)$$

$$b^{k+1} = b^k + \varepsilon \frac{\partial J_{AE}(\theta)}{\partial b} \quad (2.13)$$

donde ε es la tasa de aprendizaje. Entonces, un parámetro θ razonable puede finalmente obtenerse por el proceso jerárquico de las RBM.

Después de entrenar cada RBM, la información aprendida reside en la capa oculta, la cual puede ser utilizada como la entrada de una capa mayor para generar información esencial y obtener su parámetro θ hasta completar el entrenamiento de toda la red DAE.

Ajuste fino de DAE

Luego del preentrenamiento de las capas ocultas del DAE con las RBMs, se actualizan los pesos y los sesgos de cada capa oculta de las RBMs de multi-capas, y la estructura de la red DAE es construída. El preentrenamiento de la red DAE es un aprendizaje no supervisado de la data de los datos. Así, los resultados del aprendizaje pueden utilizarse como valores predeterminados para el aprendizaje supervisado.

El aprendizaje supervisado se lleva a cabo con el algoritmo de retropropagación o *backpropagation* (BP), utilizado para mejorar la representación de las características de los datos y optimizar los parámetros de las capas ocultas en el ajuste fino. Debido a que el entrenamiento con ajuste fino sólo necesita una búsqueda local basada en los parámetros obtenidos en el preentrenamiento, el tiempo de convergencia de la optimización se reduce significativamente. Tras el ajuste fino, se obtienen los parámetros de la red DAE globalmente optimizados.

Para las redes DAE de componentes de turbinas eólicas, el mapeado de entrada a salida es uno-a-uno, y cada ítem posee el mismo significado físico. \hat{X} mostrado en la Figura 2.6 es la reconstrucción de la entrada X , que en este trabajo corresponde a las variables de SCADA. Por lo tanto, el estado de salud de componentes de turbinas eólicas puede ser evaluado analizando la relación entre X y \hat{X} .

Evaluación de la condición y detección de anomalías

Cuando el conjunto de variables SCADA de las turbinas eólicas presenta un comportamiento normal, estas deberían satisfacer su pertinencia en el sentido físico. Una vez ocurrida la anormalidad, la pertinencia interna entre estas variables es destruída. Por lo tanto, los datos X en una condición anormal se desviarán de la reconstrucción \hat{X} estimada por la red DAE. El residual (r) y el error de reconstrucción (R_e) se definen por las siguientes ecuaciones:

$$r = \hat{X} - X \quad (2.14)$$

$$R_e = \|\hat{X} - X\|_2 \quad (2.15)$$

El residual puede reflejar el cambio de cada variable de operación, mientras que el error de reconstrucción describe el comportamiento general del sistema, por lo que se utiliza como índice de evaluación general de su condición.

Para monitorizar la tendencia de R_e y detectar su variación de anomalías, se establece un umbral adaptativo como criterio de decisión de alarmas. Cuando R_e excede el el umbral en repetidas ocasiones, significa que una falla incipiente podría ocurrir.

Por ejemplo, después de definir la red DAE para una turbina eólica, el R_e se calcula constantemente como muestra la Figura 2.8 . Es posible observar que R_e no es estacionario incluso cuando la turbina opera con normalidad. Debido a las fluctuaciones de la velocidad del viento, especialmente las ráfagas, se producen varios R_e extremos, por lo que si se define un umbral constante se podrían detectar falsas alarmas. Es por lo anterior, que al modelo de red DAE se le añade un umbral adaptativo basado en teoría del valor extremo.

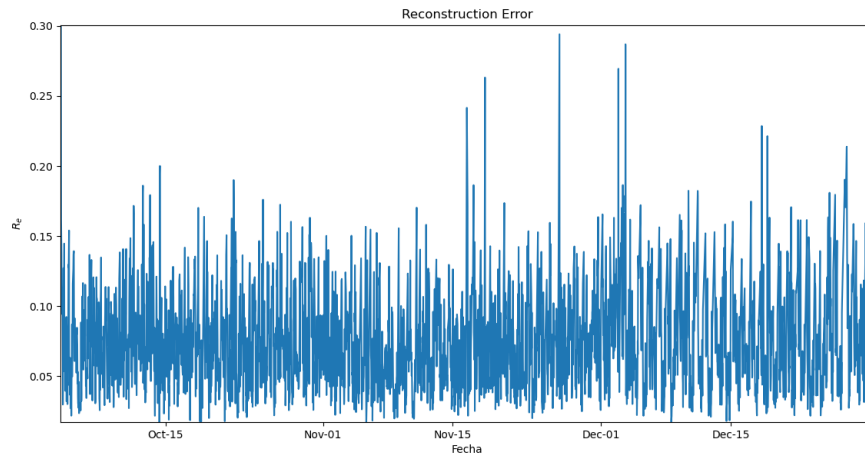


Figura 2.8: R_e de una gearbox

Umbral adaptativo basado en Teoría del Valor Extremo

Considerando la no estacionariedad y los múltiples puntos de valor extremo de condición evaluando el índice R_e , se aplica la *teoría del valor extremo* (EVT, por sus siglas en inglés) para analizar la distribución de los valores extremos en R_e y para diseñar un umbral para detección de anomalías y evaluación de condición.

El objetivo de la EVT es encontrar la ley de eventos extremos (por ejemplo, la ley de máxima temperatura diaria, o la ley de máxima altura de marea mensual) [20]. Un resultado de Fisher, Tippett [21] y luego de Gnedenko [22] establece que, bajo una condición débil, estos eventos extremos tienen el mismo tipo de distribución, independiente de la original. Estas leyes extremas son denominadas Distribuciones de Valor

Extremo (EVD, por sus siglas en inglés) y tienen la siguiente forma:

$$G_\gamma : x \mapsto \exp\left\{-\left(1 + \gamma x\right)^{-\frac{1}{\gamma}}\right\}, \quad \gamma \in \mathbb{R}, \quad 1 + \gamma x > 0 \quad (2.16)$$

Todos los extremos de distribuciones estándar comunes siguen tal distribución y el *índice de valor extremo* γ depende de esta ley original. Por ejemplo, si X_1, \dots, X_n son n variables independientes idénticamente distribuidas, entonces $M_n = \max_{1 \leq i \leq n} X_i$ probablemente siga una EVD cuyo índice de valor extremo γ está dado por la distribución inicial. Este resultado se puede asimilar entendiendo que, para la mayoría de las distribuciones las probabilidades decrecen cuando los eventos son extremos, es decir $\mathbb{P}(X > x) \rightarrow 0$ cuando x aumenta. La función $\bar{F}(x) = \mathbb{P}(X > x)$ representa la *cola* de la distribución de X . En realidad, no hay muchas formas posibles para esta cola y G_γ intenta ajustarlas.

Al ajustar una EVD a la cola de la distribución de entrada desconocida, es posible evaluar la probabilidad de potenciales eventos extremos. En particular, de una probabilidad dada q es posible calcular z_q tal que $\mathbb{P}(X > z_q) < q$. Para resolver este problema, el camino natural es estimar γ . Una método para realizar esta estimación sin tener un supuesto fuerte de la distribución de X ni conocimiento previo sobre esta, es el enfoque *Peaks-Over-Threshold* (POT).

El enfoque POT se basa en el teorema de Pickands-Balkema-de Haan [23, 24], que se puede resumir en la siguiente expresión:

$$\bar{F}_t(x) = \mathbb{P}(X - t > x | X > t) \underset{t \rightarrow \tau}{\sim} \left(1 + \frac{\gamma x}{\sigma(t)}\right)^{-\frac{1}{\gamma}} \quad (2.17)$$

El resultado de este teorema muestra que el exceso sobre un umbral t , escrito como $X - t$, seguramente siga una Distribución Pareto Generalizada (GPD) con parámetros γ, σ . De hecho la GPD necesita un tercer parámetro, la ubicación μ , pero en este caso se considera nula [20]. En vez de ajustar una EVD a los valores extremos de X , el enfoque POT intenta ajustar una GPD a los excedentes $X - t$.

En el caso de tener estimados $\hat{\gamma}$ y $\hat{\sigma}$, el cuantil puede ser calculado a través de:

$$z_q \cong t + \frac{\hat{\sigma}}{\hat{\gamma}} \left(\left(\frac{qn}{N_t} \right)^{-\hat{\gamma}} - 1 \right) \quad (2.18)$$

donde t es un umbral elevado, q es la probabilidad deseada, n es el número total de observaciones, N_t es el número de *peaks*, es decir, el número de X_i tal que $X_i > t$.

Algunos métodos que pueden ser utilizados para realizar la estimación de γ y σ son el Método de los Momentos (MOM) o los Momentos de Probabilidad Ponderada (PWM), pero estos son menos eficientes y robustos que la estimación de Máxima Verosimilitud [25]. En el trabajo de Siffer et al. [20], la optimización de los parámetros de máxima verosimilitud se realiza mediante el procedimiento de Grimshaw [26], el cual reduce el problema de optimización de dos variables a una ecuación de una sola variable.

Siffer et al. propone dos algoritmos POT, el caso estacionario (SPOT) y el caso con desviaciones (DSPOT). En el presente trabajo interesa el segundo método, debido a la naturaleza cambiante de las condiciones del viento. SPOT que asume que la distribución de las variable X_i no cambia a lo largo del tiempo, mientras que DSPOT aplica este supuesto a ventanas de tiempo de tamaño d . El algoritmo utiliza el cambio de va-

riable $X'_i = X_i - M_i$, donde M_i modela el comportamiento local en el tiempo i . La implementación utiliza un promedio móvil $M_i = (1/d) \cdot \sum_{k=1}^d X_{i-k}^*$, con $X_{i-1}^*, \dots, X_{i-d}^*$ las últimas d observaciones normales.

Capítulo 3

Metodología

La metodología de trabajo propuesta para cumplir los objetivos planteados es la siguiente:

1. Se gestiona el acceso al sistema SCADA para poder descargar el registro AWS de las 36 turbinas de PE Calama, desde octubre de 2021 hasta diciembre de 2022, así como los datos de las variables de monitoreo para el mismo periodo de tiempo.
2. A partir del registro AWS, se realiza un estudio de las alarmas que más tiempo permanecen activas. Se deben filtrar las alarmas que no tengan relación con comportamientos anómalos de la turbina, como comandos ejecutados por los operadores; cualquier alarma cuyo origen venga del exterior de la turbina. Esto permite identificar los subsistemas de las turbinas que presentan mayor cantidad de anomalías y que, por lo tanto, serán el foco de atención de este trabajo.
3. Para profundizar aún más el estudio, se etiquetan las alarmas a las cuales le sigue el estado de *Aerogenerador en NO COMUNICA*, que permite inferir a qué subsistemas se les realizan intervenciones y así poder focalizar aún más el trabajo. La falta de un registro de mantenimiento adecuado hace de esta tarea una necesidad.
4. De forma paralela al punto anterior, se investiga con personal de la compañía acerca de fallas críticas que no se puedan identificar por medio del registro AWS. El fin de esto es hallar turbinas que hayan presentado una situación fuera de lo común que sea de interés como caso de estudio.
5. En base a la información recolectada en los pasos anteriores, se determinan los subsistemas críticos de las turbinas para los cuales se construirán los modelos de aprendizaje de máquinas.
6. Luego, se realiza una revisión bibliográfica de temas pertinentes al trabajo, librerías de Python y papers de investigación que se enfoquen en el monitoreo de la condición de los subsistemas seleccionados, y que idealmente utilicen algoritmos de aprendizaje de máquinas para detección de anomalías.
7. Se seleccionan modelos de aprendizaje de máquinas (algoritmos típicos de detección de novedades) y uno de aprendizaje profundo que haya demostrado tener buenos resultados en turbinas eólicas, que cuente además con respaldo para facilitar el desarrollo de este trabajo.
8. En primera instancia cada turbina se trabaja por separado. Para cada subsistema se crea un conjunto de datos que contenga variables de interés. Estas se seleccionan inspiradas en la bibliografía y la experiencia de trabajadores de la compañía.

9. Se realiza un preprocesamiento de los datos en el cual, en primer lugar, se eliminan todos los índices que contengan datos vacíos. Luego, se eliminan los datos que estén asociados a detenciones de las turbinas, debido a que no aportan en nada aprendizaje de los modelos; esto se puede hacer con facilidad eliminando los índices donde la *velocidad del rotor* es cero. Además, se normalizan los datos debido a las diferencias de magnitud entre algunas de las variable y para intentar mejorar el rendimiento de los modelos.
10. Se crea un conjunto de entrenamiento que sea representativo del estado de operación normal de la turbina. El primer mes de funcionamiento de PE Calama puede ser un buen indicador del estado de salud de los equipos, por lo que todos los modelos utilizan un conjunto de entrenamiento con los datos de operación de este mes (ya que los datos son 10-minutales, en un mes se tienen 4320 datos por variable).
11. El conjunto de prueba contiene los datos de todos los meses posteriores al mes de entrenamiento.
12. En primer lugar se entrenan modelos basados en algoritmos de detección de novedades. De forma iterativa se ajustan los parámetros y se comparan resultados. La única métrica para evaluar el desempeño de estos modelos es la comparación visual, debido a que los datos no cuentan con etiquetado. Los modelos que detecten únicamente novedades en el momento donde se presentan las alarmas son seleccionados como posibles candidatos.
13. Luego, se trabaja con una red neuronal seleccionada durante la revisión bibliográfica. Se ajusta el número de neuronas, la cantidad de capas, las funciones de activación, entre otros parámetros, para buscar el mejor rendimiento según los datos disponibles. En paralelo se evalúa el rendimiento en el conjunto de validación, siendo este el 5% del conjunto de entrenamiento, procurando que el modelo no se sobreajuste y sea capaz de generalizar su aprendizaje.
14. La red neuronal primero se pre-entrena de forma que se guarden los pesos entre las distintas capas. Luego de esto se realiza un *ajuste fino* donde se ajustan parámetros para llegar a la versión final del modelo.
15. En estas iteraciones se busca cumplir el mismo objetivo que con los algoritmos de detección de novedades, es decir, detectar como novedosos los datos que coinciden con el momento donde se activan las alarmas, e idealmente que esto ocurra con anticipación para permitir la acción temprana de los operadores de las turbinas. La evaluación del modelo, al igual que en el caso anterior, se realiza de manera visual mediante gráficos, y luego se cuantifica por medio de una estimación del tiempo que con el cual el modelo logra anticiparse a estas alarmas.
16. Finalmente se analizan los resultados, seleccionando el modelo que mejor se ajuste al comportamiento de las turbinas y que mejor resuelve el problema planteado. En base a esto se desarrollan las conclusiones de trabajo.

3.1. Recursos

Los recursos a utilizar en el presente trabajo son los siguientes:

- *Spyder*: Entorno de desarrollo integrado (IDE) *open-source* para programación científica en lenguaje Python, en el cual es sencillo importar librerías de relevancia como *NumPy*, *SciPy*, *Matplotlib*, *pandas*, entre otras.

- Documentación técnica de ENGIE.
- Registros AWS del SCADA de Parque Eólico Calama
- Conjuntos de variables diez-minutales del SCADA de Parque Eólico Calama.
- *Papers* y tesis relacionadas.

Capítulo 4

Desarrollo y discusión de resultados

4.1. Análisis de alarmas

Como se menciona en la Metodología, el primer paso de este trabajo es encontrar los subsistemas críticos en base a la frecuencia de fallas que presentan las turbinas. Debido a la ausencia de un registro de fallas y mantenimiento apropiado, la única opción que se tiene para lograr este objetivo es estudiar el registro AWS de alarmas de PE Calama, provisto por el sistema SCADA desarrollado por Siemens Gamesa.

Durante los primeros meses de trabajo, no se tenía acceso directo al sistema SCADA, por lo que la forma de obtener los registros AWS fue solicitarlos con el Jefe de Planta de PE Calama. El registro AWS fue provisto con información desde febrero de 2022 hasta diciembre de 2022.

A partir de conversaciones con personal de la compañía, se optó por enfocar este análisis en ciertos grupos de alarmas y no el total de estas. En particular, se escogen como sistemas aquellos que pueden presentar fallas mecánicas, los cuales son: **Sistema yaw, grupo hidráulico, gearbox y generador.**

El primer análisis se lleva a cabo contando el tiempo total en el cual se presentan las alarmas durante el periodo entre febrero y diciembre de 2022. Los resultados de este conteo se muestran en la Tabla 4.1. Similarmente, se ordenan las turbinas en un listado según el tiempo total en el cual presentan alarmas activas, resultado que se muestra en la Tabla 4.2.

Tabla 4.1: Listado de alarmas ordenadas por tiempo total

Ranking	Tiempo (hh:mm:ss)	Alarma
1	14852:54:17	423 Aviso filtro principal de la multiplicadora sucio
2	1563:58:44	417 Filtro principal multiplicadora sucio
3	614:54:43	222 Muy baja presión grupo hidráulico
4	400:01:00	108 Desenrollamiento automático
5	301:26:08	427 Aviso filtro auxiliar multiplicadora sucio
6	280:21:21	217 Aviso de mal funcionamiento de la válvula del control de carga

7	277:39:01	211 Filtro hidráulico HUB sucio
8	212:26:36	203 Baja presión grupo hidráulico
9	187:47:28	205 Nivel bajo de aceite de refrigeración en grupo hidráulico
10	156:30:10	223 Temperatura muy alta aceite grupo hidráulico
11	153:48:58	219 Temperatura alarma grupo hidráulico
12	125:03:33	410 OGS
13	107:06:45	401 Baja presión multiplicadora
14	87:21:44	115 Alta presión freno del yaw
15	85:50:15	426 Fallo lectura velocidad rotor
16	83:29:52	110 Error cableado sensor de giro
17	75:24:23	438 Alarma de accionamiento de bomba de multiplicadora
18	70:07:09	548 Confirmación de fallo de ventiladores del generador. Discordancia entre la activación de la señal
19	65:48:57	405 Rotor o alta velocidad generador en STOP o EMERGENCIA
20	57:26:13	547 Confirmación de fallo de ventiladores del generador. Uno o más ventiladores fallan y la temperatura

Tabla 4.2: Listado de turbinas ordenadas por tiempo total de alarmas.

Nº turbina	Tiempo total (hh:mm:ss)	Nº turbina	Tiempo total (hh:mm:ss)	Nº turbina	Tiempo total (hh:mm:ss)
19	4634:20:47	5	201:58:54	13	73:02:28
14	2715:30:13	8	172:42:52	33	73:00:11
4	2655:01:45	29	161:35:38	16	72:14:04
22	1814:35:33	23	157:33:29	10	72:02:34
3	1567:52:49	24	155:34:03	15	62:45:47
1	1023:04:22	18	137:08:52	11	62:32:46
17	945:17:26	30	112:54:14	25	53:11:59
28	909:07:42	32	102:57:46	36	50:43:41
6	728:13:01	31	95:45:39	7	47:28:00
35	376:41:26	26	94:22:17	12	39:23:25
27	367:27:28	20	86:27:15	21	12:37:18
2	238:45:20	34	74:47:23	9	6:46:36

Estos resultados permiten obtener un primer acercamiento hacia el entendimiento sobre las fallas recurrentes de las turbinas. Se puede observar que alarmas asociadas a filtros sucios de la multiplicadora (gearbox) están presentes una cantidad de tiempo gigantesca; sólo la alarma 423 corresponde a un total de 1.70 años aproximadamente. Sin embargo, el origen de esta alarma está estrechamente relacionado con la ubicación geográfica de PE Calama; al estar en medio del desierto, el parque se encuentra expuesto a la

arena que acarrea el viento, la cual se filtra al interior del nacelle de las turbinas y ensucia constantemente los filtros. Debido a que es una alarma que tiene mayor relación con causales ambientales y no corresponde a una falla detectable con las variables entregadas por el SCADA, se descartan del análisis.

Por otro lado, el grupo hidráulico presenta las alarmas 222, 203, 205 y 223 con bastantes horas acumuladas. La única variable disponible del grupo hidráulico es la **presión**, sin embargo, la alta frecuencia de alarmas asociadas a este sistema hace que estudiarlo sea tremendamente interesante. De la gearbox se tienen las alarmas 410 y 401 que se presentan por más de 100 horas en total. En el generador se aprecia que hay bastante tiempo donde se alerta que el ventilador presenta mal funcionamiento, lo cual también puede ser un caso de estudio de interés.

Para focalizar aún más este análisis se vuelve a hacer un listado de alarmas, pero esta vez considerando sólo aquellas a las cuales les sigue un Estado de *Aerogenerador en NO COMUNICA*, debido a que esto indica que el operador cortó la fibra de comunicación entre la sala de control y la turbina para que se realice una intervención al interior de esta última. Este es el único indicador de mantenimiento que se tiene en este trabajo, y por lo tanto es esencial para determinar las fallas críticas de las turbinas. El método para realizar este filtrado es el siguiente:

- En el registro AWS se añade la columna *NO COMUNICA*.
- En esta columna se introduce una fórmula que sigue el siguiente principio:

$$\text{No comunica} = \begin{cases} 1 & : \text{Fecha fin alarma} = \text{Fecha inicio NO COMUNICA} \\ & : \text{Caso contrario} \end{cases} \quad (4.1)$$

- Se coloca un filtro sobre esta columna y se escogen sólo las celdas que tienen valor 1.
- Este método asegura que las alarmas filtradas hayan aparecido exactamente antes de que se haya realizado el mantenimiento y sirven como indicador de falla.

Al aplicar esta metodología al AWS se obtiene el siguiente listado de alarmas ordenadas por tiempo acumulado:

Tabla 4.3: Listado de alarmas que derivan en un estado NO COMUNICA, ordenadas por tiempo acumulado.

Ránking	Tiempo total (hh:mm:ss)	Alarma
1	97:49:39	222 Muy baja presión grupo hidráulico
2	64:27:20	223 Temperatura muy alta aceite grupo hidráulico
3	56:14:30	548 Confirmación de fallo de ventiladores del generador. Discordancia entre la activación de la seña
4	49:22:47	401 Baja presión multiplicadora
5	40:32:24	545 Confirmación de fallo del aviso de los ventiladores del generador.
6	33:43:19	205 Nivel bajo de aceite de refrigeración en grupo hidráulico
7	28:03:49	514 Aviso hueco de tension en red
8	25:08:37	438 Alarma de accionamiento de bomba de multiplicadora
9	14:42:51	116 Orientando y alta presión freno del yaw
10	14:32:53	219 Temperatura alarma grupo hidráulico
11	12:16:21	541 Presión baja de agua del generador
12	11:28:58	547 Confirmación de fallo de ventiladores del generador. Uno o más ventiladores fallan y la temperat
13	10:25:50	217 Aviso de mal funcionamiento de la válvula del control de carga
14	9:50:45	426 Fallo lectura velocidad rotor
15	9:20:58	417 Filtro principal multiplicadora sucio
16	7:11:30	207 Alta temperatura aceite hidráulico
17	4:20:53	211 Filtro hidráulico HUB sucio
18	3:21:11	110 Error cableado sensor de giro
19	1:49:23	410 OGS
20	1:19:10	203 Baja presión grupo hidráulico
21	0:29:55	407 Alta velocidad rotor
22	0:19:56	406 Alta velocidad generador
23	0:04:59	114 Baja presión freno del yaw
24	0:04:58	127 Posibles fugas en circuito yaw

La Tabla 4.3 confirma la necesidad de estudiar el grupo hidráulico; las primeras 2 alarmas que se presentan por mayor cantidad de tiempo vienen de este sistema, por lo que es uno de los que más mantenimiento requiere y en consecuencia, le resta disponibilidad al parque. Siguiendo la misma idea, esta tabla también confirman la criticidad del generador y de la gearbox, ya que permanecen bastante tiempo con alertas que derivan en intervenciones. Ahora, para continuar focalizando el análisis se pasa a identificar las 2 alarmas que más acumulan tiempo por cada turbina. El resultado de esto se muestra en la Tabla 4.4, donde se muestran sólo las primeras 10 turbinas de mayor tiempo acumulado.

Tabla 4.4: Listado de turbinas ordenadas por su 1º y 2º alarma predecesora de estado NO COMUNICA de mayor tiempo acumulado.

Turbina	Alarma N°1	Tiempo (hh:mm:ss)	Alarma N°2	Tiempo (hh:mm:ss)
29	401 Baja presión multiplicadora	42:01:41	108 Desenrollamiento automático	1:39:19
2	548 Confirmación de fallo de ventiladores del generador. Discordancia entre la activación de la señal	41:16:43	545 Confirmación de fallo del aviso de los ventiladores del generador.	40:32:24
5	223 Temperatura muy alta aceite grupo hidráulico	24:40:03	205 Nivel bajo de aceite de refrigeración en grupo hidráulico	13:14:10
7	222 Muy baja presión grupo hidráulico	21:28:09	110 Error cableado sensor de giro	3:21:11
25	438 Alarma de accionamiento de bomba de multiplicadora	19:11:07		
1	223 Temperatura muy alta aceite grupo hidráulico	15:44:55	222 Muy baja presión grupo hidráulico	5:01:23
17	548 Confirmación de fallo de ventiladores del generador. Discordancia entre la activación de la señal	14:57:47	222 Muy baja presión grupo hidráulico	1:30:39
10	514 Aviso hueco de tensión en red	14:48:02		
27	116 Orientando y alta presión freno del yaw	14:37:51	219 Temperatura alarma grupo hidráulico	7:44:36
6	222 Muy baja presión grupo hidráulico	13:38:46		

La Tabla 4.4 permite observar de mejor manera qué turbinas pueden ser seleccionadas como casos de estudio para probar los modelos. La turbina N°29 presenta fallas importantes relacionadas a la presión en la gearbox, lo que da pistas de las variables que podrían estudiarse en un modelo de aprendizaje de máquinas; la turbina N°25 igualmente tiene una alarma con origen en la gearbox. Por otro lado, la turbina N°2 presenta fallos en los ventiladores del generador, por lo que las variables de temperatura de este podrían ser un buen foco de estudio. Las turbinas N°1, 5, 6, 7 y 27 tienen importantes alertas provenientes del grupo hidráulico, por lo que pueden ser un caso de estudio para un modelo de este subsistema.

Caso de la turbina N°9

La turbina N°9 es un caso particular, porque esta tuvo una de las fallas más catastróficas de PE Calama, la rotura de una de sus palas, y viendo el registro AWS es sencillo notar que ninguna alarma del SCADA pudo haber alertado sobre este evento, de hecho se puede notar en la Tabla 4.2 que esta turbina es la que menos tiempo acumulado presenta alarmas (aunque una de las razones detrás de esto es que la turbina dejó de funcionar después de la rotura, que ocurrió el 26 de julio de 2022). De manera similar, dentro del listado

de variables en la Tabla 2.1 no hay nada que tenga una relación clara con el estado de las palas, lo cual lleva a cuestionar si existe alguna manera de construir un modelo capaz de predecir el evento con la información disponible.

El movimiento de las palas está controlado por el sistema pitch, el cual hace que estas roten en torno a su eje para maximizar la velocidad de giro del rotor en función de la dirección y velocidad del viento. El sistema pitch, a su vez realiza su trabajo mediante actuadores hidráulicos, lo cual orienta un poco la búsqueda de variables que puedan ser utilizadas en un modelo para detectar fallas de palas. Un plano del grupo hidráulico y de los sistemas que este controla pueden apreciarse en la Figura 4.1.

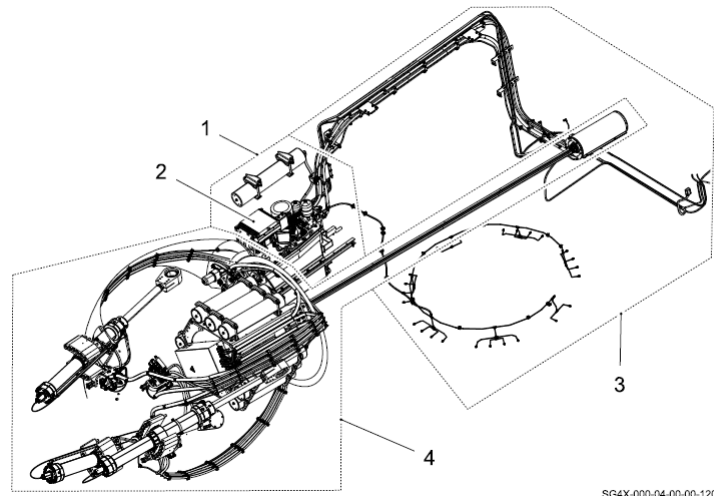


Figura 4.1: Componentes principales del grupo hidráulico. 1: Unidad hidráulica. 2: Sistema de acondicionamiento térmico. 3: Circuito hidráulico. 4: Sistema de control pitch

Dicho lo anterior, el sistema sobre el cual una falla de pala probablemente tenga mayor influencia es el grupo hidráulico. Igualmente, cualquier anomalía en el estado de salud de las palas puede tener incidencia directa sobre la potencia generada por la turbina; al no tener palas en buen estado no se puede aprovechar al máximo la potencia del viento, por lo que en el generador podría verse reflejada una falla de pala.

4.2. Casos de estudio

A partir de las conclusiones hechas en la sección anterior, se determinan primero las turbinas con las cuales se construyen los modelos de aprendizaje de máquinas y los subsistemas principales en los cuales se desea detectar las novedades.

Tabla 4.5: Casos de estudio

Turbina	Sistema de interés
2	Generador
9	Pala
27	Grupo hidráulico
29	Gearbox

Luego, se determinan las variables que componen los conjuntos de datos que se arman para cada sistema, con inspiración tomada de los conjuntos utilizados en el trabajo de Zhao et al.[17] y de conversaciones con trabajadores de Engie con experiencia en turbinas eólicas.

Tabla 4.6: Variables de sistema gearbox

Nº	Variable	Notación
1	Temperatura rodamiento gearbox (°C)	T_{rg}
2	Temperatura aceite gearbox (°C)	T_{ag}
3	Potencia activa (kW)	P
4	Velocidad generador (rpm)	Ω_g
5	Velocidad viento (m/s)	v_w
6	Temperatura ambiente (°C)	T_a

Tabla 4.7: Variables de sistema grupo hidráulico

Nº	Variable	Notación
1	Presión grupo hidráulico (bar)	p_{gh}
2	Potencia activa (kW)	P
3	Velocidad rotor (rpm)	Ω_r
4	Velocidad generador (rpm)	Ω_g
5	Ángulo pitch (°)	α
6	Posición nacelle (°)	γ
7	Velocidad viento (m/s)	v_w

Tabla 4.8: Variables de sistema generador

Nº	Variable	Notación
1	Temperatura rodamiento drive-end (°C)	T_{rde}
2	Temperatura rodamiento non-drive-end (°C)	T_{rnde}
3	Temperatura devanados 1 (°C)	T_{d1}
4	Temperatura devanados 2 (°C)	T_{d2}
5	Temperatura devanados 3 (°C)	T_{d3}
6	Temperatura anillos colectores (°C)	T_{ac}
7	Potencia activa (kW)	P
8	Velocidad generador (rpm)	Ω_g
9	Velocidad viento (m/s)	v_w
10	Temperatura ambiente (°C)	T_a

4.3. Inicialización y preprocesamiento de datos

Habiendo seleccionado los casos de estudio para el trabajo, se crea el código en *Spyder* y se importan las librerías que se utilizan para leer los datos del SCADA. Los X-Minutales del SCADA se descargan como archivos de valores separados por comas, con datos que abarcan el periodo entre octubre de 2021 (puesta en marcha de PE Calama) y diciembre de 2022.

Código 4.1: Inicialización de código

```

1 import numpy as np #Numpy para efectuar operaciones matemáticas
2 import pandas as pd #Pandas para procesar archivos .csv
3 import matplotlib.pyplot as plt #Matplotlib para generar gráficos
4
5 #Se importan variables de operación de turbinas
6 xminutal00 = pd.read_csv('X-Minutal00.csv',sep=';') #Se importa el archivo X-Minutal deseado,
   ↪ reemplazando 00 por el número respectivo de la turbina.
7 #El archivo X-Minutal debe estar en la misma carpeta del código. En caso contrario, debe colocarse
   ↪ la ubicación completa del archivo.
8 xminutal00 = xminutal00.dropna() #Se eliminan las filas con celdas vacías
9 xminutal00 = xminutal00.reset_index(drop=True) #Se reinician los índices del DataFrame luego de
   ↪ eliminar las filas vacías

```

A continuación, se definen los conjuntos de entrenamiento y de prueba para los modelos con los cuales se trabaja. Para el conjunto de entrenamiento se espera tener uno que represente correctamente la operación normal de uno de estos equipos bajo todas las condiciones climáticas posibles, por lo que tener datos de un año completo o más sería lo ideal. Bajo este razonamiento, se opta por seleccionar la turbina más saludable como turbina de entrenamiento para los modelos, lo cual se traduce como la turbina que presenta menor presencia de alarmas y fallas, y desde los resultados de la sección anterior se tiene que esta turbina es la número **12**, con 153 horas de detención en general.

Código 4.2: Conjuntos de entrenamiento y de prueba

```

1 #Elección de datos según turbina
2
3 #Turbina de entrenamiento
4 Entrenador = xminutal12
5 Entrenador.loc[Entrenador['Average Rotor Speed 10M (rpm)']<=1,'Average Rotor Speed 10M (rpm)']
   ↪ = np.nan # Se ubican todas las detenciones de las turbinas a partir de la velocidad del rotor
   ↪ y los valores cercanos a la detención, por lo que se colocan como vacías todas las celdas donde
   ↪ la velocidad es menor a 1 rpm.
6 Entrenador = Entrenador.dropna() # Se eliminan las nuevas filas con celdas vacías
7 Entrenador = Entrenador.reset_index(drop=True) #Se reinician los índices
8 IndiceEntrenamiento = Entrenador.shape[0] #Se toma el tamaño del conjunto como la cantidad de
   ↪ filas para más adelante.
9
10 #Turbina de prueba
11 Tester = xminutal00 #Reemplazar 00 por el número de cualquier turbina
12 Tester.loc[Tester['Average Rotor Speed 10M (rpm)']<=1,'Average Rotor Speed 10M (rpm)'] = np.nan
13 Tester = Tester.dropna()
14 Tester = Tester.reset_index(drop=True)
15 IndiceTester = IndiceEntrenamiento+1 #Es necesario tener los índices donde termina el conjunto de
   ↪ entrenamiento y donde inicia el conjunto de prueba, ya que estos se unen a continuación.
16
17 #Unión de conjuntos de entrenamiento y de prueba.
18 Datos = pd.concat([Entrenador,Tester],axis=0)
19 Datos = Datos.reset_index(drop=True)

```


Del conjunto general de datos, luego se extrae cada una de las variables para armar más adelante los grupos de variables por sistema. En este punto es muy relevante recalcar que en los datos cuyo valor es superior a 1000, el SCADA tiene errores con la separación de decimales y de miles, ya que ambos los coloca como puntos, lo cual es imposible de interpretar por las funciones de NumPy. Esto ocurre con los valores de potencia y velocidad de generador. Para resolver el problema se debe crear una función dentro del código que elimina el punto que SCADA utiliza para indicar los miles.

Código 4.3: Extracción de variables y eliminación de errores

```

1 #Se extraen variables de operación
2 PresiónGH = Datos['Hidraulic group pressure 10M (bar)']
3 RotorVelProm= Datos['Average Rotor Speed 10M (rpm)']
4 GeneradorVelProm = Datos['Average Generator Speed 10M (rpm)']
5 TempRodamientoDriveEnd = Datos['Bearing D.E. Temperature 10M (F)']
6 TempRodamientoNonDriveEnd = Datos['Bearing N.D.E. Temperature 10M (F)']
7 TempRodamientoGearbox = Datos['Gearbox bearing temperature 10M (F)']
8 TempAceiteGearbox = Datos['Gearbox oil temperature 10M (F)']
9 PotenciaActiva = Datos['Average Active Power 10M (kW)']
10 TempAmbiente = Datos['Average Ambient Temperature 10M (F)']
11 BobinaGenerador1= Datos['Generator windings temperature 1 10M (F)']
12 BobinaGenerador2= Datos['Generator windings temperature 2 10M (F)']
13 BobinaGenerador3= Datos['Generator windings temperature 3 10M (F)']
14 AnilloGenerador= Datos['Generators sliprings temperature 10M (F)']
15 Viento = Datos['Average Wind Speed 10M (m/s)']
16 Pitch = Datos['Average Pitch Angle 10M (°)']
17 Voltaje = Datos['Average Grid Voltage 10M (V)']
18 Nacelle = Datos['Average Nacelle Position 10M (°)']
19
20 #Se extrae además el tiempo y se convierte a formato fecha
21 Tiempo = pd.to_datetime(Datos["Date"]);
22
23 #Se crea función para eliminar puntos en datos con valor sobre 1000.
24 #Los datos vienen con errores, toma el punto como decimal y además para separar los miles, ej:
    ↪ 1500.5 viene como 1.500.5
25
26 def eliminar_punto(texto):
27     if len(texto) >= 7 and texto[1] == '.':
28         return texto[0] + texto[2:]
29     elif len(texto) >= 7 and texto[2] == '.':
30         if texto[6] == '.':
31             return texto[0:1] + texto[3:]
32     return texto
33
34 #Se elimina el punto en potencia y velocidad generador
35 GeneradorVelProm = GeneradorVelProm.apply(eliminar_punto)
36 GeneradorVelProm = GeneradorVelProm.astype('float64')
37
38 PotenciaActiva=PotenciaActiva.apply(eliminar_punto)
39 PotenciaActiva=PotenciaActiva.astype('float64')

```

Tras esto, se crean los conjuntos de datos por sistema a modelar y se separan por conjunto de entrenamiento y de prueba, a partir de los índices rescatados en los pasos anteriores.

Código 4.4: Creación de DataFrame por cada sistema

```
1
2 #Set de datos gearbox
3 Gearbox = pd.DataFrame({"Potencia activa promedio (kW)":PotenciaActiva,
4                         "Velocidad promedio generador (rpm)":GeneradorVelProm,
5                         "Temperatura aceite gearbox (°F)":TempAceiteGearbox,
6                         "Temperatura rodamiento gearbox (°F)":TempRodamientoGearbox,
7                         "Temperatura ambiente (°F)":TempAmbiente,
8                         "Velocidad viento (m/s)": Viento})
9
10 #Set de datos generador
11 Generador = pd.DataFrame({"Potencia activa (kW)":PotenciaActiva,
12                          "Voltaje de Red (V)": Voltaje,
13                          "Velocidad Viento (m/s)":Viento
14                          "Temperatura Rodamiento D.E. (°F)":TempRodamientoDriveEnd,
15                          "Temperatura Rodamiento N.D.E. (°F)":TempRodamientoNonDriveEnd,
16                          "Temperatura Anillos Colectores (°F)": AnilloGenerador,
17                          "Temperatura Devanados 1 (°F)": BobinaGenerador1,
18                          "Temperatura Devanados 2 (°F)": BobinaGenerador2,
19                          "Temperatura Devanados 3 (°F)": BobinaGenerador3,
20                          "Temperatura Ambiente (°F)":TempAmbiente})
21
22 #Set de datos grupo hidráulico
23 GrupoHidraulico = pd.DataFrame({"Velocidad Viento (m/s)":Viento,
24                                  "Ángulo Pitch (°)":Pitch,
25                                  "Posición Nacelle (°)": Nacelle,
26                                  "Potencia Activa (kW)":PotenciaActiva,
27                                  "Velocidad Rotor (rpm)":RotorVelProm,
28                                  "Velocidad Generador (rpm)":GeneradorVelProm,
29                                  "Presión Grupo Hidráulico (bar)":PresiónGH})
30
31 # Se elige set de datos para entrenamiento y prueba
32 SetDatos = Sistema #Cualquiera de los 3 creados
33 DatosMes = 6*24*30 #Total de datos por mes de 30 días
34
35 #Se definen conjuntos de entrenamiento y de prueba
36 datos_train=SetDatos.to_numpy()[0:IndiceEntrenamiento] #Desde el primer dato hasta el último de
37     ↪ la turbina de entrenamiento
38
39 datos_test=SetDatos.to_numpy()[IndiceTester:] #Desde el primer dato de la turbina de prueba hasta
40     ↪ el último.
41
42 #Lo mismo para tener como índices las fechas de cada conjunto.
43 t_train = Tiempo[0:IndiceEntrenamiento]
44 t_test=Tiempo[IndiceTester:]
```

Finalmente, se aplica una normalización a ambos conjuntos de datos mediante la función *MinMaxScaler*

[27], que básicamente aplica a los datos lo siguiente:

$$\bar{X}_i = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad \forall i \in \{1, \dots, n\} \quad (4.2)$$

Código 4.5: Normalización de datos

```
1 from sklearn.preprocessing import MinMaxScaler
2
3 scaler = MinMaxScaler()
4
5 #Datos de entrenamiento normalizados
6 datos_train_norm = scaler.fit_transform(datos_train)
7 datos_test_norm = scaler.fit_transform(datos_test)
```

4.4. Modelos basados en algoritmos de detección de novedades

Los algoritmos de detección de novedades introducidos en el capítulo de Antecedentes se pueden implementar en Python a través de la librería Scikit [15].

Código 4.6: Importación de algoritmos de detección de novedades

```
1 from sklearn.covariance import EllipticEnvelope
2 from sklearn.ensemble import IsolationForest
3 from sklearn.neighbors import LocalOutlierFactor
4 from sklearn import svm
```

4.4.1. Elliptic Envelope

El uso de Elliptic Envelope tan solo requiere definir el porcentaje de datos contaminados en el conjunto de entrenamiento; el objeto introducido por Scikit recibe el parámetro *contamination*, que representa lo anterior. Se realizan pruebas para los 10 casos de estudio modificando este parámetro, que en el código se define como *outlier fraction*.

Tabla 4.9: Valores de outlier fraction utilizados

Outlier Fraction		
0.01	0.001	0.0001

A continuación se muestra el código que define al modelo. Se crea un modelo Elliptic Envelope, el cual se ajusta a los datos de entrenamiento y luego se utiliza para predecir el comportamiento de los datos de prueba. La predicción separa los valores que se encuentran fuera de la elipse y los marca como novedades.

Código 4.7: Código Elliptic Envelope

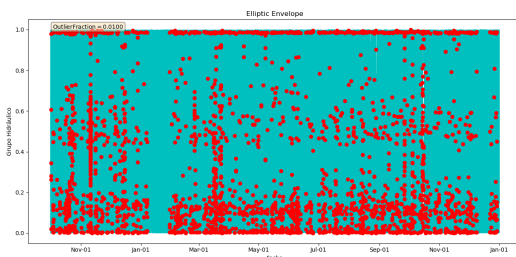
```
1 # parámetros para modelos
2 outlier_frac=0.0001; #fracción de valores anómalos, por defecto es 0.1
```

```

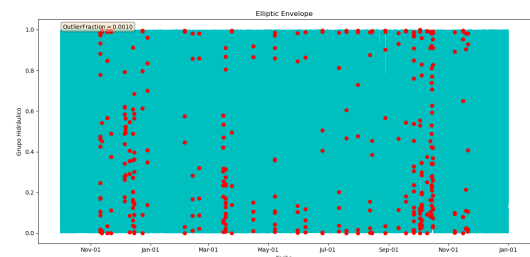
3 Modelo=EllipticEnvelope(contamination=outlier_frac)
4 leyenda = r'\$\mathrm{Outlier Fraction}=\%.4f\$' \% (outlier_frac, ) #Leyenda para insertar en
   ↪ los gráficos
5 title='Elliptic Envelope' #Título para los gráficos
6
7 Modelo.fit(datos_train_norm)
8 Yp=Modelo.predict(datos_test_norm)
9
10 xp0=datos_test_norm[np.where(Yp==1)[0,:]]#Datos de prueba donde no se detectan novedades
11 xp1=datos_test_norm[np.where(Yp==-1)[0,:]]#Datos de prueba donde sí se detectan novedades
12
13 t0=t_test[Yp==1]#Fechas donde no se detectan novedades
14 t1=t_test[Yp==-1]#Fechas donde sí se detectan novedades
15
16 props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)#Propiedades para el cuadro de leyenda
   ↪ .
17
18 fig, ax = plt.subplots()
19 ax.plot(t0,xp0,'c')
20 ax.plot(t1,xp1,'ro')
21 ax.text(0.05, 0.98, leyenda ,transform = ax.transAxes, fontsize=10, verticalalignment='top', bbox=
   ↪ props)
22 ax.set_title(title)
23 ax.set_xlabel('Fecha') # Etiqueta del eje x
24 ax.xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%b-%d'))
25 ax.set_ylabel('Datos', wrap=True)
26 plt.show()

```

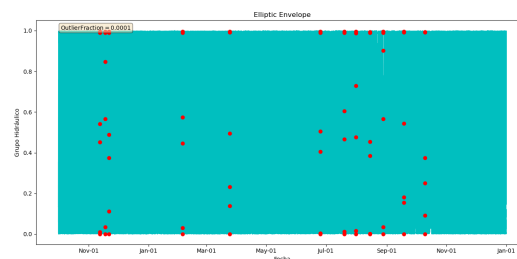
Resultados Turbina N°2 Sistema Grupo Hidráulico



(a) outlier_fraction = 0.01



(b) outlier_fraction = 0.001



(c) outlier_fraction = 0.0001

Figura 4.2: Resultados Elliptic Envelope para Turbina N°2. Sistema Grupo Hidráulico

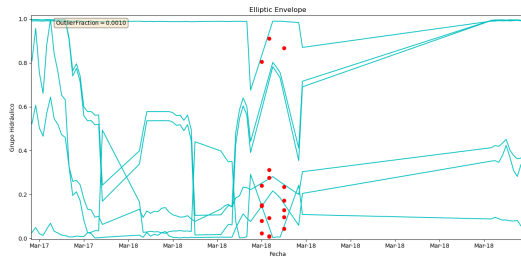
Es posible notar a partir de la Figura 4.2 que claramente la cantidad de Contaminación en los datos de entrenamiento influye directamente en la cantidad de novedades detectadas por Elliptic Envelope. Por un lado con un valor de Contaminación de 0.01 se presentan demasiadas detecciones, lo cual dificulta establecer una relación entre estas y las alarmas de por sí. Yendo al otro extremo, con un valor de Contaminación de 0.0001 las detecciones son muy pocas, por lo que su uso como herramienta de detección temprana de fallas sería nulo. Finalmente, en el caso medio de Contaminación igual a 0.001, el gráfico es más claro mostrando los momentos donde la turbina presenta un comportamiento anormal.

La turbina N°2 presenta fallas asociadas a alarmas del grupo hidráulico en las siguientes fechas:

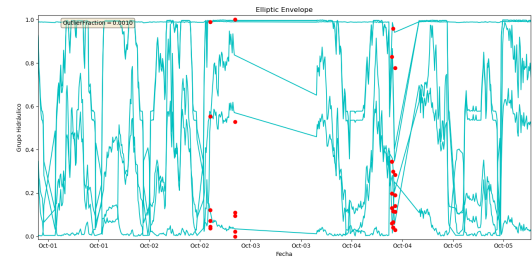
Tabla 4.10: Fallas Grupo Hidráulico Turbina N°2

Alarma	Estado posterior	Fecha Inicio	Fecha Fin	Duración
222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/03/18 07:48:30.555	2022/03/18 16:00:46.779	8:12:16
222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/09/29 19:08:20.489	2022/09/30 12:15:43.379	17:07:23
222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/09/30 19:20:27.090	2022/09/30 19:39:15.001	0:18:48
222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/10/02 20:10:43.355	2022/10/03 15:25:24.692	19:14:41
222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/10/04 10:54:11.667	2022/10/04 15:48:12.751	4:54:01

La primera de estas alarmas ocurre el 18 de marzo, y el resto entre el 30 de septiembre y el 4 de octubre. Si se hace un acercamiento en el gráfico a los datos de esta fecha se ve que, según la Figura 4.3, las detecciones no anticipan la falla y detectan sólo el momento previo a la detención. El modelo Elliptic Envelope detecta correctamente las detenciones, mas no las predice, por lo que su utilidad es la misma que el sistema de alarmas ya instalado en las turbinas, con la frecuencia de muestreo de datos disponible.



(a) Alarma 18 de marzo Turbina N°2



(b) Alarmas del 30 de septiembre al 4 de octubre

Figura 4.3: Alarmas Grupo Hidráulico detectadas con modelo Elliptic Envelope

4.4.2. Isolation Forest

Este algoritmo requiere igualmente definir el parámetro *contamination*, pero además necesita que se especifiquen otros, en particular *n_estimators* que es la cantidad de árboles aleatorios que se quiere generar. Por otro lado se pueden definir *max_samples* (número de muestras que se toma desde el conjunto de datos para entrenar los árboles aleatorios) y *max_features* (número de características que se toma desde el conjunto de datos para entrenar los árboles aleatorios), pero ambos parámetros se dejan con sus valores predeterminados.

Dicho lo anterior, se prueba para cada caso de estudio 9 combinaciones detalladas en la Tabla .

Tabla 4.11: Parámetros para Isolation Forest

outlier_fraction	0.01	0.001	0.0001
n_estimators	50	100	200

En el caso de Isolation Forest, el código del modelo es casi idéntico al de Elliptic Envelope, por lo que no se incluye la parte donde se crean las predicciones.

Código 4.8: Código Isolation Forest

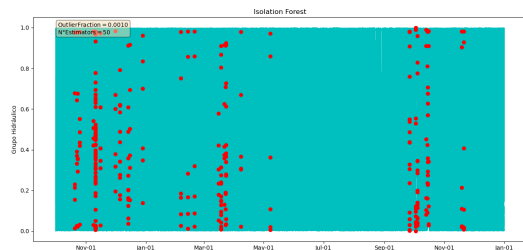
```

1 # parámetros para modelos de machine learning
2 outlier_frac=0.0001; #fracción de valores anómalos, por defecto es 0.1
3 n_estimators = 100; #Cambiar entre 50, 100 y 200
4
5 # Modelo=IsolationForest(n_estimators=n_estimators,max_samples='auto',contamination=
6   ↪ outlier_frac)
7 leyenda = '\n'.join((r'\$\mathrm{Outlier Fraction}=\%.4f\$' \% (outlier_frac, ),r'\$\mathrm{N}^\circ
8   ↪ Estimators}=\%.0f\$' \% (n_estimators, )))
9 title='Isolation Forest'
10
11 Modelo.fit(datos_train_norm)
12 Yp=Modelo.predict(datos_test_norm)

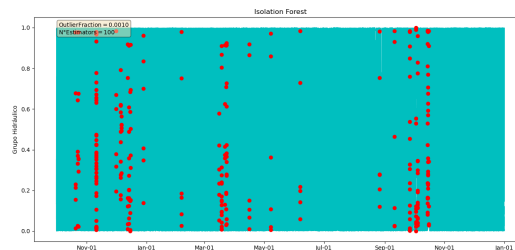
```

Resultados Turbina N°2 Sistema Grupo Hidráulico

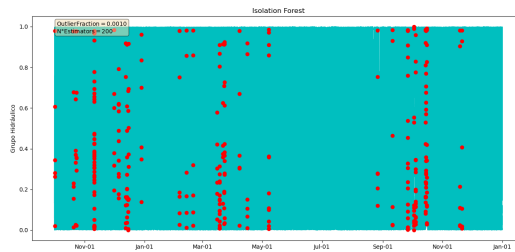
El ejercicio en este caso es idéntico al de Elliptic Envelope, y por las mismas razones que se determina que el caso que más vale la pena estudiar es aquel con Contamination igual a 0.001.



(a) $n_{\text{estimators}} = 50$



(b) $n_{\text{estimators}} = 100$

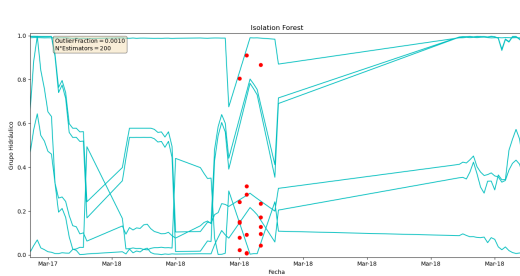


(c) $n_{\text{estimators}} = 200$

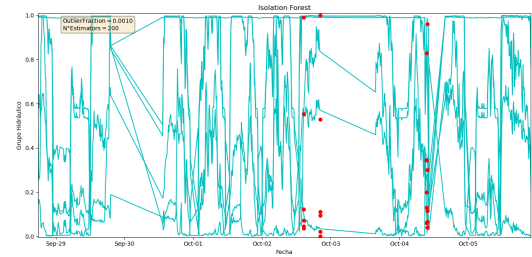
Figura 4.4: Resultados Isolation Forest para Turbina N°2. Sistema Grupo Hidráulico, con $\text{outlier_fraction} = 0.001$

Los resultados de Isolation Forest son similares a los de Elliptic Envelope, según la Figura 4.4 en el

sentido de que visualmente no presentan diferencias significativas. Para estudiar si se logran anticipar las fallas se realiza el mismo ejercicio que en el caso de Elliptic Envelope, pero para no sobrecargar el informe con gráficos, sólo se muestra uno de los casos, y ya que no hay uno que destaque por sobre el resto se ejemplifica con $n_estimators = 200$



(a) Alarma 18 de marzo Turbina N°2



(b) Alarmas del 30 de septiembre al 4 de octubre Turbina N°2

Figura 4.5: Alarmas Grupo Hidráulico detectadas con modelo Isolation Forest, con $outlier_fraction = 0.001$

Se puede observar de la Figura 4.5 que los resultados son casi idénticos a los del modelo Elliptic Envelope, por lo que Isolation Forest tampoco aportaría valor al sistema de monitoreo de Engie como herramienta de anticipación, ya que detecta únicamente las anomalías en el instante previo a la detención.

4.4.3. Local Outlier Factor

En el caso del algoritmo Local Outlier Factor, también se debe definir *contamination*. Ahora, además de esto debe definirse el parámetro $n_neighbors$, que indica con cuántos vecinos más cercanos se quiere comparar la densidad local. Es necesario indicar además el parámetro $novelty == True$, ya que este algoritmo se utiliza de forma predeterminada para detección de anomalías y no de novedades.

Se prueba para cada caso las 9 combinaciones posibles entre los parámetros de la Tabla 4.12.

Tabla 4.12: Parámetros para Local Outlier Factor

outlier_fraction	0.01	0.001	0.0001
n_neighbors	5	10	20

Similar a los 2 modelos anteriores, el código de Local Outlier Factor se muestra a continuación:

Código 4.9: Código Local Outlier Factor

```

1 # parámetros para modelos de machine learning
2 outlier_frac=0.0001; #fracción de valores anómalos, por defecto es 0.1
3 n_neighbors = 20; #Cambiar entre 5, 10 y 20
4
5 Modelo=LocalOutlierFactor(n_neighbors=n_neighbors,contamination=outlier_frac,novelty=True)
6 leyenda = '\n'.join((r'\$\\mathrm{Outlier Fraction}=\% .4f\$' \% (outlier_frac, ),r'\$\\mathrm{N}^{\circ}
   \leftrightarrow Neighbors}=\% .0f\$' \% (n_neighbors, )))
7 title='Local Outlier Factor'
8

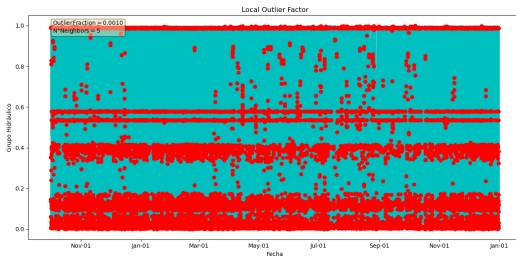
```

```
9 Modelo.fit(datos_train_norm)
10 Yp=Modelo.predict(datos_test_norm)
```

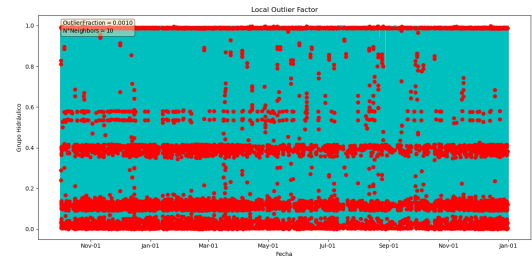
Resultados Turbina N°2

Sistema Grupo Hidráulico

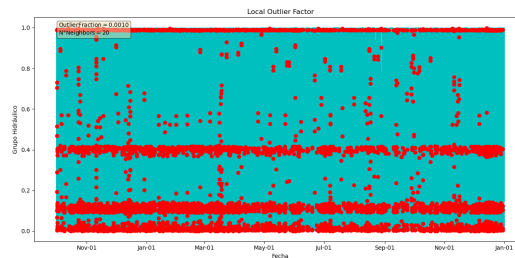
Al igual que en con Isolation Forest, se ilustran los resultados con Contamination igual a 0.001.



(a) $n_neighbors = 5$



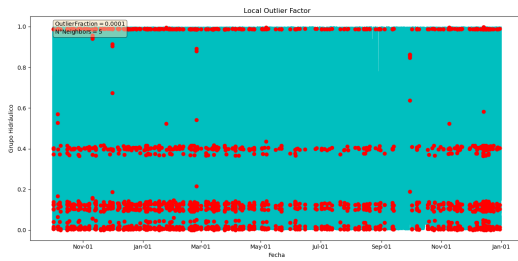
(b) $n_neighbors = 10$



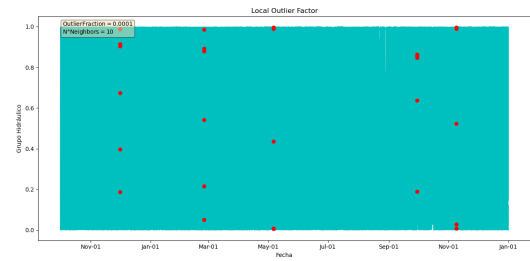
(c) $n_neighbors = 20$

Figura 4.6: Resultados Local Outlier Factor para Turbina N°2. Sistema Grupo Hidráulico, con outlier_fraction = 0.001

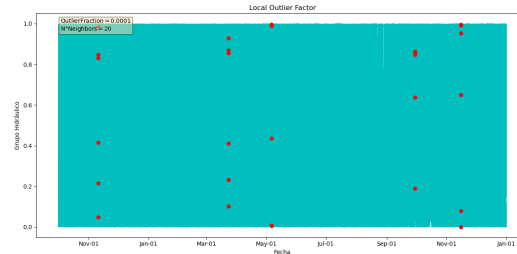
Inmediatamente, al observar la Figura 4.7 es posible concluir que el nivel de Contamination es muy alto para este modelo, ya que reconoce casi todos los datos como novedades. Al reducir el nivel de Contamination se obtiene lo siguiente:



(a) $n_neighbors = 5$



(b) $n_neighbors = 10$



(c) $n_neighbors = 20$

Figura 4.7: Resultados Local Outlier Factor para Turbina N°2. Sistema Grupo Hidráulico, con $outlier_fraction = 0.0001$

Nuevamente se identifican problemas. En el caso de $n_neighbors = 5$ se siguen detectando demasiadas novedades. Por otro lado, para $n_neighbors = 10$ y $n_neighbors = 20$ ocurre lo contrario, que se detectan muy pocas, por lo que Local Outlier Factor queda completamente descartado.

4.4.4. One-Class Support Vector Machine

El algoritmo OC SVM no requiere indicar el parámetro *contamination*, pero si requiere elegir un *Kernel* y definir el parámetro *nu*, el cual establece un límite superior para la fracción de errores de entrenamiento y un límite inferior para la fracción de vectores de soporte. Debe estar en el intervalo entre 0 y 1.

Se prueba para cada caso las 9 combinaciones posibles entre los parámetros de la Tabla 4.13.

Tabla 4.13: Parámetros para OC-Support Vector Machine

Kernel	polynomial	rbf	sigmoid
Nu	0.1	0.01	0.001

Por último, al código de OC-SVM sólo se le cambian los parámetros iniciales como ya se había mencionado.

Código 4.10: Código One-Class Support Vector Machine

```

1 # parámetros para modelos de machine learning
2 nu = 0.01; #Cambiar entre 0.1, 0.001 y 0.0001
3 kernel = "poly"# Cambiar entre 'poly', 'rbf' y 'sigmoid'
4
5 Modelo=svm.OneClassSVM(nu=nu,kernel=kernel,gamma='auto')
6 leyenda = '\n'.join((r'\$nu=\%.4f\$' \% (nu, ),r'\$mathrm{Kernel}=\%s\$' \% (kernel)))

```

```

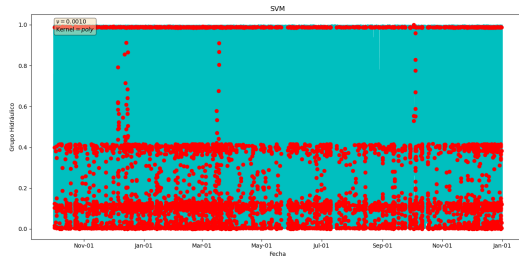
7 title='SVM'
8
9 Modelo.fit(datos_train_norm)
10 Yp=Modelo.predict(datos_test_norm)

```

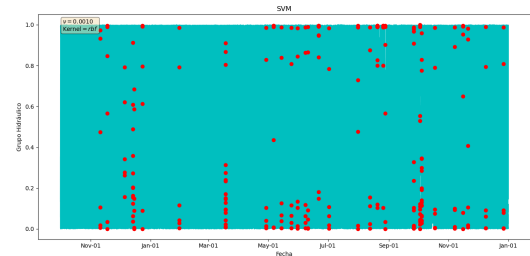
Resultados Turbina N°2

Sistema Grupo Hidráulico

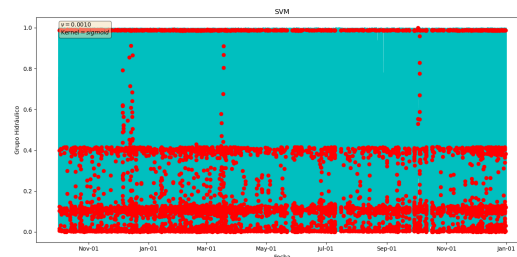
En el caso del algoritmo OC-SVM se muestran los resultados de los 3 Kernel seleccionados para un único valor ν , que será 0.001.



(a) Kernel = Polynomial (grado 3)



(b) Kernel = RBF



(c) Kernel = Sigmoid

Figura 4.8: Resultados One-Class SVM para Turbina N°2. Sistema Grupo Hidráulico, con $\nu = 0.001$

De las Figuras 4.8.a y 4.8.c se puede observar que los resultados son similares a los de Local Outlier Factor, por lo que se descartan para un posible modelo. Por otro lado, la Figura 4.8.b muestra resultados semejantes a los de Elliptic Envelope e Isolation Forest, y por lo tanto es posible concluir que las fallas indicadas para esos casos tampoco son anticipadas con OC-SVM.

4.5. Modelo basado en Deep Autoencoder

La idea detrás del uso de una red DAE surge a partir de los resultados poco claros obtenidos con los modelos de detecciones de novedades. Tras una búsqueda bibliográfica sobre aplicaciones del aprendizaje de máquinas en parques eólicos, se encuentra el trabajo de Zhao et al. [17], cuya base se encuentra explicada en el capítulo de Antecedentes. La red DAE de ese trabajo se preentrena con RBMs, procedimiento desarrollado por Hinton y Salakhutdinov [19] para la reducción de dimensionalidad de datos en el procesamiento de imágenes. Este último trabajo hizo público su código construido en MATLAB, el cual fue posteriormente adaptado a lenguaje Python por Jeremy Lane [28].

Al igual que en el caso anterior, se utiliza como conjunto de entrenamiento las variables de operación de la turbina N°12, cuyos datos abarcan el periodo entre octubre de 2021 y diciembre de 2022. Para entrenar redes neuronales es necesario definir un conjunto de validación, este puede ser establecido tanto manualmente como de forma aleatoria; en este caso se opta por la segunda opción, tomando el conjunto de validación como un porcentaje de los datos de entrenamiento.

Para el preentrenamiento con RBMs se importa primero una librería de redes neuronales creada para Python, la cual es **Keras** [29], aplicada sobre el framework **TensorFlow** [30]. El código desarrollado por Lane [28] define la *clase* Autoencoder, la cual realiza preentrenamiento por capas con RBMs, luego desarrolla las capas en un modelo Deep Autoencoder. El Deep Autoencoder es retornado como un modelo Keras, el cual puede ser guardado y ajustado finamente con el método del gradiente descendente.

Código 4.11: Importación de módulos de Keras y clase Autoencoder

```
1 from tensorflow.keras.models import Model
2 from Autoencoder import *
```

La arquitectura de la red DAE, es decir, el número de capas del encoder y el tamaño de cada una de estas se define con la *clase* Autoencoder.

Código 4.12: Generación de modelo Autoencoder

```
1 #La dimensión de la primera (y última capa) es igual a la cantidad de variables de entrenamiento.
2 #Para el tamaño de las siguientes capas se toman como referencia ejemplos similares.
3 pretrained_autoencoder = Autoencoder(layer_dims = [datos_train.shape[1],layer_dims])
```

Tras esto se preentrena el autoencoder, para lo cual se debe definir el número de *epochs* y el número de muestras.

Código 4.13: Preentrenamiento con RBMs

```
1 #Epoch: un corte arbitrario, generalmente definido como "una pasada por todo el conjunto de datos",
   ↳ utilizado para separar el entrenamiento en fases distintas, lo que resulta útil para el registro y
   ↳ la evaluación periódica.
2 #Sample: un elemento del conjunto de datos.
3 pretrained_autoencoder.pretrain(datos_train_norm.T,epochs = epochs,num_samples =
   ↳ num_samples)
4 pretrained_model = pretrained_autoencoder.unroll()
```

Luego de obtener resultados suficientemente buenos, el modelo preentrenado se guarda en la carpeta donde se encuentra ubicado el código.

Código 4.14: Guardar modelo preentrenado

```
1 pretrained_model.save('pretrained_model.keras',save_format='keras')
```

A continuación se realiza el ajuste fino de los modelos, buscando maximizar la precisión y minimizar la pérdida de información.

Código 4.15: Ajuste fino de modelos

```
1 from tensorflow.keras.models import load_model
```

```

2 from tensorflow.keras.callbacks import Callback
3 from tensorflow.keras.optimizers import Adam # Optimizador que aplica el algoritmo de gradiente
  ↪ descendente
4 from livelossplot import PlotLossesKeras # Para graficar Precisión y Pérdida en cada epoch
5
6 finetuned_model=load_model('pretrained_model.keras',compile=True)
7
8 finetuned_model.compile(loss='mean_squared_error',optimizer=Adam(),metrics=['accuracy'])
9
10 #Notar que al ser Autoencoder, el input y el output son el mismo conjunto de datos
11 #validation_split es el porcentaje del conjunto de entrenamiento que será utilizado como conjunto de
  ↪ validación.
12 finetuned_model.fit(datos_train_norm,datos_train_norm,epochs=epochs,
13                    validation_split= validation_split, verbose=1,
14                    callbacks=[PlotLossesKeras()])
15
16 #Por último se guarda el modelo definitivo
17 finetuned.save('finetuned_model.keras',save_format='keras')

```

El paso final que se lleva a cabo con el Autoencoder es obtener la predicción de los datos de testeo con el modelo generado y calcular el error de reconstrucción de la predicción con respecto a los datos originales.

Código 4.16: Predicción y cálculo de Error de Reconstrucción

```

1 import numpy as np
2
3 Ys = finetuned_model.predict(datos_test_norm)
4
5 Re = np.linalg.norm(Ys-datos_test_norm,axis=1) #Error de reconstrucción
6 Residual = Ys-datos_test_norm #Residuales

```

Posteriormente, se instala la librería *ads-evt* tomada del trabajo realizado por Siffer et al. [20]. La implementación original del método Peaks-Over-Threshold no se encuentra disponible, sin embargo un repositorio en GitHub [31] se encargó de recopilar la información del trabajo en la librería mencionada. Luego de instalar la librería, se importa al código.

Código 4.17: Importación de librería *ads-evt* y definición de conjuntos sobre los cuales se aplica el umbral

```

1 #se importa ads-evt
2 from typing import List
3 import ads_evt as spot
4
5 # Se convierte el error de reconstrucción a Data Series, convirtiendo los índices en las fechas del
  ↪ conjunto de prueba.
6 ReSeries = pd.Series(Re1, index = t_test)
7 t_test = t_test.reset_index(drop=True)
8
9 #Se debe calibrar el umbral con un conjunto inicial, seleccionando el primer mes de datos.
10 init_data = ReSeries[0:DatosMes]

```

```

11 data = ReSeries[DatosMes:]
12
13 #Se utiliza el resto de datos para evaluar el umbral
14 tdata = t_test[DatosMes:]
15 tdata = tdata.reset_index(drop=True)

```

Para generar los gráficos se deben definir la probabilidad q y la ventana temporal d . Además, se puede seleccionar el tipo de método POT que se quiere aplicar; se tiene el caso estacionario SPOT, la versión doble de este bSPOT que calcula el umbral superior e inferior, el caso con desviaciones DSPOT, y su versión doble bDSPOT.

Sobre los gráficos, para identificar las detenciones de la turbina causadas por alarmas se colocan etiquetas en forma de rectángulos con un número asociado a las alarmas (en base a un listado). Este etiquetado requiere un trabajo manual de búsqueda de los índices de la fecha donde inician las detenciones, y para el análisis de cada turbina deben cambiarse estas etiquetas por las que correspondan, entonces se debe prestar atención a este paso.

Código 4.18: Generación del gráfico

```

1 #Parámetros de los umbrales
2 proba = q
3 depth = d
4
5 #Número de turbina
6 turbina = n
7
8 #Selección de método POT (pueden tenerse varios a la vez)
9 models: List[spot.SPOTBase] = [
10     #spot.SPOT(q=proba),
11     spot.dSPOT(q=proba,depth=depth),
12     # spot.biSPOT(q=proba),
13     # spot.bidSPOT(q=proba,depth=depth,n_points=2),
14 ]
15
16 #Etiquetado de alarmas
17 import matplotlib.patches as mpatches
18
19 fig, ax = plt.subplots()
20
21 left, bottom, width, height = (i_1, 0, 20, 2) #Aquí, i_1 es el índice de la fecha del evento alarma/
    ↪ detención 1
22 rect=mpatches.Rectangle((left,bottom),width,height,alpha=0.1,facecolor="red")
23 plt.gca().add_patch(rect)
24 plt.text(i_1, 0.5,'1',fontsize=12, color="black", wrap=True)
25 #
26 #
27 #
28 left, bottom, width, height = (i_n, 0, 20, 2) #Aquí, i_n es el índice de la fecha del evento alarma/
    ↪ detención n

```

```

29 rect=mpatches.Rectangle((left,bottom),width,height,alpha=0.1,facecolor="red")
30 plt.gca().add_patch(rect)
31 plt.text(i_n, 0.5,'1',fontsize=12, color="black", wrap=True)
32
33 for alg in models:
34     alg.fit(init_data=init_data,data=data)
35     alg.initialize()
36     results=alg.run()
37     #Plot
38     figs=alg.plot(results)
39     plt.xlabel("Fecha")
40     plt.ylabel(r'$R_e$')
41     leyenda = '\n'.join((r'\$Turbina=\%.0f\$' \% (turbina, ),r'\$q=\%.4f\$' \% (proba, ),r'\$d
    ↪ =\%.0f\$' \% (depth,)))
42     plt.text(0.05, 0.98, leyenda , fontsize=10, verticalalignment='top', bbox= props)
43     plt.title("DSPOT")
44     plt.legend(loc = 'upper right')
45     plt.tight_layout()
46     plt.show()

```

Por último, para lograr una buena comprensión del gráfico generado, se muestra uno de los resultados en la Figura 4.9 y luego se detalla cada elemento de este.

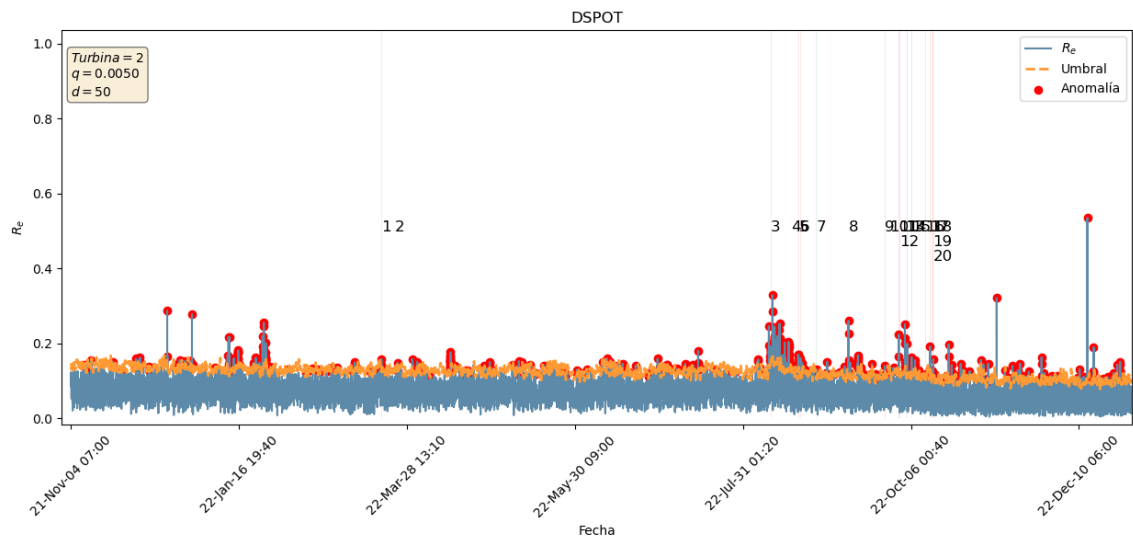
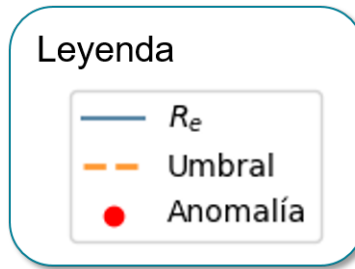


Figura 4.9: Ejemplo Gráfico R_e v/s Fecha + Umbral DSPOT

- Eje horizontal: Fechas en formato YY:MM:DD hh:mm
- Eje vertical: $R_e = \|\hat{X} - X\|_2$
- Leyenda:



4.5.1. Arquitectura Modelo Gearbox

Se construye esta red con la siguiente arquitectura, considerando que las etapas de *encoding* y *decoding* son idénticas, por lo que esta última no se especifica.

Tabla 4.14: Arquitectura Modelo Gearbox

Capa (tipo)	Cantidad de Neuronas	Función de activación
Input	6	
Dense	256	<i>sigmoid</i>
Dense	128	<i>sigmoid</i>
Dense	64	<i>sigmoid</i>
Dense	32	<i>sigmoid</i>
Dense	16	<i>sigmoid</i>
Dense	8	<i>sigmoid</i>

Preentrenamiento

Se utilizan los siguientes parámetros:

- *epochs* = 40
- *num_samples* = 1000

Ajuste fino

Una vez completado el preentrenamiento, el modelo se guarda en formato *.keras*. A este modelo se le aplica ajuste fino con los parámetros :

- **model.compile**
 - *loss* = *mean_squared_error*
 - *optimizer* = Adam(1e-5)
- **model.fit**
 - *epochs* = 40
 - *validation_split* = 0.05
 - *verbose* = 1

Con estos parámetros, el modelo logra la curva de aprendizaje que se muestra en la Figura 4.10. Se probaron otras combinaciones, las cuales se resumen en la Tabla 4.15, pero el que se detalla arriba fue que obtuvo los mejores resultados de Exactitud y Pérdida.

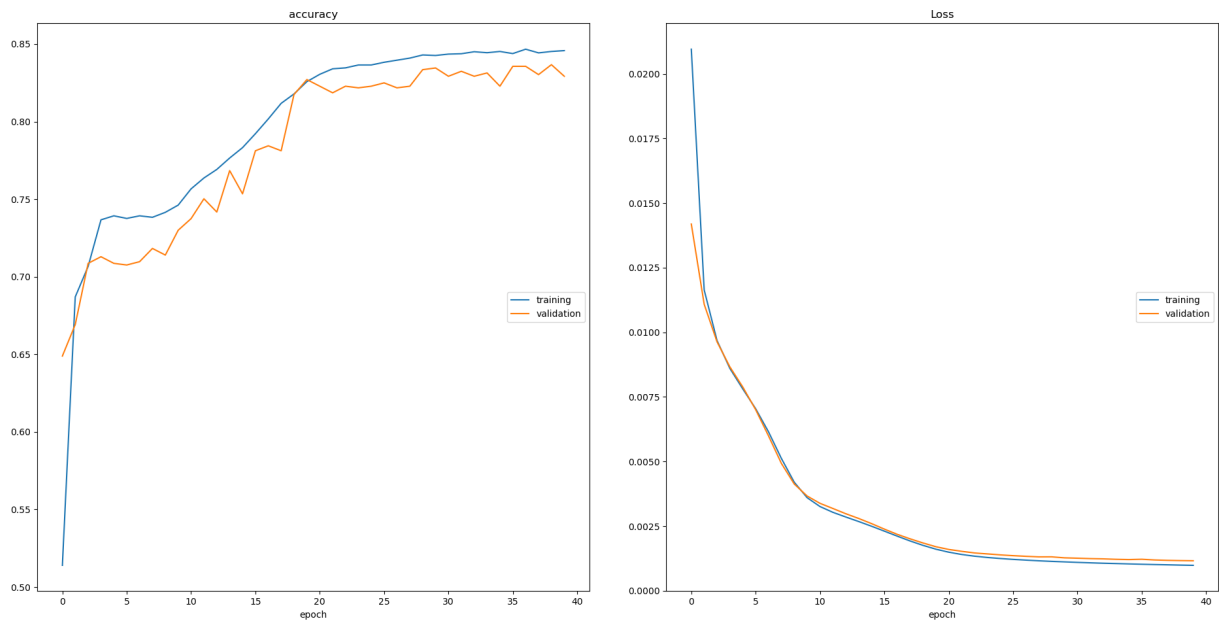


Figura 4.10: Curvas de aprendizaje del modelo Gearbox

Tabla 4.15: Combinaciones utilizadas para Modelo Gearbox

Arquitectura		Preentrenamiento		Ajuste fino	
Tamaño	Capas Ocultas Encoder/Decoder	Epochs	Nº Samples	Epochs	Optimizer
	[128,64,32,16,8]	40	1000	10	Adam()
	[128,64,32,16,8]	40	1000	10	Adam(1e-5)
	[256,128,64,32,16,8]	40	1000	10	Adam()
	[256,128,64,32,16,8]	40	1000	30	Adam(1e-5)

4.5.2. Arquitectura Modelo Grupo Hidráulico

La arquitectura de este modelo es la siguiente, recordando que las etapas de *encoding* y *decoding* son idénticas:

Tabla 4.16: Arquitectura Modelo Grupo Hidráulico

Capa (tipo)	Cantidad de Neuronas	Función de activación
Input	7	
Dense	128	<i>sigmoid</i>
Dense	64	<i>sigmoid</i>
Dense	32	<i>sigmoid</i>
Dense	16	<i>sigmoid</i>
Dense	8	<i>sigmoid</i>

Preentrenamiento

Se utilizan los siguientes parámetros:

- *epochs* = 40
- *num_samples* = 5000

Ajuste fino

Para el ajuste fino del Modelo Grupo Hidráulico se aplica lo siguiente:

- **model.compile**
 - *loss* = *mean_squared_error*
 - *optimizer* = Adam()
- **model.fit**
 - *epochs* = 8
 - *validation_split* = 0.05
 - *verbose* = 1

La curva de aprendizaje lograda se muestra en la Figura 4.11, mientras que las otras combinaciones probadas se resumen en la Tabla 4.17.

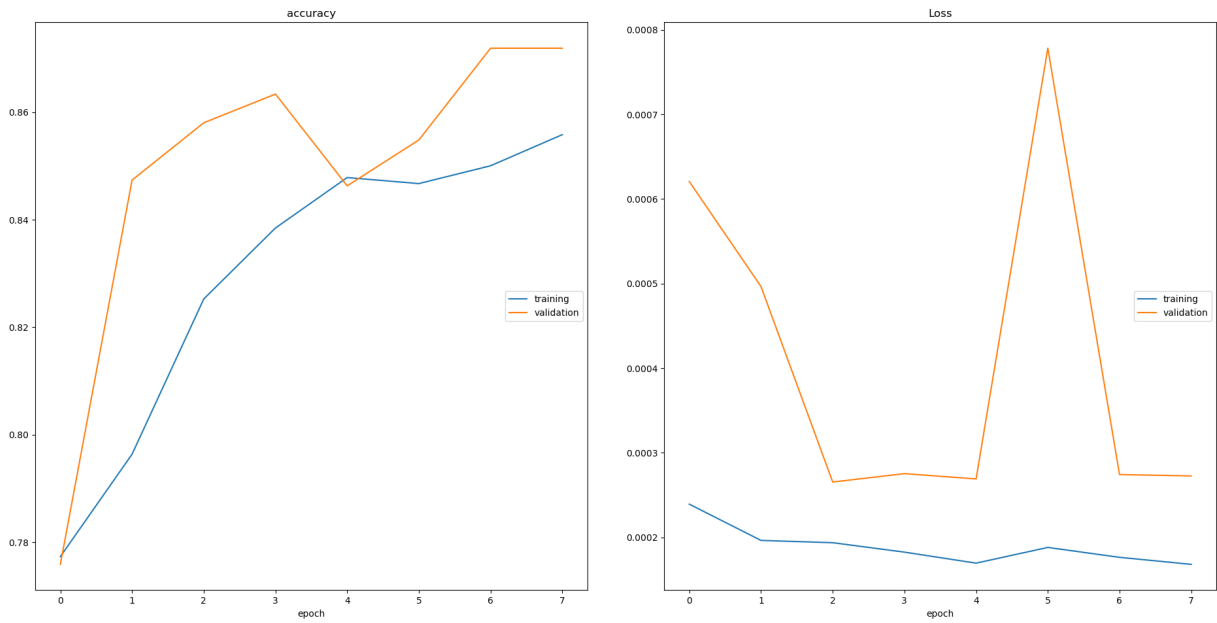


Figura 4.11: Curvas de aprendizaje del modelo Grupo Hidráulico

Tabla 4.17: Combinaciones utilizadas para entrenamiento de Modelo Grupo Hidráulico

Arquitectura Tamaño Capas Ocultas Encoder/Decoder	Preentrenamiento		Ajuste fino	
	Epochs	Nº Samples	Epochs	Optimizer
[128,64,32,16,8]	40	1000	10	Adam()
[128,64,32,16,8]	40	5000	10	Adam()
[128,64,32,16,8]	40	5000	8	Adam(1e-4)
[128,64,32,16,8]	40	5000	8	Adam(1e-5)
[512,256,128,64,32,16,8]	20	5000	10	Adam(1e-5)
[512,256,128,64,32,16,8]	20	5000	20	Adam(1e-5)
[512,256,128,64,32,16,8]	40	1000	10	Adam()
[1024, 512,256,128,64,32,16,8]	20	5000	10	Adam()

4.5.3. Arquitectura Modelo Generador

Se especifica por último la arquitectura del modelo Generador en las siguientes líneas de texto.

Tabla 4.18: Arquitectura Modelo Generador

Capa (tipo)	Cantidad de Neuronas	Función de activación
Input	10	
Dense	128	<i>sigmoid</i>
Dense	64	<i>sigmoid</i>
Dense	32	<i>sigmoid</i>
Dense	16	<i>sigmoid</i>
Dense	8	<i>sigmoid</i>

Preentrenamiento

Se utilizan los siguientes parámetros:

- *epochs* = 40
- *num_samples* = 1000

Ajuste fino

Para el ajuste fino del Modelo Grupo Hidráulico se aplica lo siguiente:

- **model.compile**
 - *loss* = *mean_squared_error*
 - *optimizer* = Adam()
- **model.fit**
 - *epochs* = 30
 - *validation_split* = 0.05
 - *verbose* = 1

El modelo Generador logra la curva de aprendizaje mostrada en la Figura 4.12. Las combinaciones de parámetros probadas se muestran en la Tabla 4.19.

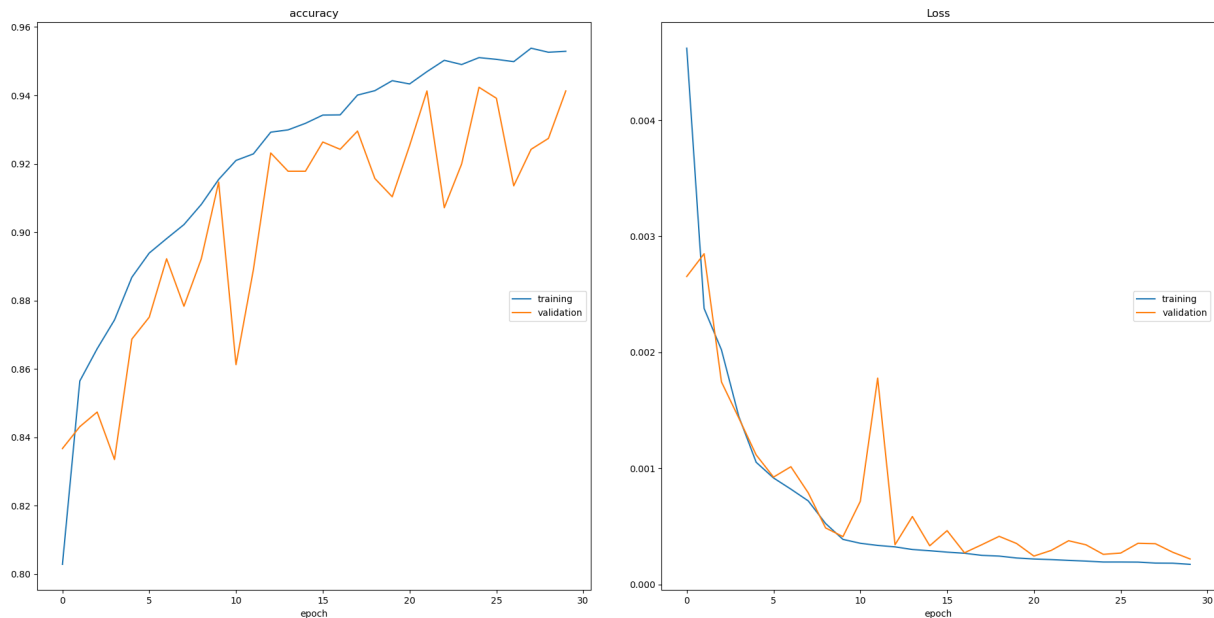


Figura 4.12: Curvas de aprendizaje del modelo Generador

Tabla 4.19: Combinaciones utilizadas para entrenamiento de Modelo Generador

Arquitectura	Preentrenamiento		Ajuste fino		
	Tamaño Capas Ocultas Encoder/Decoder	Epochs	Nº Samples	Epochs	Optimizer
	[128,64,32,16,8]	40	1000	10	Adam()
	[128,64,32,16,8]	40	1000	30	Adam()

4.5.4. Resultados Turbina N°2

Se enumeran primero las alarmas predecesoras de estado NO COMUNICA identificadas para esta turbina en la Tabla 4.20. Con este listado se busca determinar si los modelos Gearbox, Grupo Hidráulico y/o Generador son capaces de anticipar el instante donde se corta comunicación con la turbina. No se anticipa la falla en sí, sino que el momento en que el operador decide realizar un mantenimiento, lo que se puede considerar como consecuencia de la falla. El resultado considerado como anticipación exitosa es que se genere una serie de detecciones previas al estado NO COMUNICA, y no una detección puntual. Se espera que a una falla le preceda un comportamiento anormal que se mantiene en el tiempo hasta que esta se produzca, lo cual es fácilmente identificable a través de una serie de detecciones de parte de los modelos. Cuando se muestra una detección que se encuentre separada del resto por una cantidad de tiempo considerable (detección puntual), es probable que esté relacionado con una falla repentina del parque u algún otro factor externo a la turbina en sí, por lo que no debería interpretarse como un indicador de falla.

Tabla 4.20: Alarmas identificadas para Turbina N°2

Número	Alarma	Estado posterior	Fecha Inicio	Fecha Fin	Duración
1	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/03/18 07:48:30.555	2022/03/18 16:00:46.779	8:12:16
2	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/03/23 15:08:30.759	2022/03/23 16:47:27.076	1:38:56
3	823 Velocidad de emergencia incorrecta	Aerogenerador en NO COMUNICA	2022/08/10 08:40:50.868	2022/08/10 18:10:08.829	9:29:18
4	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/08/18 11:33:12.958	2022/08/18 13:33:01.408	1:59:48
5	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/08/21 09:23:02.431	2022/08/21 13:48:57.528	4:25:55
6	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/08/22 05:59:02.388	2022/08/22 18:36:01.564	12:36:59
7	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/08/28 12:40:29.288	2022/08/28 19:23:04.538	6:42:35
8	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/09/10 04:54:25.067	2022/09/10 18:08:40.688	13:14:16
	6026 Bajo voltaje de red				
	6607 Hueco de tensión				
	6420 No aparece alimentación 400 V				
	6601 Data Tracker Grabado				
9	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/09/24 10:25:50.422	2022/09/24 15:34:02.935	5:08:13
10	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/09/26 08:53:56.736	2022/09/26 11:10:21.037	2:16:24
11	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/09/29 19:08:20.489	2022/09/30 12:15:43.379	17:07:23
12	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/09/30 19:20:27.090	2022/09/30 19:39:15.001	0:18:48
13	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/10/02 20:10:43.355	2022/10/03 15:25:24.692	19:14:41
14	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/10/04 10:54:11.667	2022/10/04 15:48:12.751	4:54:01
15	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/06 05:40:22.746	2022/10/06 12:31:46.904	6:51:24
16	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/11 11:00:53.988	2022/10/11 16:40:13.366	5:39:19
17	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/13 08:41:34.855	2022/10/13 18:07:33.696	9:25:59
18	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/14 15:46:03.387	2022/10/14 17:48:13.644	2:02:10
19	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/14 19:57:17.172	2022/10/15 14:33:34.459	18:36:17
20	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/15 18:31:07.850	2022/10/16 16:34:19.837	22:03:12

Lo que se muestra a continuación es un ejemplo de detección por modelo, idealmente aquel que tenga mayor tiempo de anticipación o aquel que tenga asociada la detención de mayor duración. La totalidad de las detecciones se encuentra en el Anexo.

En primer lugar, la Figura 4.13 muestra la detección temprana de las alarmas N°1 y N°2.

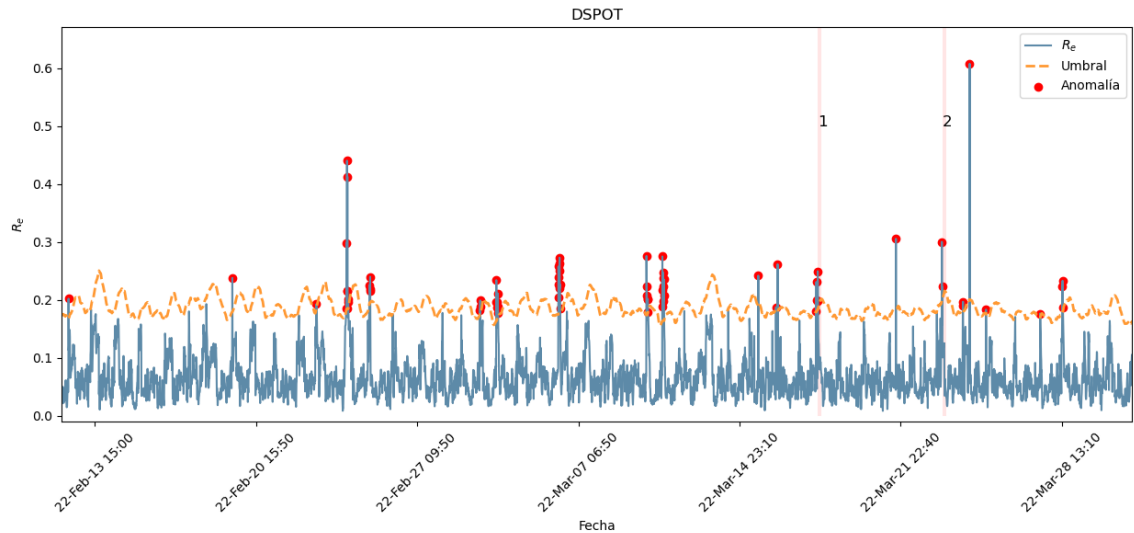


Figura 4.13: Detección de alarmas 1 y 2 por modelo Gearbox en Turbina N°2

Para la alarma N°1, se muestra una serie de detecciones que comienza en los últimos días de febrero lo que significarían varias semanas de anticipación, mientras que para la alarma N°2 las detección ocurre unos días antes, aunque podrían considerarse como parte del mismo comportamiento detectado previamente. Dicho esto, ambas alarmas podrían haberse detectado con una anticipación calculada en la Tabla 4.21.

Tabla 4.21: Anticipación de alarmas 1 y 2 de Turbina N°2 por modelo Gearbox

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
1	222 Muy baja presión grupo hidráulico	2-22-22 3:50 PM	3-18-22 7:30 PM	579:40:00
2	116 Orientando y alta presión freno del yaw	3-21-22 10:40 PM	3-23-22 3:00 PM	40:20:00

Luego, se tiene un resultado del modelo Generador, el cual es de suma importancia ya que está relacionado a un fallo de ventiladores, que está asociado a una de las alarmas de mayor tiempo acumulado en este identificadas en este trabajo. Este se muestra en la Figura 4.14

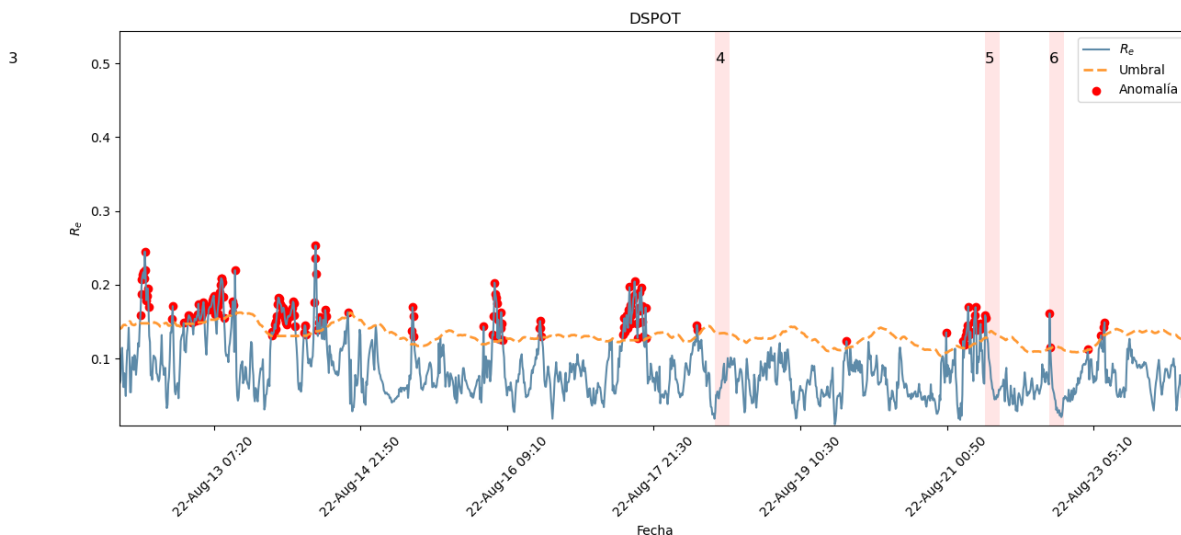


Figura 4.14: Detección de alarmas 4, 5 y 6 por modelo Generador en Turbina N°2

En este caso el comportamiento anormal es notorio y es posible detectarlo con más de 1 semana de anticipación, y además es posible asociarlo a las 3 fallas, ya que ocurren con poco tiempo de diferencia. El resultado de estas detecciones está en la Tabla 4.22.

Tabla 4.22: Anticipación de alarmas 4, 5 y 6 de Turbina N°2 por modelo Generador

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
4	548 Confirmación de fallo de ventiladores del generador.	8-9-22 8:30 AM	8-18-22 2:00 PM	221:30:00
5	547 Confirmación de fallo de ventiladores del generador.	8-9-22 8:30 AM	8-21-22 9:30 AM	289:00:00
6	548 Confirmación de fallo de ventiladores del generador.	8-9-22 8:30 AM	8-22-22 5:30 AM	309:00:00

Por último, se tiene el ejemplo de detección del modelo Grupo Hidráulico en la Figura 4.15. Este se anticipa a un conjunto de alarmas que se agrupan en el número 8. Las razones detrás de la aparición de esta alarmas pueden ser varias y la verdad es que es difícil identificar la causa, pero es claro el hecho de que existe un comportamiento fuera de lo normal previo a esta falla.

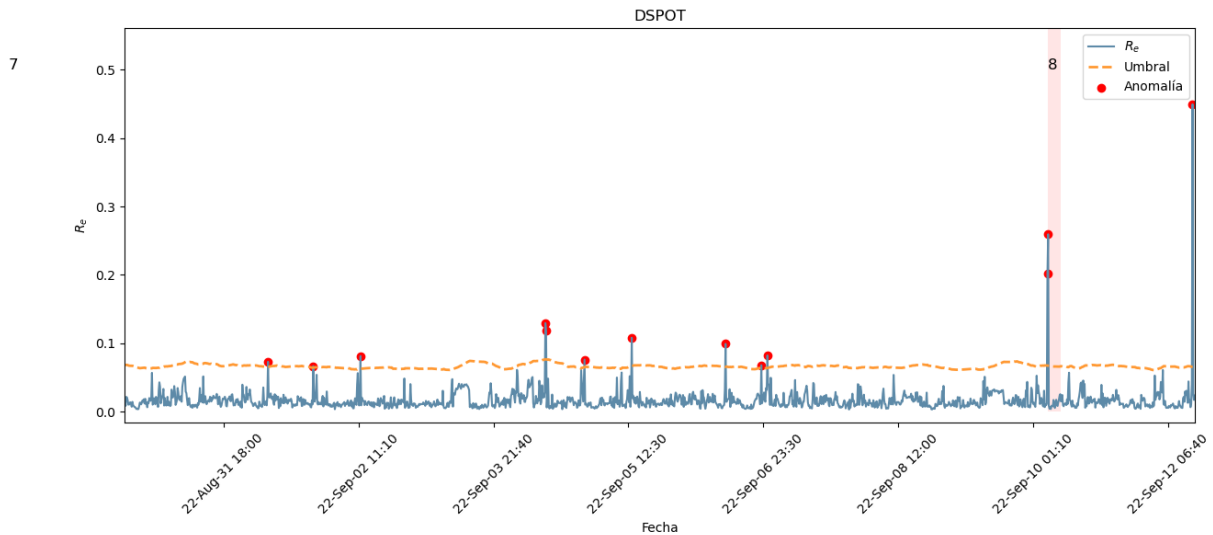


Figura 4.15: Detección de alarma 8 por modelo Grupo Hidráulico en Turbina N°2

Desde la primera detección, esta falla pudo haberse anticipado con más de una semana, como indica la Tabla 4.23

Tabla 4.23: Anticipación de alarma 8 de Turbina N°2 por modelo Grupo Hidráulico

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
8	6022 Baja Frecuencia de Red	8-31-22 6:00 PM	9-10-22 4:50 AM	226:50:00
	6026 Bajo voltaje de red			
	6607 Hueco de tensión			
	6420 No aparece alimentación 400 V			
	6601 Data Tracker Grabado			

A modo de resumen, se presentan en la Tabla 4.24 los resultados de cada uno de los modelos con respecto al listado de alarmas. En general, el trabajo cumple con detectar la totalidad de las alarmas, y si bien no todas las detecciones poseen una exactitud suficiente, con los 3 modelos en conjunto es posible cubrir en buena medida el estado de salud de la turbina.

Tabla 4.24: Resumen de resultados de los modelos para Turbina N°2

Número	Alarma	Modelo Gearbox	Modelo GH	Modelo Generador
1	222 Muy baja presión grupo hidráulico	Sí	Sí	Sí
2	116 Orientando y alta presión freno del yaw	Sí	Sí	No
3	823 Velocidad de emergencia incorrecta	Sí	No	Sí
4	548 Confirmación de fallo de ventiladores del generador.	Sí	Sí	Sí
5	547 Confirmación de fallo de ventiladores del generador.	Sí	Sí	Sí
6	548 Confirmación de fallo de ventiladores del generador.	Sí	Sí	Sí
7	548 Confirmación de fallo de ventiladores del generador.	Sí	Sí	No
	6022 Baja Frecuencia de Red			
	6026 Bajo voltaje de red			
8	6607 Hueco de tensión	No	Sí	Sí
	6420 No aparece alimentación 400 V			
	6601 Data Tracker Grabado			
9	548 Confirmación de fallo de ventiladores del generador.	Sí	No	Sí
10	548 Confirmación de fallo de ventiladores del generador.	Sí	Sí	Sí
11	222 Muy baja presión grupo hidráulico	Sí	No	Sí
12	222 Muy baja presión grupo hidráulico	Sí	No	Sí
13	222 Muy baja presión grupo hidráulico	Sí	Sí	Sí
14	222 Muy baja presión grupo hidráulico	Sí	Sí	Sí
15	548 Confirmación de fallo de ventiladores del generador.	No	Sí	Sí
16	548 Confirmación de fallo de ventiladores del generador.	Sí	Sí	Sí
17	547 Confirmación de fallo de ventiladores del generador.	Sí	Sí	Sí
18	547 Confirmación de fallo de ventiladores del generador.	Sí	Sí	Sí
19	547 Confirmación de fallo de ventiladores del generador.	Sí	Sí	Sí
20	547 Confirmación de fallo de ventiladores del generador.	Sí	Sí	Sí

4.5.5. Resultados Turbina N°9

Se debe recordar que esta turbina presenta una rotura de pala durante 2022, por lo que al estar fuera de servicio en gran parte del año no muestra una alta tasa de fallas/alarmas. Las pocas alarmas que derivan en un mantenimiento junto con la rotura de pala se muestran en la Tabla 4.25

Tabla 4.25: Alarmas identificadas para Turbina N°9

Número	Alarma	Estado posterior	Fecha Inicio	Fecha Fin	Duración
1	6026 Bajo voltaje de red	Aerogenerador en NO COMUNICA	2022/03/15 10:59:46.666	2022/03/15 17:56:11.744	6:56:25
	6607 Hueco de tensión				
	6420 No aparece alimentación 400 V				
	1887 Fallo del sistema de refrigeración del panel de potencia				
	1890 Fallo del sistema de refrigeración del panel de control				
	6601 Data Tracker Grabado				
2	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/05/05 10:57:21.261	2022/05/05 14:42:41.011	3:45:20
	6026 Bajo voltaje de red				
	6607 Hueco de tensión				
	6253 Alarma de secuencia de red de tensión				
	6420 No aparece alimentación 400 V				
	1887 Fallo del sistema de refrigeración del panel de potencia				
	1890 Fallo del sistema de refrigeración del panel de control				
	6601 Data Tracker Grabado				
3	Rotura de pala	Aerogenerador en NO COMUNICA	2022/07/26 14:09:16.493	2022/07/31 23:59:59.998	129:50:44
			2022/08/01 00:00:00.000	2022/08/31 23:59:59.998	744:00:00
			2022/09/01 00:00:00.000	2022/09/30 23:59:59.998	720:00:00
			2022/10/01 00:00:00.000	2022/10/31 23:59:59.998	744:00:00
			2022/11/01 00:00:00.000	2022/11/30 23:59:59.998	720:00:00
			2022/12/01 00:00:00.000	2022/12/31 23:59:59.998	744:00:00

A continuación se muestran resultados de los modelos Gearbox y Grupo Hidráulico para la rotura de pala. En este caso, el modelo Generador no detecta la falla, por lo que se omite de los resultados.

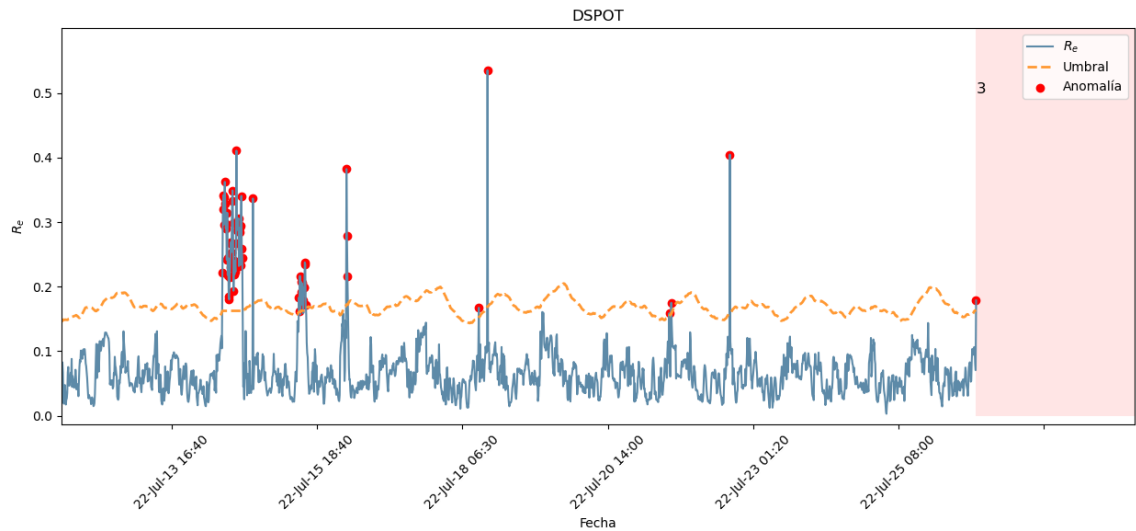


Figura 4.16: Detección de rotura de pala por modelo Gearbox en Turbina N°9

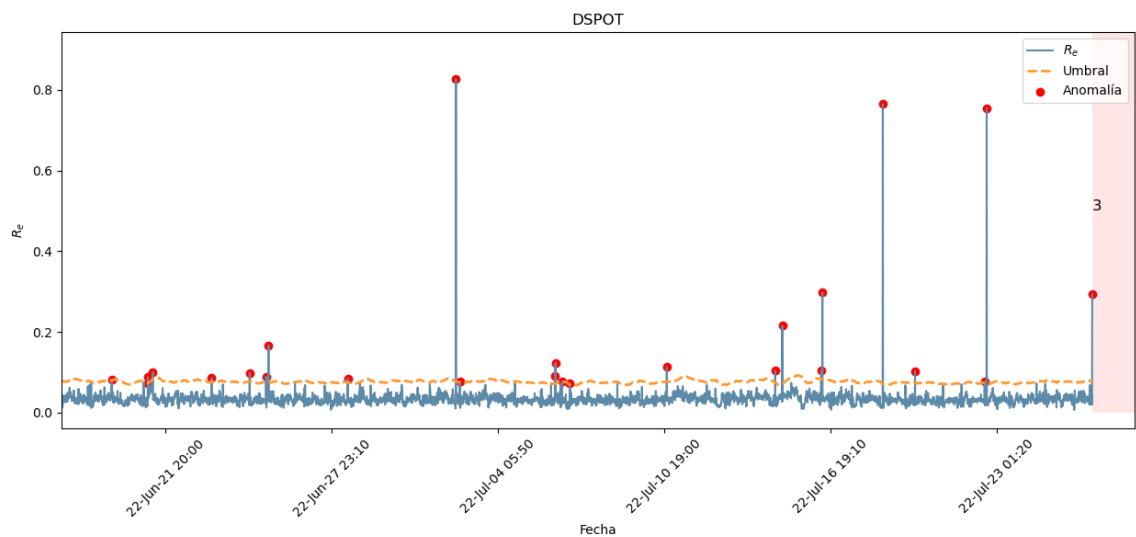


Figura 4.17: Detección de rotura de pala por modelo Grupo Hidráulico en Turbina N°9

Como es posible ver de las Figuras 4.16 y 4.17, previo al corte de comunicación por la rotura de pala, se detectan varias anomalías en el error de reconstrucción, y cualquiera podría estar relacionada a la rotura de pala, ya que no se tiene claridad sobre el momento exacto en el cual ocurrió este evento. Sin embargo, es posible descartar cierta información; la acumulación de anomalías cerca del 13 de julio en la Figura 4.16 está asociada a un fallo general de planta (de características desconocidas), y no debería considerarse como indicador de la rotura de pala.

El resultado obtenido por el modelo Grupo Hidráulico, en la Figura 4.17, muestra detecciones de todo tipo de magnitud, por lo que resulta complejo establecer cuál de estas podría corresponder al inicio de la

rotura, o si todas están relacionadas. No es claro qué tan fuerte es la relación entre las variables ocupadas en el sistema Grupo Hidráulico y el estado de las palas, pero lo que sí es seguro es que el modelo detecta que algo anda mal con la turbina. Desde la primera detección de este gráfico hasta que se detiene la operación de la turbina pasa más de 1 mes, como indica la Tabla 4.26

Tabla 4.26: Anticipación de rotura de pala por modelo Grupo Hidráulico en Turbina N°9

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
3	Rotura de pala	6-24-22 9:00 PM	7-26-22 2:00 PM	761:00:00

A modo general, los resultados de los tres modelos para las alarmas (y la rotura de pala) que derivan en mantenimiento se resumen en la Tabla 4.27

Tabla 4.27: Resumen de resultados de los modelos para Turbina N°9

Número	Alarma	Modelo Gearbox	Modelo GH	Modelo Generador
1	6026 Bajo voltaje de red	Sí	Sí	No
	6607 Hueco de tensión			
	6420 No aparece alimentación 400 V			
	1887 Fallo del sistema de refrigeración del panel de potencia			
	1890 Fallo del sistema de refrigeración del panel de control			
	6601 Data Tracker Grabado			
2	6022 Baja Frecuencia de Red	Sí	Sí	No
	6026 Bajo voltaje de red			
	6607 Hueco de tensión			
	6253 Alarma de secuencia de red de tensión			
	6420 No aparece alimentación 400 V			
	1887 Fallo del sistema de refrigeración del panel de potencia			
	1890 Fallo del sistema de refrigeración del panel de control			
6601 Data Tracker Grabado				
3	Rotura de pala	Sí	Sí	No

4.5.6. Resultados Turbina N°27

La turbina 27 se toma como caso de estudio debido a sus fallas de grupo hidráulico, siendo la alarma *205 Nivel bajo de aceite de refrigeración en grupo hidráulico* la que más afecta la disponibilidad de la turbina, como indica la Tabla 4.28. Hay una dificultad importante con respecto a la detección temprana de esta alarma, ya que entre las variables que componen el modelo Grupo Hidráulico, no hay nada que mida el nivel del aceite de refrigeración, está únicamente la presión del grupo hidráulico, por lo que la única manera de detectarla sería identificando el efecto de este bajo nivel de aceite sobre el resto de las variables.

Tabla 4.28: Alarmas identificadas para Turbina N°27

Número	Alarma	Estado posterior	Fecha Inicio	Fecha Fin	Duración
1	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/03/11 15:41:10.891	2022/03/11 17:19:56.613	1:38:46
2	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/05/20 08:24:26.480	2022/05/20 15:03:33.434	6:39:07
	6026 Bajo voltaje de red				
	6607 Hueco de tensión				
	6420 No aparece alimentación 400 V				
	6601 Data Tracker Grabado				
3	205 Nivel bajo de aceite de refrigeración en grupo hidráulico	Aerogenerador en NO COMUNICA	2022/06/04 22:34:44.842	2022/06/06 13:17:40.613	38:42:56
4	201 Máximo tiempo bombeo	Aerogenerador en NO COMUNICA	2022/07/13 12:16:50.970	2022/07/13 14:40:48.450	2:23:57
5	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/07/30 11:37:50.836	2022/07/30 19:12:26.255	7:34:35
	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/07/30 19:36:13.803	2022/07/30 19:52:46.985	0:16:33
	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/07/30 19:53:37.292	2022/07/31 10:31:29.246	14:37:52
6	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/09/10 04:53:54.968	2022/09/10 18:07:20.180	13:13:25
	6026 Bajo voltaje de red				
	6607 Hueco de tensión				
	6420 No aparece alimentación 400 V				
	6601 Data Tracker Grabado				
7	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/12/06 04:36:10.524	2022/12/06 16:35:31.946	11:59:21
8	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/12/07 09:03:04.560	2022/12/07 20:00:08.528	10:57:04

Desafortunadamente, la detección más clara de la alarma 3 es la que se muestra en la Figura 4.18 obtenida con el modelo Gearbox, que muestra unos cuantos eventos puntuales antes de que se corte la comunicación con la turbina, por lo que es complicado asegurar que el modelo ha detectado correctamente la falla. Por otro lado, el grupo de alarmas identificado con el número 2 sí es precedido por una serie de detecciones anormales en el error de reconstrucción, y es posible afirmar que el modelo sí ha detectado un comportamiento fuera de lo normal que puede estar asociado a esta detención, sin embargo el significado de las alarmas que componen este grupo es poco claro, incluso para trabajadores de Engie, por lo que no es posible identificar qué es lo que ha fallado en el sistema.

Si se considera la primera de las detecciones para cada caso por separado, el tiempo de anticipación que permitiría el modelo es el que se detalla en la Tabla 4.29, recordando que para la alarma 3 se debería tener cierta reserva con la interpretación del resultado.

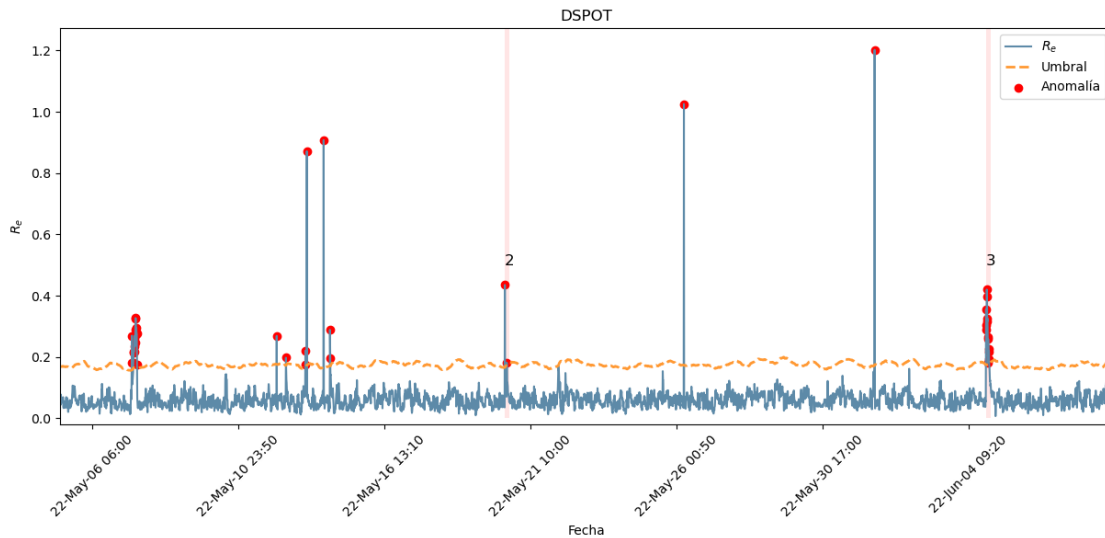


Figura 4.18: Detección de Alarmas 2 y 3 por modelo Gearbox en Turbina N°27

Tabla 4.29: Anticipación de Alarmas 2 y 3 por modelo Gearbox en Turbina N°27

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
2	6022 Baja Frecuencia de Red	5-14-22 5:30 AM	5-20-22 8:00 AM	146:30:00
	6026 Bajo voltaje de red			
	6607 Hueco de tensión			
	6420 No aparece alimentación 400 V			
	6601 Data Tracker Grabado			
3	205 Nivel bajo de aceite de refrigeración en grupo hidráulico	6-2-22 1:00 PM	6-4-22 10:30 PM	57:30:00

Luego, por parte del modelo Grupo Hidráulico se tiene la detección de la alarma *116 Orientando y alta presión freno del yaw*, identificada con el número 5 (este es un grupo de 3 detenciones que ocurren de forma casi consecutiva). El freno del yaw está accionado por actuadores hidráulicos, por lo que la relación es directa con la presión del grupo hidráulico, y resulta claro cuando se observa la Figura 4.19, donde el error de reconstrucción muestra una serie de anomalías antes de la primera de las alarmas de este grupo. El tiempo total de anticipación a estas fallas que se podría haber tenido con la primera de las detecciones se detalla en la Tabla 4.30.

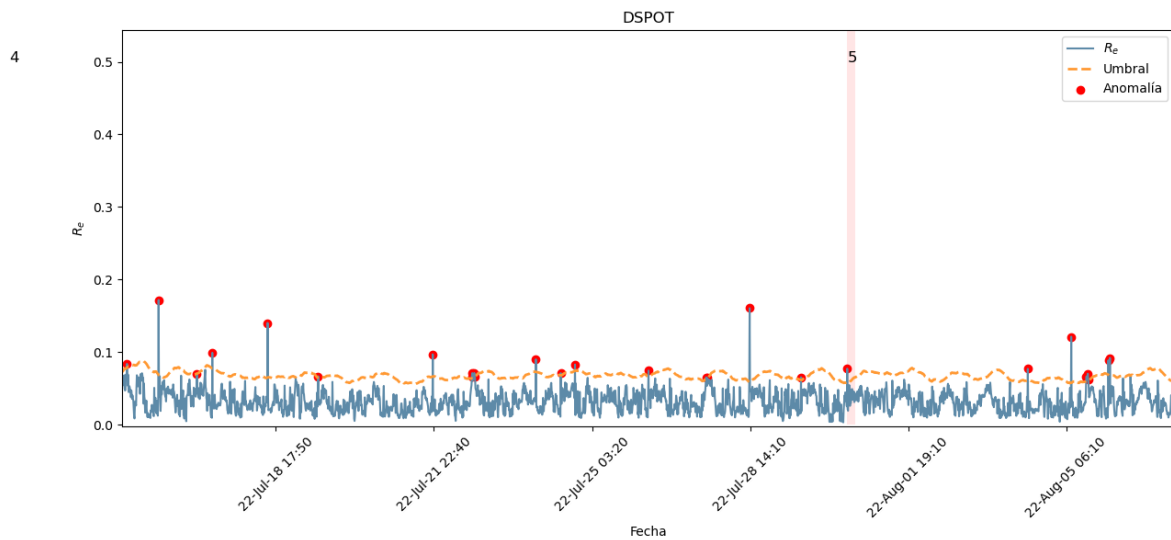


Figura 4.19: Detección de Alarma 5 por modelo Grupo Hidráulico en Turbina N°27

Tabla 4.30: Anticipación de Alarma 5 por modelo Grupo Hidráulico en Turbina N°27

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
5	116 Orientando y alta presión freno del yaw	7-18-22 12:30 PM	7-30-22 11:00 AM	286:30:00
	116 Orientando y alta presión freno del yaw	7-18-22 12:30 PM	7-30-2022 19:30	295:00:00
	116 Orientando y alta presión freno del yaw	7-18-22 12:30 PM	7-30-2022 19:30	295:00:00

Por último, el modelo Generador permite detectar con claridad las detenciones asociadas a las alarmas *222 Muy baja presión grupo hidráulico* identificadas con los números 7 y 8. Si se observa la Figura 4.20, es claro que el error de reconstrucción presenta un comportamiento anormal en el periodo de tiempo cercano a la falla, y es uno de los resultados más claros de este trabajo. De hecho, al compararlo con otros de los resultados asociados a esta misma alarma, el comportamiento del error de reconstrucción difiere bastante; mientras en otros casos las detecciones se observan en menor cantidad y con mayor separación entre sí, en este caso se encuentran más agrupadas, por lo que se puede pensar que esta falla no es del grupo hidráulico

y proviene de otro sistema de la turbina. Ya que no se tiene el registro de fallas es imposible asegurarlo, pero de todas formas es un resultado interesante. Esta falla podría haberse anticipado con el tiempo indicado en la Tabla 4.31.

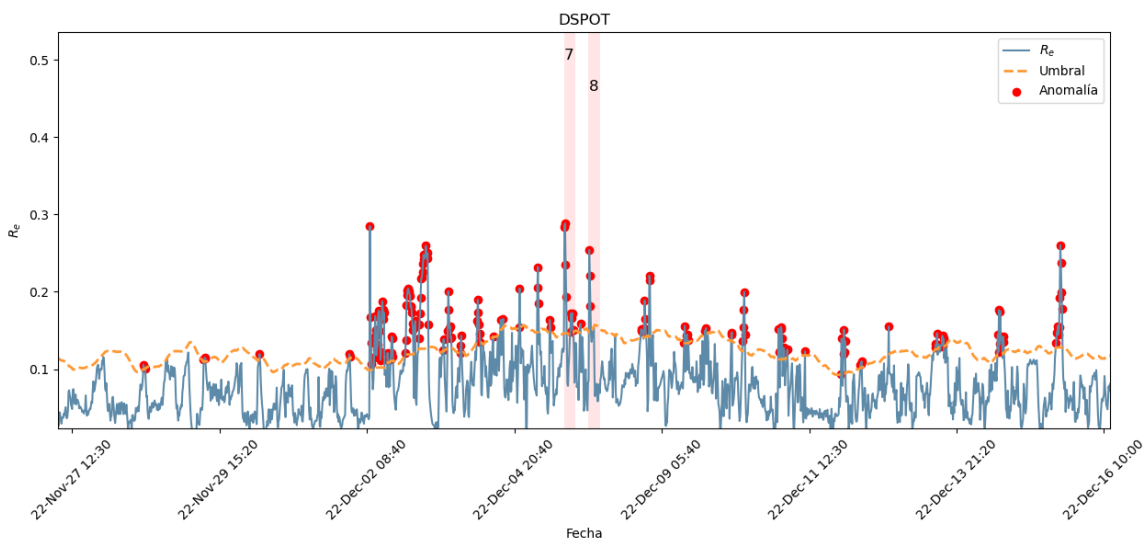


Figura 4.20: Detección de Alarmas 7 y 8 por modelo Generador en Turbina N°27

Tabla 4.31: Anticipación de Alarmas 7 y 8 por modelo Generador en Turbina N°27

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
7	222 Muy baja presión grupo hidráulico	12-2-22 8:00 PM	12-6-22 4:00 AM	80:00:00
8	222 Muy baja presión grupo hidráulico			

En resumen, como se muestra en la Tabla 4.32, todas las detenciones son anticipadas por los modelos, aunque con diferencias en la precisión y claridad de estas. El modelo Grupo Hidráulico es el que mejor rinde en esta turbina, ya que logra detectar todos los eventos, y corresponde al resultado deseado debido a que este caso de estudio se había seleccionado por las fallas de grupo hidráulico.

Tabla 4.32: Resumen de resultados de los modelos para Turbina N°27

Número	Alarma	Modelo Gearbox	Modelo GH	Modelo Generador
1	116 Orientando y alta presión freno del yaw	Sí	Sí	No
2	6022 Baja Frecuencia de Red 6026 Bajo voltaje de red 6607 Hueco de tensión 6420 No aparece alimentación 400 V 6601 Data Tracker Grabado	Sí	Sí	Sí
3	205 Nivel bajo de aceite de refrigeración en grupo hidráulico	Sí	Sí	Sí
4	201 Máximo tiempo bombeo	No	Sí	Sí
5	116 Orientando y alta presión freno del yaw 116 Orientando y alta presión freno del yaw 116 Orientando y alta presión freno del yaw	Sí Sí Sí	Sí Sí Sí	Sí Sí Sí
6	6022 Baja Frecuencia de Red 6026 Bajo voltaje de red 6607 Hueco de tensión 6420 No aparece alimentación 400 V 6601 Data Tracker Grabado	Sí	Sí	No
7	222 Muy baja presión grupo hidráulico	Sí	Sí	Sí
8	222 Muy baja presión grupo hidráulico	Sí	Sí	Sí

4.5.7. Resultados Turbina N°29

Como último caso de estudio se tiene la turbina número 29. Esta fue seleccionada por la cantidad de fallas asociadas a alarmas del sistema Gearbox, como se muestra además en la Tabla 4.33, que muestra en reiteradas ocasiones la presencia de la alarma *401 Baja presión multiplicadora*. Un punto importante a considerar en este caso es que el 22 de febrero de 2022, la turbina pasa por un cambio de gearbox por motivos que no son claros, aunque sí es claro que luego de que la turbina es puesta en servicio comienzan a emitirse las alarmas de baja presión, lo que podría deberse a que la turbina sufre un desajuste por el cambio de gearbox, el cual claramente no se soluciona ya que la alarma aparece en varias ocasiones durante 2022. Aparentemente, esta alteración en el funcionamiento de la gearbox no afecta en mayor medida la generación de potencia de la turbina, excepto cuando debe cortarse la comunicación por el mantenimiento.

El otro punto a considerar es que el cambio de gearbox inutiliza el modelo Gearbox en el caso de esta turbina, debido a que este fue construido teniendo como base la gearbox que poseen el resto de las turbinas. Este cambio de gearbox cambia por completo el estado normal de operación del sistema Gearbox, por lo que el modelo ya no funciona, pero se verá que los otros 2 modelos logran detectar las fallas por su cuenta de forma correcta, demostrando la independencia entre cada uno de estos.

Tabla 4.33: Alarmas identificadas para Turbina N°29

Número	Alarma	Estado posterior	Fecha Inicio	Fecha Fin	Duración
1	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	26-02-2022 17:16	26-02-2022 19:54	2:38:02
	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	26-02-2022 20:02	27-02-2022 16:36	20:34:41
2	410 OGS	Aerogenerador en NO COMUNICA	2022/03/10 15:03:14.524	2022/03/10 15:41:30.476	0:38:16
3	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/03/10 17:49:45.213	2022/03/10 18:32:36.941	0:42:52
4	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/03/25 03:05:13.259	2022/03/25 10:41:49.528	7:36:36
5	440 Alarma de accionamiento de ventilador de bomba de multiplicadora	Aerogenerador en NO COMUNICA	2022/04/12 20:10:29.392	2022/04/13 15:17:15.865	19:06:46
	2116 Cambio de reset a Local				
	915 Paso Pausa-Local Reset				
6	217 Aviso de mal funcionamiento de la válvula del control de carga		2022/05/09 00:03:41.431	2022/05/09 07:46:17.496	7:42:36
7	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/05/27 00:27:21.278	2022/05/27 14:30:12.883	14:02:52
8	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/06/03 07:35:55.177	2022/06/03 12:07:16.591	4:31:21
	6026 Bajo voltaje de red				
	6607 Hueco de tensión				
	6253 Alarma de secuencia de red de tensión				
	6420 No aparece alimentación 400 V				
9	115 Alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/06/08 07:17:16.544	2022/06/08 11:28:32.855	4:11:16
10	343 Turbina funcionando con hielo	Aerogenerador en NO COMUNICA	2022/06/17 13:09:50.738	2022/06/17 14:55:37.435	1:45:47
11	722 Alarma Fallo Voleta	Aerogenerador en NO COMUNICA	2022/07/06 10:49:46.816	2022/07/06 14:14:09.592	3:24:23
12	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/09/10 04:52:55.615	2022/09/10 18:06:10.803	13:13:15
	6607 Hueco de tensión				
	6026 Bajo voltaje de red				
	6420 No aparece alimentación 400 V				
13	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/09/11 11:59:49.370	2022/09/11 21:01:08.734	9:01:19
14	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/09/23 04:51:16.918	2022/09/23 08:59:15.040	4:07:58
15	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/10/14 01:36:53.589	2022/10/14 11:12:47.070	9:35:53
16	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/10/18 05:13:07.376	2022/10/18 10:07:53.321	4:54:46
17	410 OGS	Aerogenerador en NO COMUNICA	2022/11/30 09:52:27.191	2022/11/30 17:21:22.941	7:28:56

El primero de los ejemplos viene del modelo Grupo Hidráulico, que detecta las alarmas *401 Baja presión multiplicadora* identificadas con el número 1, lo cual se puede apreciar en la Figura 4.21 donde el error de reconstrucción muestra detecciones cuya magnitud va en aumento. El tiempo con el cual se podría haber anticipado esta detención se detalla en la Tabla 4.34

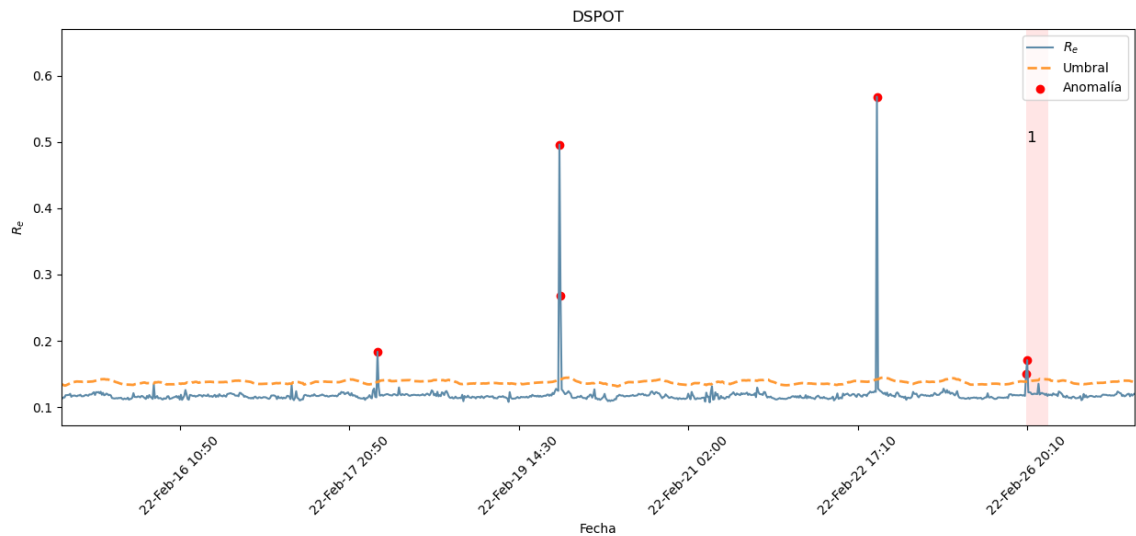


Figura 4.21: Detección de Alarma 1 por modelo Grupo Hidráulico en Turbina N°29

Tabla 4.34: Anticipación de Alarma 1 por modelo Grupo Hidráulico en Turbina N°29

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
1	401 Baja presión multiplicadora	2-17-22 8:50 PM	26-02-2022 17:16	212:26:23
	401 Baja presión multiplicadora	2-17-22 8:50 PM	26-02-2022 20:02	215:12:13

Luego se tiene como ejemplo la detección de la alarma *440 Alarma de accionamiento de ventilador de bomba de multiplicadora* por parte del modelo Generador en la Figura 4.22. Este es un ejemplo claro de éxito del modelo ya que son múltiples las veces que el error de reconstrucción sobrepasa el umbral. Considerando la primera de las detecciones, se podría haber anticipado esta detención con el tiempo indicado en la Tabla 4.35

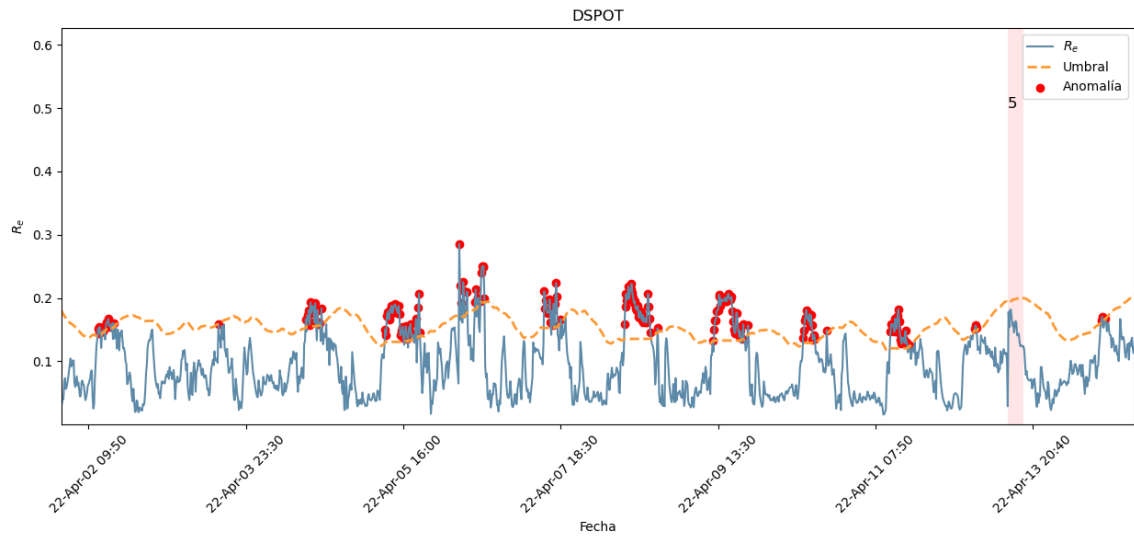


Figura 4.22: Detección de Alarma 5 por modelo Generador en Turbina N°29

Tabla 4.35: Anticipación de Alarma 5 por modelo Generador en Turbina N°29

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
5	440 Alarma de accionamiento de ventilador de bomba de multiplicadora	04-04-2022 14:30	2022/04/12 20:10:29.392	197:40:29
	2116 Cambio de reset a Local			
	915 Paso Pausa-Local Reset			

El último ejemplo viene nuevamente del modelo Grupo Hidráulico, que detecta correctamente las detenciones asociadas a las alarmas 15 y 16. La Figura 4.23 muestra pequeñas detecciones que repentinamente aumentan su magnitud antes de la detención de la turbina, por lo que para ambos casos se puede establecer que el modelo anticipó el evento. El tiempo con el que se podría haber tomado acción temprana se detalla en la Tabla 4.36.

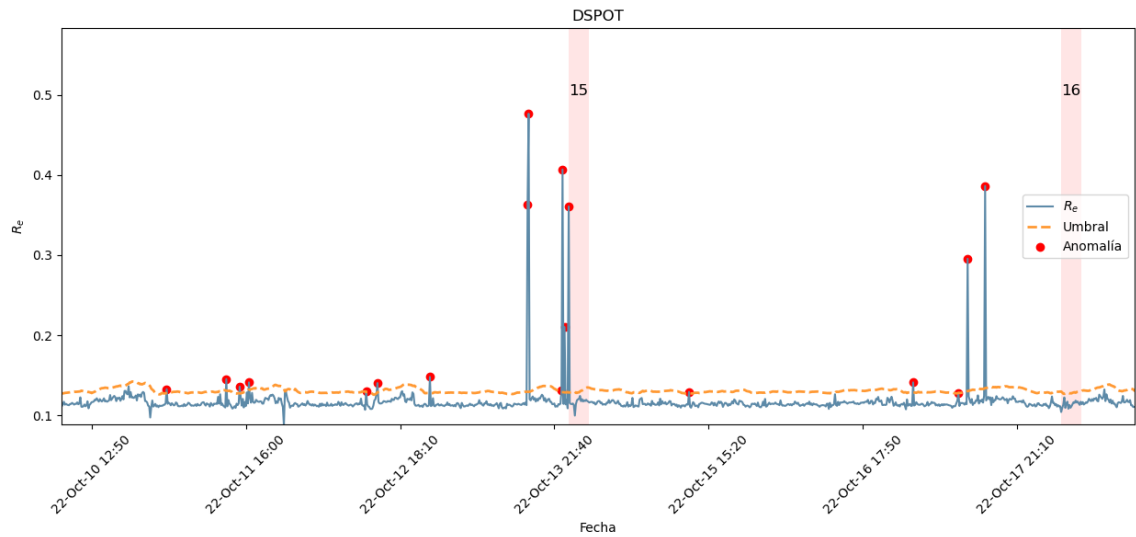


Figura 4.23: Detección de Alarmas 15 y 16 por modelo Grupo Hidráulico en Turbina N°29

Tabla 4.36: Anticipación de Alarmas 15 y 16 por modelo Grupo Hidráulico en Turbina N°29

Número	Alarma	Fecha Inicio	Fecha Fin	Tiempo total anticipación (hh:mm:ss)
15	222 Muy baja presión grupo hidráulico	11-10-2022 15:00	2022/10/14 01:36:53.589	58:36:54
16	401 Baja presión multiplicadora	15-10-2022 15:20	2022/10/18 05:13:07.376	61:53:07

Finalmente, como indica la Tabla 4.37 se anticipan todas los eventos de detención a pesar de no contar con el modelo Gearbox, por lo que se confirma la funcionalidad de los modelos, que definitivamente tienen margen de mejora ya que la precisión de las detecciones no es la misma para todos los eventos.

Tabla 4.37: Resumen de resultados de los modelos para Turbina N°29

Número	Alarma	Modelo Gearbox	Modelo GH	Modelo Generador
1	401 Baja presión multiplicadora	No aplica	Sí	Sí
	401 Baja presión multiplicadora	No aplica	Sí	No
2	410 OGS	No aplica	Sí	No
3	401 Baja presión multiplicadora	No aplica	Sí	No
4	222 Muy baja presión grupo hidráulico	No aplica	Sí	
5	440 Alarma de accionamiento de ventilador de bomba de multiplicadora	No aplica	Sí	Sí
	2116 Cambio de reset a Local			
	915 Paso Pausa-Local Reset			
6	217 Aviso de mal funcionamiento de la válvula del control de carga	No aplica	No	Sí
7	401 Baja presión multiplicadora	No aplica	Sí	Sí
8	6022 Baja Frecuencia de Red	No aplica	Sí	Sí
	6026 Bajo voltaje de red			
	6607 Hueco de tensión			
	6253 Alarma de secuencia de red de tensión			
	6420 No aparece alimentación 400 V			
9	115 Alta presión freno del yaw	No aplica	Sí	Sí
10	343 Turbina funcionando con hielo	No aplica	Sí	Sí
11	722 Alarma Fallo Veleta	No aplica	Sí	Sí
12	6022 Baja Frecuencia de Red	No aplica	Sí	No
	6607 Hueco de tensión			
	6026 Bajo voltaje de red			
	6420 No aparece alimentación 400 V			
13	401 Baja presión multiplicadora	No aplica	Sí	Sí
14	401 Baja presión multiplicadora	No aplica	Sí	Sí
15	222 Muy baja presión grupo hidráulico	No aplica	Sí	No
16	401 Baja presión multiplicadora	No aplica	Sí	No
17	410 OGS	No aplica	Sí	Sí

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

De los resultados obtenidos es posible establecer lo siguiente:

- El uso de modelos de aprendizaje de máquinas requiere una base de información que debe ser recolectada con planificación y con suficiente anticipación. Este trabajo aprovechó la presencia del sistema SCADA en el Parque Eólico para tomar información de fines de 2021 y todo 2022.
- Los algoritmos clásicos de detección de anomalías/novedades son de utilidad en análisis de vibraciones y de aceite, pero la variación aleatoria de la velocidad del viento en el caso de las turbinas eólicas restringe su rendimiento. Además, el SCADA no cuenta con datos de vibraciones ni de aceite, lo que limita más el uso de los algoritmos mencionados. Si bien es posible utilizar las variables que contiene el SCADA, el comportamiento variable de estas dificulta la capacidad de los algoritmos de encontrar relaciones entre estas y definir cuándo se encuentra en un estado de operación fuera de lo normal. Las relaciones entre estas variables de operación requieren ser modeladas por algoritmos de mayor complejidad como las redes neuronales utilizadas en este trabajo.
- Con respecto a los algoritmos de detección de novedades, los que mostraron mejores resultados fueron Elliptic Envelope y Isolation Forest, ya que no presentaron gráficos limpios y sin un exceso de novedades detectadas, a diferencia de Local Outlier Factor y OC-SVM que detectaron demasiados datos como novedosos. El problema con estos algoritmos es que no consiguieron anticiparse al momento de las detenciones y sólo identificaron el instante donde estas se producían, lo cual no habría permitido cumplir el objetivo del trabajo.
- Con respecto a los modelos basados en redes neuronales Deep Autoencoder, el rendimiento de estos mejora considerablemente al preentrenar con RBMs. Antes de aplicar preentrenamiento, se obtuvieron errores de reconstrucción de muy baja calidad. Por otro lado, para mejorar más el rendimiento de los modelos en el ajuste fino, se tuvo cuidado al determinar el número de *epochs*, seleccionando un valor que no provoque un sobreajuste al observar con detención las curvas de aprendizaje.
- Como se observa en el caso del modelo Grupo Hidráulico, aumentar el tamaño de la red no implica necesariamente una mejora en el rendimiento de los modelos. Los modelos con mayor tamaño presentaron sobreajuste. En casos como este, la mejora del rendimiento no depende de la complejidad de la red, sino que de la cantidad y calidad de los datos, la elección de la función de pérdida, el optimizador y otros hiperparámetros.

- Los modelos basados en estructura Deep Autoencoder muestran un rendimiento satisfactorio, ya que en principio detectan las detenciones (fallas) en su gran mayoría, con diferentes niveles de precisión en cuanto a la cantidad de veces que el error de reconstrucción sobrepasa el umbral y la magnitud con la cual lo hace. Estas anomalías podrían deberse a las alarmas que se presentan en el momento, así como podrían ser errores del modelo. Debido a esto, se debe pensar en realizar un análisis más exhaustivo antes de apresurarse y concluir que lo que se observa en el gráfico es una falla. No obstante, los modelos desarrollados son una buena herramienta para apoyar el monitoreo de la condición de un volumen considerable de turbinas, permitiendo identificar y acotar el número de equipos que requieren un análisis más avanzado de su condición por un humano, tanto en términos de mantenimiento como de operación.
- Una mayor disponibilidad de datos mejoraría el rendimiento de estos modelos. Se debe considerar que fueron entrenados con datos de una única turbina en un periodo de 1 año y 3 meses, por lo que tener información de varios años de operación normal robustecería la capacidad de los modelos de detectar cuándo se presenta un comportamiento anormal. Esto permitiría incluso entrenar los modelos con datos de la totalidad de las turbinas del parque eólico.
- En línea con el punto anterior, la elección de los datos de entrenamiento está limitada por la falta de un registro de fallas adecuado. No se puede elegir un conjunto de entrenamiento repleto de fallas, y tener claridad acerca del momento en el que se detecta una falla permitiría filtrar correctamente los datos que no deben formar parte del conjunto de entrenamiento para los modelos.
- Un buen registro de fallas y mantenimiento permite además identificar con mayor claridad cuáles detecciones de los modelos son por fallas y cuáles no. En varios casos los modelos detectaron anomalías que no estaban cerca de ninguna alarma, las cuales podrían ser falsos positivos o fallas repentinas que no se encontraron en el registro AWS.
- Una mayor cantidad de variables por sistema también mejoraría el rendimiento de los modelos. Sólo el modelo Generador incluyó una cantidad de variables significativa, mientras que el modelo Grupo Hidráulico, uno de los sistemas más afectados, sólo tenía disponible una variable de presión (además del resto de variables generales de la turbina).
- Independiente del tipo de resultados obtenidos, este trabajo es de relevancia para Engie, en el sentido de que durante la noche no hay personal en la planta, por lo que cualquier detección de los modelos durante las horas de día que se anticipe con un tiempo de 12 horas permitiría tomar medidas en el momento donde estén los operadores en terreno.
- Para este trabajo se desarrollan y evalúan modelos de aprendizaje *offline*. Una implementación *online* requeriría la integración con alguna plataforma informática adecuada donde sea posible ejecutar estos modelos de forma continua y que permita leer y escribir información desde y hacia las bases de datos implementadas en Engie.

5.2. Trabajo futuro

Se propone como trabajo futuro:

- Idealmente, el análisis de alarmas debería ser una tarea automatizada, debido a la gran cantidad de equipos que se requiere estudiar. En este trabajo se realizó de forma manual, por lo que se propone diseñar un código que lo efectúe de forma automática.

- Explorar otras formas de preprocesamiento y procesamiento de datos, buscando mejorar los datos de entrenamiento y de prueba de los modelos. Añadido a esto, sería deseable tener a disposición un registro adecuado de fallas, aunque eso se escapa de los alcances de este trabajo y corresponde a una gestión que debe hacer Engie.
- Ya que a la fecha ya se tiene registro del año 2023 completo, es posible entrenar los modelos Deep Autoencoder con la información entre 2021 y 2023 de las 36 turbinas, aunque esto definitivamente requerirá crear modelos con una configuración distinta.
- Lograr el diagnóstico de fallas y pronóstico de vida útil de los componentes de las turbinas sería una meta ambiciosa para este trabajo y de suma relevancia para Engie. Esta tarea está restringida por la ausencia de un registro de fallas, pero una vez este se obtenga, el trabajo sería posible.
- Probar otros tipos de Autoencoders y enfoques, como modelos basados en *Convolutional Neural Network (CNN)*, *Long Short-Term Memory (LSTM)*, *Gated Recurrent Unit (GRU)*, o *Graph Convolutional Networks (GCN)*.

Bibliografía

- [1] Walford, C., “Wind turbine reliability: understanding and minimizing wind turbine operation and maintenance costs.,” 2006, [doi:10.2172/882048](https://doi.org/10.2172/882048).
- [2] Hatch, C., “Improved wind turbine condition monitoring using acceleration enveloping,” Orbit, 2004.
- [3] Mathew, S., Wind energy: Fundamentals, resource analysis and economics. Springer Verlag, 2006.
- [4] Beri, H. y Yao, Y., “Numerical simulation of unsteady flow to show self-starting of vertical axis wind turbine using fluent,” Journal of Applied Sciences, vol. 11, no. 6, p. 962–970, 2011, [doi:10.3923/jas.2011.962.970](https://doi.org/10.3923/jas.2011.962.970).
- [5] Aslam Bhutta, M. M., Hayat, N., Farooq, A. U., Ali, Z., Jamil, S. R., y Hussain, Z., “Vertical axis wind turbine – a review of various configurations and design techniques,” Renewable and Sustainable Energy Reviews, vol. 16, no. 4, p. 1926–1939, 2012, [doi:10.1016/j.rser.2011.12.004](https://doi.org/10.1016/j.rser.2011.12.004).
- [6] Bassi, L., “Industry 4.0: Hope, hype or revolution?,” 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI), 2017, [doi:10.1109/rtsi.2017.8065927](https://doi.org/10.1109/rtsi.2017.8065927).
- [7] Factoría Electrónica, “Tecnologías habilitadoras de la industria 4.0,” 2023, <https://www.factoriadelfuturo.com/tecnologias-habilitadoras>.
- [8] Manfré, M., Creation of a Machine Learning model for the Predictive Maintenance of an engine equipped with a rotating shaft. Electronic engineering master degree, Politecnico di Torino, 2020.
- [9] Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., y Siegel, D., “Prognostics and health management design for rotary machinery systems—reviews, methodology and applications,” Mechanical Systems and Signal Processing, vol. 42, no. 1–2, p. 314–334, 2014, [doi:10.1016/j.ymssp.2013.06.004](https://doi.org/10.1016/j.ymssp.2013.06.004).
- [10] Kadry, S., “Diagnostics and prognostics of engineering systems,” IGI Global, 2012, [doi:10.4018/978-1-4666-2095-7](https://doi.org/10.4018/978-1-4666-2095-7).
- [11] Alarcón, L., Desarrollo e implementación de modelos de aprendizaje de máquinas para identificación de fallas en el monitoreo de la condición de activos en Planta Celulosa Nueva Aldea. Memoria para optar al título de ingeniero civil mecánico, Universidad de Chile, 2022.
- [12] Pannu, A., “Artificial intelligence and its application in different areas,” International Journal of Engineering and Innovative Technology, vol. 4, p. 79–84, 2015.
- [13] Raschka, S. y Mirjalili, V., Python machine learning: Aprendizaje Automático y Aprendizaje Profundo Con Python, scikit-learn Y tensorflow. Marcombo, 2019.
- [14] Goldstein, M. y Uchida, S., “A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data,” PLOS ONE, vol. 11, no. 4, 2016, [doi:10.1371/journal.pone.0152173](https://doi.org/10.1371/journal.pone.0152173).
- [15] “2.7 novelty and outlier detection.,” https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection.

lier-detection.

- [16] Kilaru, V., “One class classification using support vector machines,” 2022, <https://www.analyticsvidhya.com/blog/2022/06/one-class-classification-using-support-vector-machines/>.
- [17] Zhao, H., Liu, H., Hu, W., y Yan, X., “Anomaly detection and fault analysis of wind turbine components based on deep learning network,” *Renewable Energy*, vol. 127, p. 825–834, 2018, [doi:10.1016/j.renene.2018.05.024](https://doi.org/10.1016/j.renene.2018.05.024).
- [18] Krizhevsky, A., Sutskever, I., y Hinton, G. E., “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, p. 84–90, 2017, [doi:10.1145/3065386](https://doi.org/10.1145/3065386).
- [19] Hinton, G. E. y Salakhutdinov, R. R., “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, p. 504–507, 2006, [doi:10.1126/science.1127647](https://doi.org/10.1126/science.1127647).
- [20] Siffer, A., Fouque, P.-A., Termier, A., y Largouet, C., “Anomaly detection in streams with extreme value theory,” *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, [doi:10.1145/3097983.3098144](https://doi.org/10.1145/3097983.3098144).
- [21] Fisher, R. A. y Tippett, L. H., “Limiting forms of the frequency distribution of the largest or smallest member of a sample,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 24, no. 2, p. 180–190, 1928, [doi:10.1017/s0305004100015681](https://doi.org/10.1017/s0305004100015681).
- [22] Gnedenko, B., “Sur la distribution limite du terme maximum d’une serie aleatoire,” *The Annals of Mathematics*, vol. 44, no. 3, p. 423, 1943, [doi:10.2307/1968974](https://doi.org/10.2307/1968974).
- [23] Balkema, A. A. y de Haan, L., “Residual life time at great age,” *The Annals of Probability*, vol. 2, no. 5, 1974, [doi:10.1214/aop/1176996548](https://doi.org/10.1214/aop/1176996548).
- [24] Pickands III, J., “Statistical inference using extreme order statistics,” *The Annals of Statistics*, vol. 3, no. 1, 1975, [doi:10.1214/aos/1176343003](https://doi.org/10.1214/aos/1176343003).
- [25] Beirlant, J., Goegebeur, Y., Teugels, J., y Segers, J., “Statistics of extremes,” *Wiley Series in Probability and Statistics*, 2004, [doi:10.1002/0470012382](https://doi.org/10.1002/0470012382).
- [26] Grimshaw, S. D., “Computing maximum likelihood estimates for the generalized pareto distribution,” *Technometrics*, vol. 35, no. 2, p. 185–191, 1993, [doi:10.1080/00401706.1993.10485040](https://doi.org/10.1080/00401706.1993.10485040).
- [27] Scikit-Learn, “Sklearn.preprocessing.MinMaxScaler.”, <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>.
- [28] Lane, J., “Deep autoencoder with rbm pretraining.”, <https://github.com/akshun4/deep-autoencoder-with-RBM-pretraining/blob/master/README.md>.
- [29] Chollet, F. *et al.*, “Keras.” <https://keras.io>, 2015.
- [30] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., y Zheng, X., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, <https://www.tensorflow.org/>. Software available from tensorflow.org.
- [31] limjct, “Anomaly detection in streams with extreme value theory.”, <https://github.com/limjct/ads-evt>

[/blob/main/README.md.](#)

Anexos

Anexo A. Resultados Turbina N°2

Tabla A.1: Alarmas identificadas para Turbina N°2

Número	Alarna	Estado posterior	Fecha Inicio	Fecha Fin	Duración
1	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/03/18 07:48:30.555	2022/03/18 16:00:46.779	8:12:16
2	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/03/23 15:08:30.759	2022/03/23 16:47:27.076	1:38:56
3	823 Velocidad de emergencia incorrecta	Aerogenerador en NO COMUNICA	2022/08/10 08:40:50.868	2022/08/10 18:10:08.829	9:29:18
4	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/08/18 11:33:12.958	2022/08/18 13:33:01.408	1:59:48
5	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/08/21 09:23:02.431	2022/08/21 13:48:57.528	4:25:55
6	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/08/22 05:59:02.388	2022/08/22 18:36:01.564	12:36:59
7	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/08/28 12:40:29.288	2022/08/28 19:23:04.538	6:42:35
8	6022 Baja Frecuencia de Red 6026 Bajo voltaje de red 6607 Hueco de tensión 6420 No aparece alimentación 400 V 6601 Data Tracker Grabado	Aerogenerador en NO COMUNICA	2022/09/10 04:54:25.067	2022/09/10 18:08:40.688	13:14:16
9	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/09/24 10:25:50.422	2022/09/24 15:34:02.935	5:08:13
10	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/09/26 08:53:56.736	2022/09/26 11:10:21.037	2:16:24
11	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/09/29 19:08:20.489	2022/09/30 12:15:43.379	17:07:23
12	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/09/30 19:20:27.090	2022/09/30 19:39:15.001	0:18:48
13	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/10/02 20:10:43.355	2022/10/03 15:25:24.692	19:14:41
14	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/10/04 10:54:11.667	2022/10/04 15:48:12.751	4:54:01
15	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/06 05:40:22.746	2022/10/06 12:31:46.904	6:51:24
16	548 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/11 11:00:53.988	2022/10/11 16:40:13.366	5:39:19
17	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/13 08:41:34.855	2022/10/13 18:07:33.696	9:25:59
18	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/14 15:46:03.387	2022/10/14 17:48:13.644	2:02:10
19	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/14 19:57:17.172	2022/10/15 14:33:34.459	18:36:17
20	547 Confirmación de fallo de ventiladores del generador.	Aerogenerador en NO COMUNICA	2022/10/15 18:31:07.850	2022/10/16 16:34:19.837	22:03:12

A.1. Modelo Gearbox

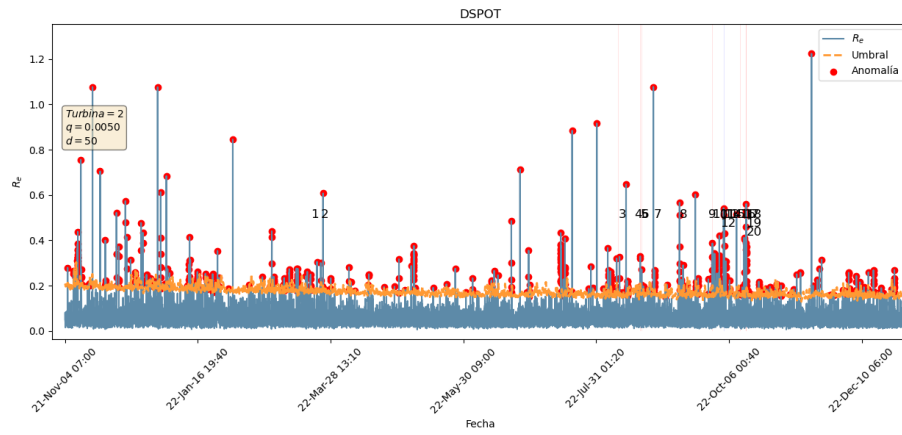
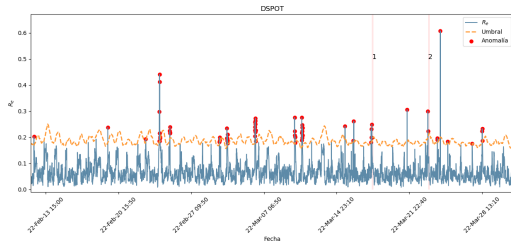
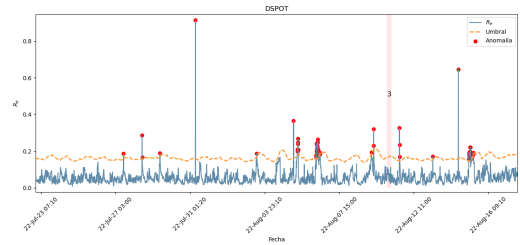


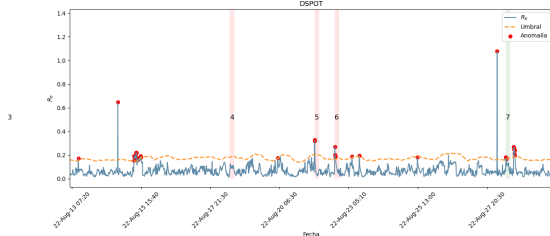
Figura A.1: Resultado General Modelo Gearbox en Turbina N°2



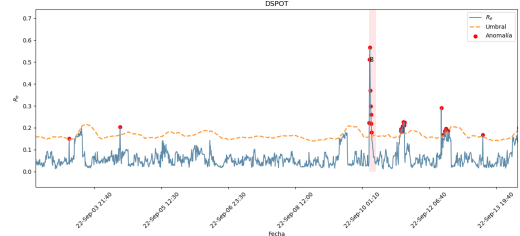
(a) Detección de Alarmas 1 y 2 por modelo Gearbox en Turbina N°2



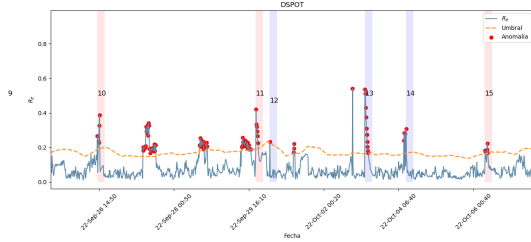
(b) Detección de Alarma 3 por modelo Gearbox en Turbina N°2



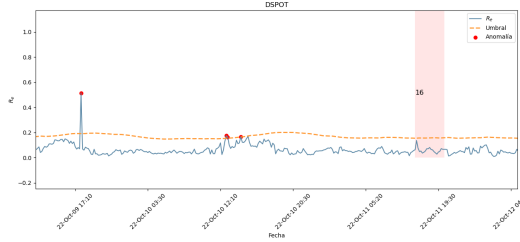
(c) Detección de Alarmas 4, 5, 6 y 7 por modelo Gearbox en Turbina N°2



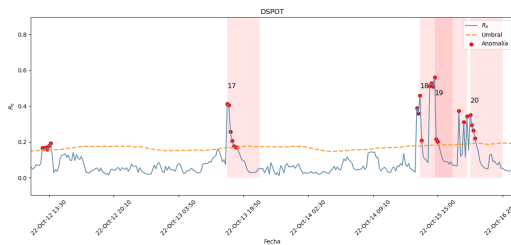
(d) Detección de Alarma 8 por modelo Gearbox en Turbina N°2



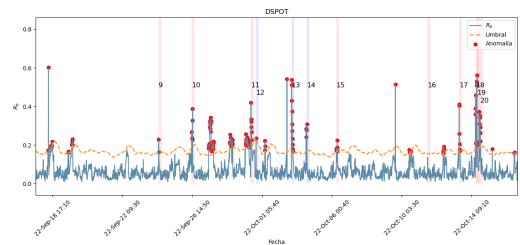
(e) Detección de Alarmas 10 a 15 por modelo Gearbox en Turbina N°2



(f) Detección de Alarma 16 por modelo Gearbox en Turbina N°2



(g) Detección de Alarmas 17, 18, 19 y 20 por modelo Gearbox en Turbina N°2



(h) Detección de Alarmas 9 a 20 por modelo Gearbox en Turbina N°2

Figura A.2: Resultados específicos de Alarmas Modelo Gearbox en Turbina N°2

A.2. Modelo Grupo Hidráulico

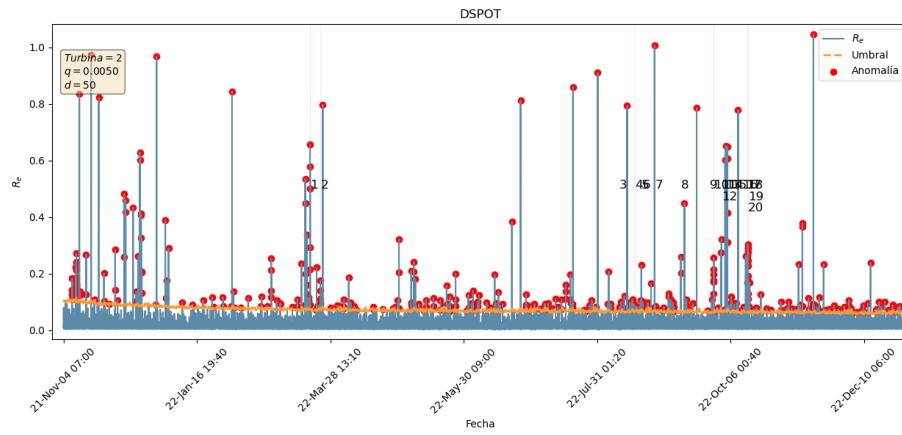
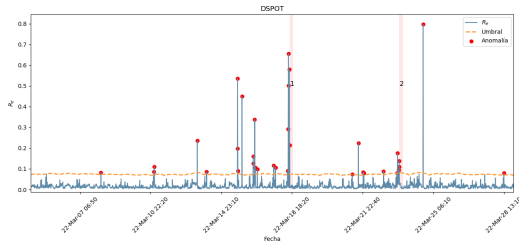
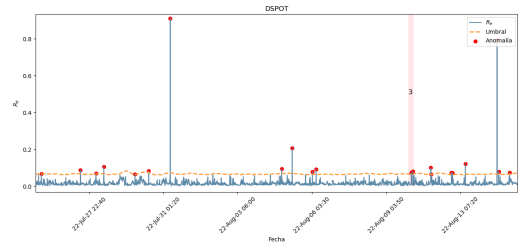


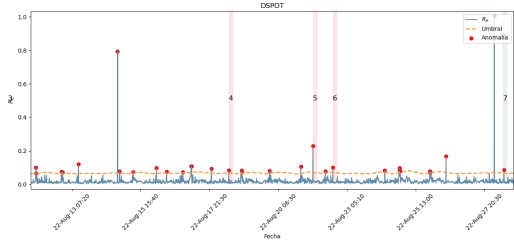
Figura A.3: Resultado General Modelo Grupo Hidráulico en Turbina N°2



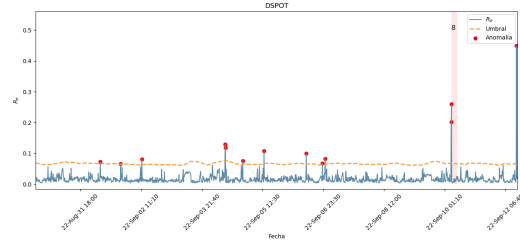
(a) Detección de Alarmas 1 y 2 por modelo Grupo Hidráulico en Turbina N°2



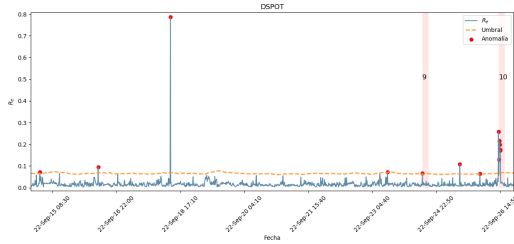
(b) Detección de Alarma 3 por modelo Grupo Hidráulico en Turbina N°2



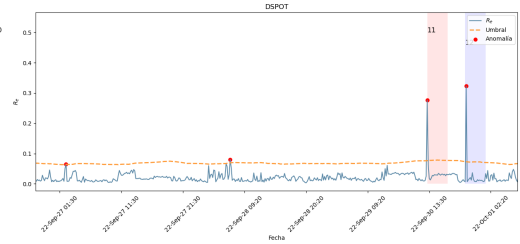
(c) Detección de Alarmas 4, 5, 6 y 7 por modelo Grupo Hidráulico en Turbina N°2



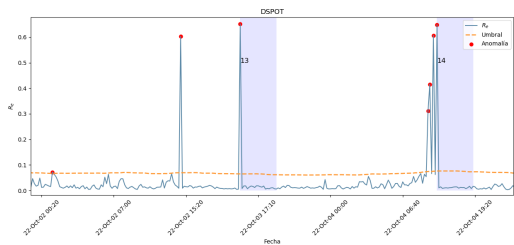
(d) Detección de Alarma 8 por modelo Grupo Hidráulico en Turbina N°2



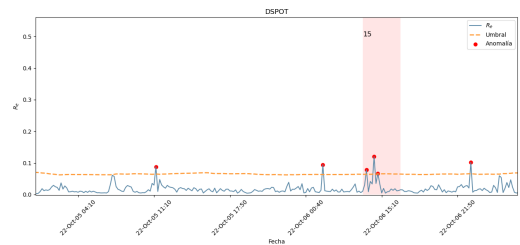
(e) Detección de Alarmas 9 y 10 por modelo Grupo Hidráulico en Turbina N°2



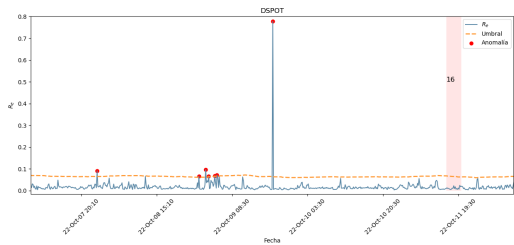
(f) Detección de Alarma 11 y 12 por modelo Grupo Hidráulico en Turbina N°2



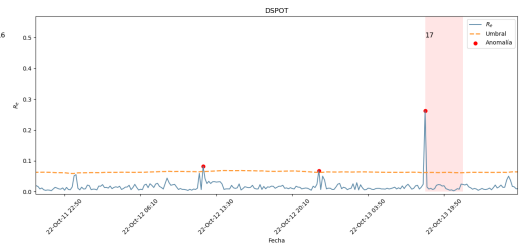
(g) Detección de Alarmas 13 y 14 por modelo Grupo Hidráulico en Turbina N°2



(h) Detección de Alarma 15 por modelo Grupo Hidráulico en Turbina N°2

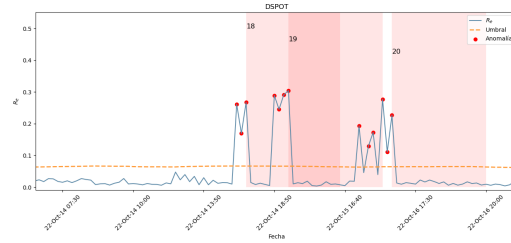


(i) Detección de Alarma 16 por modelo Grupo Hidráulico en Turbina N°2



(j) Detección de Alarma 17 por modelo Grupo Hidráulico en Turbina N°2

Figura A.4: Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°2



(k) Detección de Alarma 18, 19 y 20 por modelo Grupo Hidráulico en Turbina N°2

Figura A.4: Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°2

A.3. Modelo Generador

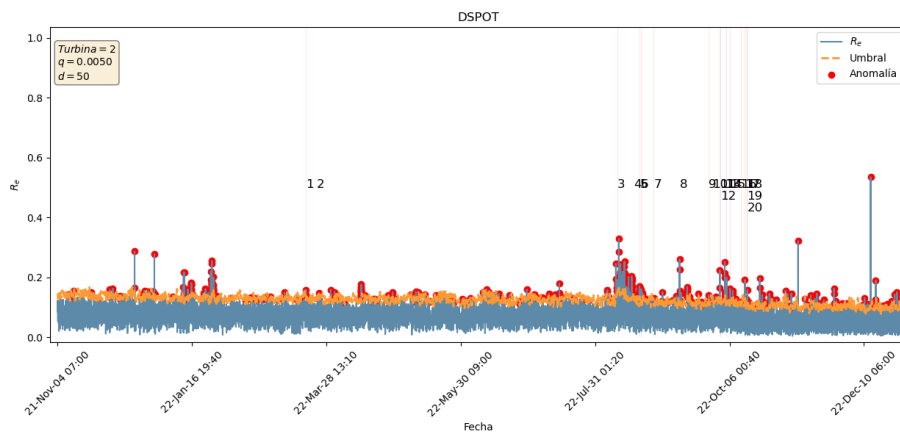
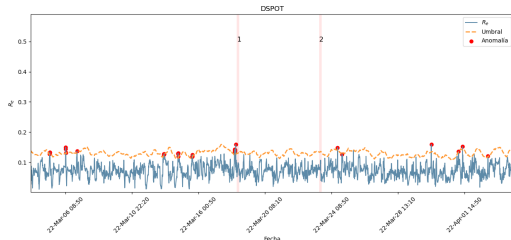
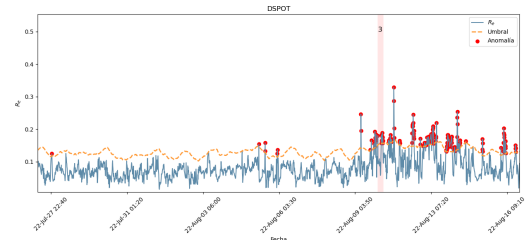


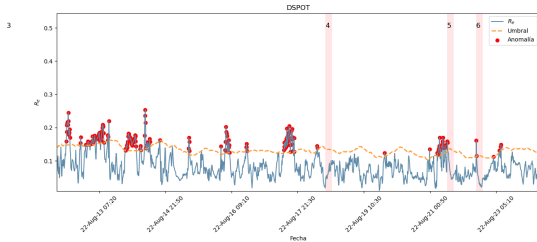
Figura A.5: Resultado General Modelo Generador en Turbina N°2



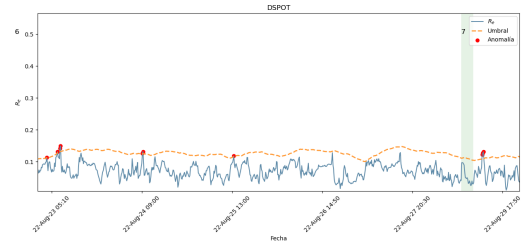
(a) Detección de Alarmas 1 y 2 por modelo Generador en Turbina N°2



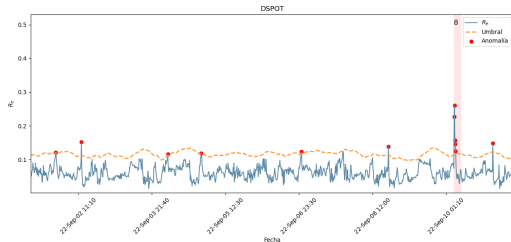
(b) Detección de Alarma 3 por modelo Generador en Turbina N°2



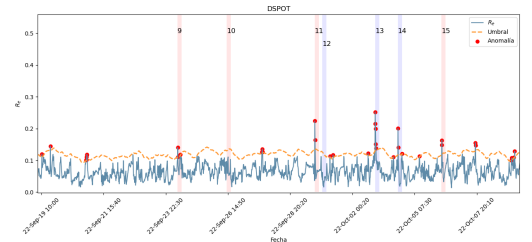
(c) Detección de Alarmas 4, 5 y 6 por modelo Generador en Turbina N°2



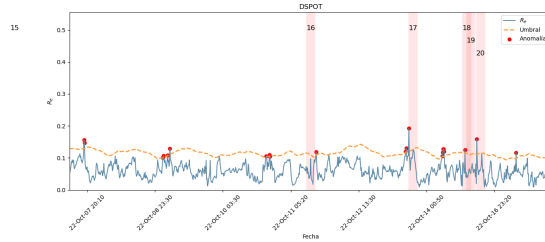
(d) Detección de Alarma 7 por modelo Generador en Turbina N°2



(e) Detección de Alarma 8 por modelo Generador en Turbina N°2



(f) Detección de Alarmas 9 a 15 por modelo Generador en Turbina N°2



(g) Detección de Alarmas 16 a 20 por modelo Grupo Hidráulico en Turbina N°2

Figura A.6: Resultados específicos de Alarmas Modelo Generador en Turbina N°2

Anexo B. Resultados Turbina N°9

Tabla B.1: Alarmas identificadas para Turbina N°9

Número	Alarma	Estado posterior	Fecha Inicio	Fecha Fin	Duración
1	6026 Bajo voltaje de red	Aerogenerador en NO COMUNICA	2022/03/15 10:59:46.666	2022/03/15 17:56:11.744	6:56:25
	6607 Hueco de tensión				
	6420 No aparece alimentación 400 V				
	1887 Fallo del sistema de refrigeración del panel de potencia				
	1890 Fallo del sistema de refrigeración del panel de control				
6601 Data Tracker Grabado					
2	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/05/05 10:57:21.261	2022/05/05 14:42:41.011	3:45:20
	6026 Bajo voltaje de red				
	6607 Hueco de tensión				
	6253 Alarma de secuencia de red de tensión				
	6420 No aparece alimentación 400 V				
1887 Fallo del sistema de refrigeración del panel de potencia					
1890 Fallo del sistema de refrigeración del panel de control					
6601 Data Tracker Grabado					
3	Rotura de pala	Aerogenerador en NO COMUNICA	2022/07/26 14:09:16.493	2022/07/31 23:59:59.998	129:50:44
			2022/08/01 00:00:00.000	2022/08/31 23:59:59.998	744:00:00
			2022/09/01 00:00:00.000	2022/09/30 23:59:59.998	720:00:00
			2022/10/01 00:00:00.000	2022/10/31 23:59:59.998	744:00:00
			2022/11/01 00:00:00.000	2022/11/30 23:59:59.998	720:00:00
			2022/12/01 00:00:00.000	2022/12/31 23:59:59.998	744:00:00

B.1. Modelo Gearbox

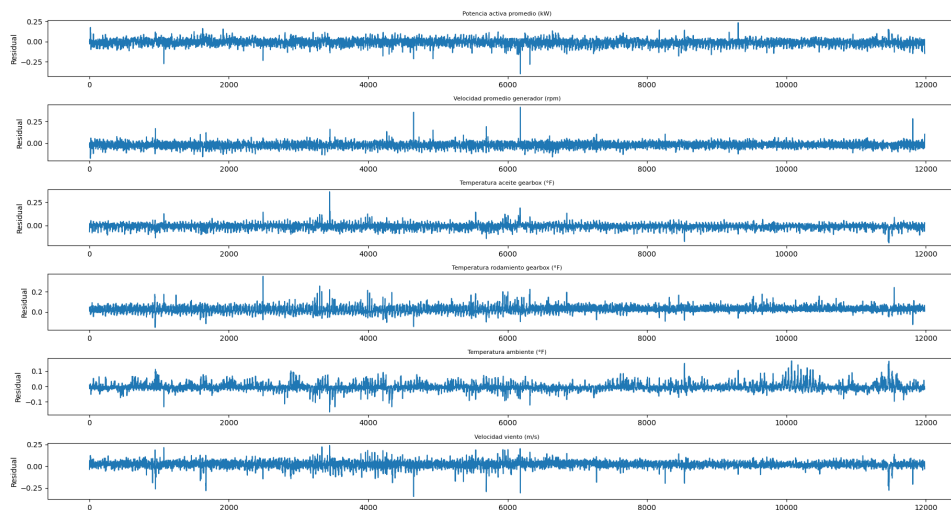


Figura B.1: Residuales Modelo Gearbox en Turbina N°9

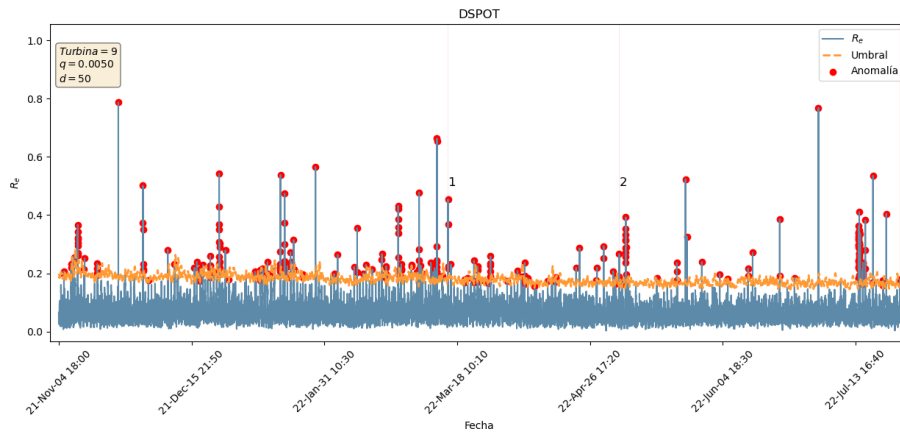
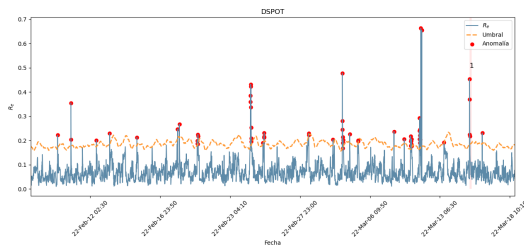
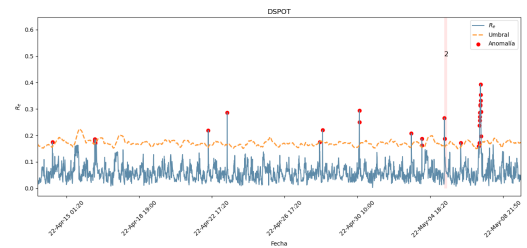


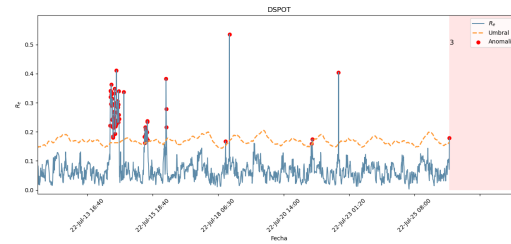
Figura B.2: Resultado General Modelo Gearbox en Turbina N°9



(a) Detección de Alarma 1 por modelo Gearbox en Turbina N°9



(b) Detección de Alarma 2 por modelo Gearbox en Turbina N°9



(c) Detección de Rotura de Pala por modelo Gearbox en Turbina N°9

Figura B.3: Resultados específicos de Alarmas Modelo Gearbox en Turbina N°9

B.2. Modelo Grupo Hidráulico

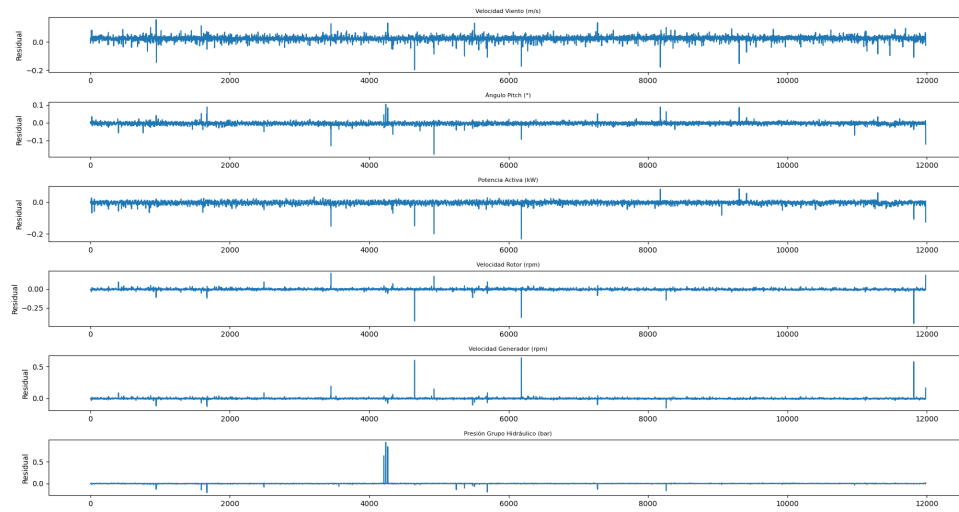


Figura B.4: Residuales Modelo Grupo Hidráulico en Turbina N°9

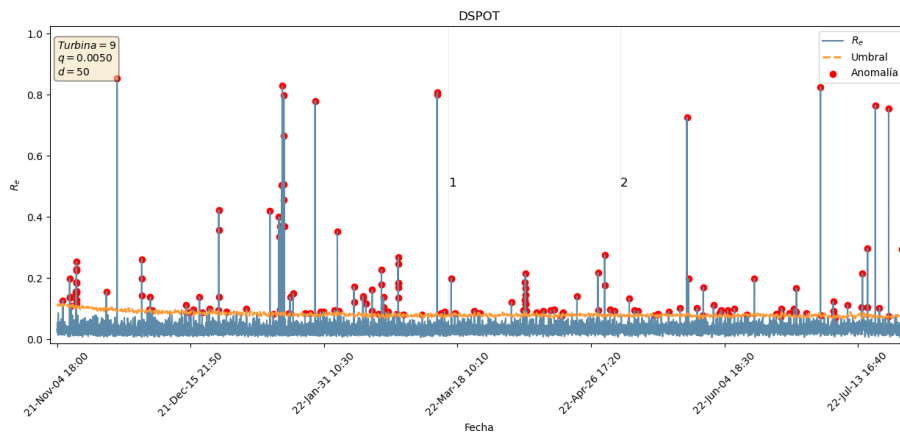
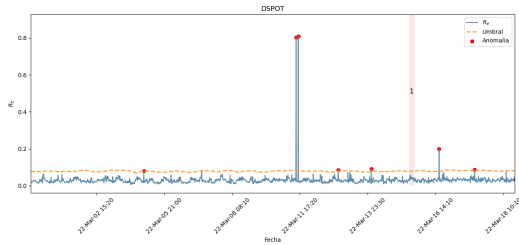
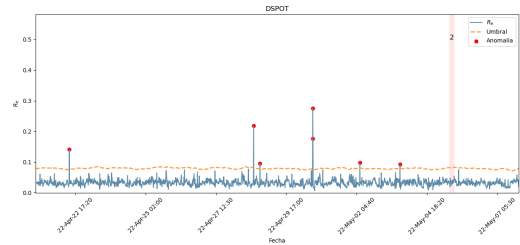


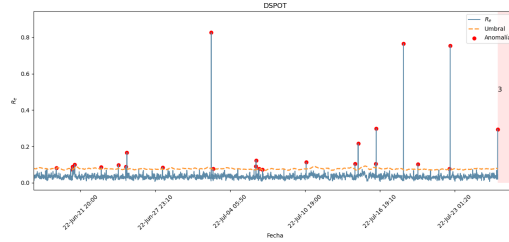
Figura B.5: Resultado General Modelo Grupo Hidráulico en Turbina N°9



(a) Detección de Alarma 1 por modelo Grupo Hidráulico en Turbina N°9



(b) Detección de Alarma 2 por modelo Grupo Hidráulico en Turbina N°9



(c) Detección de Rotura de Pala por modelo Grupo Hidráulico en Turbina N°9

Figura B.6: Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°9

B.3. Modelo Generador

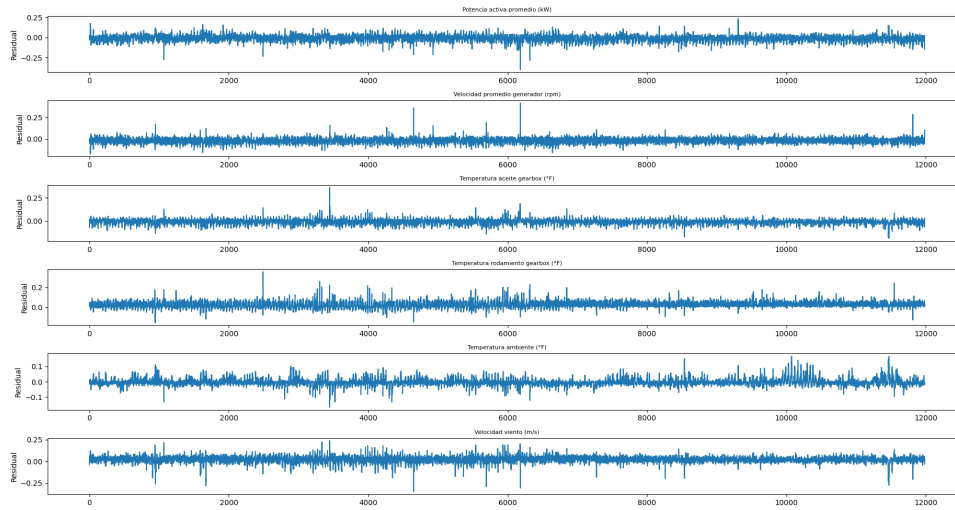


Figura B.7: Residuales Modelo Generador en Turbina N°9

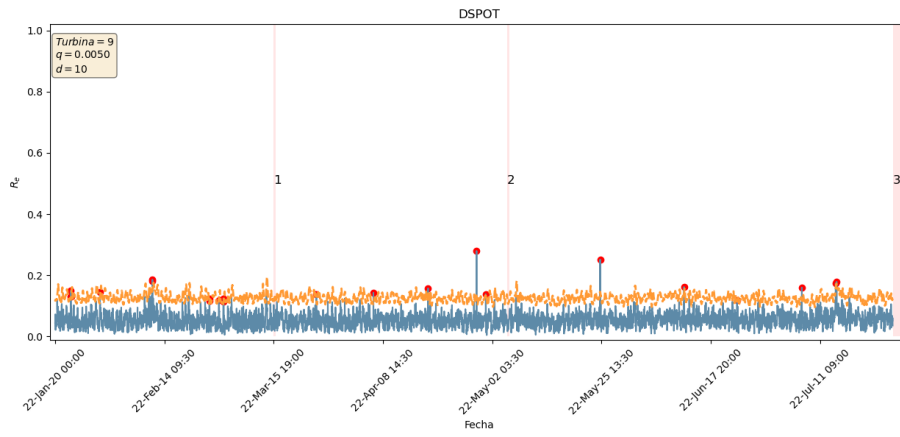
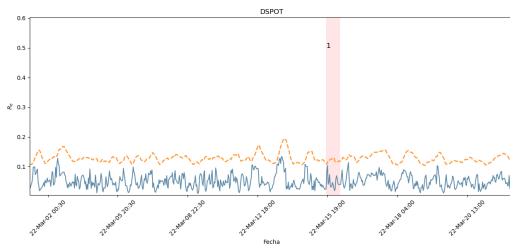
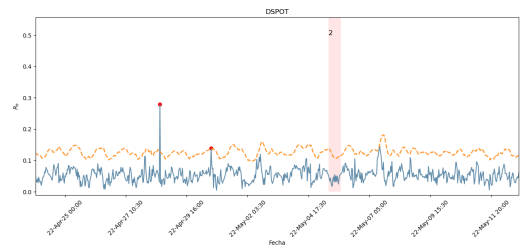


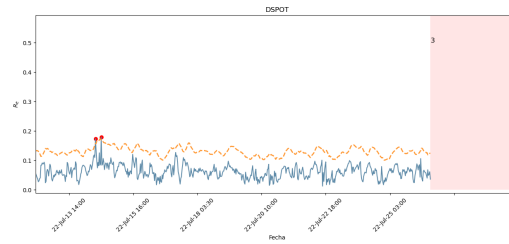
Figura B.8: Resultado General Modelo Generador en Turbina N°9



(a) Detección de Alarma 1 por modelo Generador en Turbina N°9



(b) Detección de Alarma 2 por modelo Generador en Turbina N°9



(c) Detección de Rotura de Pala por modelo Generador en Turbina N°9

Figura B.9: Resultados específicos de Alarmas Modelo Generador en Turbina N°9

Anexo C. Resultados Turbina N°27

Tabla C.1: Alarmas identificadas para Turbina N°27

Número	Alarma	Estado posterior	Fecha Inicio	Fecha Fin	Duración
1	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/03/11 15:41:10.891	2022/03/11 17:19:56.613	1:38:46
2	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/05/20 08:24:26.480	2022/05/20 15:03:33.434	6:39:07
	6026 Bajo voltaje de red				
	6607 Hueco de tensión				
	6420 No aparece alimentación 400 V				
	6601 Data Tracker Grabado				
3	205 Nivel bajo de aceite de refrigeración en grupo hidráulico	Aerogenerador en NO COMUNICA	2022/06/04 22:34:44.842	2022/06/06 13:17:40.613	38:42:56
4	201 Máximo tiempo bombeo	Aerogenerador en NO COMUNICA	2022/07/13 12:16:50.970	2022/07/13 14:40:48.450	2:23:57
5	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/07/30 11:37:50.836	2022/07/30 19:12:26.255	7:34:35
	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/07/30 19:36:13.803	2022/07/30 19:52:46.985	0:16:33
	116 Orientando y alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/07/30 19:53:37.292	2022/07/31 10:31:29.246	14:37:52
6	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/09/10 04:53:54.968	2022/09/10 18:07:20.180	13:13:25
	6026 Bajo voltaje de red				
	6607 Hueco de tensión				
	6420 No aparece alimentación 400 V				
	6601 Data Tracker Grabado				
7	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/12/06 04:36:10.524	2022/12/06 16:35:31.946	11:59:21
8	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/12/07 09:03:04.560	2022/12/07 20:00:08.528	10:57:04

C.1. Modelo Gearbox

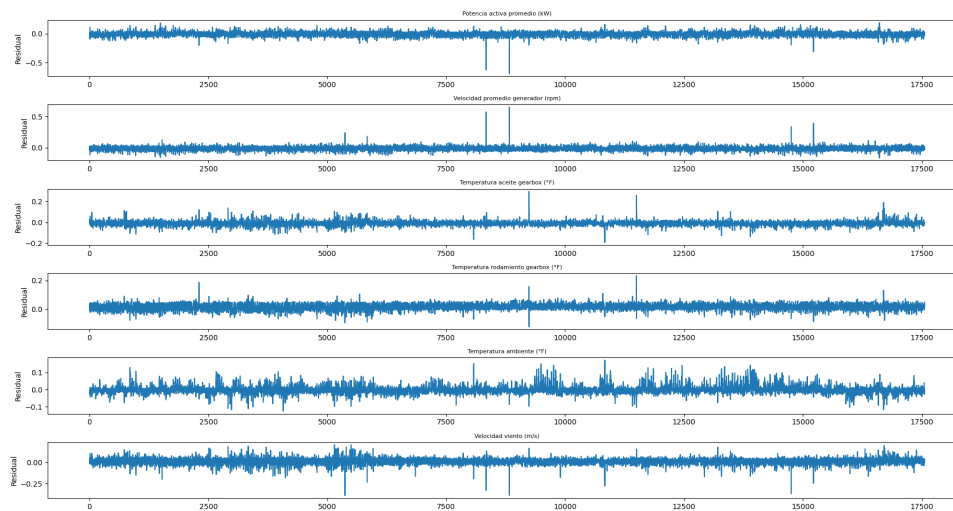


Figura C.1: Residuales General Modelo Gearbox en Turbina N°27

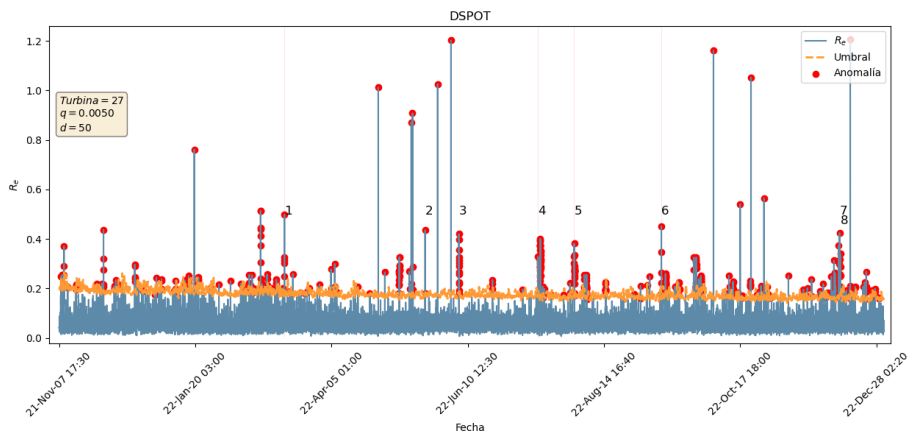
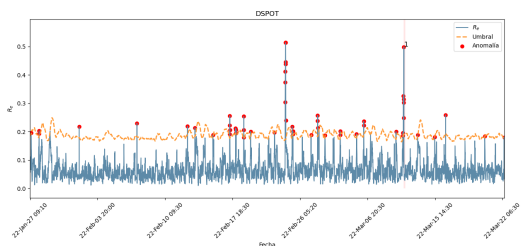
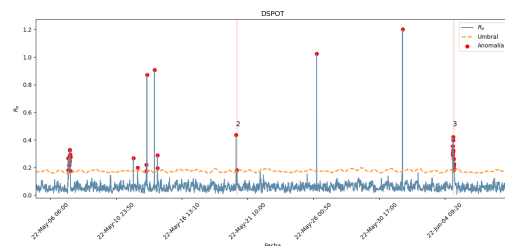


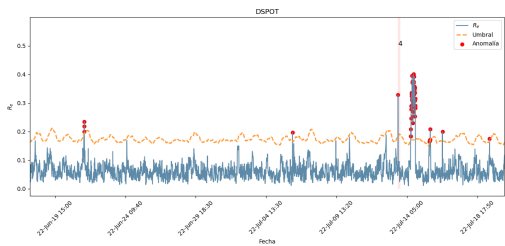
Figura C.2: Resultado General Modelo Gearbox en Turbina N°27



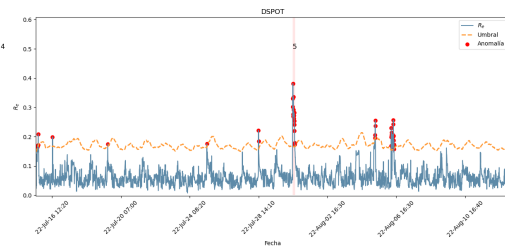
(a) Detección de Alarma 1 por modelo Gearbox en Turbina N°27



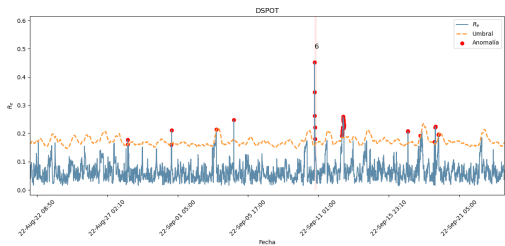
(b) Detección de Alarmas 2 y 3 por modelo Gearbox en Turbina N°27



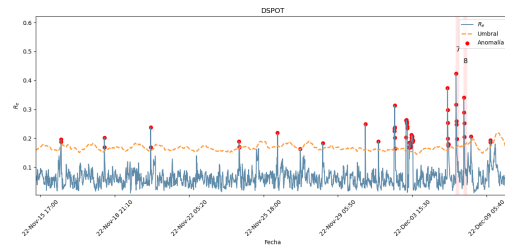
(c) Detección de Alarma 4 por modelo Gearbox en Turbina N°27



(d) Detección de Alarma 5 por modelo Gearbox en Turbina N°27



(e) Detección de Alarma 6 por modelo Gearbox en Turbina N°27



(f) Detección de Alarma 7 y 8 por modelo Gearbox en Turbina N°27

Figura C.3: Resultados específicos de Alarmas Modelo Gearbox en Turbina N°27

C.2. Modelo Grupo Hidráulico

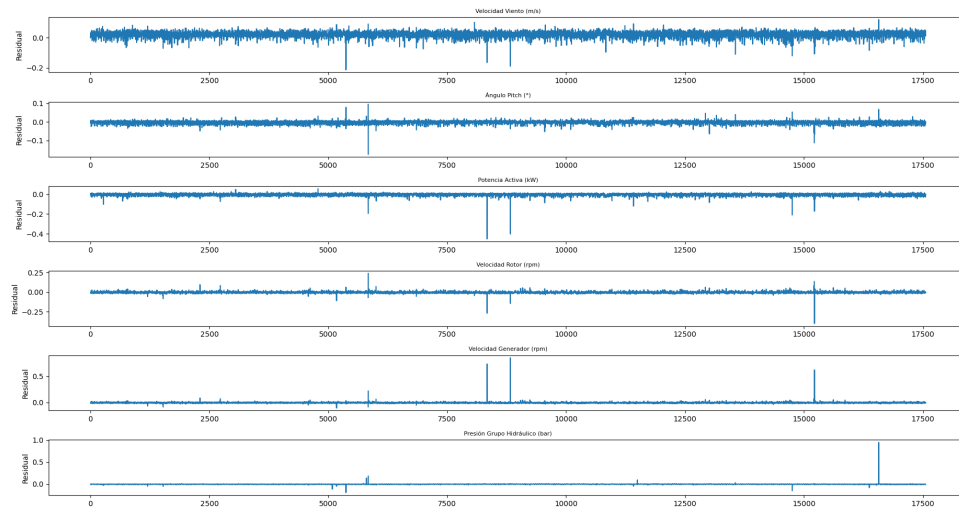


Figura C.4: Resultado General Modelo Grupo Hidráulico en Turbina N°27

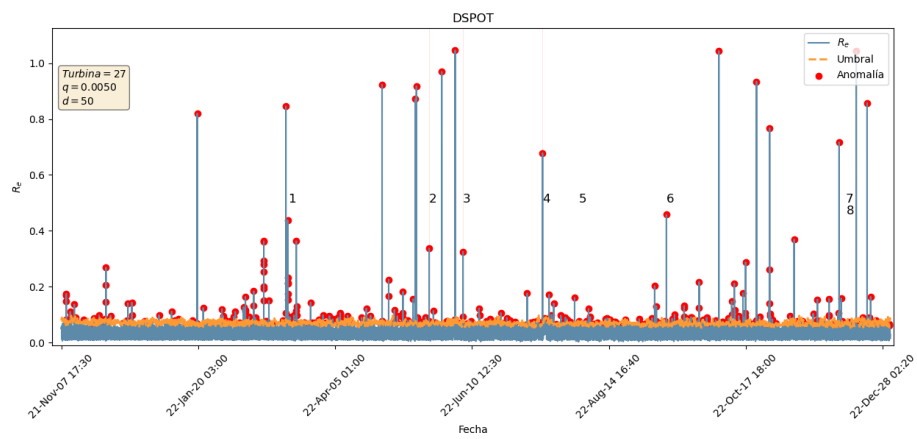
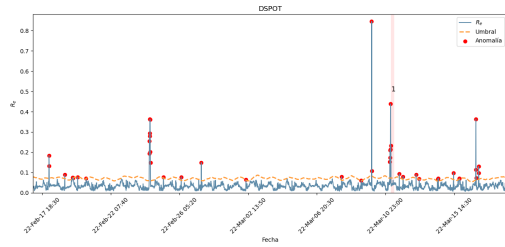
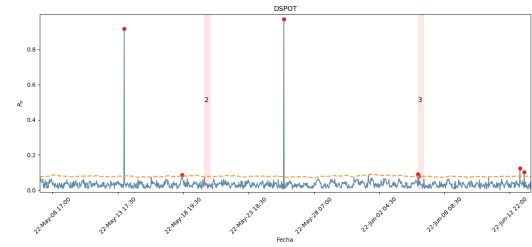


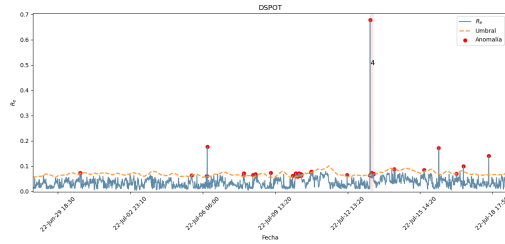
Figura C.5: Resultado General Modelo Grupo Hidráulico en Turbina N°27



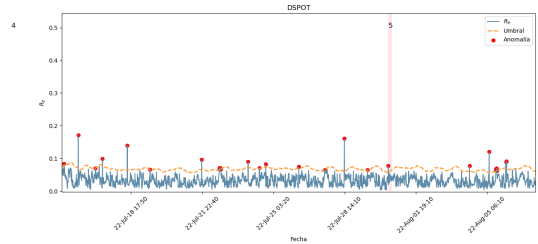
(a) Detección de Alarma 1 por modelo Grupo Hidráulico en Turbina N°27



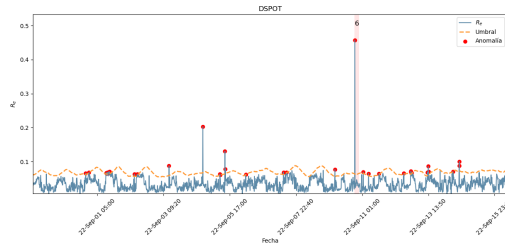
(b) Detección de Alarmas 2 y 3 por modelo Grupo Hidráulico en Turbina N°27



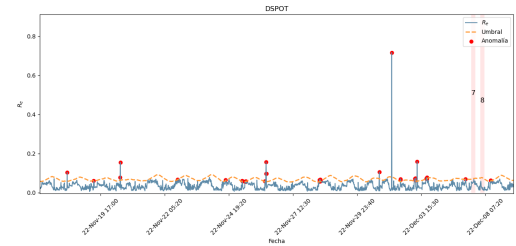
(c) Detección de Alarma 4 por modelo Grupo Hidráulico en Turbina N°27



(d) Detección de Alarma 5 por modelo Grupo Hidráulico en Turbina N°27



(e) Detección de Alarma 6 por modelo Grupo Hidráulico en Turbina N°27



(f) Detección de Alarma 7 y 8 por modelo Grupo Hidráulico en Turbina N°27

Figura C.6: Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°27

C.3. Modelo Generador

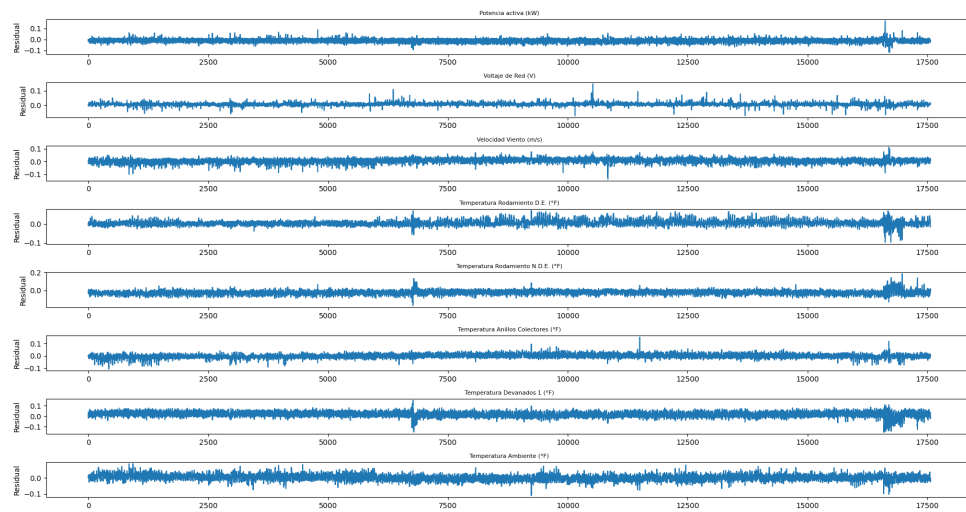


Figura C.7: Residuales Modelo Generador en Turbina N°27

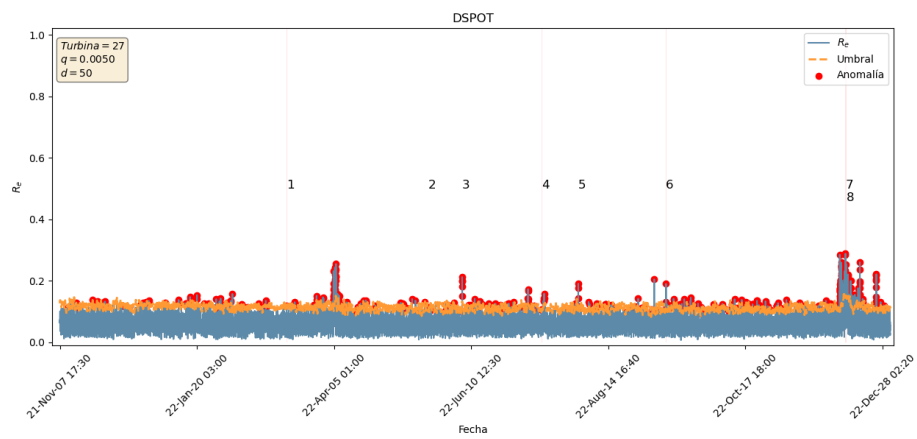
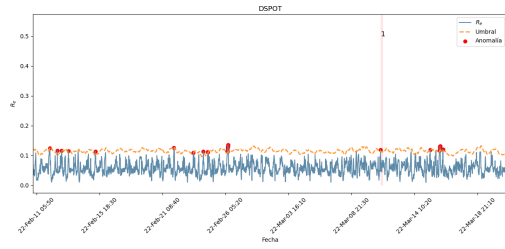
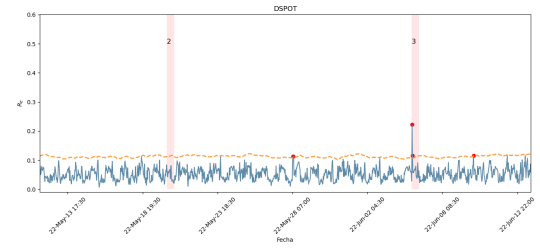


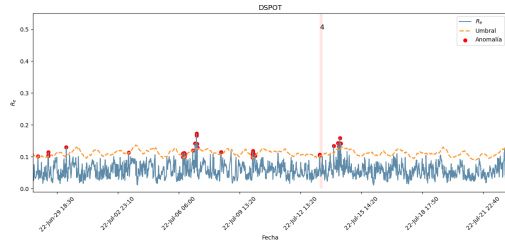
Figura C.8: Resultado General Modelo Generador en Turbina N°27



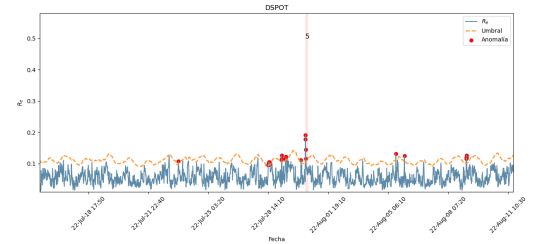
(a) Detección de Alarma 1 por modelo Generador en Turbina N°27



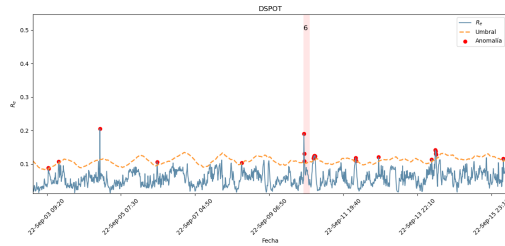
(b) Detección de Alarmas 2 y 3 por modelo Generador en Turbina N°27



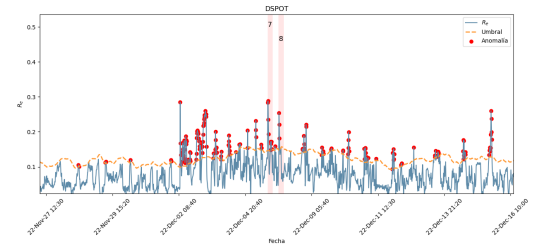
(c) Detección de Alarma 4 por modelo Generador en Turbina N°27



(d) Detección de Alarma 5 por modelo Generador en Turbina N°27



(e) Detección de Alarma 6 por modelo Generador en Turbina N°27



(f) Detección de Alarma 7 y 8 por modelo Generador en Turbina N°27

Figura C.9: Resultados específicos de Alarmas Modelo Generador en Turbina N°27

Anexo D. Resultados Turbina N°29

Tabla D.1: Alarmas identificadas para Turbina N°29

Número	Alarma	Estado posterior	Fecha Inicio	Fecha Fin	Duración
1	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	26-02-2022 17:16	26-02-2022 19:54	2:38:02
	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	26-02-2022 20:02	27-02-2022 16:36	20:34:41
2	410 OGS	Aerogenerador en NO COMUNICA	2022/03/10 15:03:14.524	2022/03/10 15:41:30.476	0:38:16
3	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/03/10 17:49:45.213	2022/03/10 18:32:36.941	0:42:52
4	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/03/25 03:05:13.259	2022/03/25 10:41:49.528	7:36:36
5	440 Alarma de accionamiento de ventilador de bomba de multiplicadora	Aerogenerador en NO COMUNICA	2022/04/12 20:10:29.392	2022/04/13 15:17:15.865	19:06:46
	2116 Cambio de reset a Local				
	915 Paso Pausa-Local Reset				
6	217 Aviso de mal funcionamiento de la válvula del control de carga		2022/05/09 00:03:41.431	2022/05/09 07:46:17.496	7:42:36
7	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/05/27 00:27:21.278	2022/05/27 14:30:12.883	14:02:52
8	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/06/03 07:35:55.177	2022/06/03 12:07:16.591	4:31:21
	6026 Bajo voltaje de red				
	6607 Hueco de tensión				
	6253 Alarma de secuencia de red de tensión				
	6420 No aparece alimentación 400 V				
9	115 Alta presión freno del yaw	Aerogenerador en NO COMUNICA	2022/06/08 07:17:16.544	2022/06/08 11:28:32.855	4:11:16
10	343 Turbina funcionando con hielo	Aerogenerador en NO COMUNICA	2022/06/17 13:09:50.738	2022/06/17 14:55:37.435	1:45:47
11	722 Alarma Fallo Veleta	Aerogenerador en NO COMUNICA	2022/07/06 10:49:46.816	2022/07/06 14:14:09.592	3:24:23
12	6022 Baja Frecuencia de Red	Aerogenerador en NO COMUNICA	2022/09/10 04:52:55.615	2022/09/10 18:06:10.803	13:13:15
	6607 Hueco de tensión				
	6026 Bajo voltaje de red				
	6420 No aparece alimentación 400 V				
13	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/09/11 11:59:49.370	2022/09/11 21:01:08.734	9:01:19
14	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/09/23 04:51:16.918	2022/09/23 08:59:15.040	4:07:58
15	222 Muy baja presión grupo hidráulico	Aerogenerador en NO COMUNICA	2022/10/14 01:36:53.589	2022/10/14 11:12:47.070	9:35:53
16	401 Baja presión multiplicadora	Aerogenerador en NO COMUNICA	2022/10/18 05:13:07.376	2022/10/18 10:07:53.321	4:54:46
17	410 OGS	Aerogenerador en NO COMUNICA	2022/11/30 09:52:27.191	2022/11/30 17:21:22.941	7:28:56

D.1. Modelo Gearbox

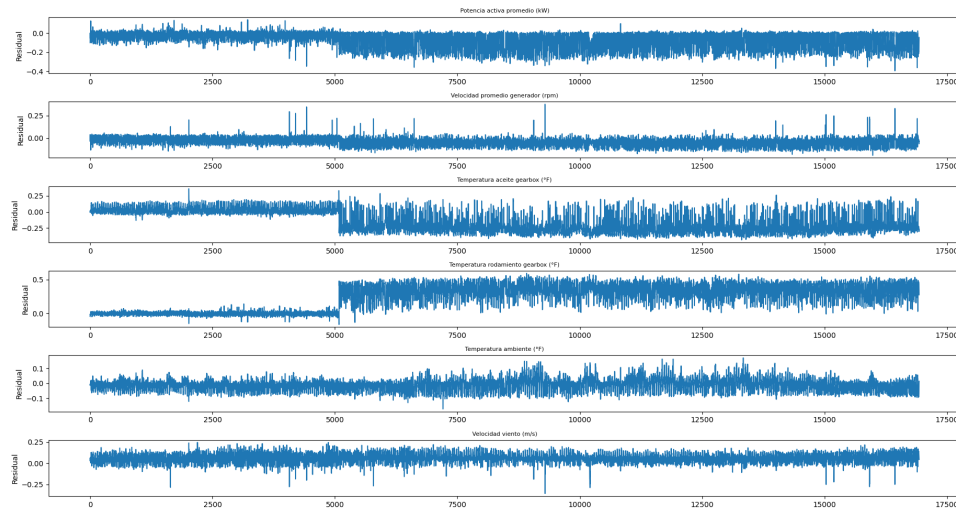


Figura D.1: Residuales Modelo Gearbox en Turbina N°29

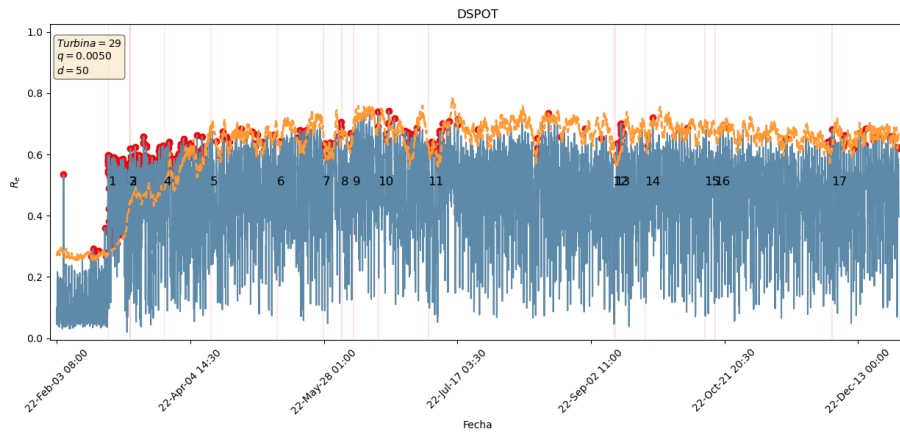


Figura D.2: Resultado General Modelo Gearbox en Turbina N°29

D.2. Modelo Grupo Hidráulico

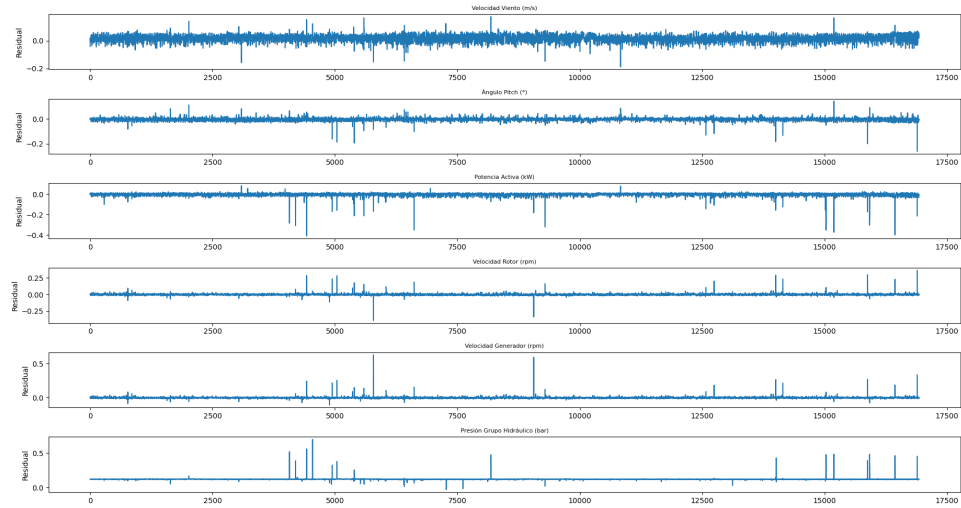


Figura D.3: Residuales Modelo Grupo Hidráulico en Turbina N°29

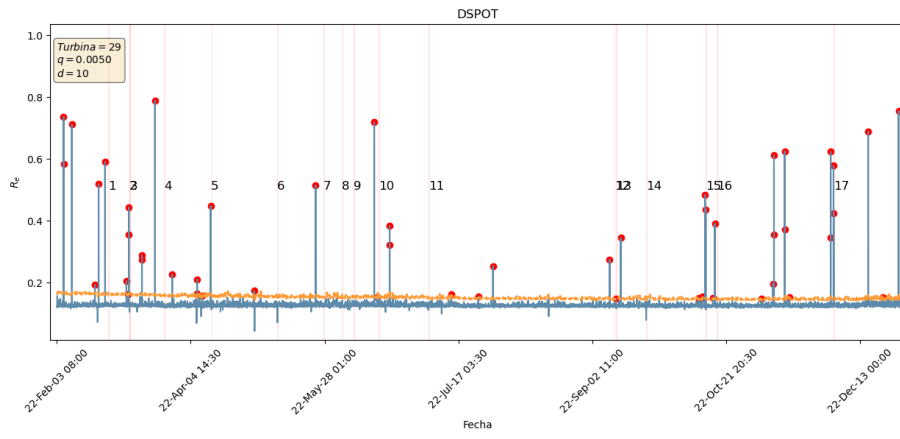
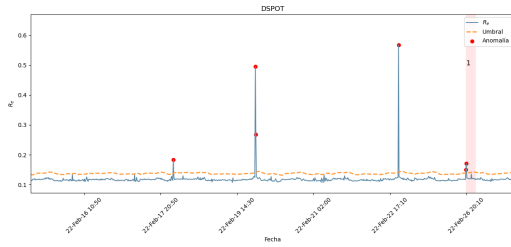
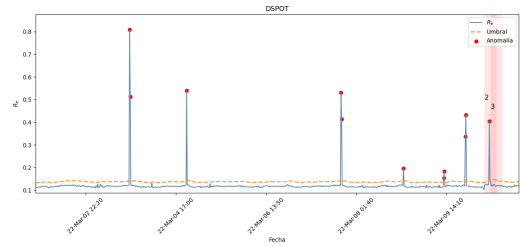


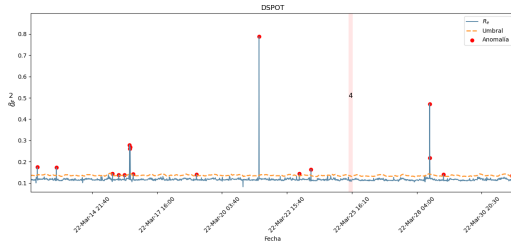
Figura D.4: Resultado General Modelo Grupo Hidráulico en Turbina N°29



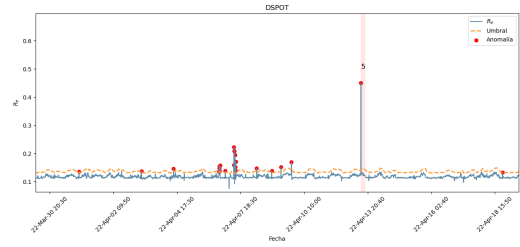
(a) Detección de Alarma 1 por modelo Grupo Hidráulico en Turbina N°29



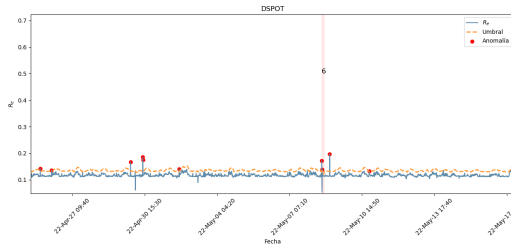
(b) Detección de Alarmas 2 y 3 por modelo Grupo Hidráulico en Turbina N°29



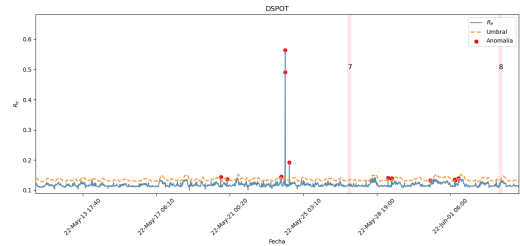
(c) Detección de Alarma 4 por modelo Grupo Hidráulico en Turbina N°29



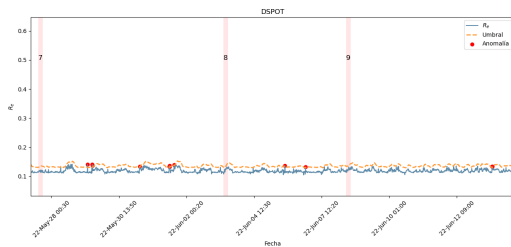
(d) Detección de Alarma 5 por modelo Grupo Hidráulico en Turbina N°29



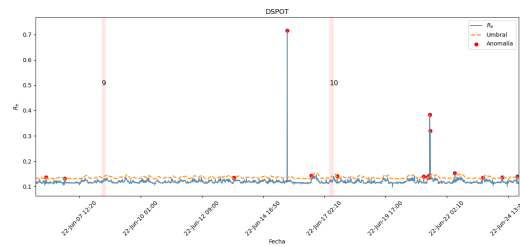
(e) Detección de Alarma 6 por modelo Grupo Hidráulico en Turbina N°29



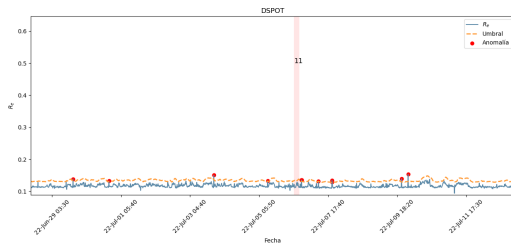
(f) Detección de Alarma 7 y 8 por modelo Grupo Hidráulico en Turbina N°29



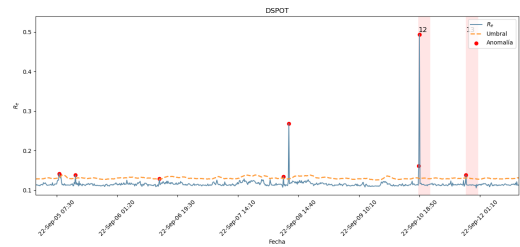
(g) Detección de Alarmas 8 y 9 por modelo Grupo Hidráulico en Turbina N°29



(h) Detección de Alarma 10 por modelo Grupo Hidráulico en Turbina N°29

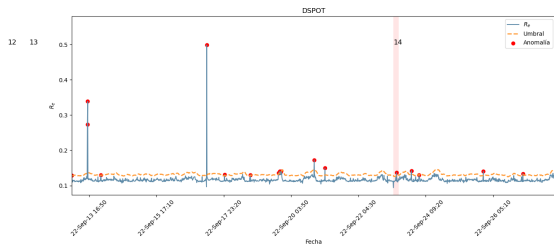


(i) Detección de Alarma 11 por modelo Grupo Hidráulico en Turbina N°29

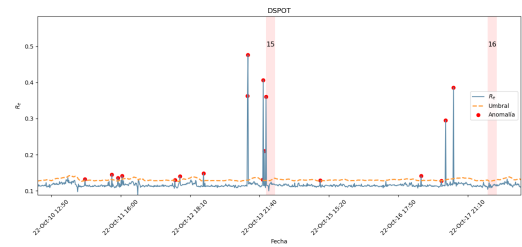


(j) Detección de Alarmas 12 y 13 por modelo Grupo Hidráulico en Turbina N°29

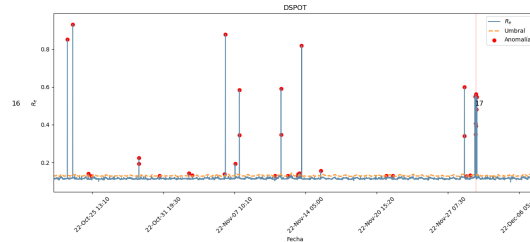
Figura D.5: Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°29



(k) Detección de Alarma 14 por modelo Grupo Hidráulico en Turbina N°29



(l) Detección de Alarmas 15 y 16 por modelo Grupo Hidráulico en Turbina N°29



(m) Detección de Alarma 17 por modelo Grupo Hidráulico en Turbina N°29

Figura D.5: Resultados específicos de Alarmas Modelo Grupo Hidráulico en Turbina N°29

D.3. Modelo Generador

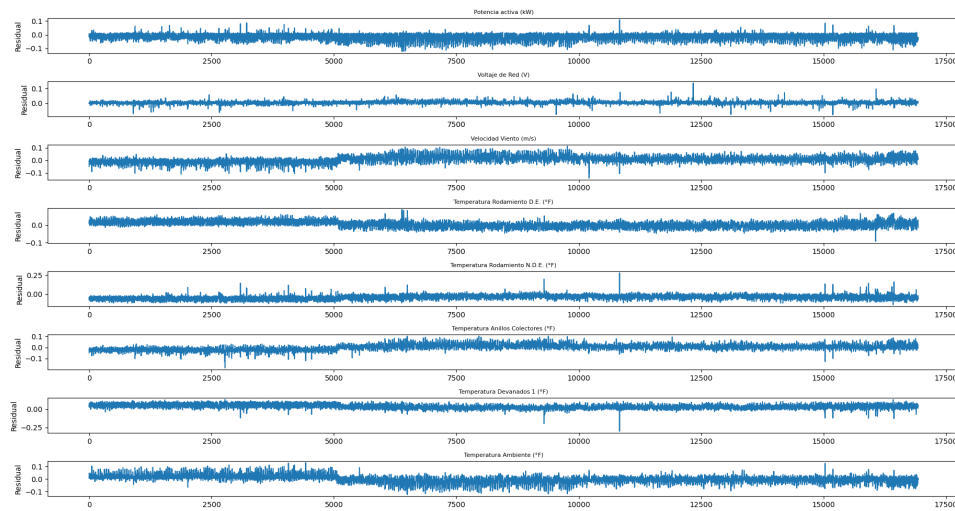


Figura D.6: Residuales Modelo Generador en Turbina N°29

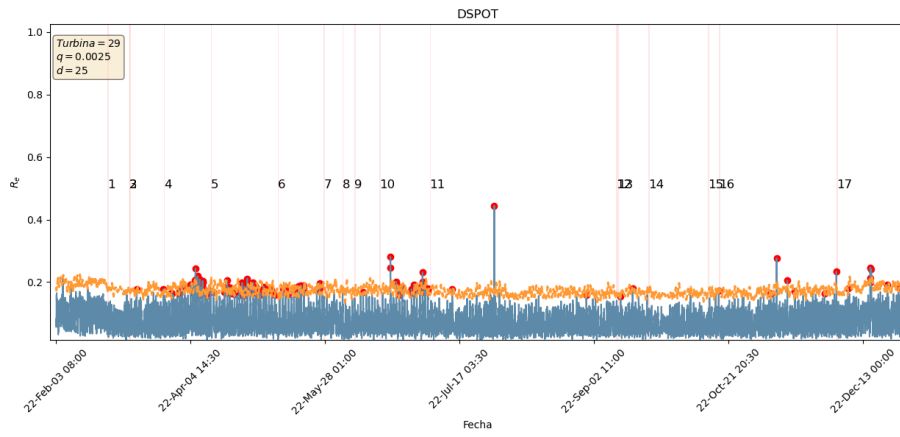
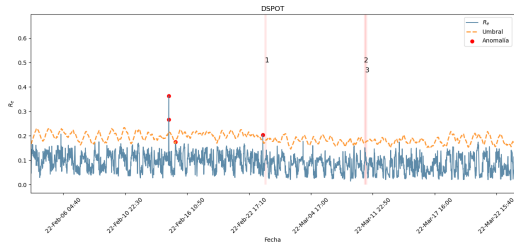
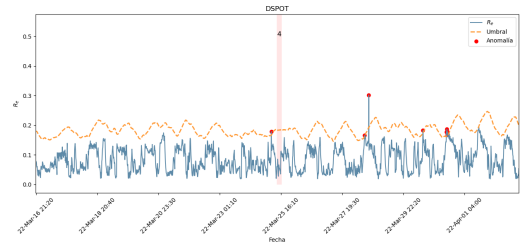


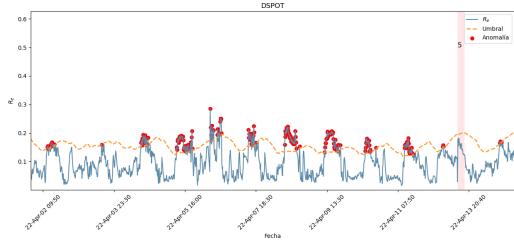
Figura D.7: Resultado General Modelo Generador en Turbina N°29



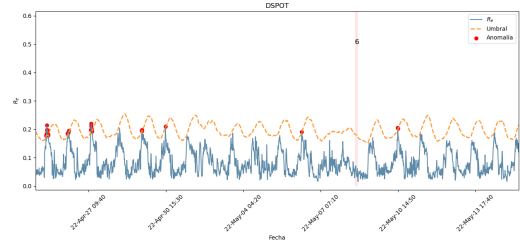
(a) Detección de Alarmas 1, 2 y 3 por modelo Generador en Turbina N°29



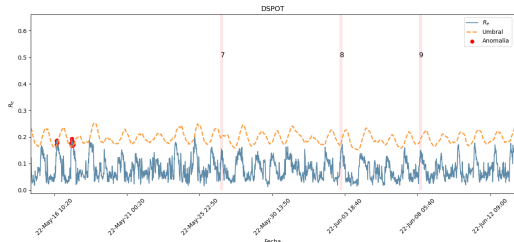
(b) Detección de Alarma 4 por modelo Generador en Turbina N°29



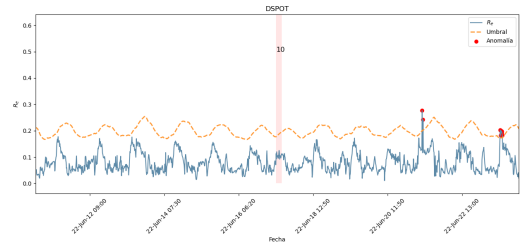
(c) Detección de Alarma 5 por modelo Generador en Turbina N°29



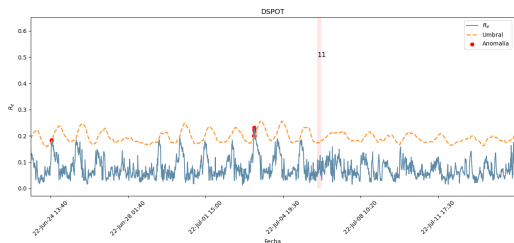
(d) Detección de Alarma 6 por modelo Generador en Turbina N°29



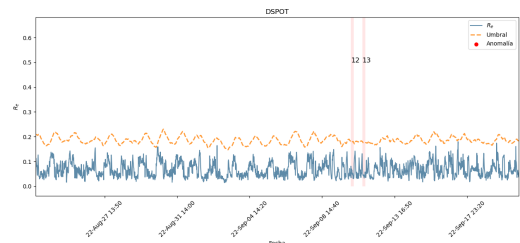
(e) Detección de Alarmas 7, 8 y 9 por modelo Generador en Turbina N°29



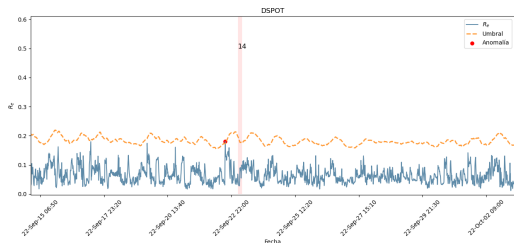
(f) Detección de Alarma 10 por modelo Generador en Turbina N°29



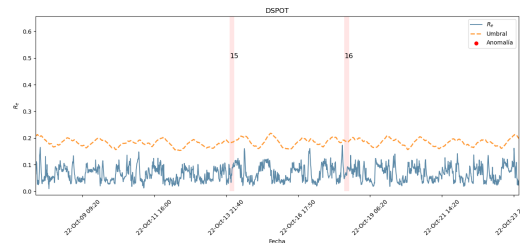
(g) Detección de Alarma 11 por modelo Generador en Turbina N°29



(h) Detección de Alarmas 12 y 13 por modelo Generador en Turbina N°29

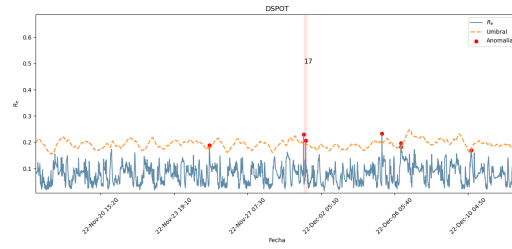


(i) Detección de Alarma 14 por modelo Generador en Turbina N°29



(j) Detección de Alarmas 15 y 16 por modelo Generador en Turbina N°29

Figura D.8: Resultados específicos de Alarmas Modelo Generador en Turbina N°29



(k) Detección de Alarma 17 por modelo Generador en Turbina N°29

Figura D.8: Resultados específicos de Alarmas Modelo Generador en Turbina N°29